



# Machine Learning for Bioinformatics & Systems Biology

## 4. Clustering & hidden Markov models

Perry Moerland      *Amsterdam UMC, University of Amsterdam*

Marcel Reinders      *Delft University of Technology*

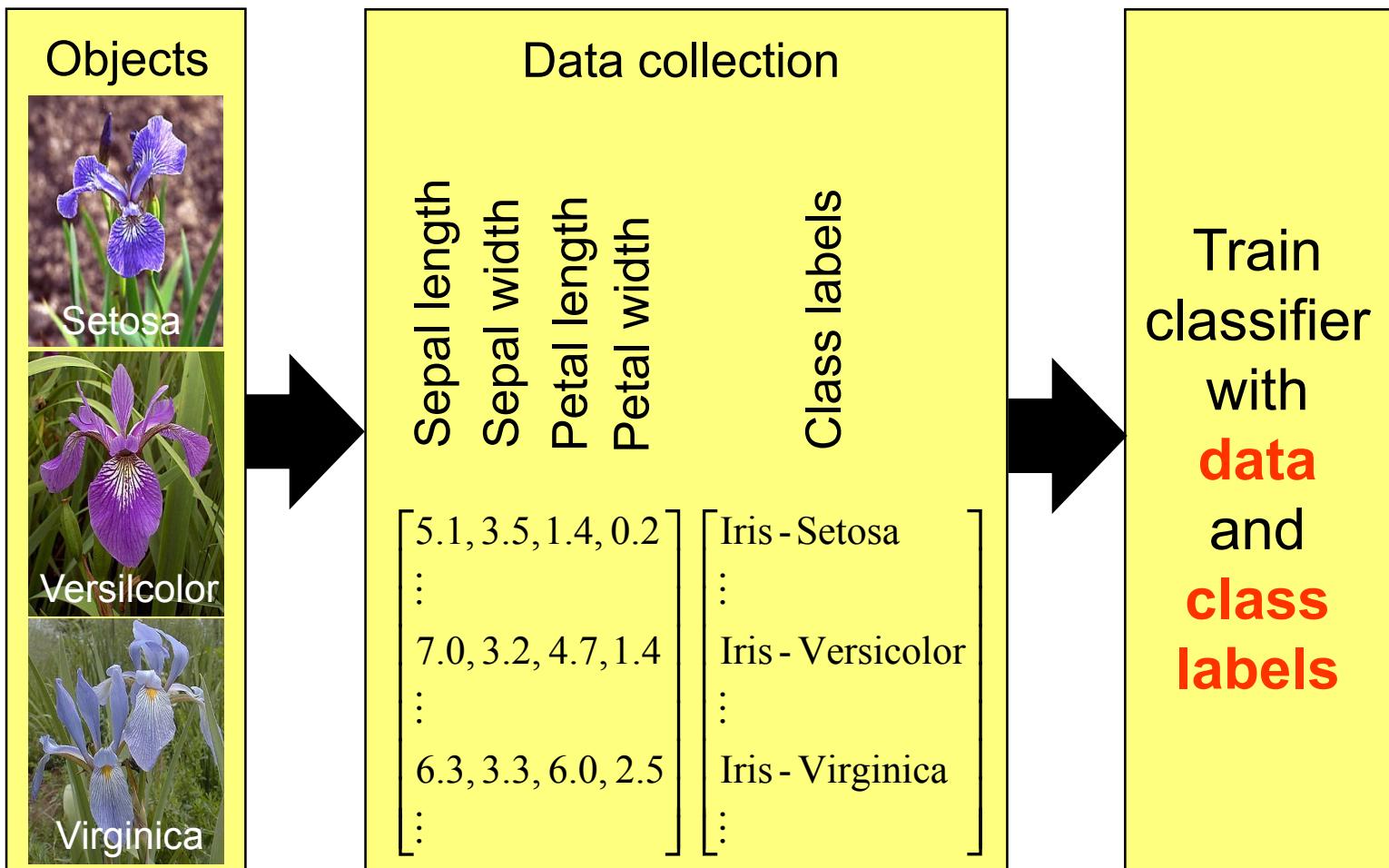
Lodewyk Wessels      *Netherlands Cancer Institute*

*Some material courtesy of Robert Duin and David Tax*

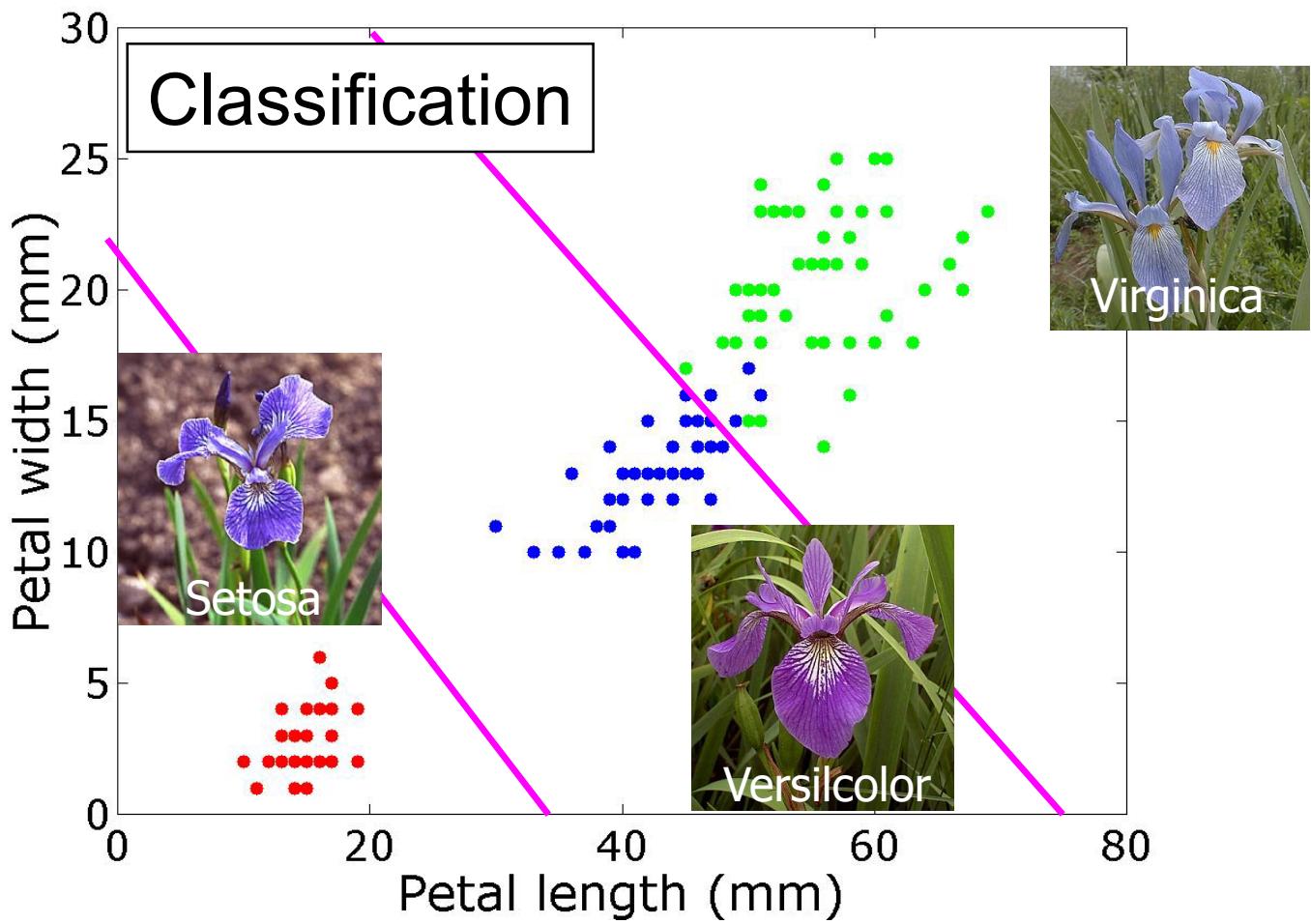
# Clustering

- Supervised vs. unsupervised learning
- Hierarchical clustering
- Sum-of-squares clustering ( $k$ -means)
- Cluster validation
- Mixtures-of-Gaussians clustering (EM algorithm)

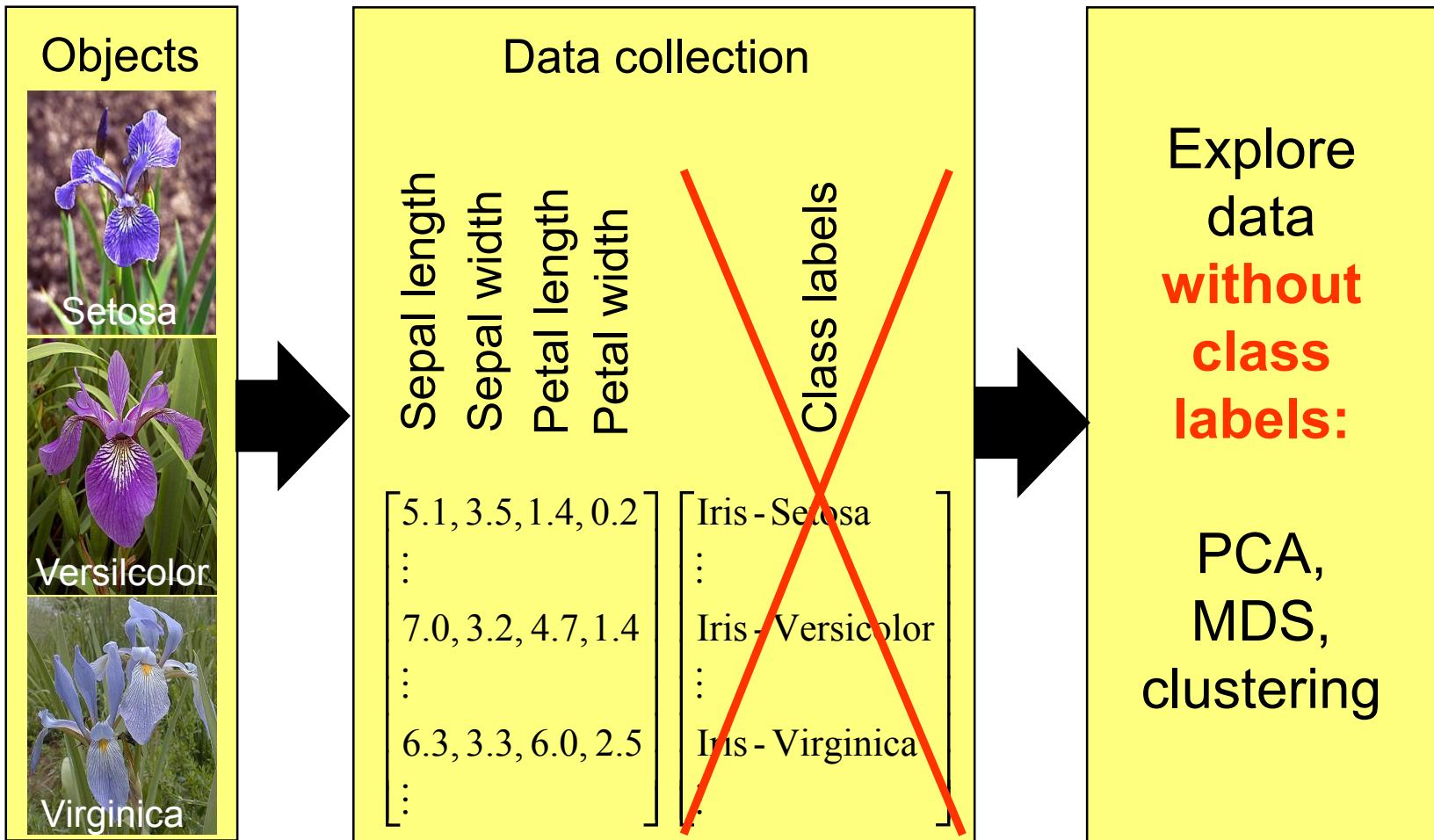
# Supervised learning



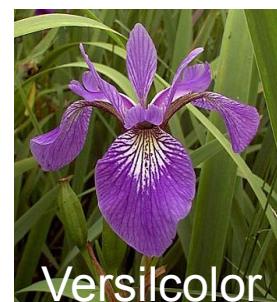
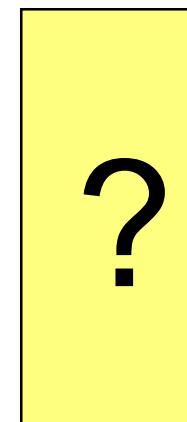
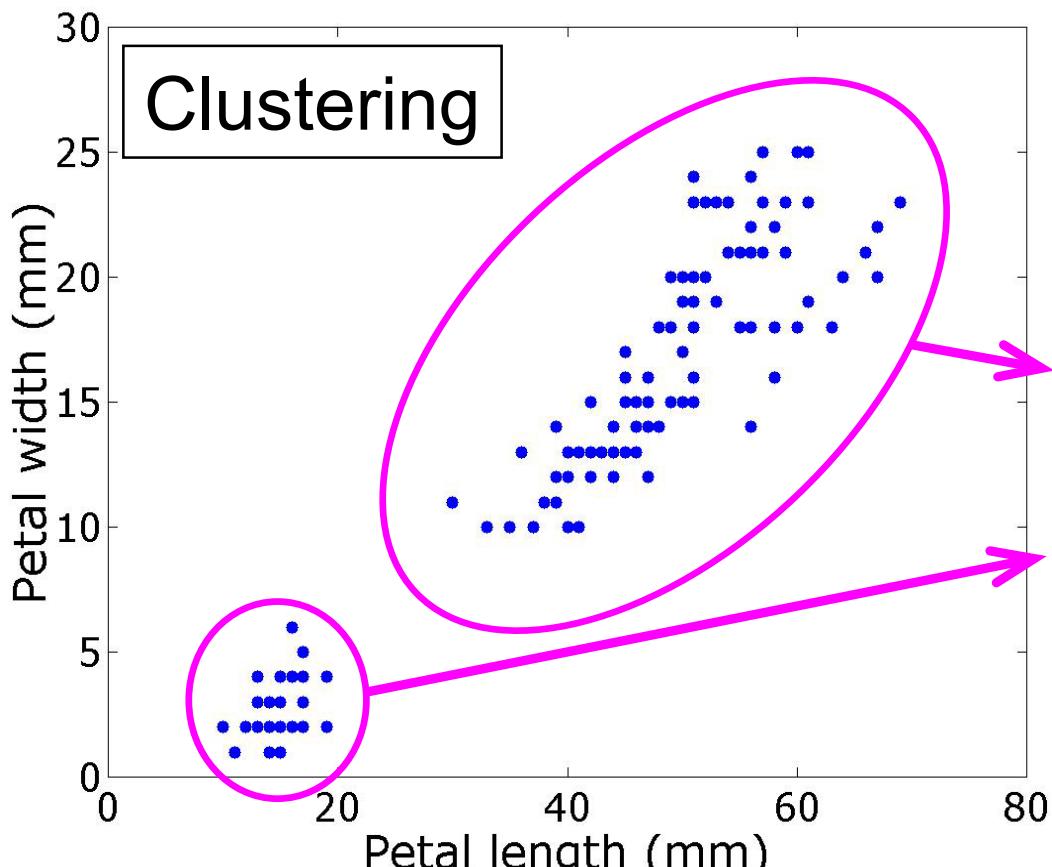
# Supervised learning (2)



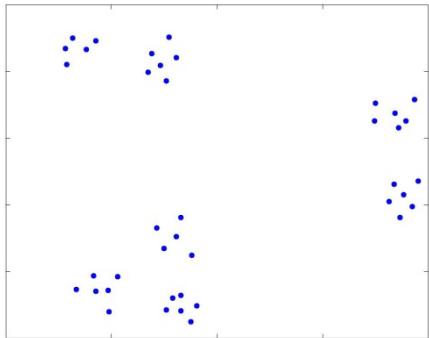
# Unsupervised learning



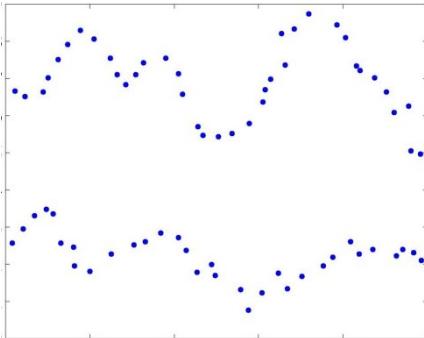
# Unsupervised learning (2)



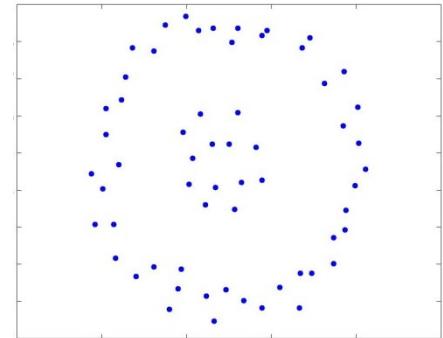
# What is a cluster?



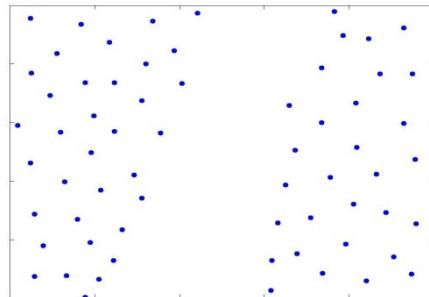
Shape: compact, convex  
Separation: large



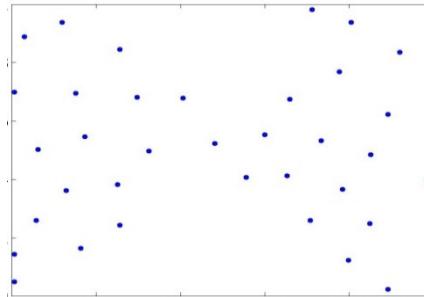
Shape: strings  
Separation: large?



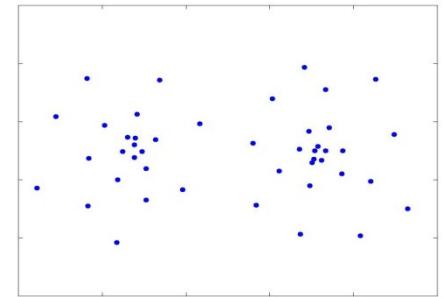
Shape: convex and circular  
Separation: large?



Shape: ?  
Separation: large?



Shape: loose, convex  
Separation: small



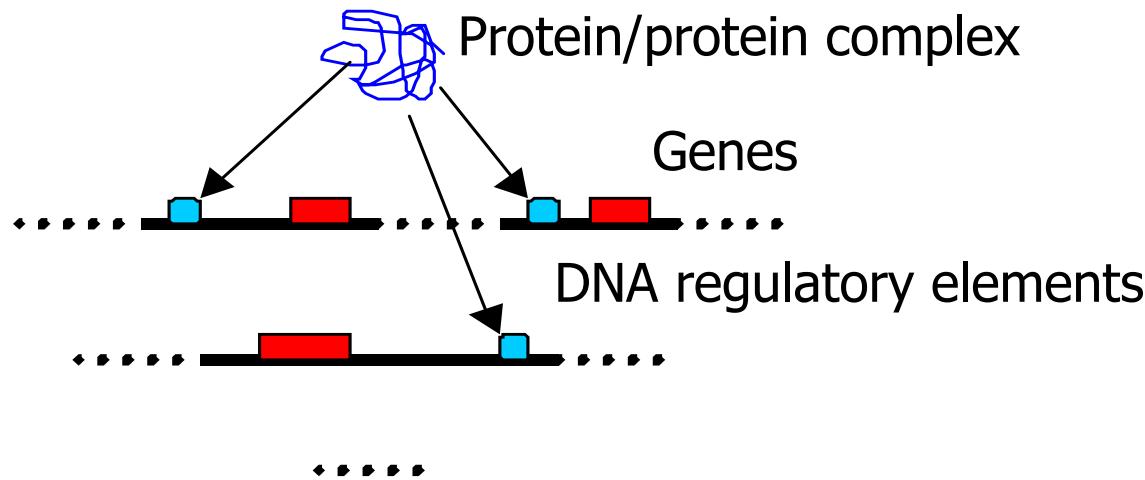
Shape: loose, convex  
Separation: small

# What is a cluster? (2)

- Clustering: finding natural groups in data...
  - which themselves are far apart
  - in which objects are close together
- Define what is “far apart” and “close together”:
  - Need a distance measure or dissimilarity measure
  - This measure should capture what we think is important for the grouping
  - The choice for a certain distance measure is often the most important choice in clustering!
- There is no such thing as *the objective clustering*

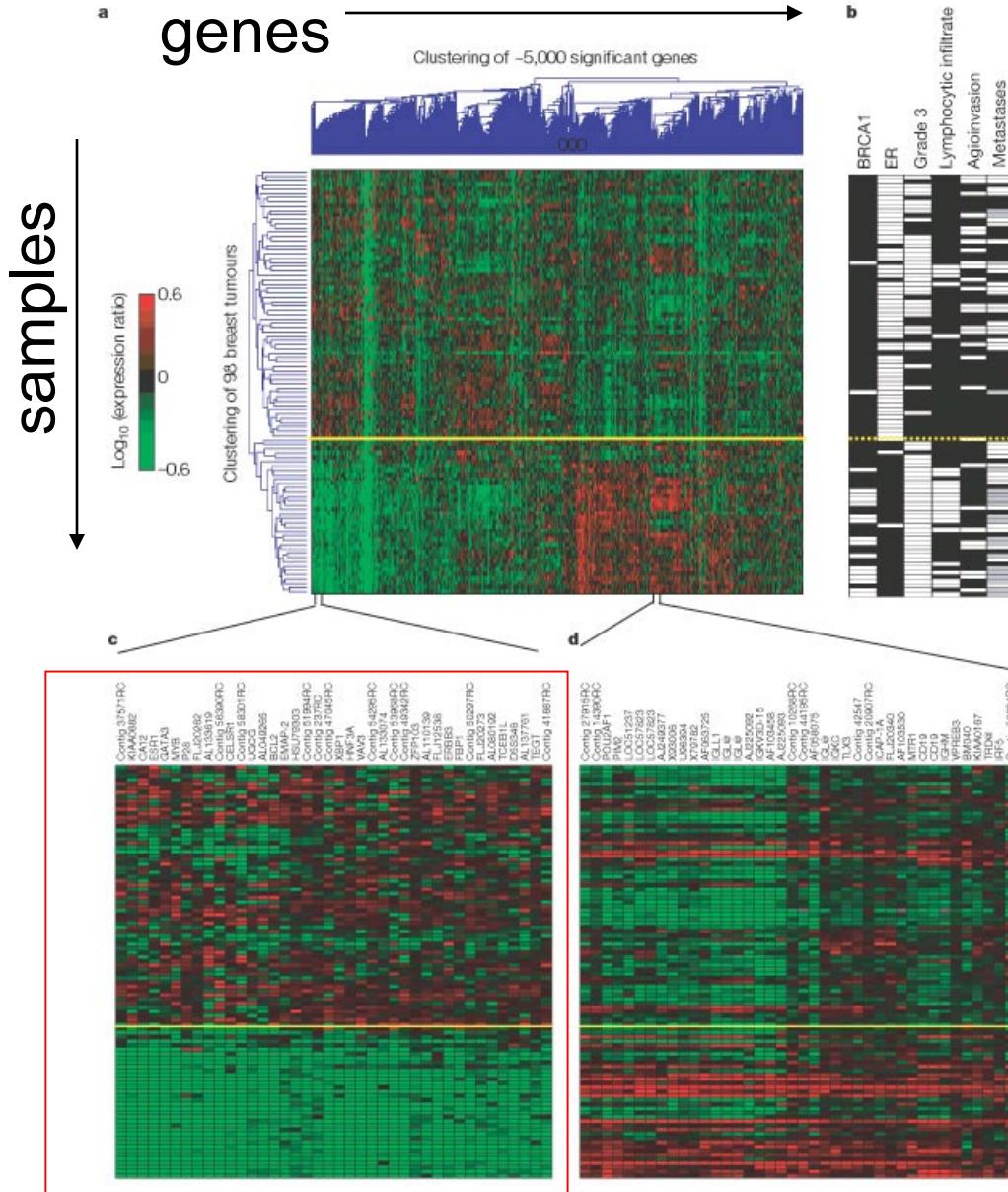
# What is a cluster in bioinformatics?

- Clustering gene expression data:
- Genes: similar ~ co-expression ~ co-regulation ~ same pathway / same function



- Samples: similar ~ same type of tissue
- Used for discovery of new subclasses (*subtypes*) in tumors

# Example: genes (and samples)



■ negative

□ positive

histopathological data

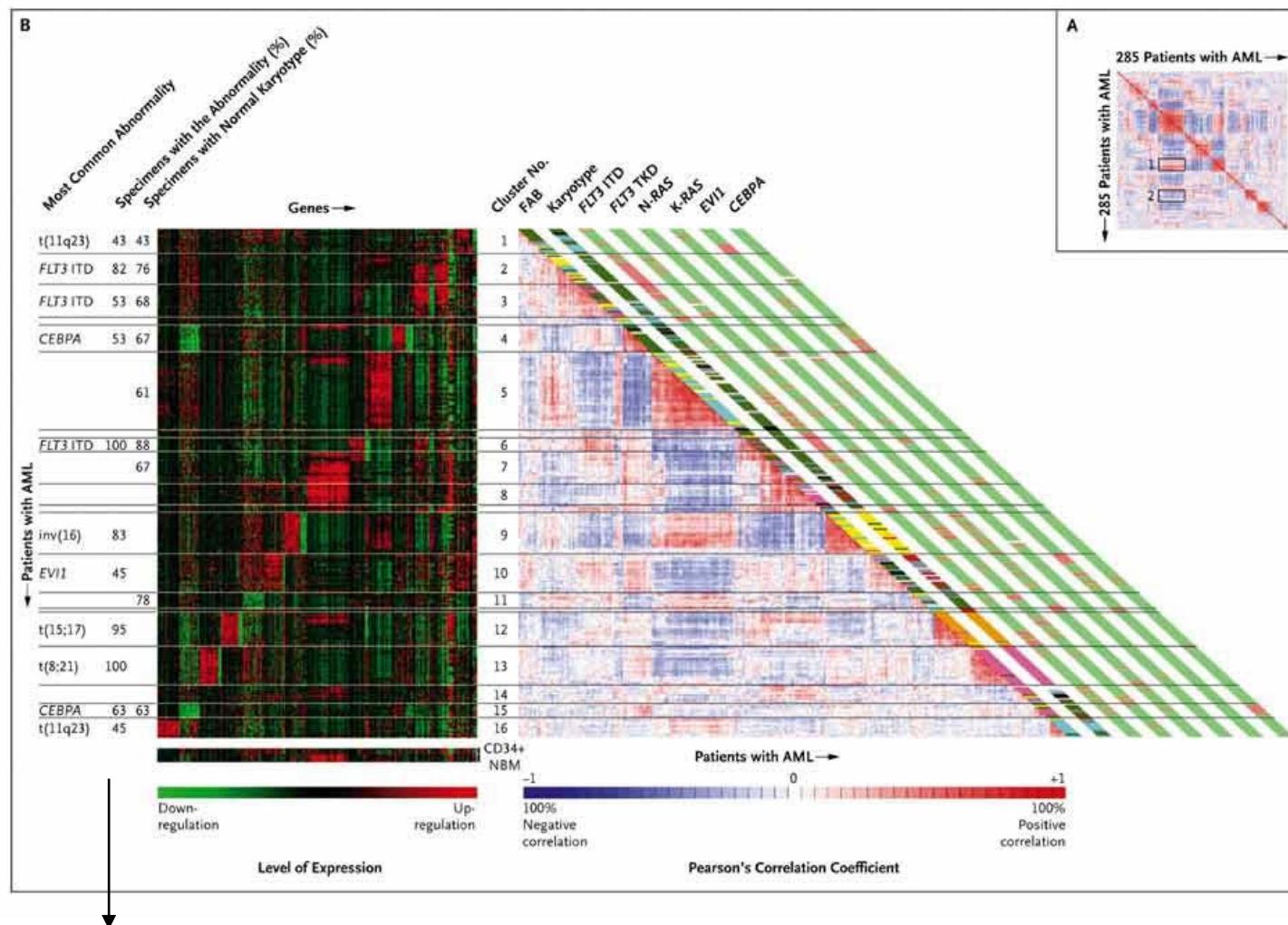
ER gene (*ESR1*) and genes co-regulated with ER, some of which are known ER target genes

Van 't Veer et al, Nature 415: 530-536 (2002)



# Example: samples

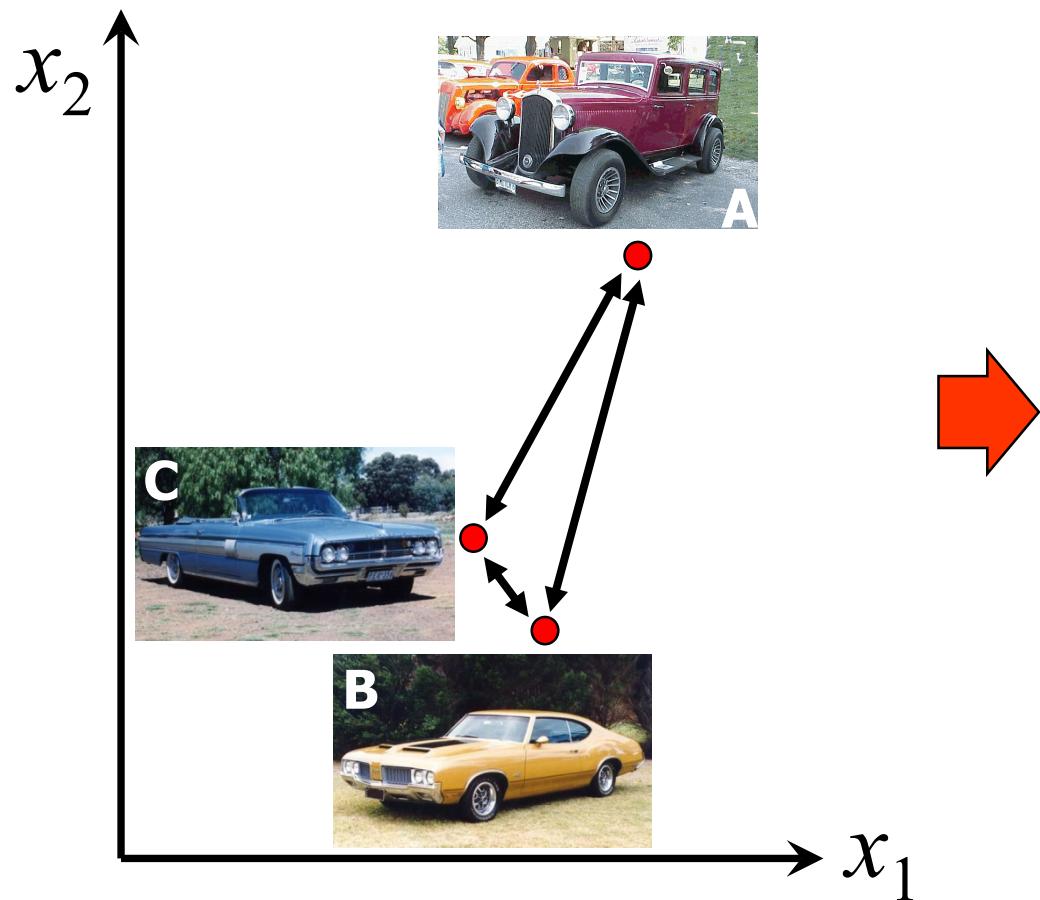
Valk et al, N Engl J Med. 2004 Apr 15;350(16):1617-28.



Identified 16 groups of patients with acute myeloid leukemia



# Dissimilarity measures



$$\begin{aligned} \mathbf{D} &= \begin{bmatrix} 0 & d(\mathbf{A}, \mathbf{B}) & d(\mathbf{A}, \mathbf{C}) \\ 0 & 0 & d(\mathbf{B}, \mathbf{C}) \\ 0 & & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 10 & 11 \\ 0 & 0 & 2 \\ 0 & & 0 \end{bmatrix} \end{aligned}$$

# Dissimilarity measures (2)

- Let  $d(r, s)$  be the dissimilarity between objects  $r$  and  $s$
- Formally, dissimilarity measures should satisfy

$$d(r, s) \geq 0, \forall r, s$$

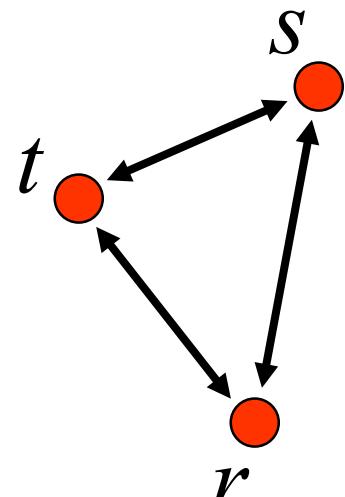
$$d(r, r) = 0, \forall r$$

$$d(r, s) = d(s, r), \forall r, s$$

- If in addition, the triangle inequality holds, the measure is a *metric*

$$d(r, t) + d(t, s) \geq d(r, s), \forall r, s, t$$

- Most often used: Euclidean distance (metric)

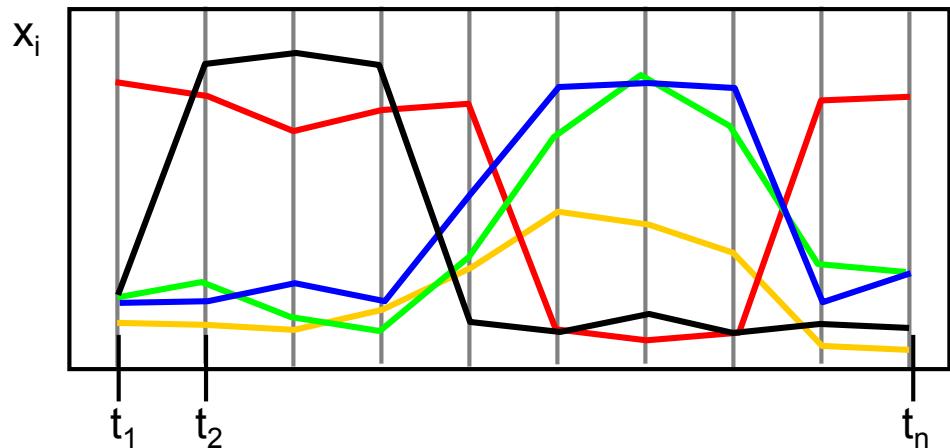


# Dissimilarity measures (3)

- Example: time series data

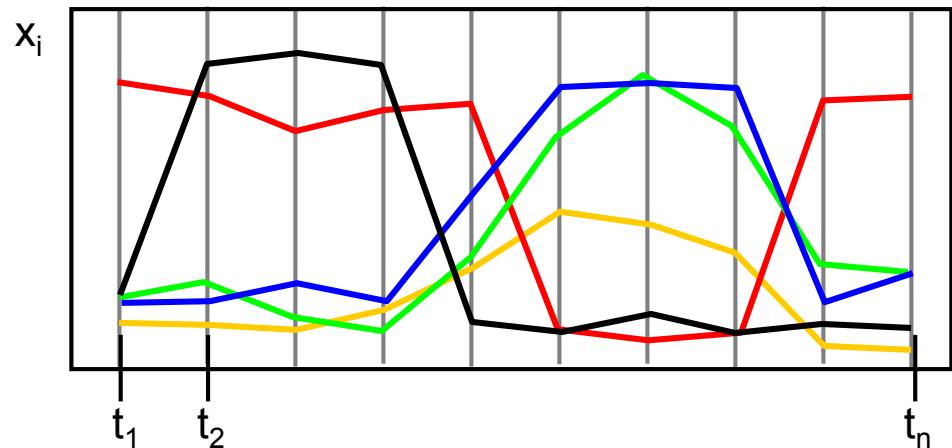
Euclidean distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^n (x_{i,t} - x_{j,t})^2$$



# Dissimilarity measures (3)

- Example:  
time series data



Euclidean distance  
match exact shape

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^n (x_{i,t} - x_{j,t})^2$$

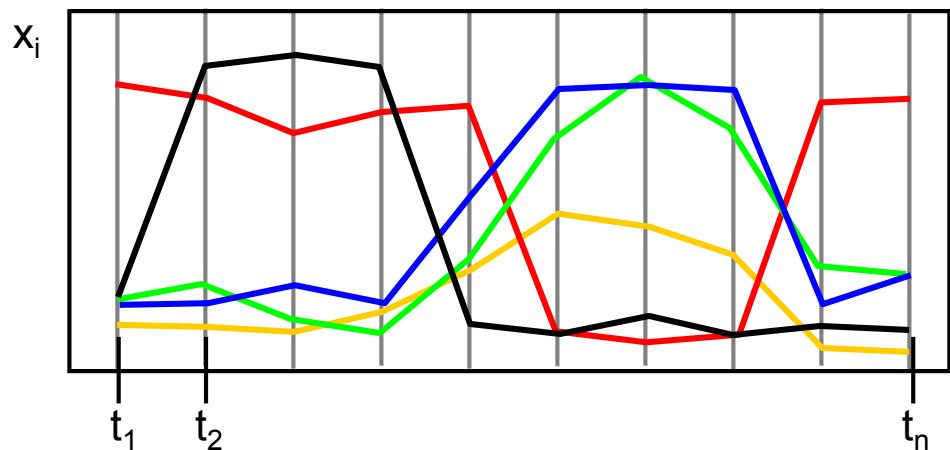
$$\begin{aligned} d(\text{blue circle}, \text{green circle}) &< d(\text{blue circle}, \text{yellow circle}) \\ d(\text{blue circle}, \text{green circle}) &<< d(\text{blue circle}, \text{red circle}) \\ d(\text{blue circle}, \text{green circle}) &<< d(\text{blue circle}, \text{black circle}) \end{aligned}$$

# Dissimilarity measures (3)

- Example: time series data

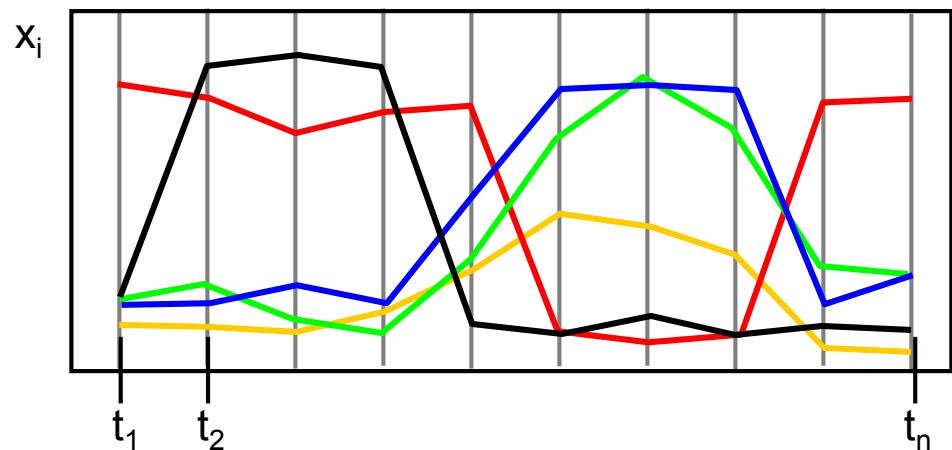
Euclidean distance  
match exact shape

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^n (x_{i,t} - x_{j,t})^2$$



# Dissimilarity measures (3)

- Example:  
time series data



Euclidean distance  
match exact shape

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^n (x_{i,t} - x_{j,t})^2$$

$d(\text{blue}, \text{green}) < d(\text{blue}, \text{yellow})$   
 $d(\text{blue}, \text{green}) \ll d(\text{blue}, \text{red})$   
 $d(\text{blue}, \text{green}) \ll d(\text{blue}, \text{black})$

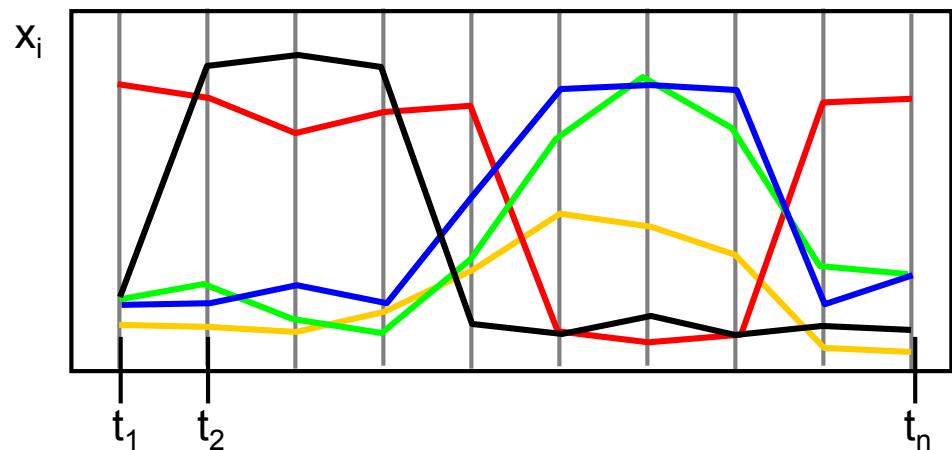
Pearson correlation  
ignore amplitude

$$\rho_{ij} = \frac{\sum_{t=1}^n (x_{i,t} - \mu_i)(x_{j,t} - \mu_j)}{\sigma_i \sigma_j}$$

$d(\text{blue}, \text{green}) \approx d(\text{blue}, \text{yellow})$   
 $d(\text{blue}, \text{green}) \ll d(\text{blue}, \text{red})$   
 $d(\text{blue}, \text{green}) \ll d(\text{blue}, \text{black})$

# Dissimilarity measures (3)

- Example:  
time series data



Euclidean distance  
match exact shape

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^n (x_{i,t} - x_{j,t})^2$$

$$\begin{aligned} d(\text{blue}, \text{green}) &< d(\text{blue}, \text{yellow}) \\ d(\text{blue}, \text{green}) &\ll d(\text{blue}, \text{red}) \\ d(\text{blue}, \text{green}) &\ll d(\text{blue}, \text{black}) \end{aligned}$$

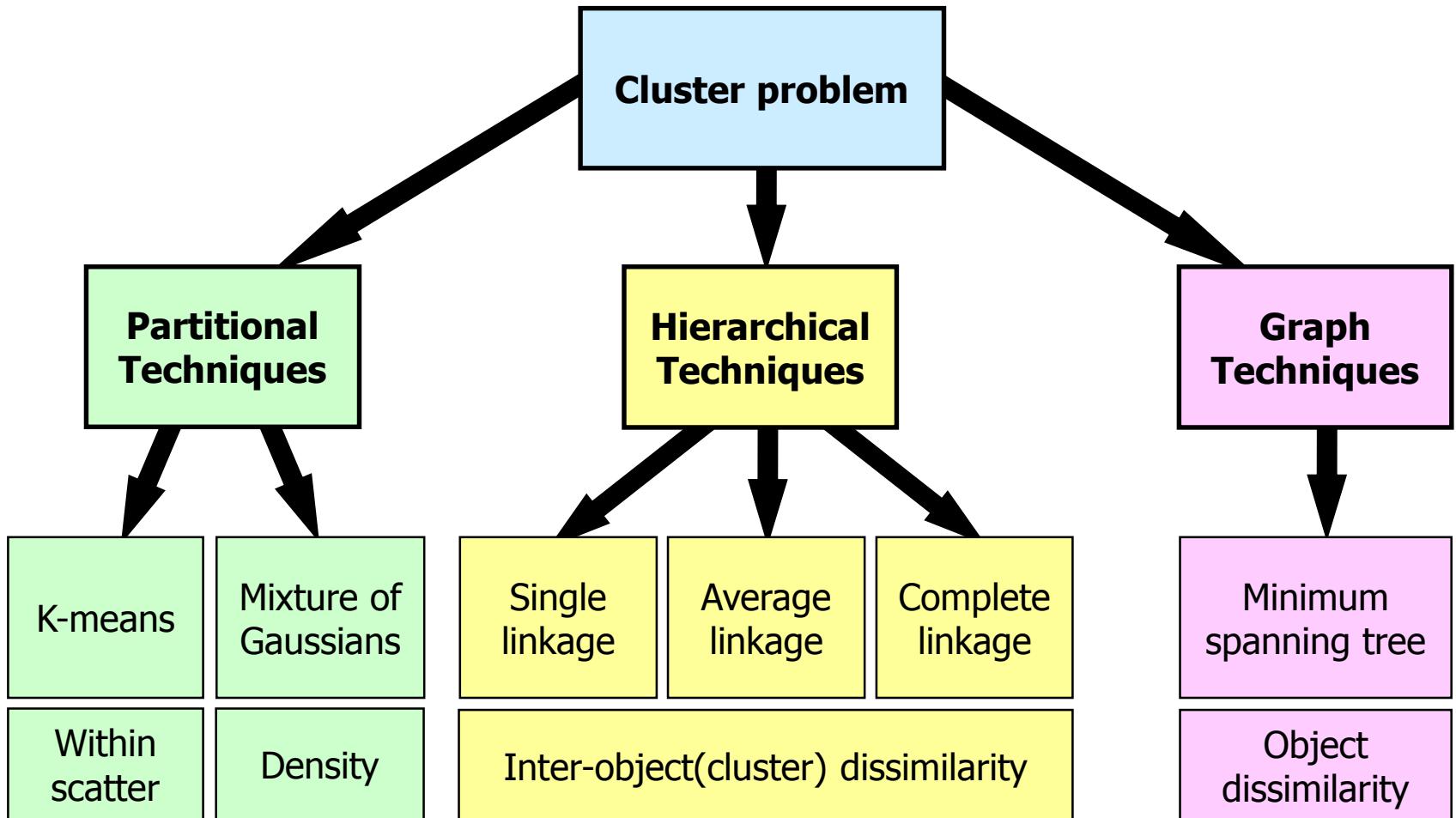
Pearson correlation  
ignore amplitude

$$\rho_{ij} = \frac{\sum_{t=1}^n (x_{i,t} - \mu_i)(x_{j,t} - \mu_j)}{\sqrt{\sum_{t=1}^n (x_{i,t} - \mu_i)^2 \sum_{t=1}^n (x_{j,t} - \mu_j)^2}}$$

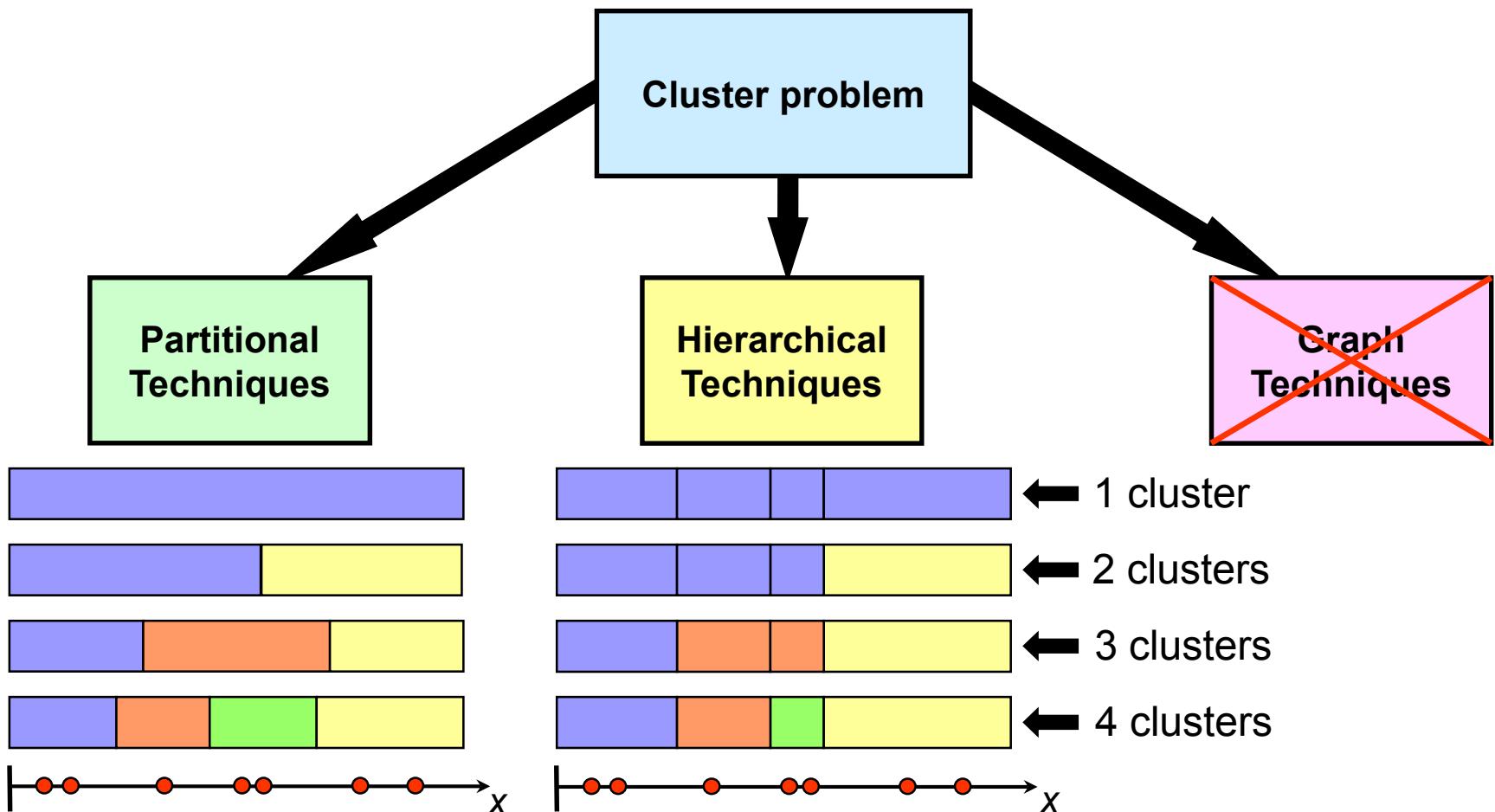
$$\begin{aligned} d(\text{blue}, \text{green}) &\approx d(\text{blue}, \text{yellow}) & d(\text{blue}, \text{green}) &\approx d(\text{blue}, \text{yellow}) \\ d(\text{blue}, \text{green}) &\ll d(\text{blue}, \text{red}) & d(\text{blue}, \text{green}) &\approx d(\text{blue}, \text{red}) \\ d(\text{blue}, \text{green}) &\ll d(\text{blue}, \text{black}) & d(\text{blue}, \text{green}) &\ll d(\text{blue}, \text{black}) \end{aligned}$$

Absolute correlation  
ignore amplitude & sign

# Clustering techniques



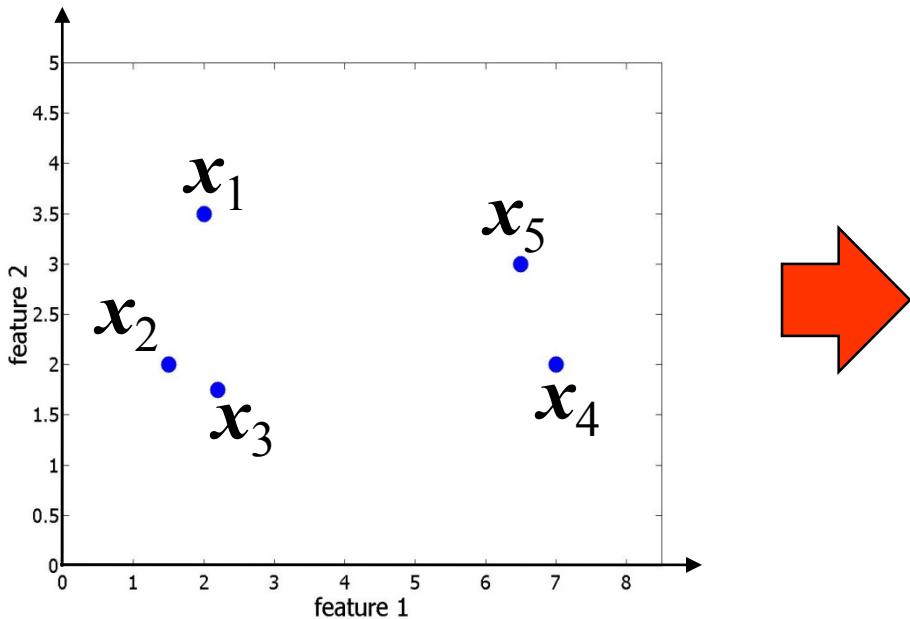
# Clustering techniques (2)



# Hierarchical clustering

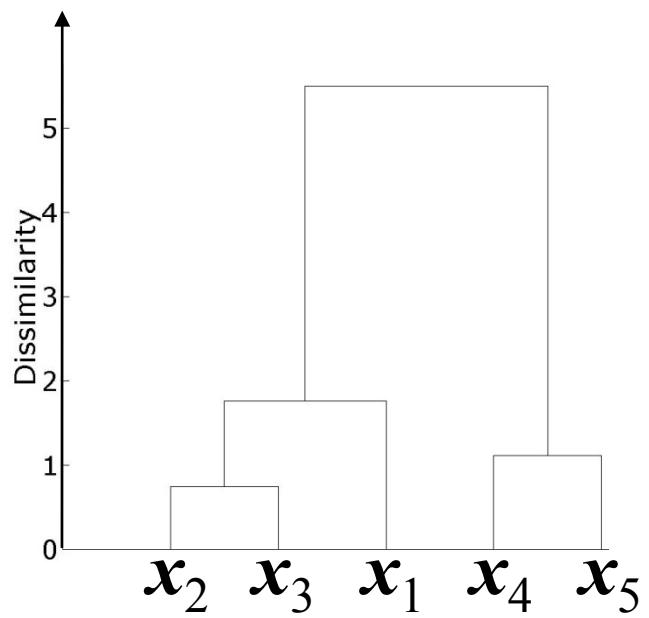
Input:

- dataset,  $\mathbf{X}$ :  $[n \times p]$ , or directly:
- dissimilarity matrix,  $\mathbf{D}$ :  $[n \times n]$
- linkage type



Output:

- dendrogram

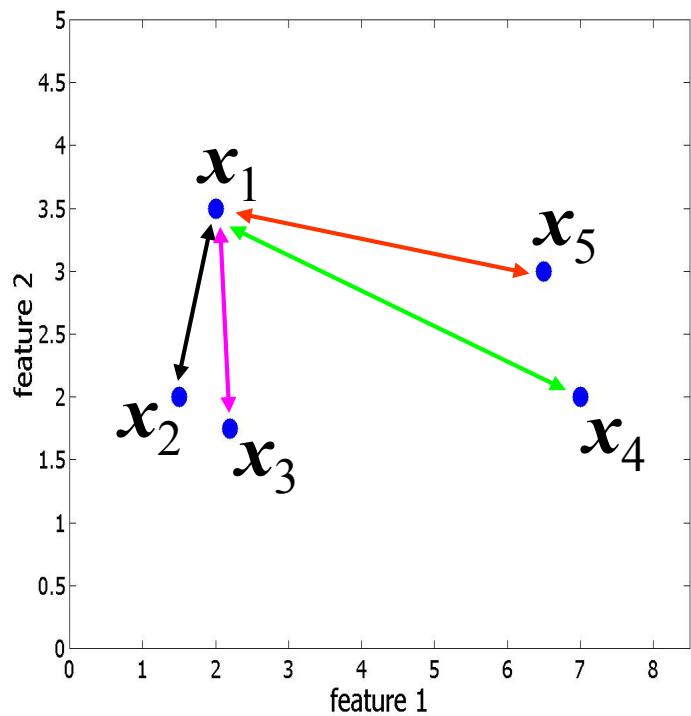


# Hierarchical clustering (2)

- **Algorithm** (agglomerative clustering)
  - Start: all objects of  $\mathbf{X}$  in a separate cluster
  - Clustering: combine the 2 clusters with the shortest distance in dissimilarity matrix,  $\mathbf{D}$
  - Distance between clusters is based on linkage type:
    - single, complete, average, ...
  - Repeat until only 1 cluster is left

# Hierarchical clustering (3)

Dataset

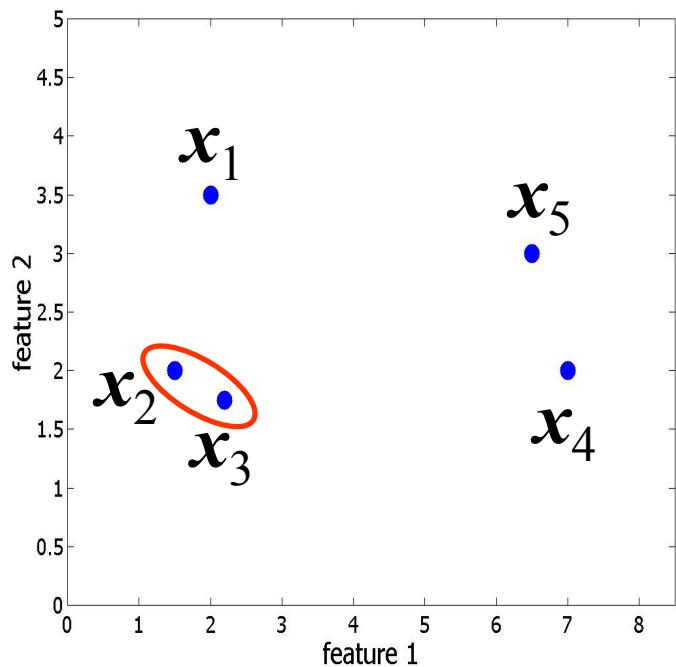


Euclidean distance matrix,  $D$

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_1$	0.00	1.58	1.76	5.22	4.53
$x_2$		0.00	0.74	5.50	5.10
$x_3$			0.00	4.81	4.48
$x_4$				0.00	1.12
$x_5$					0.00

# Hierarchical clustering (4)

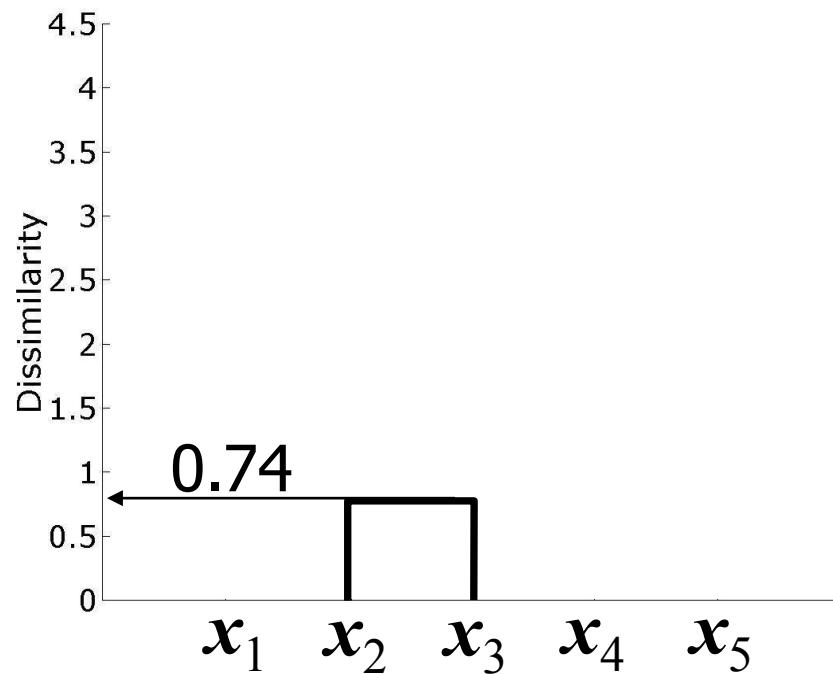
- **Step 1:**  
Find the most similar pair of objects:  $\min_{(i,j)}\{d(i,j)\} = d(2,3)$



	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_1$	0.00	1.58	1.76	5.22	4.53
$x_2$		0.00	0.74	5.50	5.10
$x_3$			0.00	4.81	4.48
$x_4$				0.00	1.12
$x_5$					0.00

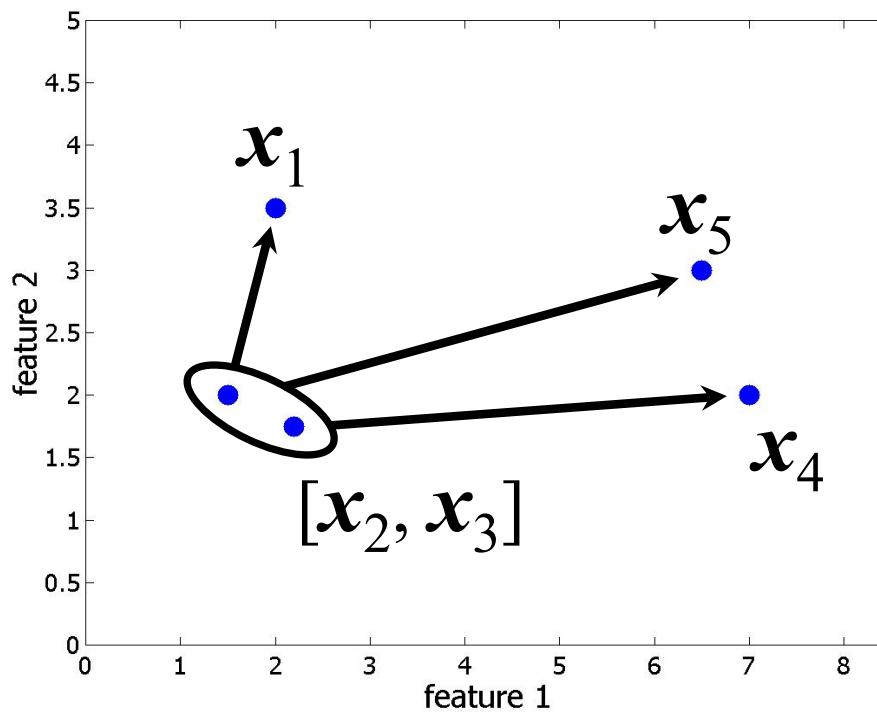
# Hierarchical clustering (5)

- **Step 2:**  
Merge  $x_2$  and  $x_3$  into a single object,  $[x_2, x_3]$ ;



# Hierarchical clustering (6)

- Step 3:  
Recompute  $D$  –  
*what is the distance between  $[x_2, x_3]$  and the rest?*

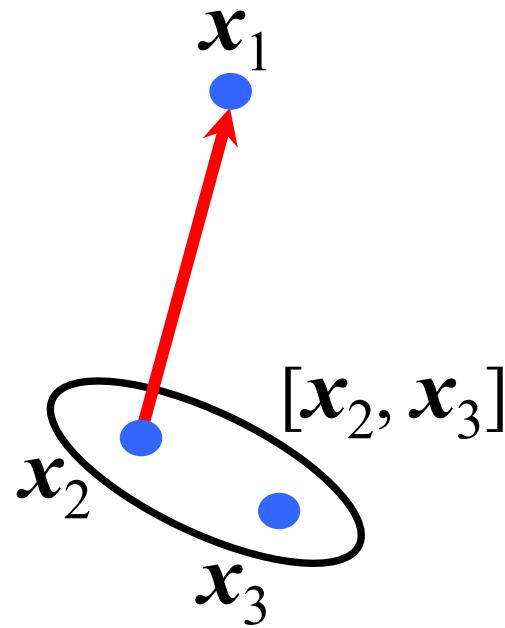


# Hierarchical clustering (7)

- Step 3:

Recompute  $D$  –

**single linkage:**  $d([x_2, x_3], x_1) = \min(d(x_1, x_2), d(x_1, x_3))$

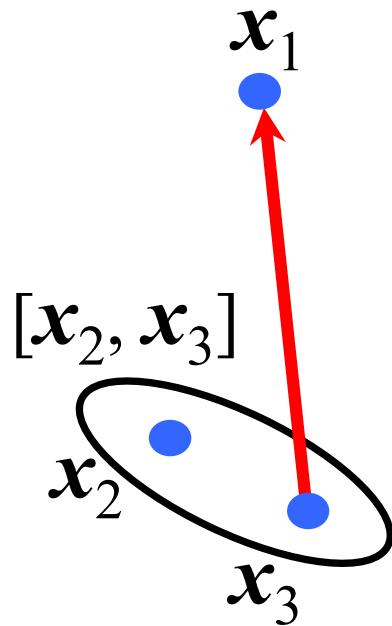


# Hierarchical clustering (8)

- Step 3:

Recompute  $D$  –

**complete linkage:**  $d([x_2, x_3], x_1) = \max(d(x_1, x_2), d(x_1, x_3))$

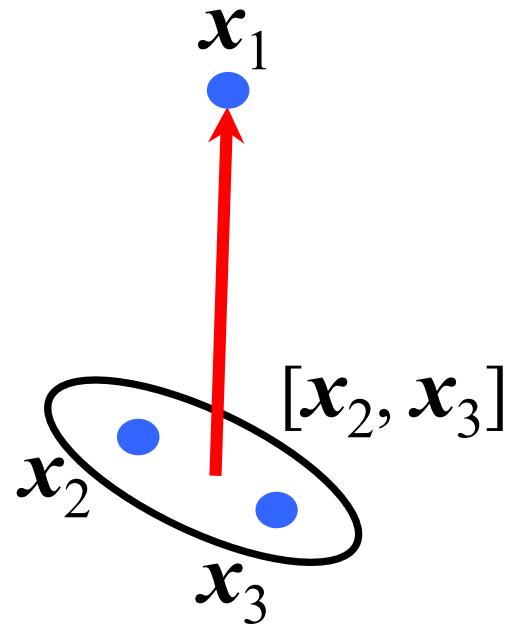


# Hierarchical clustering (9)

- Step 3:

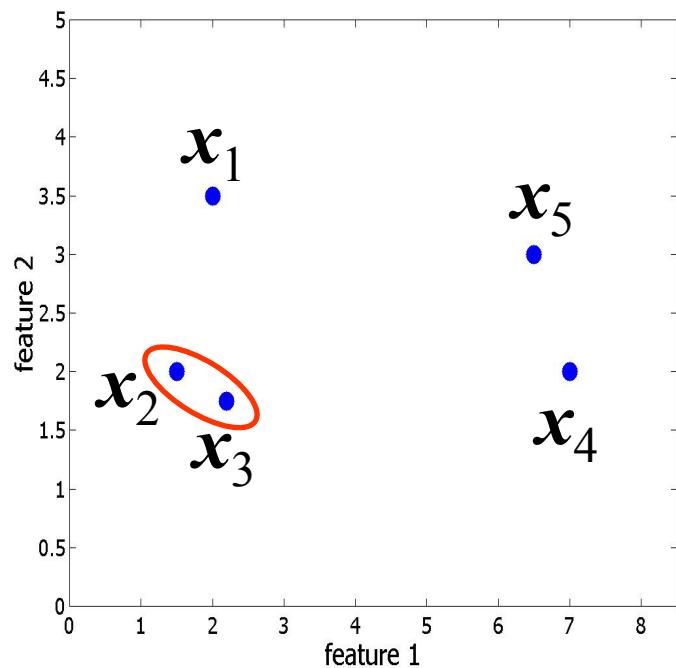
Recompute  $D$  –

**average linkage:**  $d([x_2, x_3], x_1) = \text{mean}(d(x_1, x_2), d(x_1, x_3))$



# Hierarchical clustering (10a)

- Step 3:  
Recompute  $D$  – **single linkage**:



$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	
$x_1$	0.00	1.58	1.76	5.22	4.53
$x_2$		0.00	0.74	5.50	5.10
$x_3$			0.00	4.81	4.48
$x_4$				0.00	1.12
$x_5$					0.00

# Hierarchical clustering (10b)

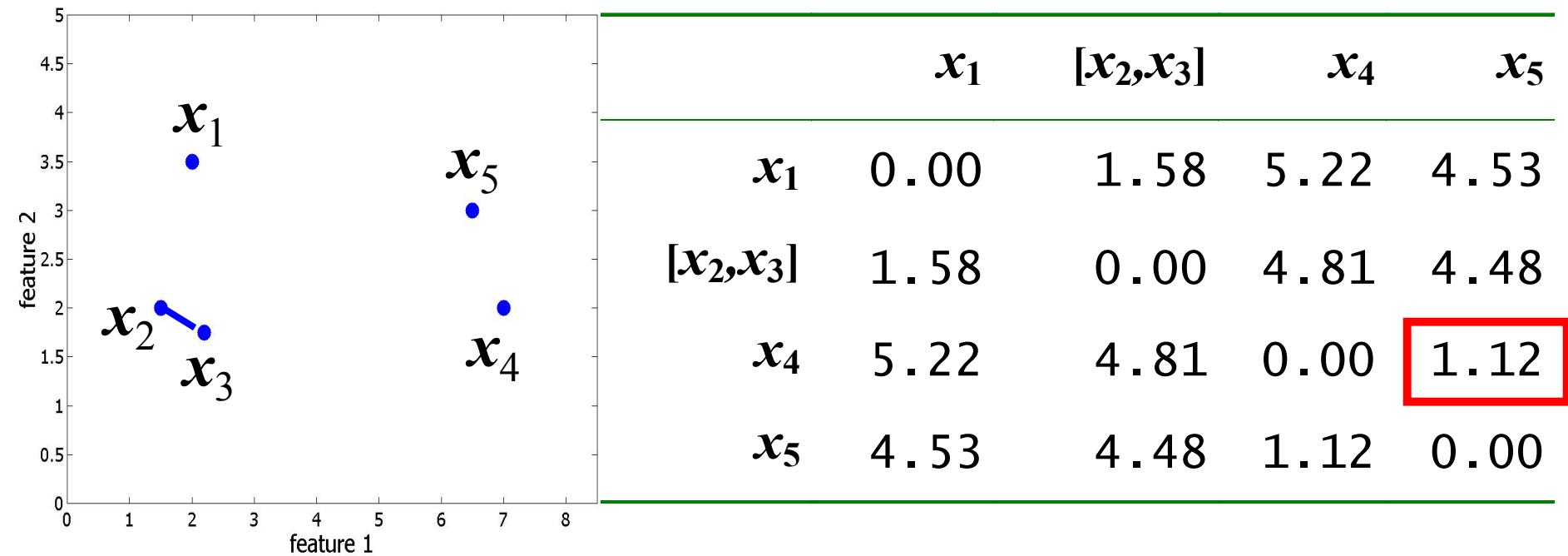
- Step 3:  
Recompute  $D$  – **single linkage**:

	$x_1$	$[x_2, x_3]$	$x_4$	$x_5$
$x_1$	0.00	1.58	5.22	4.53
$[x_2, x_3]$		0.00	4.81	4.48
$x_4$			0.00	1.12
$x_5$				0.00

# Hierarchical clustering (11)

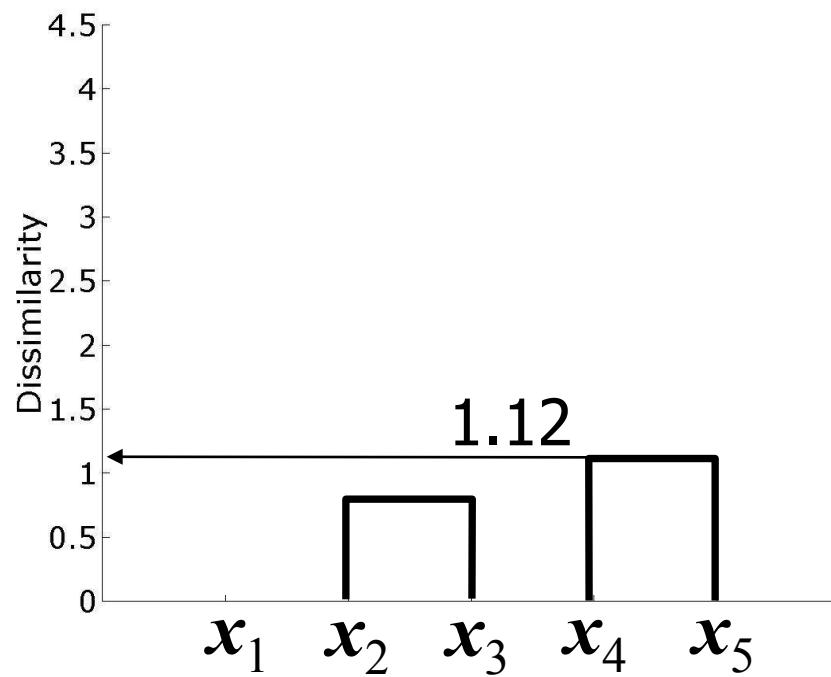
- **Repeat, step 1:**

Find the most similar pair of objects:  $\min_{(i,j)}\{d(i,j)\} = d(4,5)$



# Hierarchical clustering (12)

- **Repeat, step 2:**  
Merge  $x_4$  and  $x_5$  into a single object,  $[x_4, x_5]$ ;



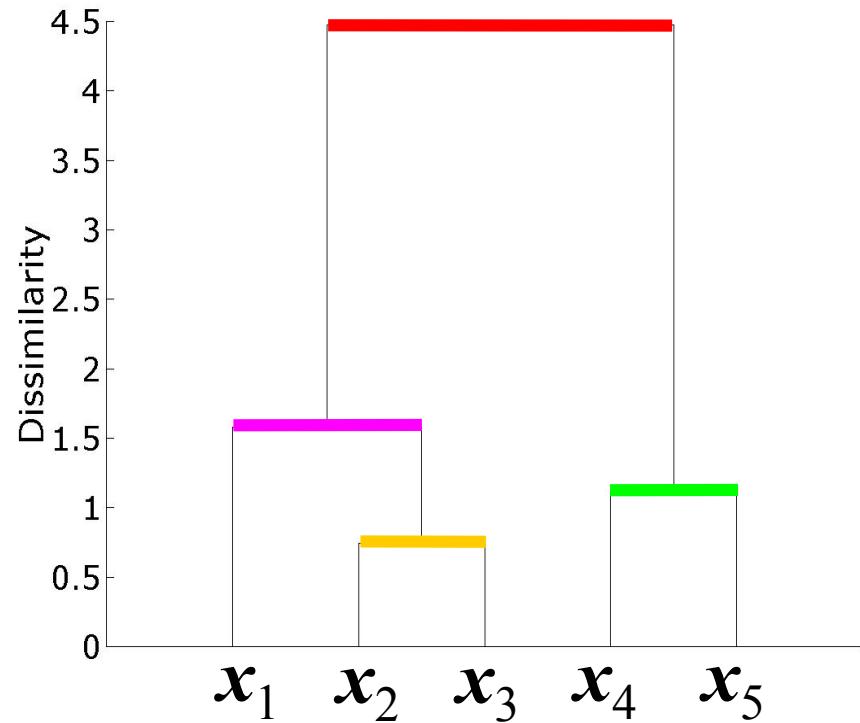
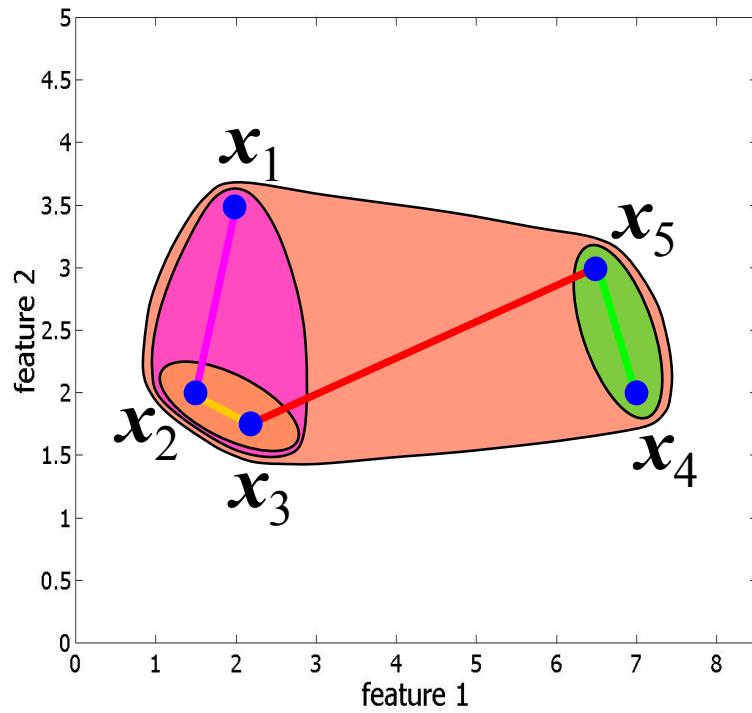
# Hierarchical clustering (13)

- Repeat, step 3:  
Recompute  $D$  (single linkage):

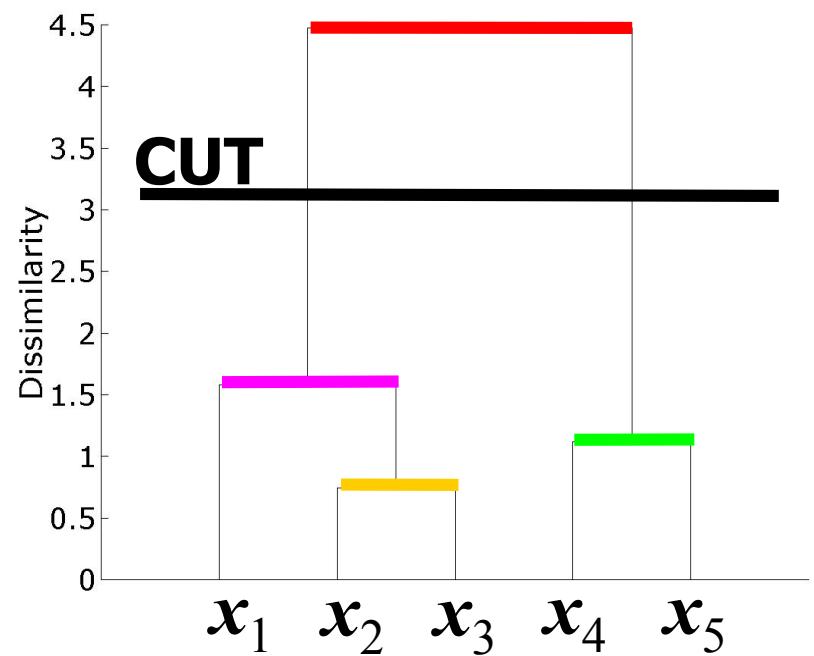
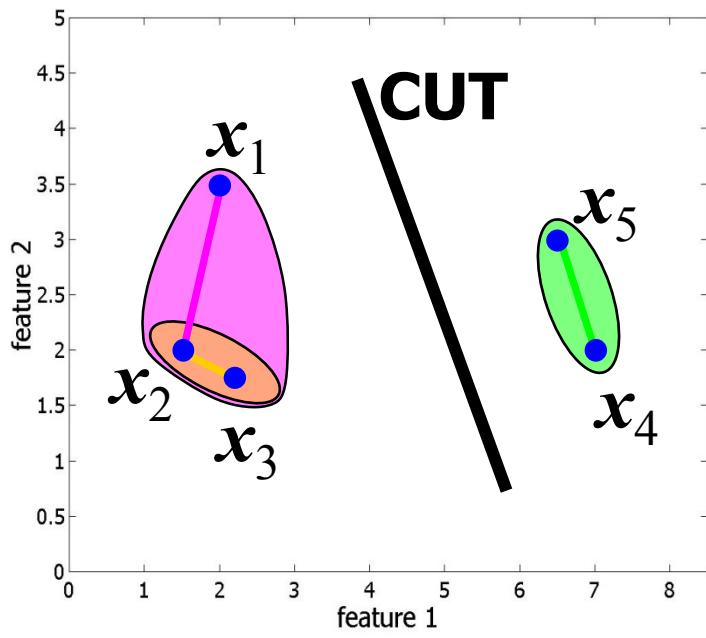
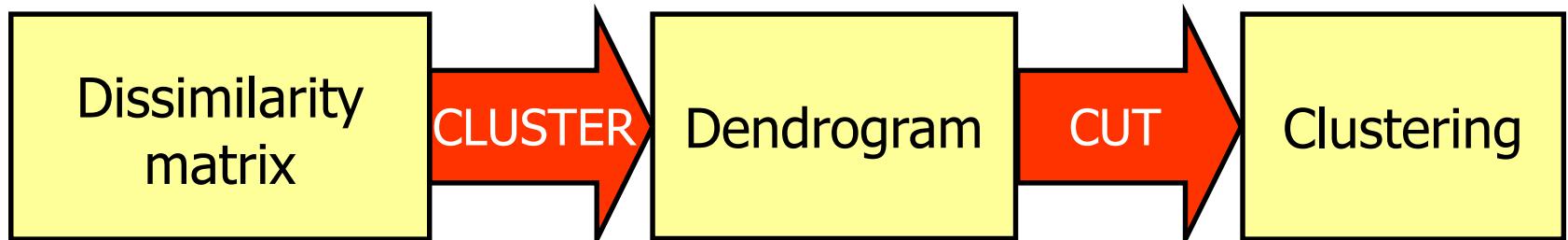
	$x_1$	$[x_2, x_3]$	$[x_4, x_5]$
$x_1$	0.00	1.58	4.53
$[x_2, x_3]$		0.00	4.48
$[x_4, x_5]$			0.00

# Hierarchical clustering (14)

- Repeat steps 1-3 until a single cluster remains

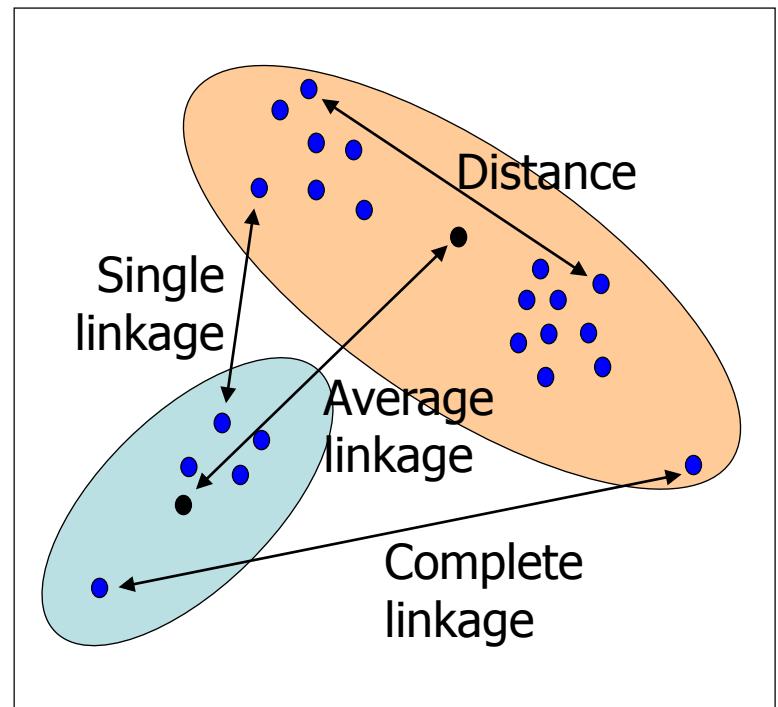


# Hierarchical clustering (15)

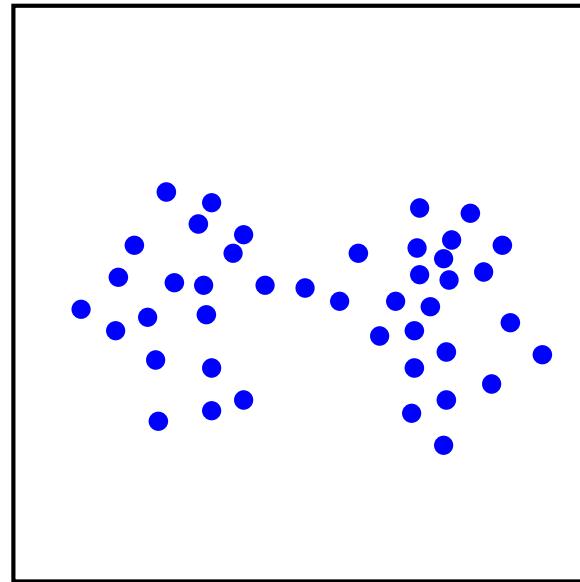
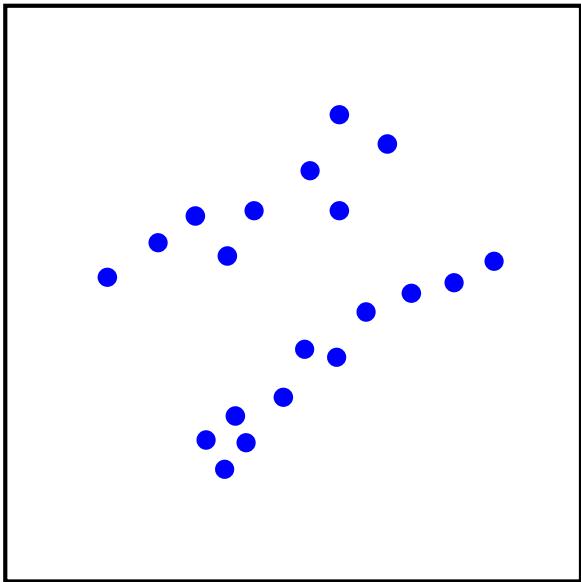


# Hierarchical clustering (16)

- Hierarchical clustering: repeatedly group closest clusters
- Important choices:
  - *Distance measure* between objects: Euclidean, correlation, Hamming, Minkowski, ...
  - *Linkage* between clusters: single, average, complete



# Linkage and cluster shape

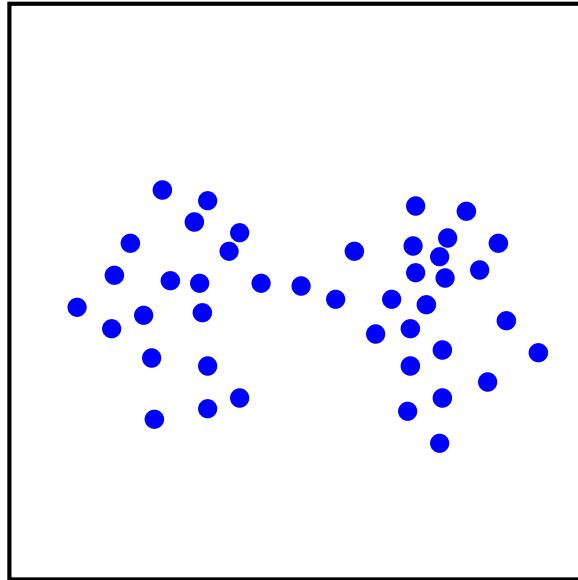
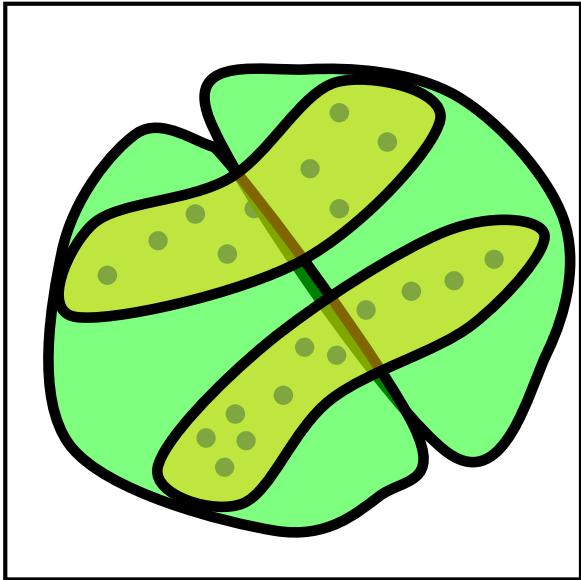


complete linkage



single linkage

# Linkage and cluster shape (2)

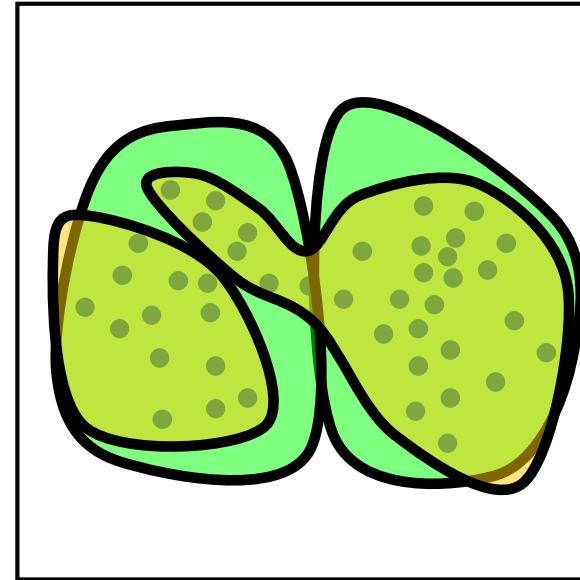
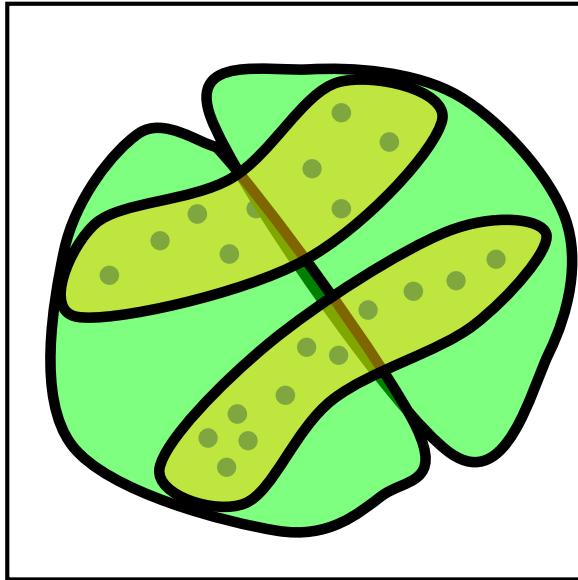


Complete linkage



Single linkage

# Linkage and cluster shape (3)



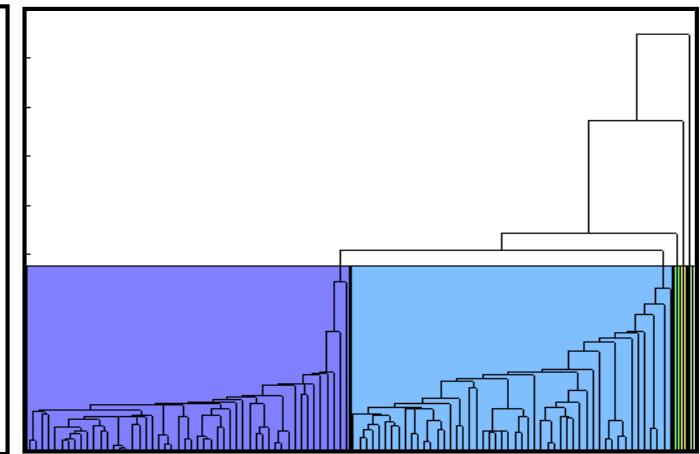
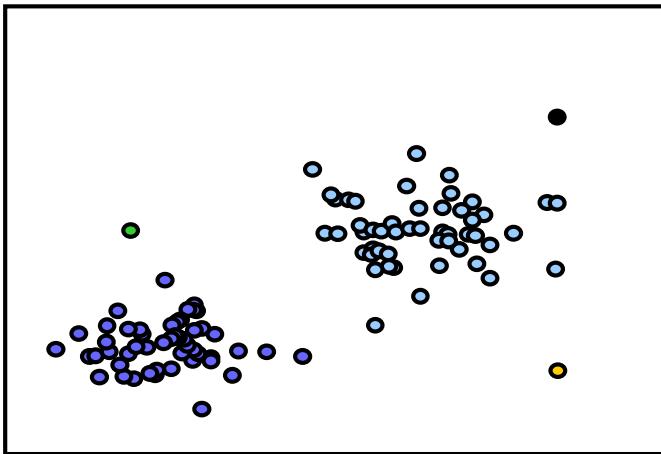
Complete linkage



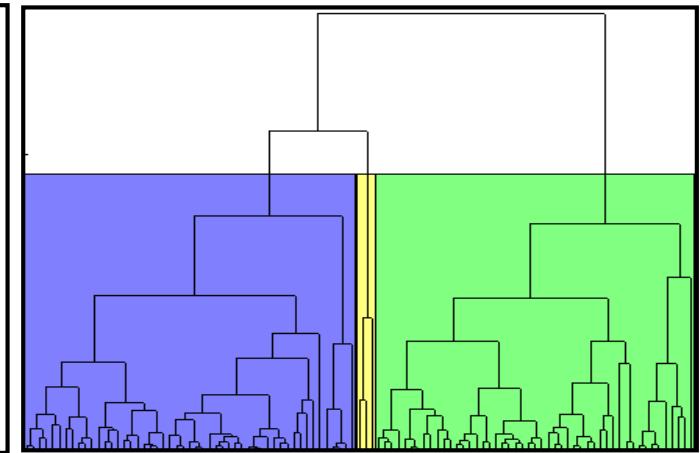
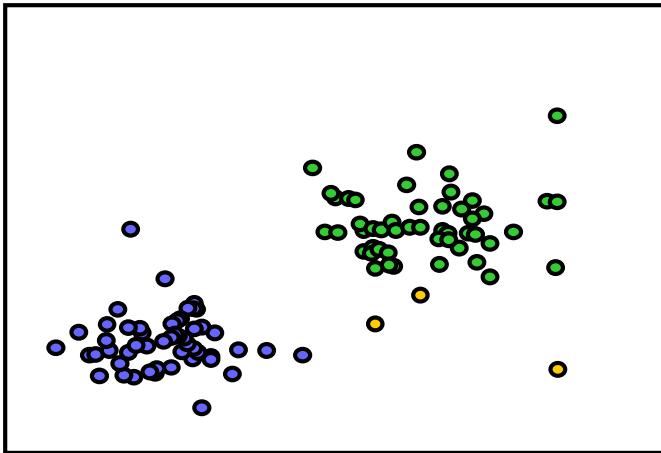
Single linkage

# Linkage and outliers

Single  
linkage

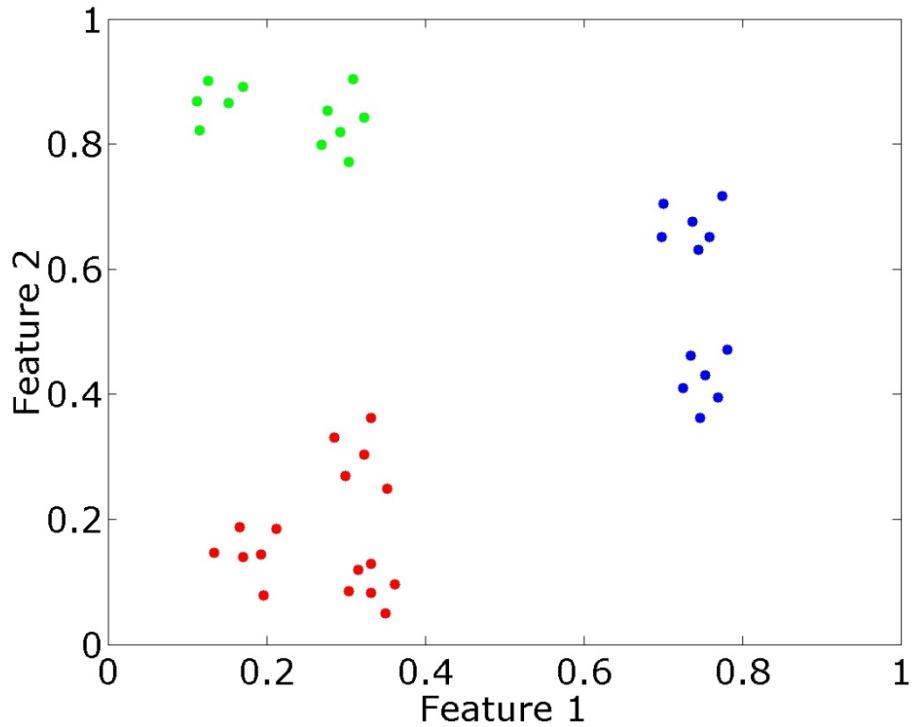
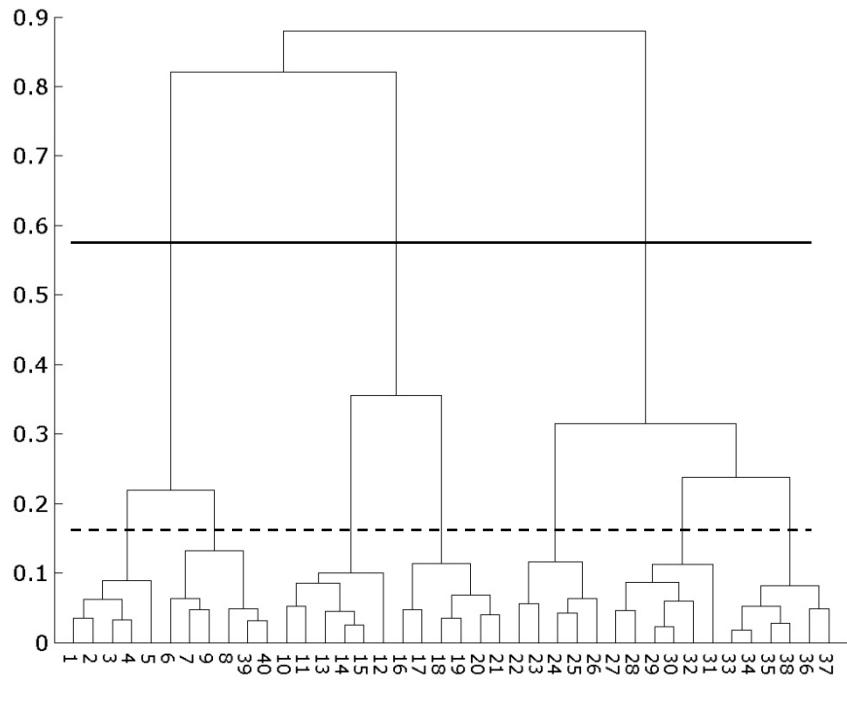


Complete  
linkage



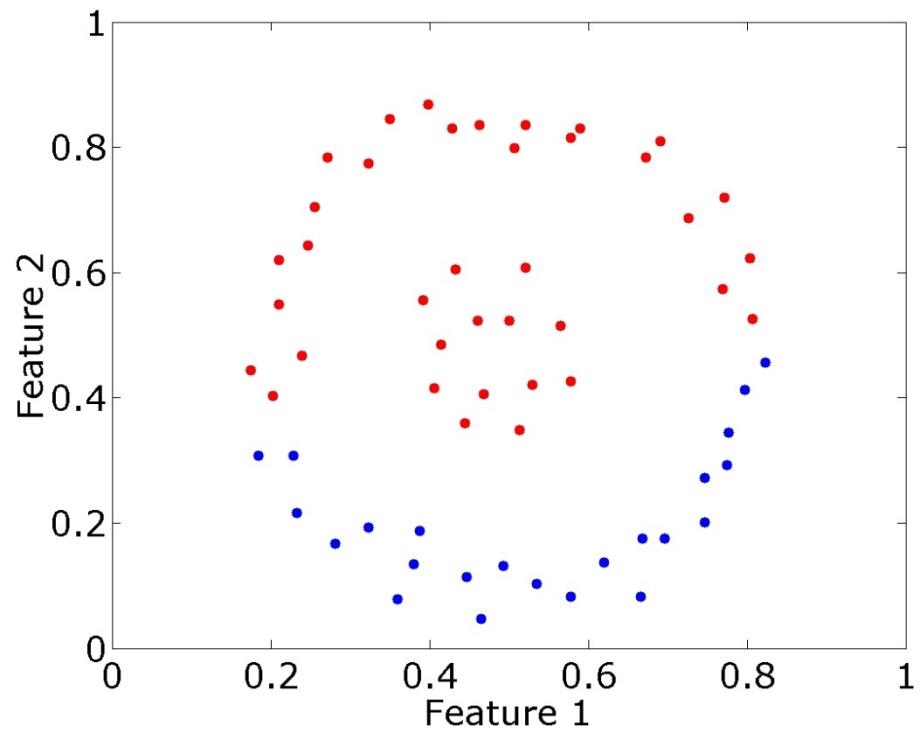
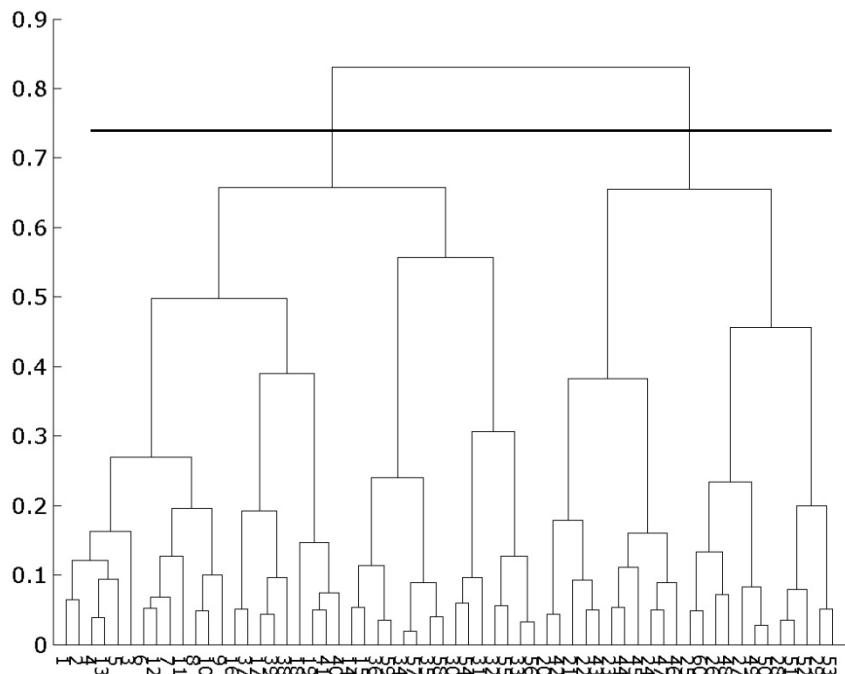
# Hierarchical clustering examples

Euclidean, complete linkage



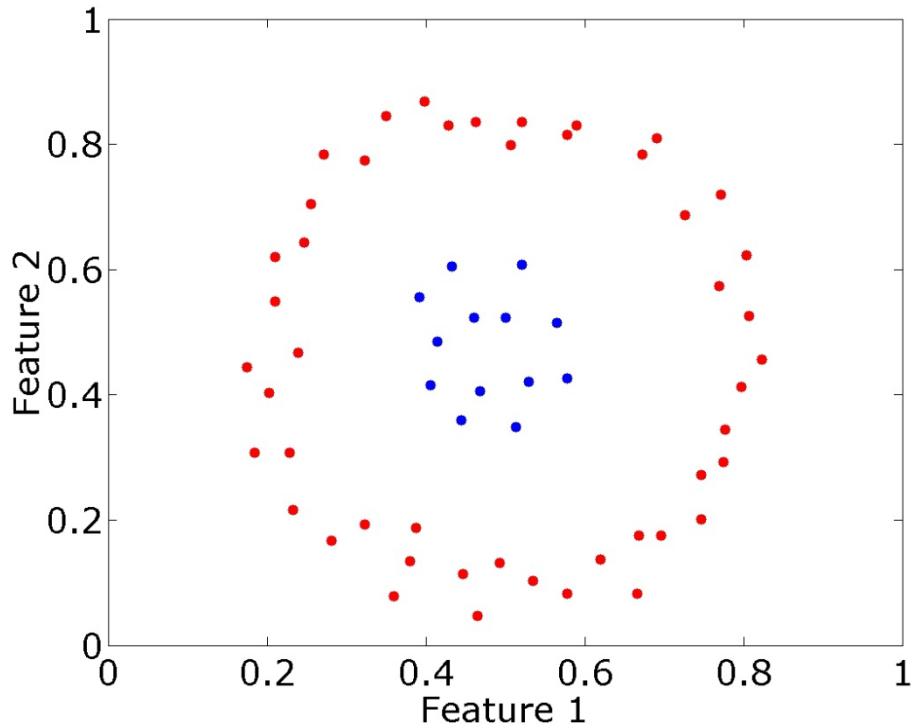
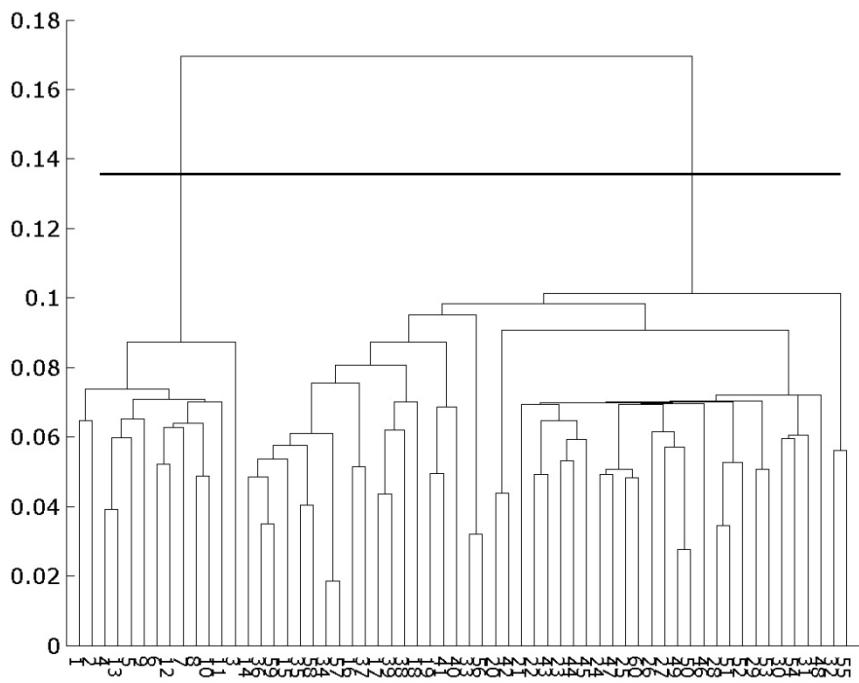
# Hierarchical clustering examples (2)

Euclidean, complete linkage



# Hierarchical clustering examples (3)

Euclidean, single linkage

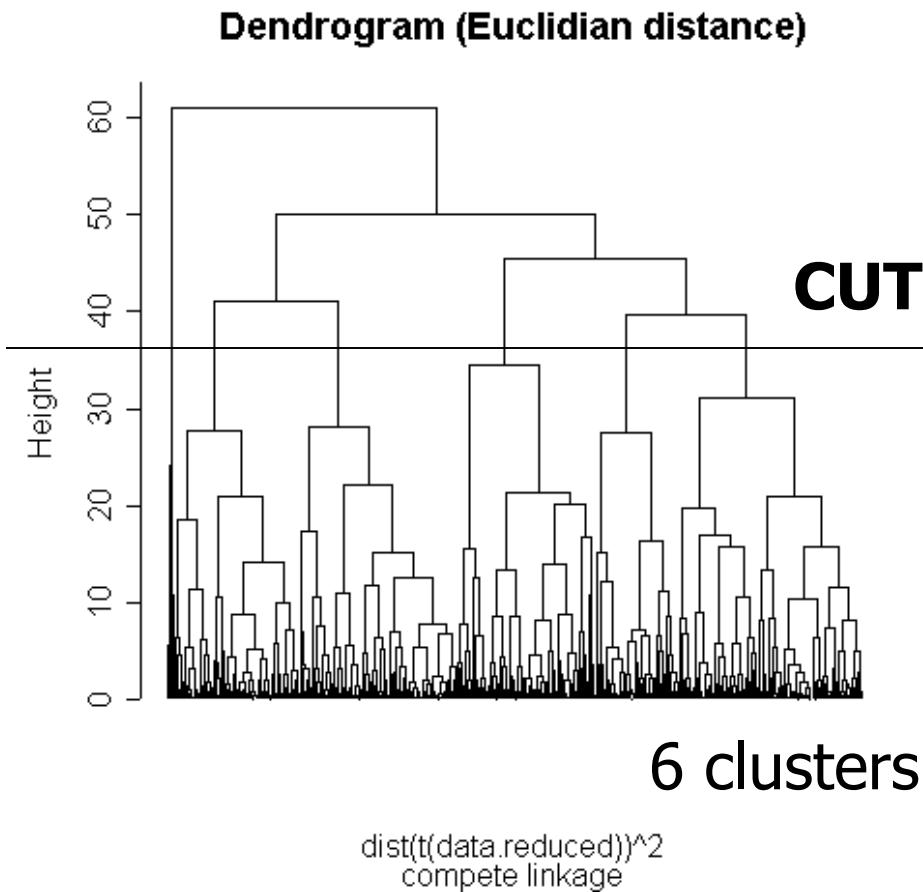


# Hierarchical clustering (17)

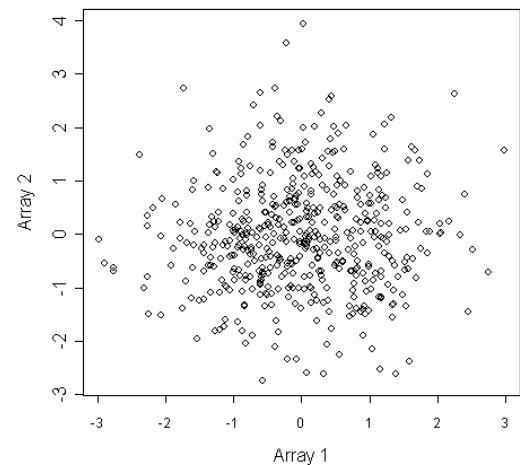
- Advantages:
  - dendrogram gives overview of all possible clusterings
  - linkage type allows to find clusters of varying shapes (convex and non-convex)
  - different dissimilarity measures can be used
- Disadvantages:
  - computationally intensive:  
 $O(n^2)$  in complexity and memory
  - clusterings limited to “hierarchical nestings”

# Hierarchical clustering: warning

- Cluster 500 genes, 5 arrays:



Data were random ...



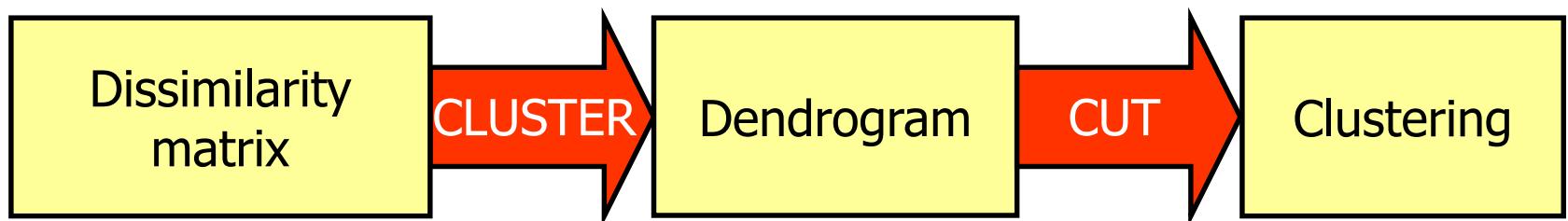
Validation is needed



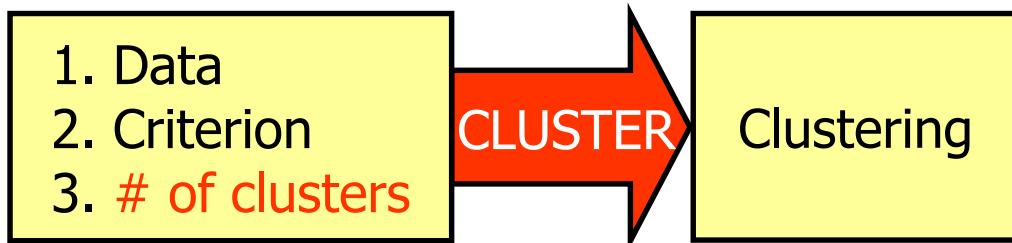
**10min break**  
**Exercise 4.1-4.7**

# Sum-of-squares clustering

- Hierarchical:



- Sum-of-squares:

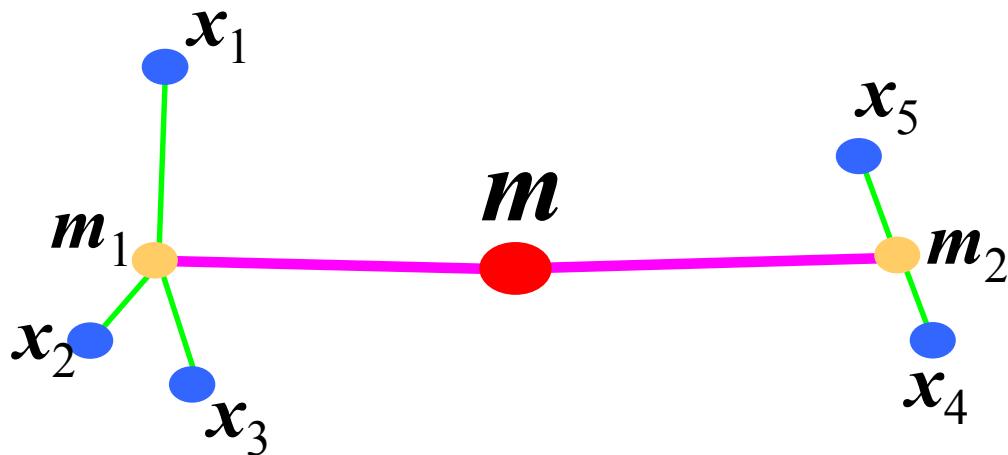


# Sum-of-squares clustering (2)

- Recall from Day 2 (& 3) (Fisher: within and between scatter):

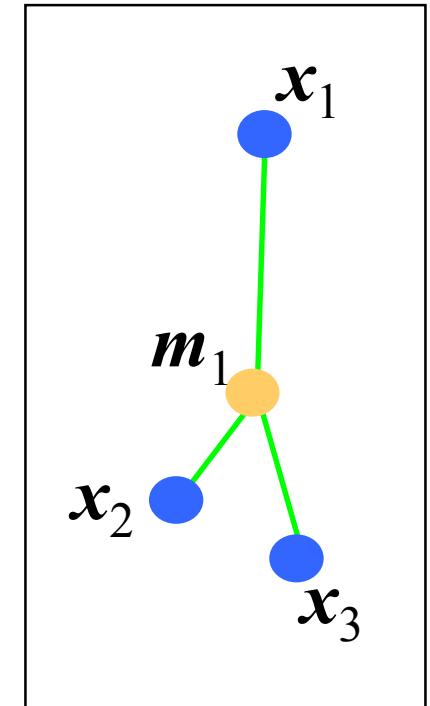
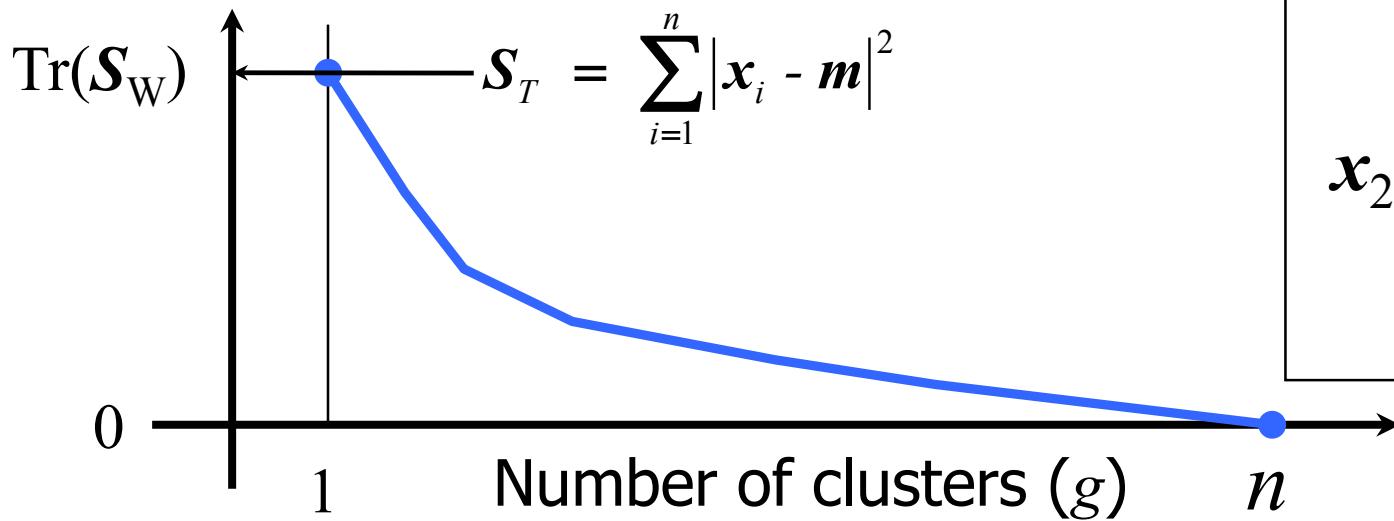
$$S_w = \sum_{i=1}^C \frac{n_i}{n} \Sigma_i \quad (n_1 = 3, n_2 = 2, n = 5, C = 2)$$

$$S_B = \sum_{i=1}^C \frac{n_i}{n} (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T, \quad \mathbf{m} = \sum_{i=1}^C \frac{n_i}{n} \mathbf{m}_i$$



# K-means

- Minimize:  $\text{Tr}(\mathbf{S}_W) = \frac{1}{n} \sum_{j=1}^g \mathbf{S}_j$   
$$\mathbf{S}_j = \sum_{i=1}^{n_j} |\mathbf{x}_i - \mathbf{m}_j|^2$$
  
(sum of per cluster variances)



## K-means (2)

- Iterative procedure to search for  $\min(\text{Tr}(\mathbf{S}_W))$ :
  1. choose number of clusters ( $g$ )
  2. position prototypes ( $m_j, j=1, \dots, g$ ) randomly
  3. assign samples to closest prototype
  4. compute mean of samples assigned to same prototype: new prototype position

Repeat steps 3 and 4 as long as prototypes move

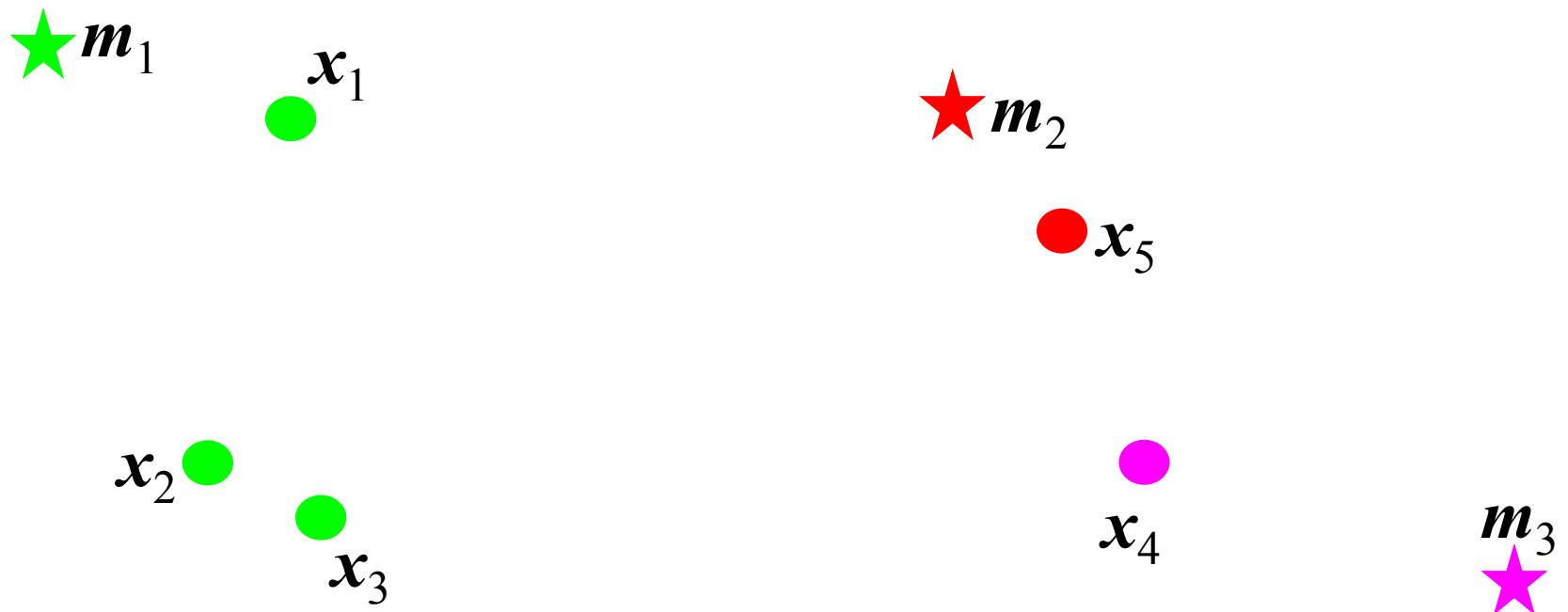
## K-means (3)

- **Step 1:** Choose number of clusters/prototypes
- **Step 2:** Position prototypes randomly



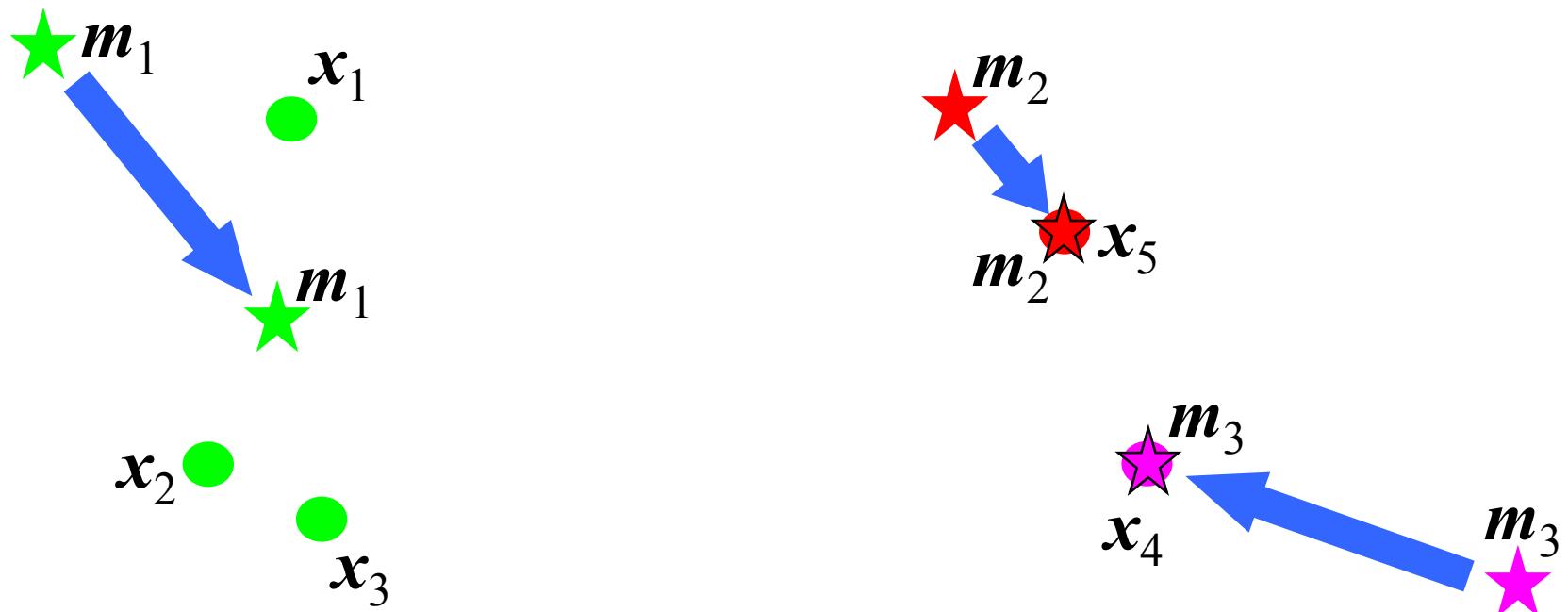
## K-means (4)

- Step 3: Assign samples to closest prototype



## K-means (5)

- **Step 4:** Compute mean of samples assigned to same prototype: new prototype positions



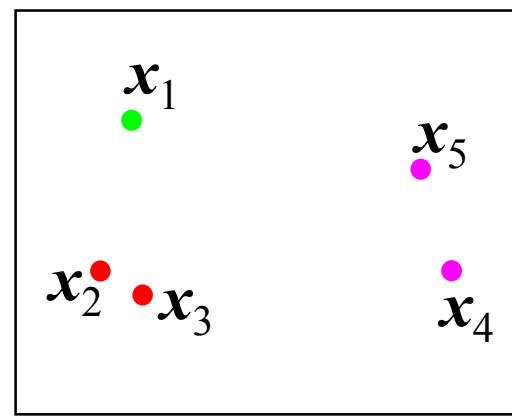
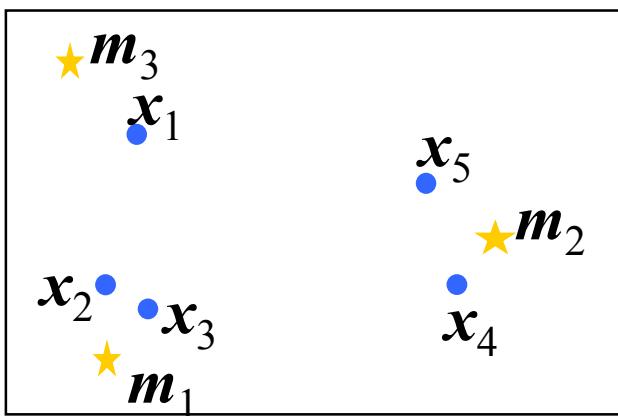
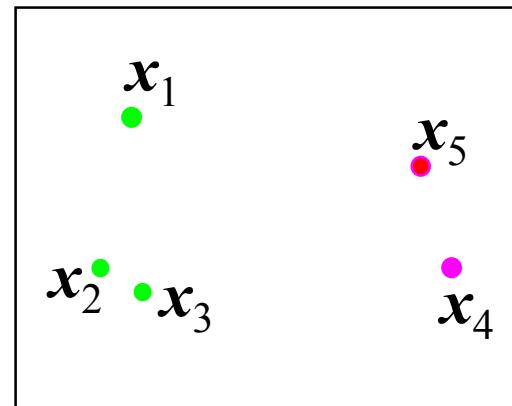
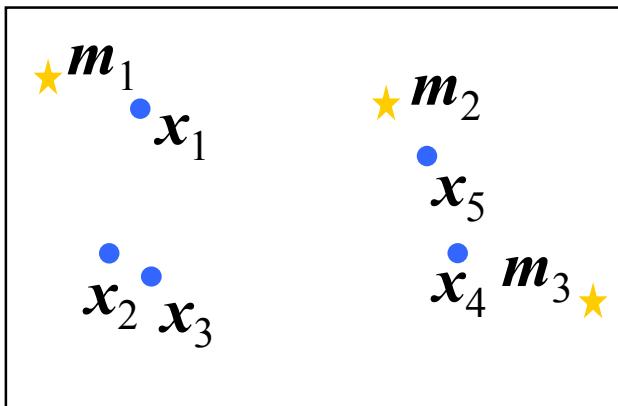
## K-means (6)

- **Repeat** as long as prototype positions change:
  - **Step 3:** Assign samples
  - **Step 4:** Recompute prototype positions



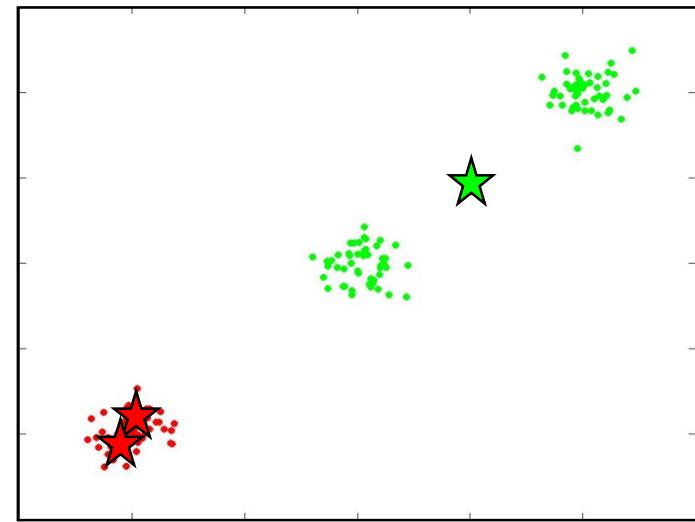
# K-means problems

- Clustering depends on initialization



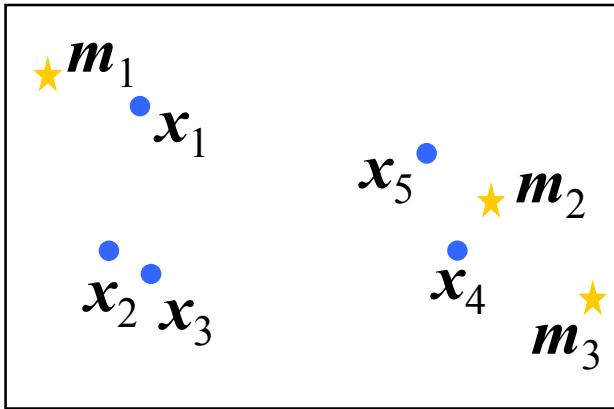
# K-means problems (2)

- Algorithm can get stuck in local minima
- Solution:
  - start from  $I$  different random initialisations
  - keep the best clustering (lowest  $\text{Tr}(S_W)$ )
  - For high-dimensional data, many restarts can be necessary (e.g.  $I = 100$ )



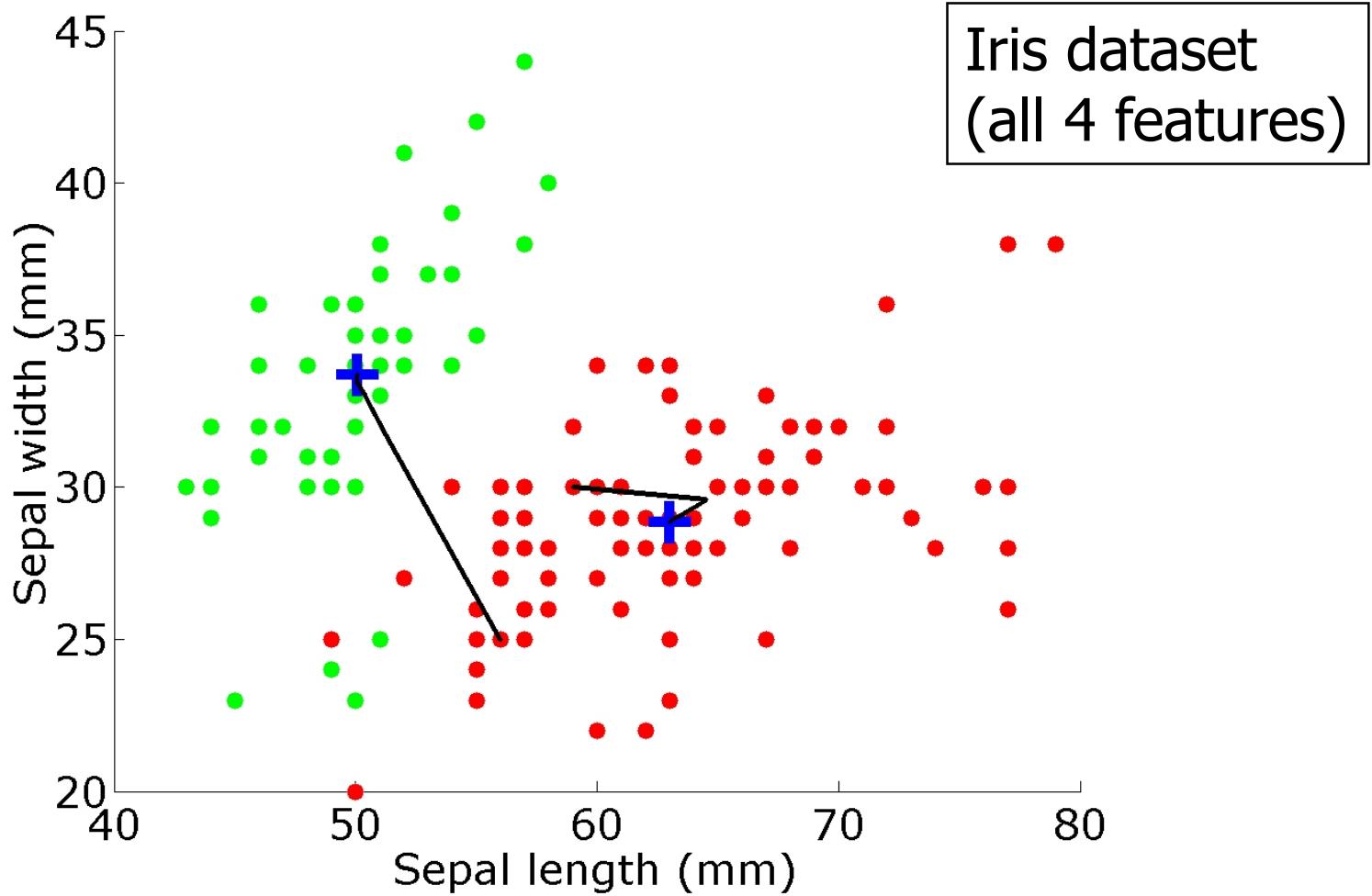
# K-means problems (3)

- Clusters can loose all samples



- Possible solution:
  - remove cluster and continue with  $g - 1$  means
  - alternatively, split largest cluster into two or add a random cluster to continue with  $g$  means

# K-means example



# Advantages/disadvantages: *K*-means

- Disadvantages:
  - Finds only convex clusters (“round shapes”)
  - Sensitive to initialization
  - Can get stuck in local minima
- Advantages:
  - Very simple
  - Fast

# Recapitulation

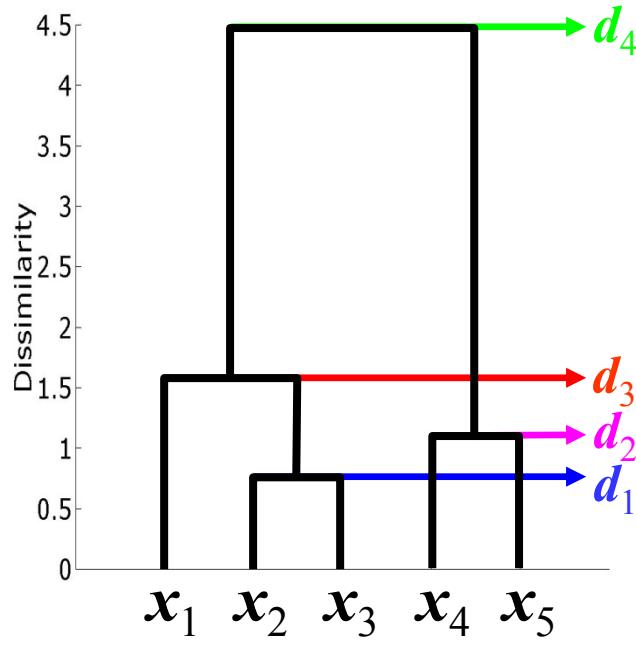
- Clustering is way to detect *natural* groups in data
- What is natural is partly subjective
- We looked at:
  - *Hierarchical clustering*
  - *Sum of squares (k-means)* clustering
- Hierarchical clustering:
  - *dendrogram* shows a complete hierarchy of possible clusterings
  - computationally intensive
- *K*-means
  - fast
  - sensitive to *initialization* and *local minima*

# Cluster validation

- Cluster validation:
  - Checking whether grouping is really present
  - Choosing the optimal number of clusters
- A difficult problem – the ground truth is not known (since we do not know the object labels)!
- Methods:
  - Distortion measures:
    - Does clustering approximate structure in data?
  - Validity measures:
    - Davies-Bouldin index
  - Fusion graph
  - Gap statistic

# Distortion measures

- How well does a dendrogram capture structure in data?



$d^*$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_1$	0	$d_3$	$d_3$	$d_4$	$d_4$
$x_2$		0	$d_1$	$d_4$	$d_4$
$x_3$			0	$d_4$	$d_4$
$x_4$				0	$d_2$
$x_5$					0

# Distortion measures (2)

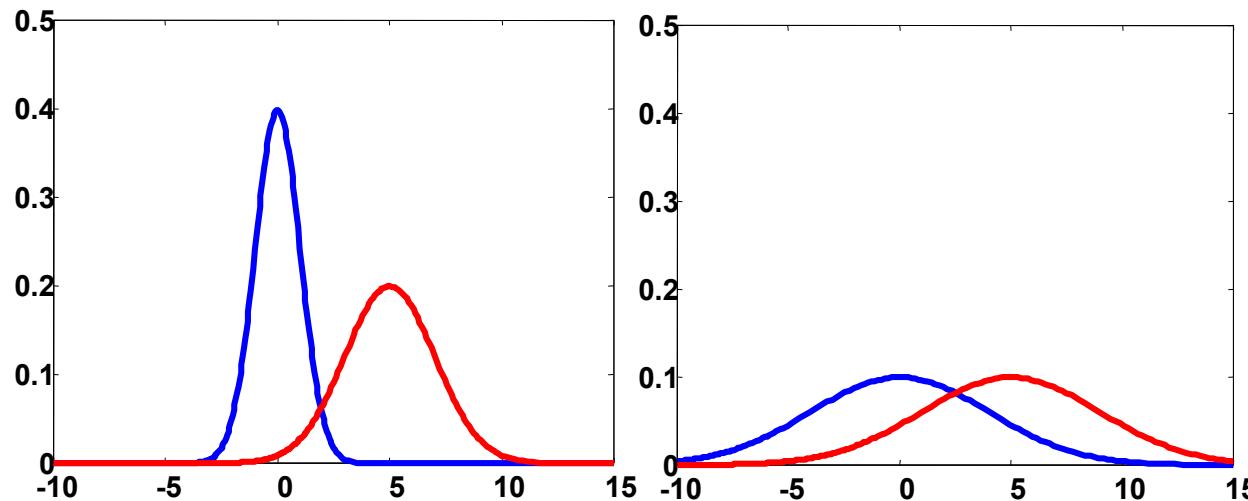
- Measure of distortion: Pearson correlation of  $d$  and  $d^*$

$$\rho(d, d^*) = \frac{\text{cov}(d, d^*)}{\sqrt{\text{var}(d)\text{var}(d^*)}} \in [-1, 1]$$

$d$						$d^*$					
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$		$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_1$	0 . 00	1 . 58	1 . 76	5 . 22	4 . 53		0	$d_3$	$d_3$	$d_4$	$d_4$
$x_2$		0 . 00	0 . 74	5 . 50	5 . 10			0	$d_1$	$d_4$	$d_4$
$x_3$			0 . 00	4 . 81	4 . 48				0	$d_4$	$d_4$
$x_4$				0 . 00	1 . 12					0	$d_2$
$x_5$					0 . 00						0

# Validity measures

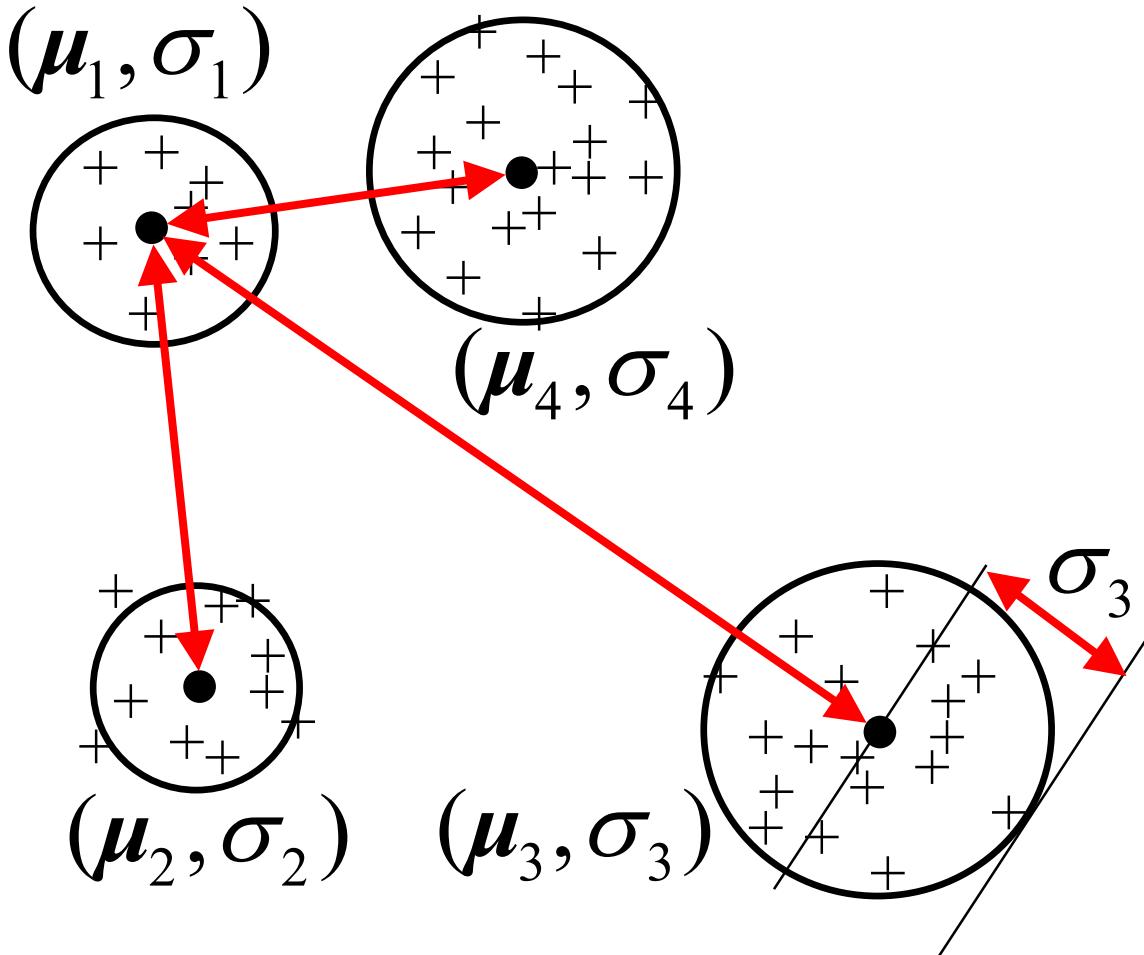
- Many are based on within and between group scatter
- The larger the between group scatter and the smaller the within group scatter, the better
- Example: Davies-Bouldin



# Davies-Bouldin index

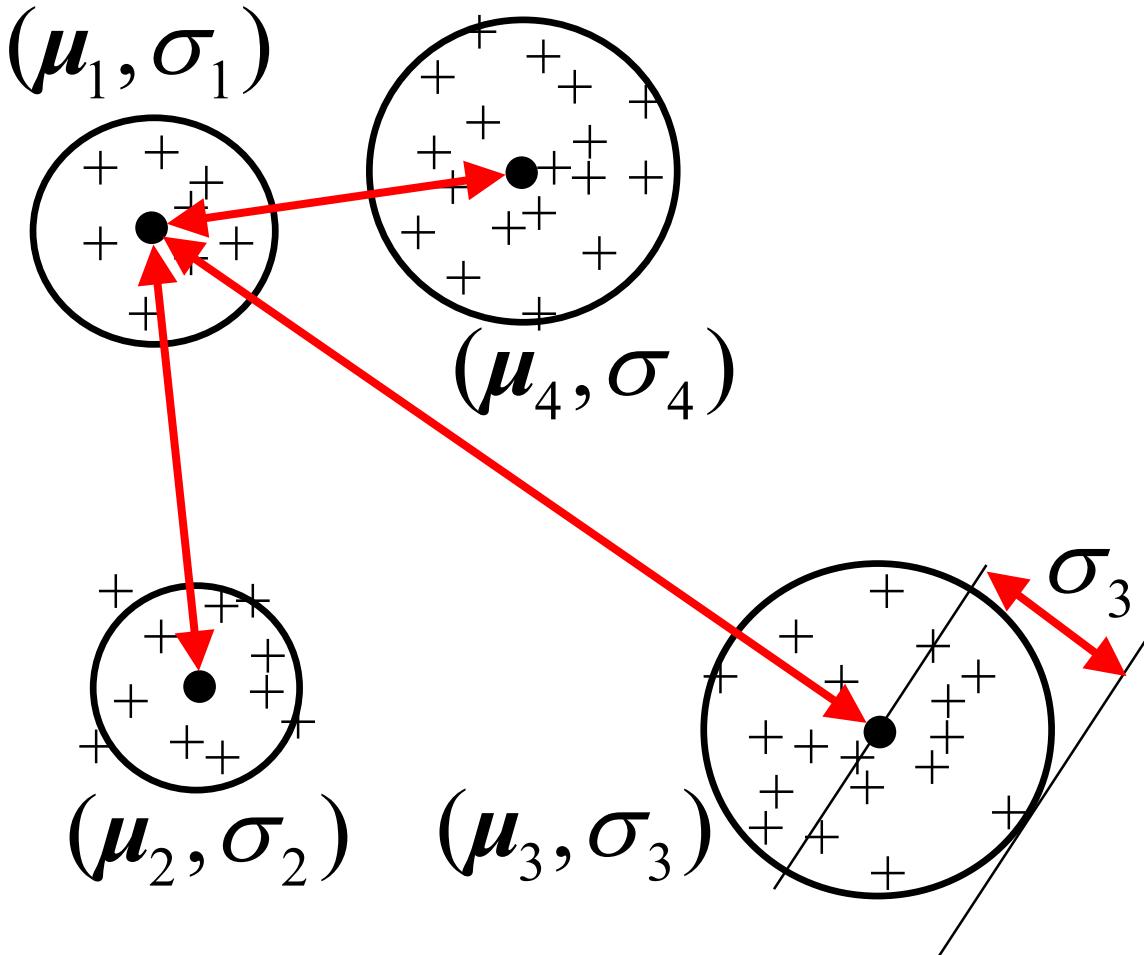
- Assumption: clusters are spherical
- For a good clustering, it should hold that:
  - objects are compactly organized within a cluster
  - clusters are far apart
- D.L. Davies and D.W. Bouldin, IEEE Transactions on Pattern Analysis and Machine Intelligence 1, pp. 224-227, 1979

# Davies-Bouldin index (2)



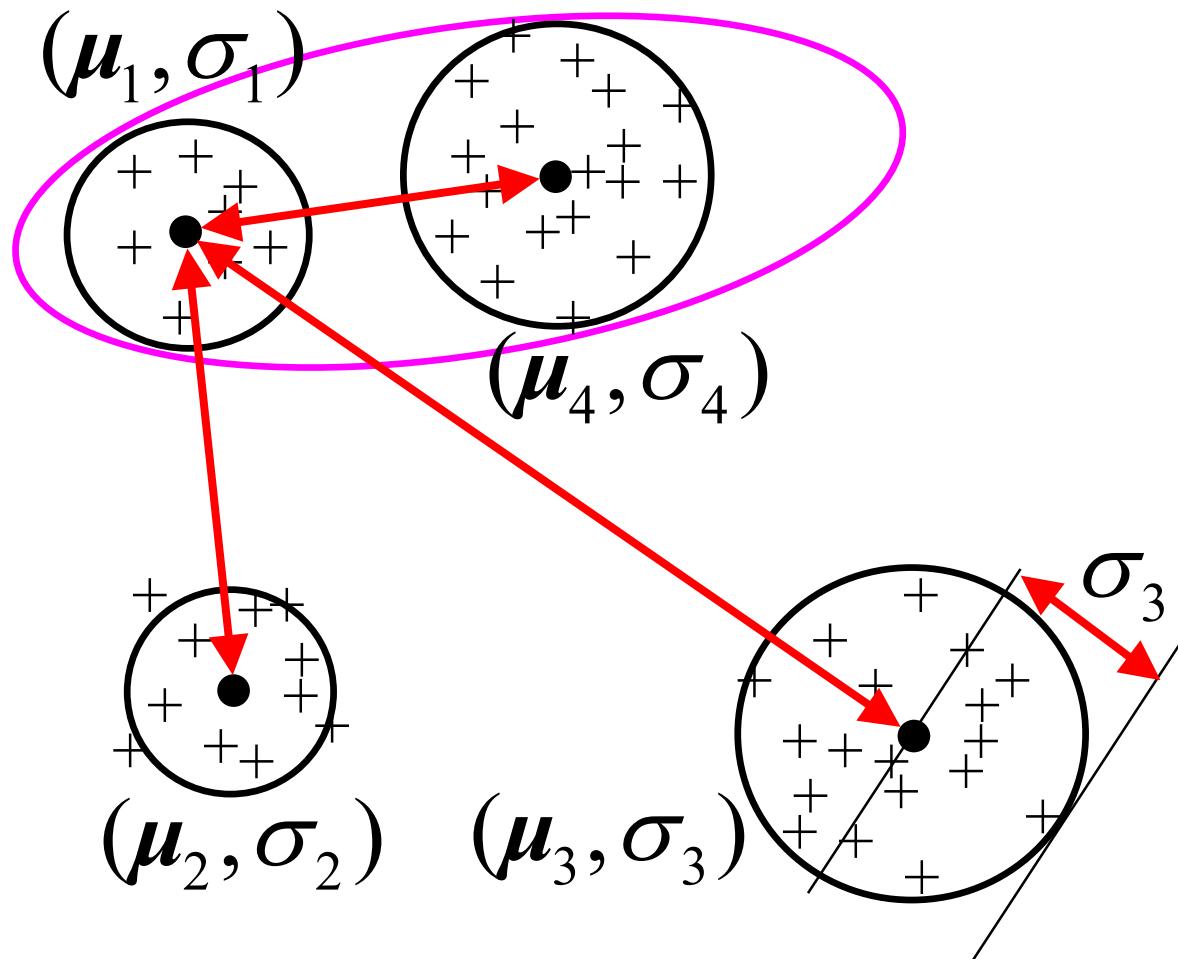
$$\sigma_j = \sqrt{\frac{1}{n_j} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2}$$
$$\mu_j = \frac{1}{n_j} \sum_{x_i \in C_j} x_i$$

# Davies-Bouldin index (3)



$$R_{jk} = \frac{\sigma_j + \sigma_k}{\|\boldsymbol{\mu}_j - \boldsymbol{\mu}_k\|}$$
$$\sigma_j = \sqrt{\frac{1}{n_j} \sum_{x_i \in C_j} \|x_i - \boldsymbol{\mu}_j\|^2}$$
$$\boldsymbol{\mu}_j = \frac{1}{n_j} \sum_{x_i \in C_j} x_i$$

# Davies-Bouldin index (4)



$$R_{jk} = \frac{\sigma_j + \sigma_k}{\|\mu_j - \mu_k\|}$$
$$R_j = \max_{k=1,..g; k \neq j} R_{jk}$$

## Davies-Bouldin index (5)

$$R_{jk} = \frac{\sigma_j + \sigma_k}{\|\boldsymbol{\mu}_j - \boldsymbol{\mu}_k\|}$$

$$R_j = \max_{k=1,..g; k \neq j} R_{jk}$$

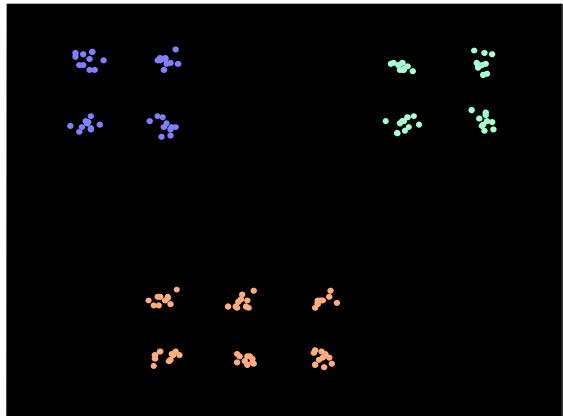
$$I_{DB} = \frac{1}{g} \sum_{j=1}^g R_j$$

Paired cluster criterion

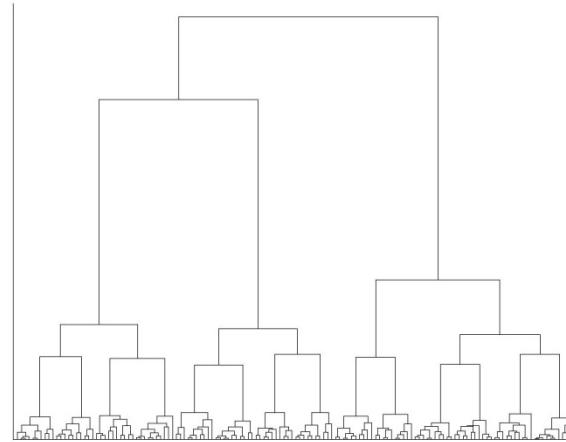
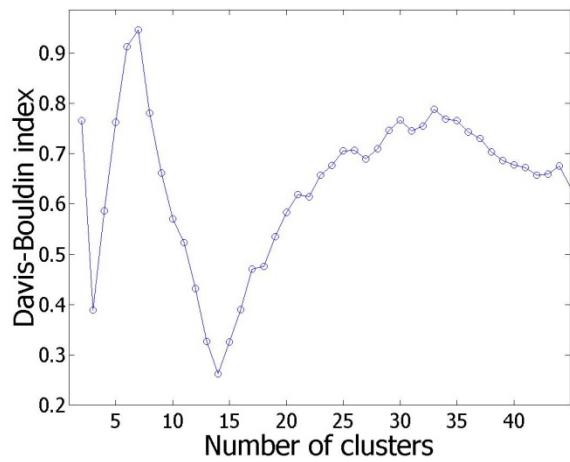
Worst-case value per cluster

Average worst-case

# Davies-Bouldin index (5)



Dataset

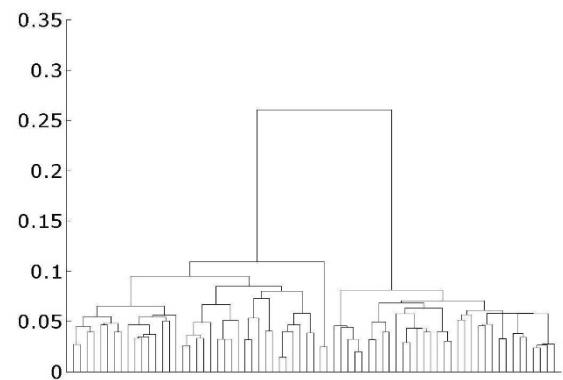
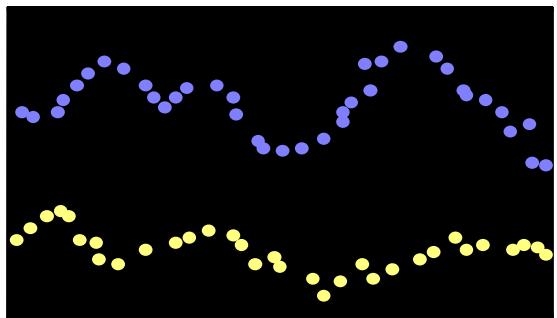


Complete link

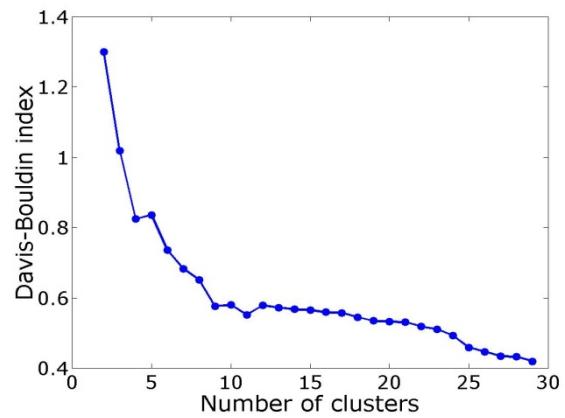
Davies-Bouldin:  
3 or 14 clusters

# Davies-Bouldin index (7)

Single link

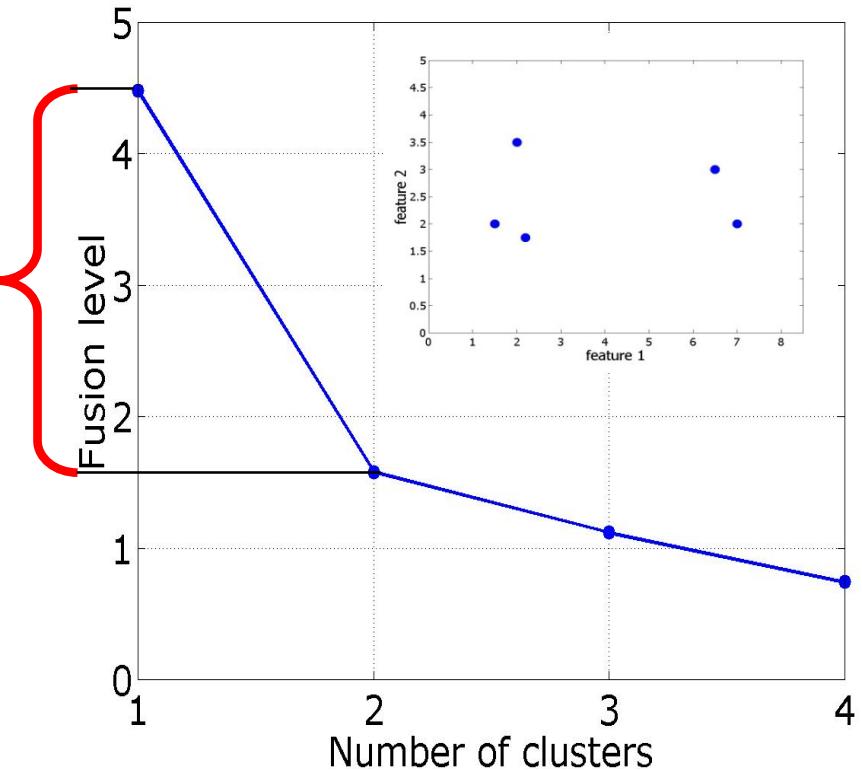
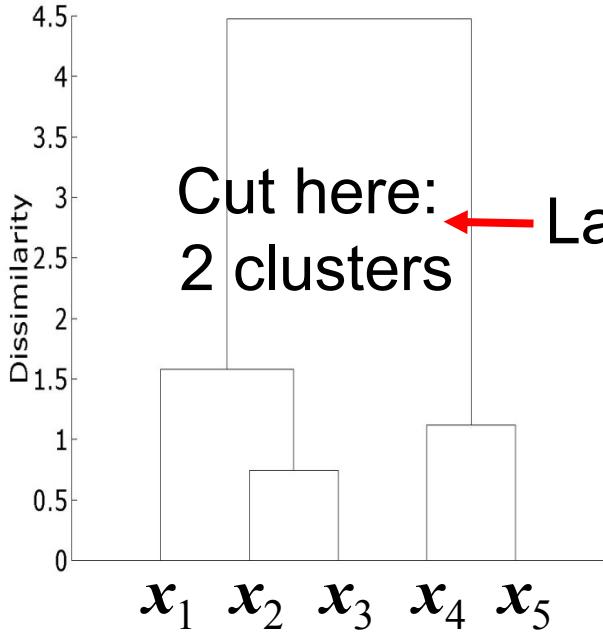


Davies-Bouldin:



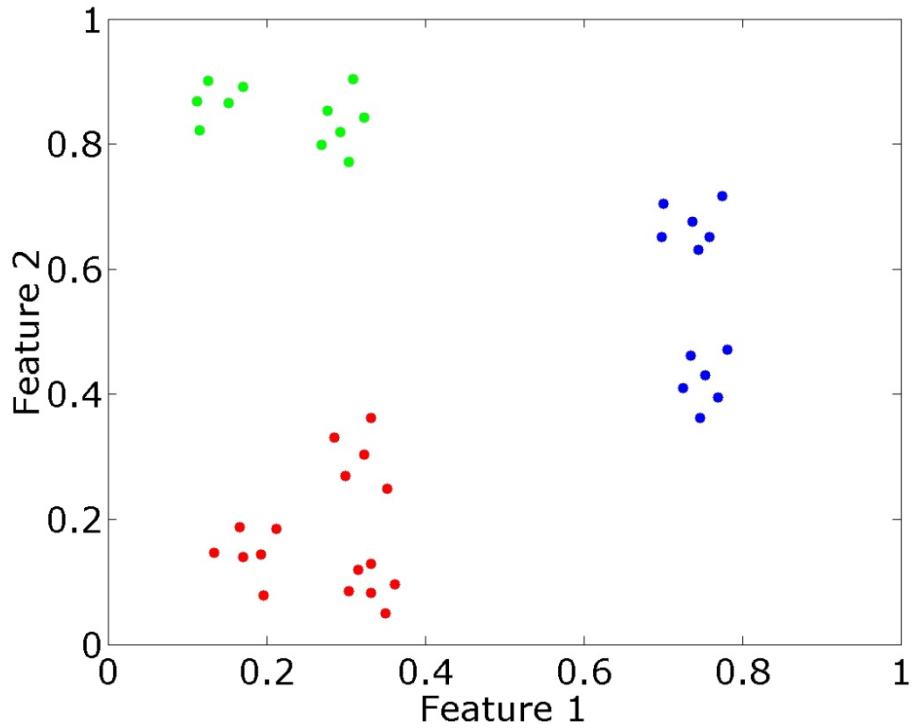
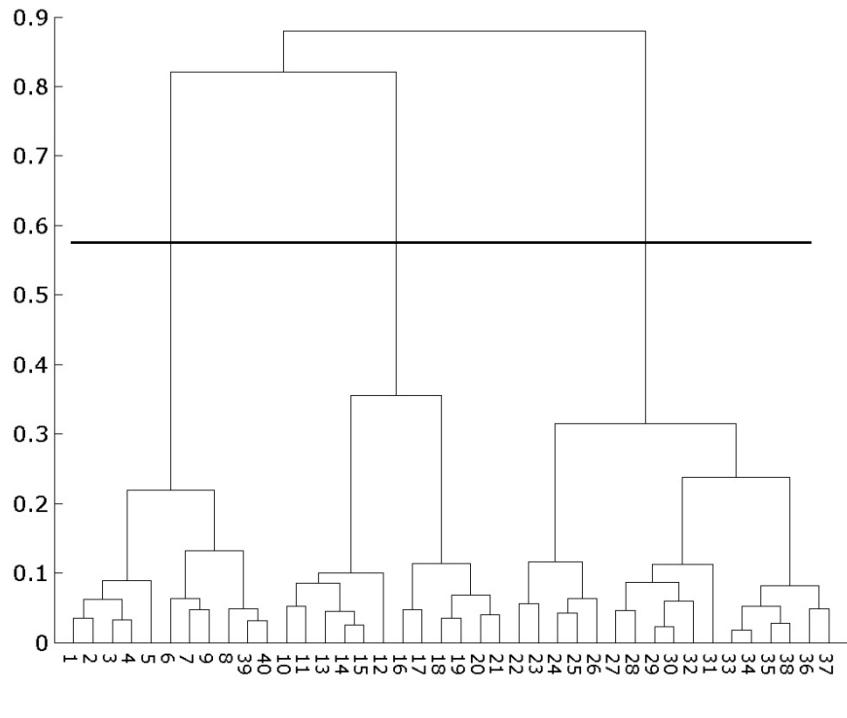
# Fusion graph

- Heuristic approach: fusion level



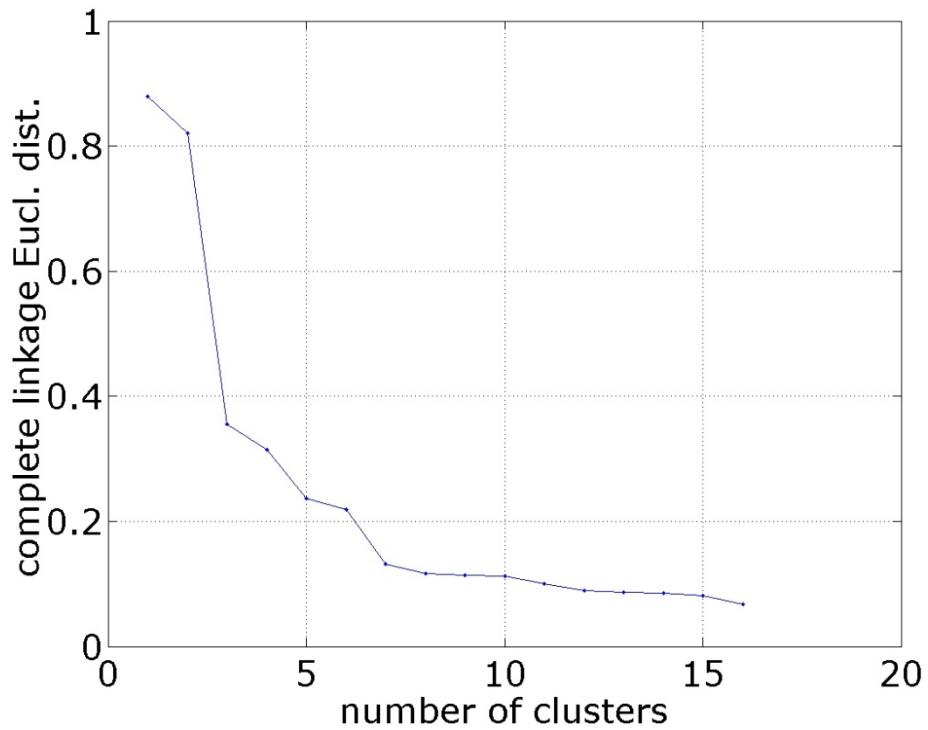
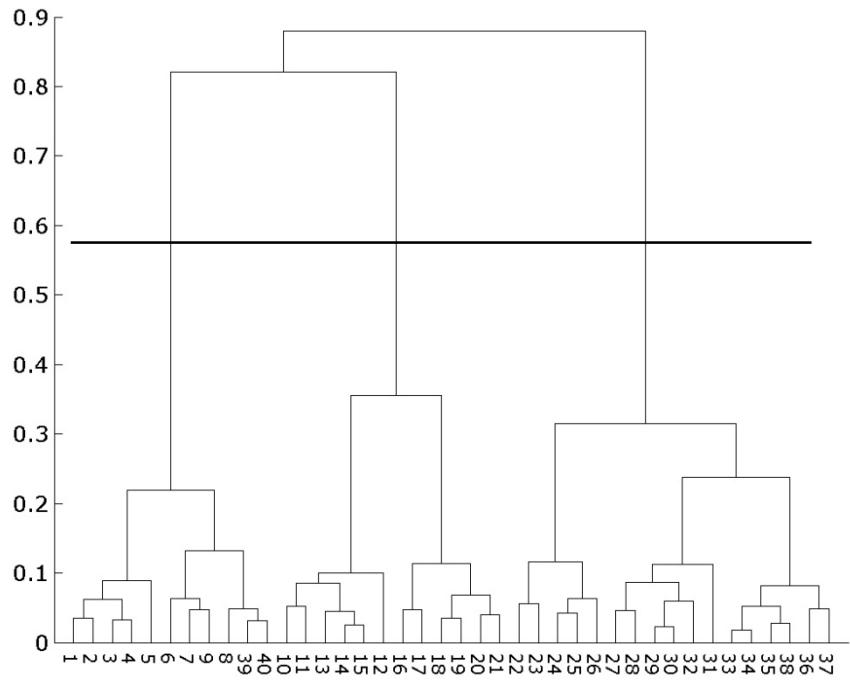
# Fusion graph (2)

(Euclidean; complete linkage)



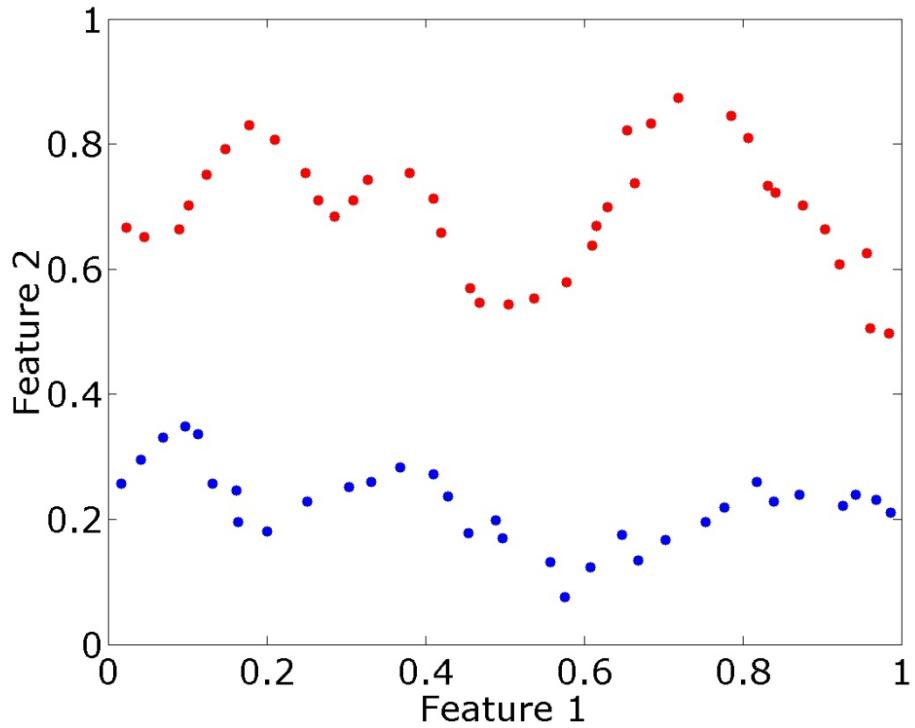
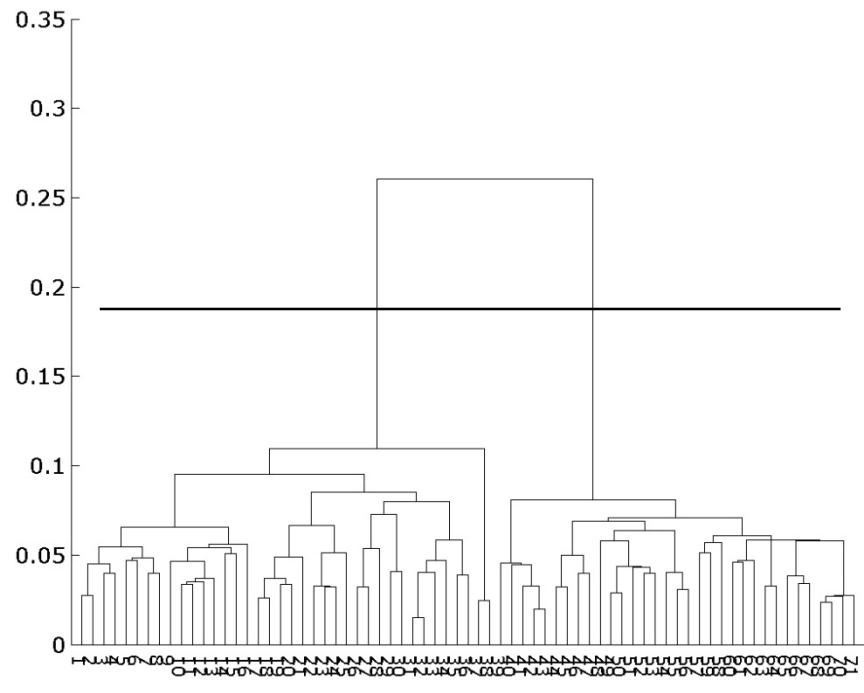
# Fusion graph (3)

(Euclidean; complete linkage)



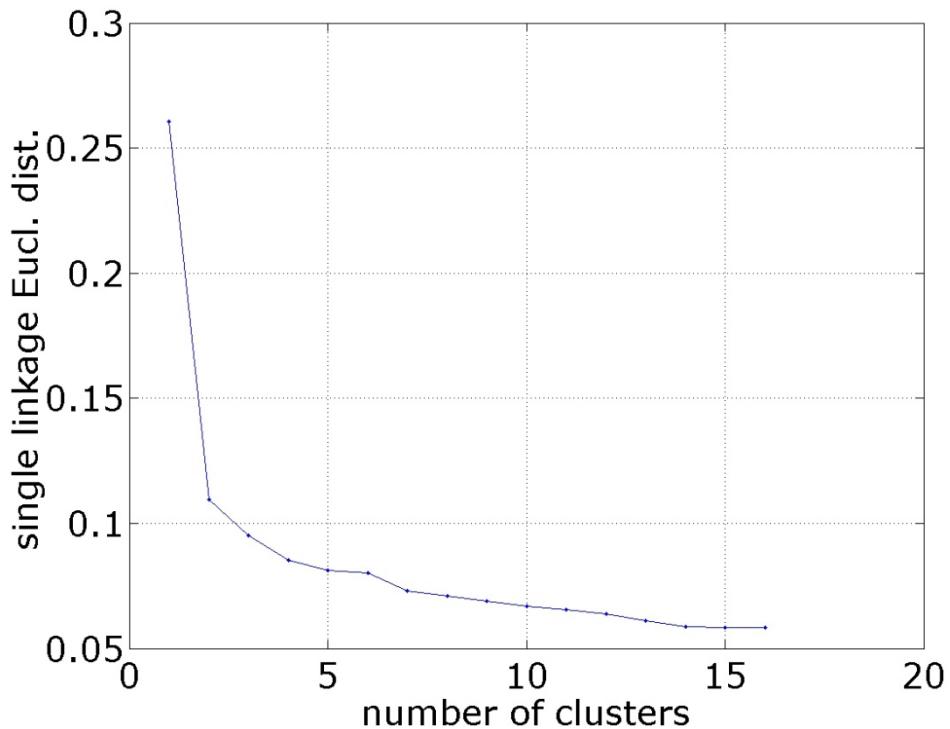
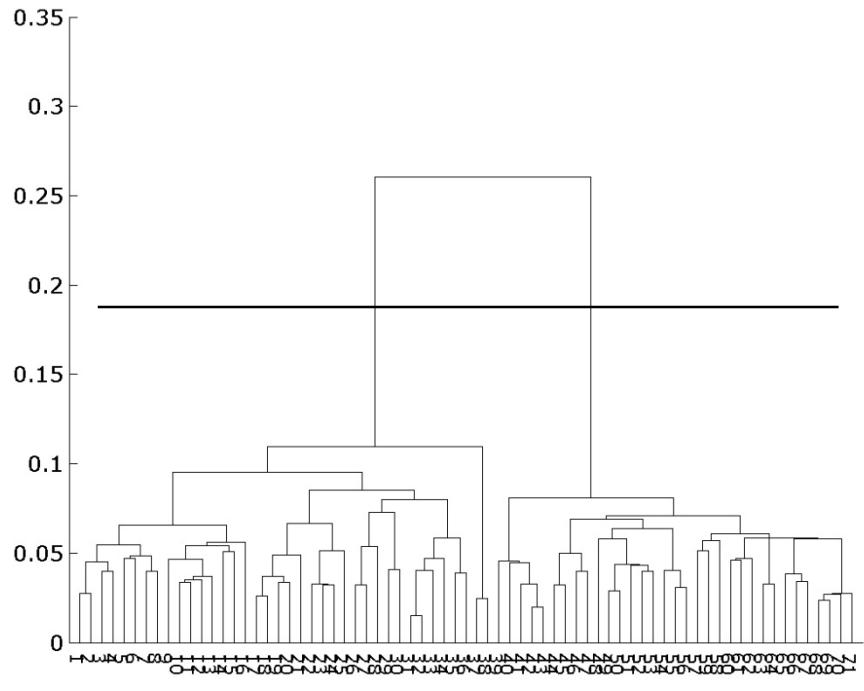
# Fusion graph (4)

(Euclidean; single linkage)



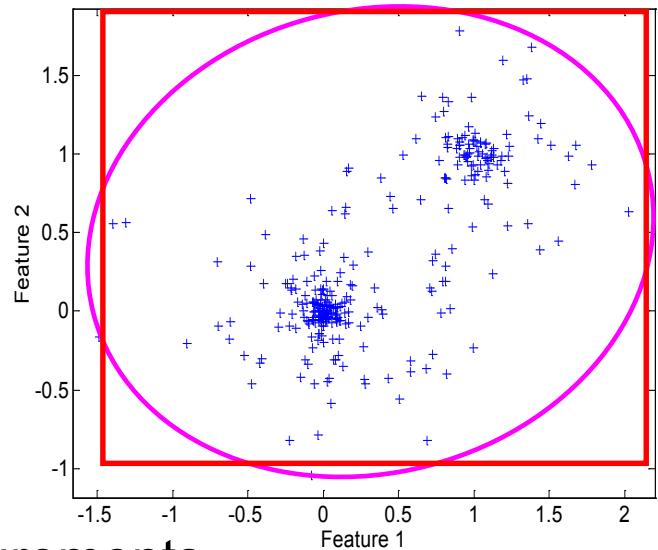
# Fusion graph (5)

(Euclidean; single linkage)



# What is a large jump?

- Compare the fusion graph of the dataset with a *null hypothesis*, i.e. a dataset where the clustering structure has been destroyed
- Different approaches:
  - Generate random data within bounding box or convex hull of data;
  - Preferable to shuffle data, i.e. not generate new data, but perturb relationships between measurements
  - For example, randomly match feature values, i.e. permute values within columns



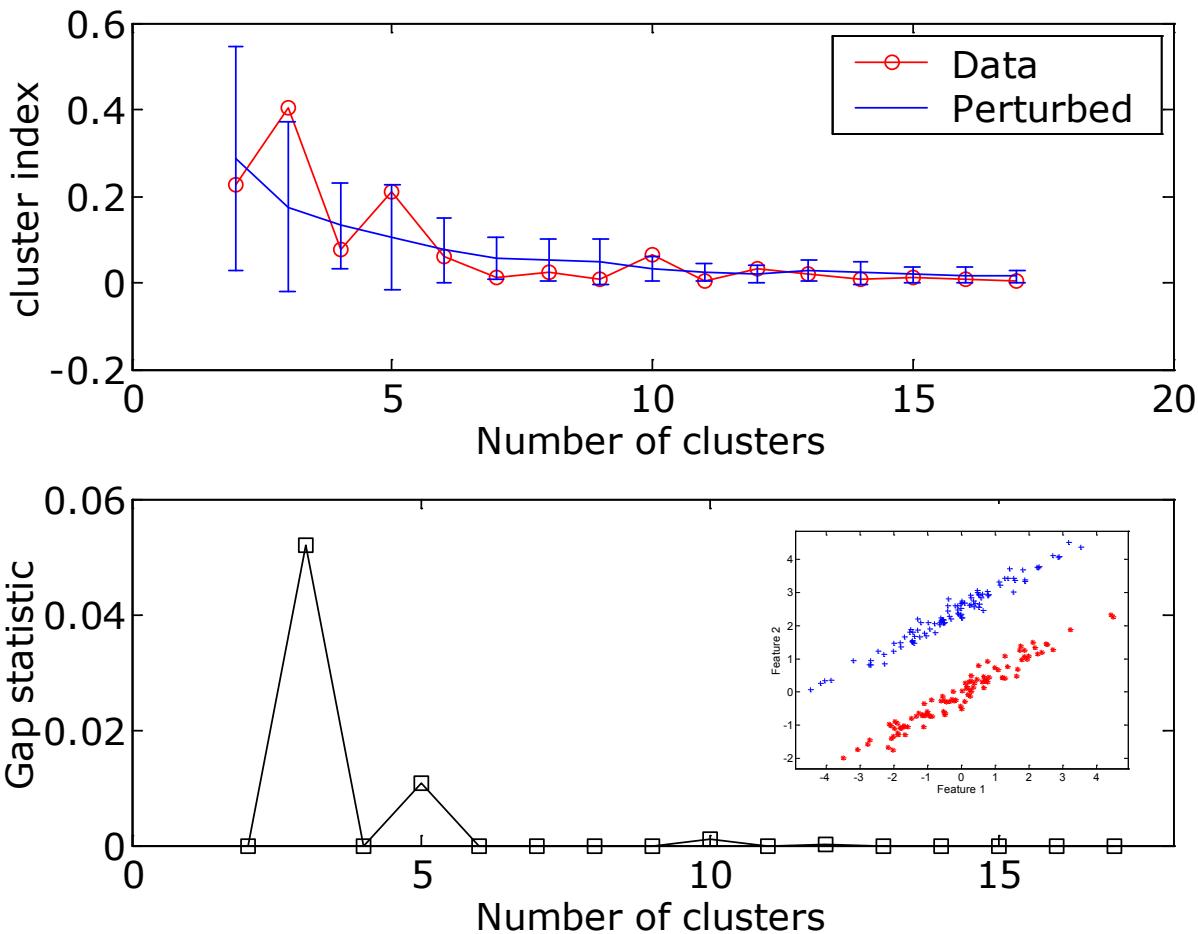
# The gap statistic

1. Generate dendrogram and extract fusion graph,  $f_j$
2. Repeat  $r$  times
  1. Perturb columns
  2. Generate dendrogram and fusion graph,  $f_{j,r}^*$
3. Compute average  $\mu_j^*$  and standard deviation  $\sigma_j^*$  of these perturbed graphs
4. Compute the difference between the data fusion graph and the average perturbed fusion graph (*gap statistic*):

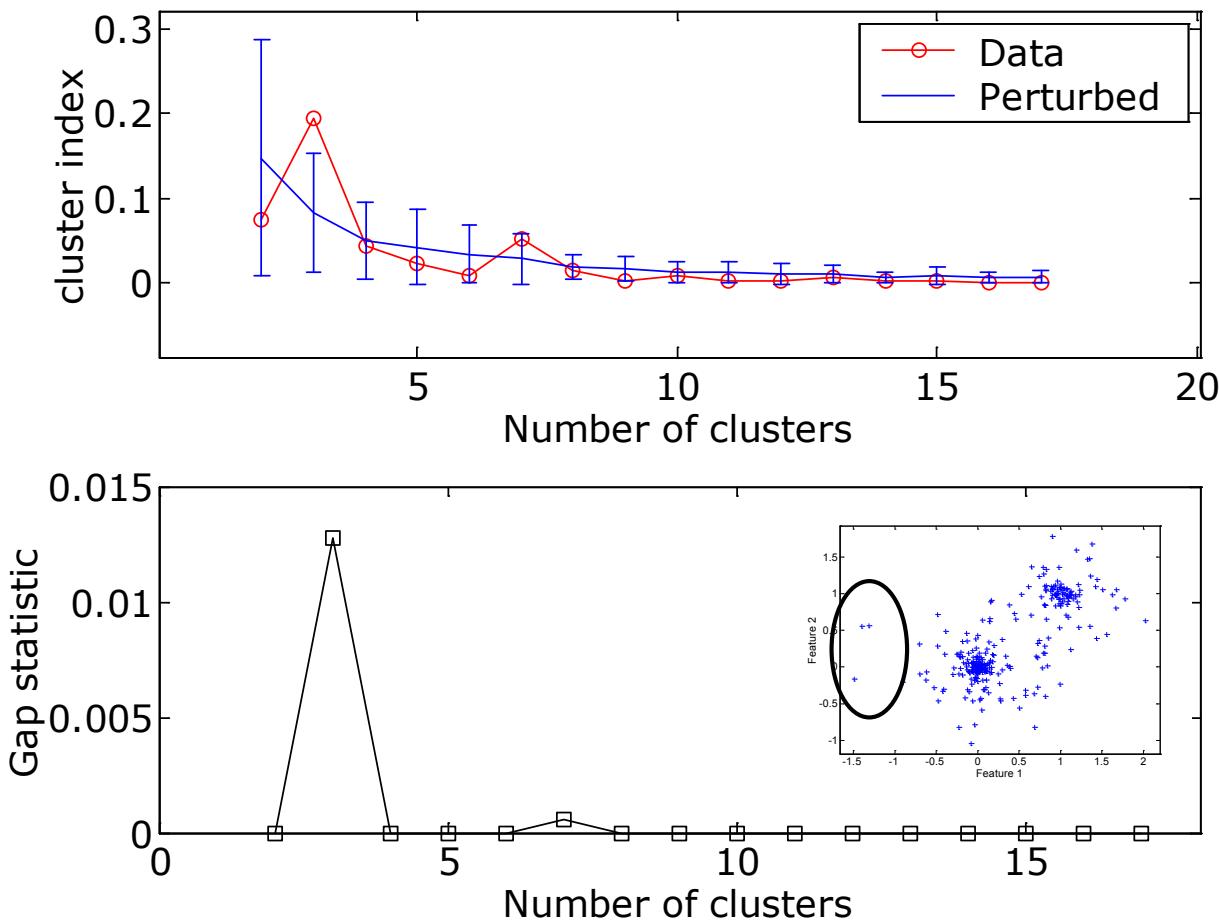
$$g_j^{gap} = \max \left\{ f_j - \mu_j^*, 0 \right\}, j = 1, 2, \dots, g$$

5. Look for large values of gap statistic  $g_j^{gap} = f_j$

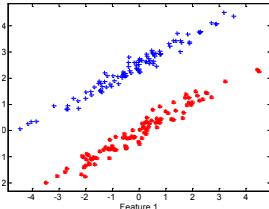
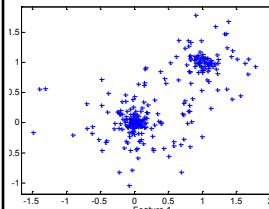
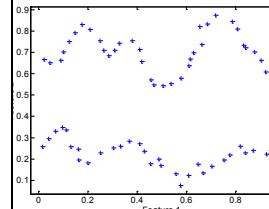
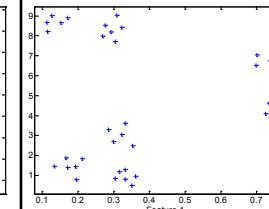
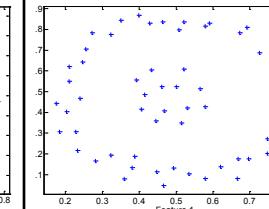
# Gap fusion graph (single linkage)



# Gap fusion graph (single linkage) (2)



# DBI vs. fusion graphs

					
DBI (s)	?	3/4	?	4	4+
DBI (c)	8+	2	5+	4	8+
Gap fusion graph (s)	3	3	2	3	2
Gap fusion graph (c)	2 (?)	2	4	3	3

# Recapitulation

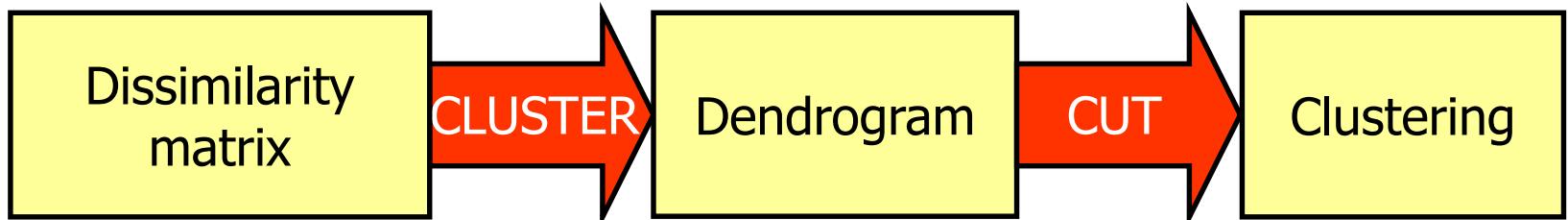
- *Cluster validation* is used for:
  - Assessing clustering
  - Deciding on the number of clusters
- Methods:
  - *Distortion measures* (dendrogram)
  - *Davies-Bouldin index*
  - *Fusion graph and gap statistic*
- When applying cluster validation, one also needs to define what a good cluster is – like in clustering itself.  
There's no free lunch...



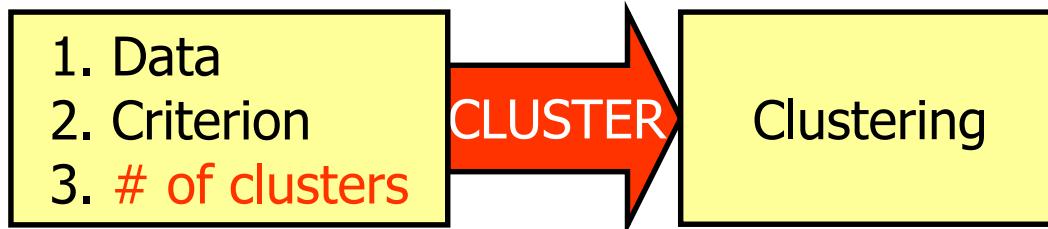
**Lunch break**  
**Exercise 4.8-4.16**

# Clustering overview

## 1. Hierarchical:



## 2. K-means:

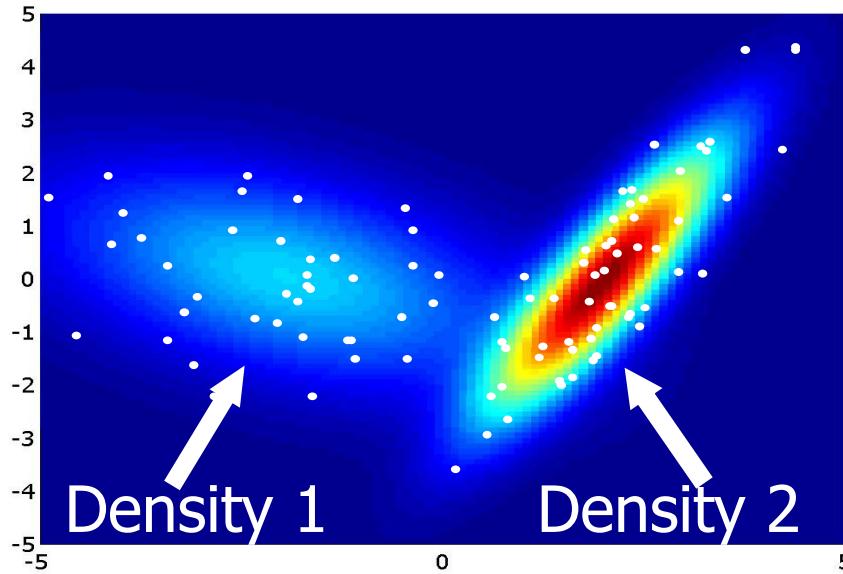


## 3. Density-based:



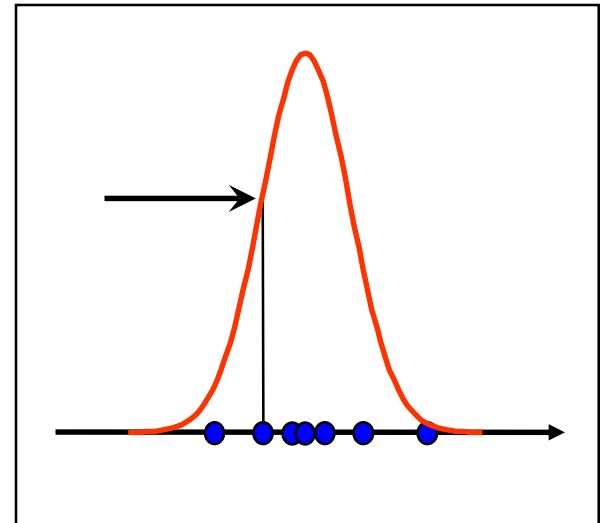
# Density-based clustering

- Each cluster is described by a probability density function
- Total dataset described by a *mixture* of density functions
- Clustering = maximizing the mixture fit
- Clusters are based on *a posteriori probabilities*



# Density-based clustering (2)

- Given:
  - $n$  independent objects:  $\{x_1, \dots, x_n\}$
  - probability density function model:
$$p(x | \theta) \sim N(\mu, \Sigma)$$
- Estimate parameters  $\theta = \{\mu, \Sigma\}$  such that model *fits* data
- Use *likelihood* as criterion: probability of observing the data set, given the model (as on Day 1, for kernel width  $h$  in Parzen density estimation)



# Estimation: maximum likelihood

- General method to estimate parameters  $\theta$  of probability distribution from data  $D = \{x_1, \dots, x_n\}$ . How?
- Maximize joint probability of the data

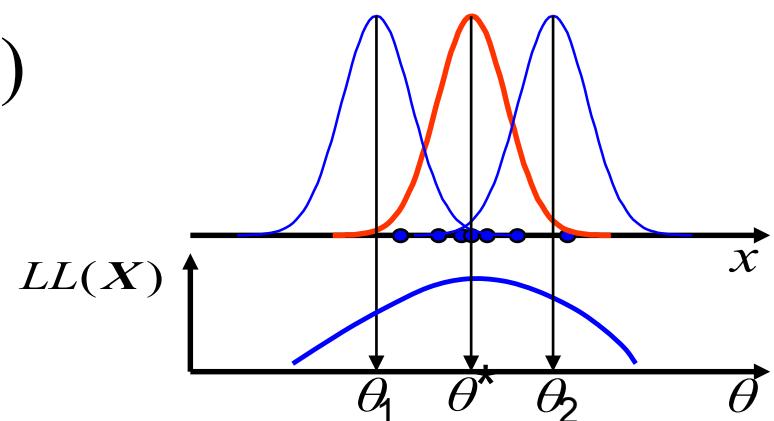
likelihood:

$$L = p(x_1, \dots, x_n | \theta) = \prod_{i=1}^n p(x_i | \theta)$$

log-likelihood:

$$LL = \sum_{i=1}^n \log \sum_Q p(x_i, Q | \theta)$$

same solution since log is monotonic



# Estimation: maximum likelihood (2)

Two possible outcomes:  $x = 0$  or  $x = 1$ .

Success ( $x = 1$ ) occurs with probability  $p$

Bernoulli distribution:  $P(x) = p^x (1 - p)^{1-x}$

Likelihood:  $P(X_1 = x_1, \dots, X_n = x_n | p) = p^{x_1} (1 - p)^{1-x_1} \dots p^{x_n} (1 - p)^{1-x_n}$

$$= p^{n_1} (1 - p)^{n-n_1}$$

↓

$$\frac{d(p^{n_1} (1 - p)^{n-n_1})}{dp} = 0$$

**# of successes**

Maximum at  $p = n_1/n$

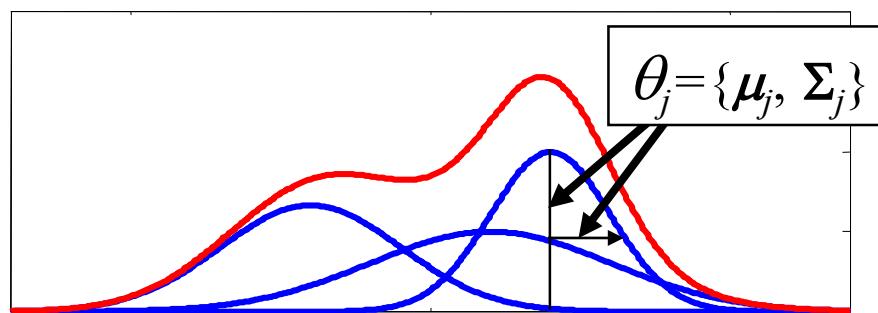
# Mixture-of-Gaussians

- Choose Gaussian as component density  $p(\mathbf{x}; \theta_j)$ :

$$p(\mathbf{x}; \theta_j) = \frac{1}{\sqrt{2\pi^p \det(\Sigma_j)}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j)\right)$$

- Describe complete data set as a mixture of  $p(\mathbf{x}; \theta)$ 's:

$$p(\mathbf{x}; \Psi) = \sum_{j=1}^g \pi_j p(\mathbf{x}; \theta_j) \quad \text{with} \quad \sum_{j=1}^g \pi_j = 1$$

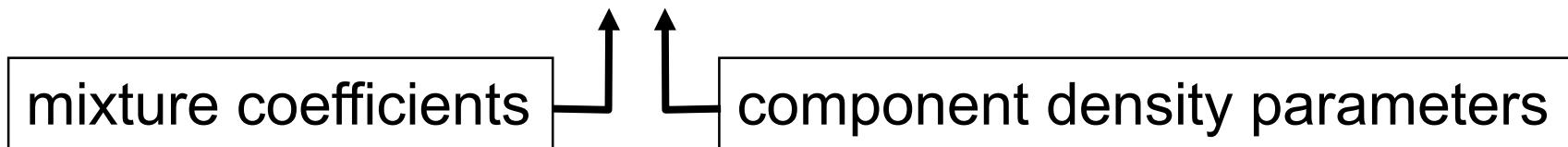


# Mixture-of-Gaussians (2)

$$p(\mathbf{x}; \Psi) = \sum_{j=1}^g \pi_j p(\mathbf{x}; \theta_j) \quad \text{with} \quad \sum_{j=1}^g \pi_j = 1$$

- Parameters:
  - Set number of clusters,  $g$
  - Estimate other parameters by maximum-likelihood:

$$\Psi = (\boldsymbol{\pi}, \boldsymbol{\theta} = \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}_{j=1\dots g})$$



**log-likelihood:**  $LL(\mathbf{X}; \Psi) = \sum_{i=1}^n \log \sum_{j=1}^g \pi_j p(\mathbf{x}_i; \theta_j)$

# EM algorithm

- **Problem:** need to simultaneously estimate two interdependent things...
  - Cluster membership of each object
  - Density parameters of each cluster:  $\pi_j, \mu_j, \Sigma_j$
- **Expectation-Maximization algorithm:**
  - General class of algorithms for this type of problem
  - Repeatedly:
    - Recalculate cluster membership of each object ( $E$ )
    - Recalculate density parameters of each cluster ( $M$ )
- Introduce a **hidden** variable  $z$  to explicitly indicate mixture components

$$\pi_j = p(z = j)$$

# Intermezzo: probabilities

$n = 20$

	die 1	•	.. •	• ..	• ..	•	•
die 2				•	•	• ..	• .. ..
	1	2	3	4	5	6	

**sum rule:**  $P(x) = \sum_y P(x, y)$

$$1/5 = P(3) = P(3, \text{ die 1}) + P(3, \text{ die 2}) = 3/20 + 1/20$$



**product rule:**  $P(x, y) = P(x | y)P(y) = P(y | x)P(x)$

$$\begin{aligned} 3/20 &= P(3, \text{ die 1}) = P(3 | \text{die 1})P(\text{die 1}) = (3/11)(11/20) = 3/20 \\ &= P(\text{ die 1} | 3)P(3) = (3/4)(4/20) = 3/20 \end{aligned}$$

# Intermezzo: Bayes' theorem

From product rule

$$P(x \mid y)P(y) = P(y \mid x)P(x)$$



$$\text{Bayes : } P(x \mid y) = \frac{P(y \mid x)P(x)}{P(y)} = \frac{P(y \mid x)P(x)}{\sum_x P(y \mid x)P(x)}$$

$$P(\text{die 1} \mid 3) = \frac{P(3 \mid \text{die 1})P(\text{die 1})}{P(3)} = \frac{(3/11)(11/20)}{4/20} = 3/4$$

# Intermezzo: game show



3 doors, 1 prize

- Rules:
  - Choose a door
  - Gameshow host opens one of the other two doors without revealing the prize
- Should you stick with your door or switch to the other closed door?

## EM algorithm (2)

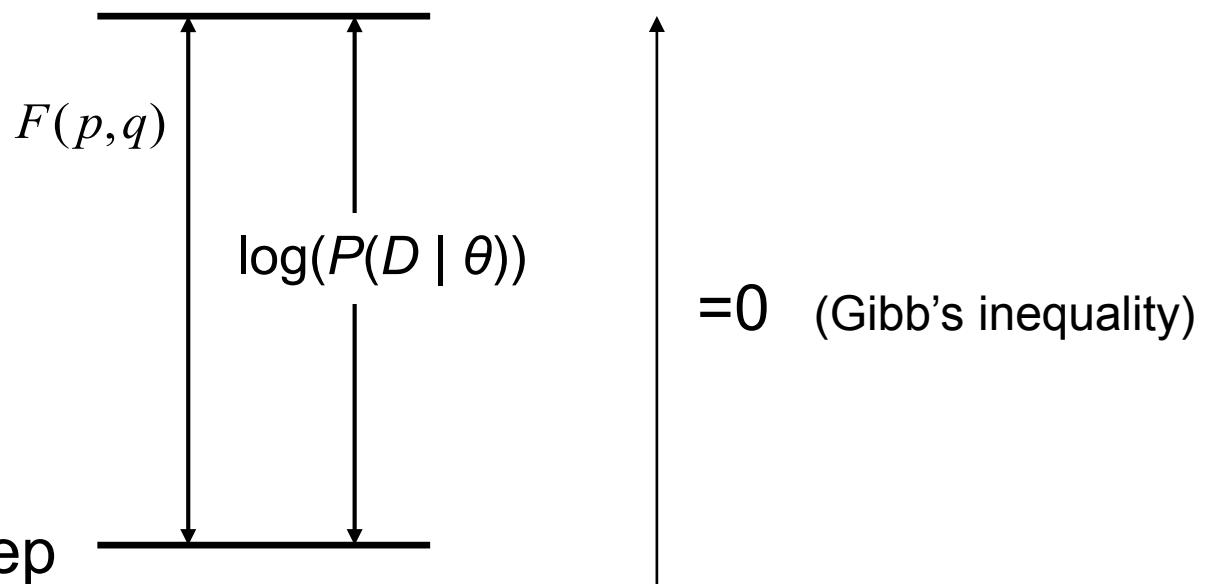
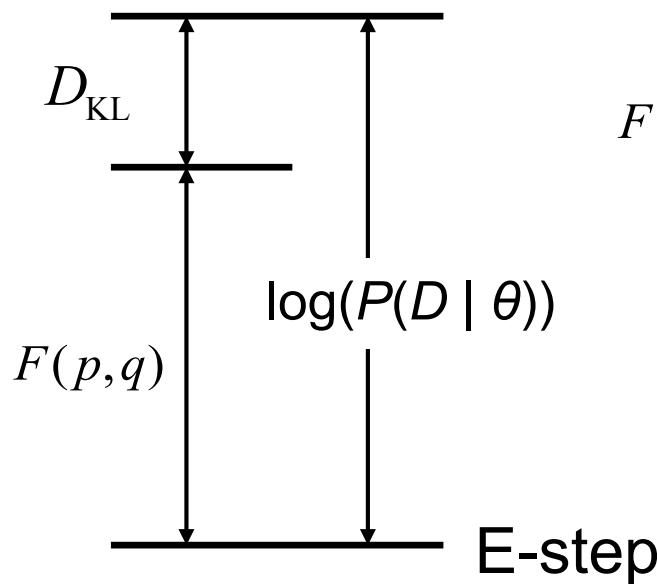
$$\begin{aligned} \log p(D) &= \sum_x \log p(x) = \sum_x \sum_z q(z) \log p(x) \\ &= \sum_{x,z} q(z) \log \frac{p(x,z)}{p(z|x)} = \sum_{x,z} q(z) \log \left( \frac{p(x,z)}{p(z|x)} \times \frac{q(z)}{q(z)} \right) \\ &= \sum_{x,z} q(z) \log \left( \frac{p(x,z)}{q(z)} \right) + \sum_{x,z} q(z) \log \left( \frac{q(z)}{p(z|x)} \right) \\ &= F(p_{\text{joint}}, q) + D_{KL}(q \parallel p_{\text{post}}) \end{aligned}$$

free energy    relative entropy ( $\geq 0$ )

arbitrary distribution    hidden variable

# EM algorithm: E-step

$$\log p(D) = \sum_{x,z} q(z) \log \left( \frac{p(x, z)}{q(z)} \right) + \boxed{\sum_{x,z} q(z) \log \left( \frac{q(z)}{p(z | x)} \right)}$$

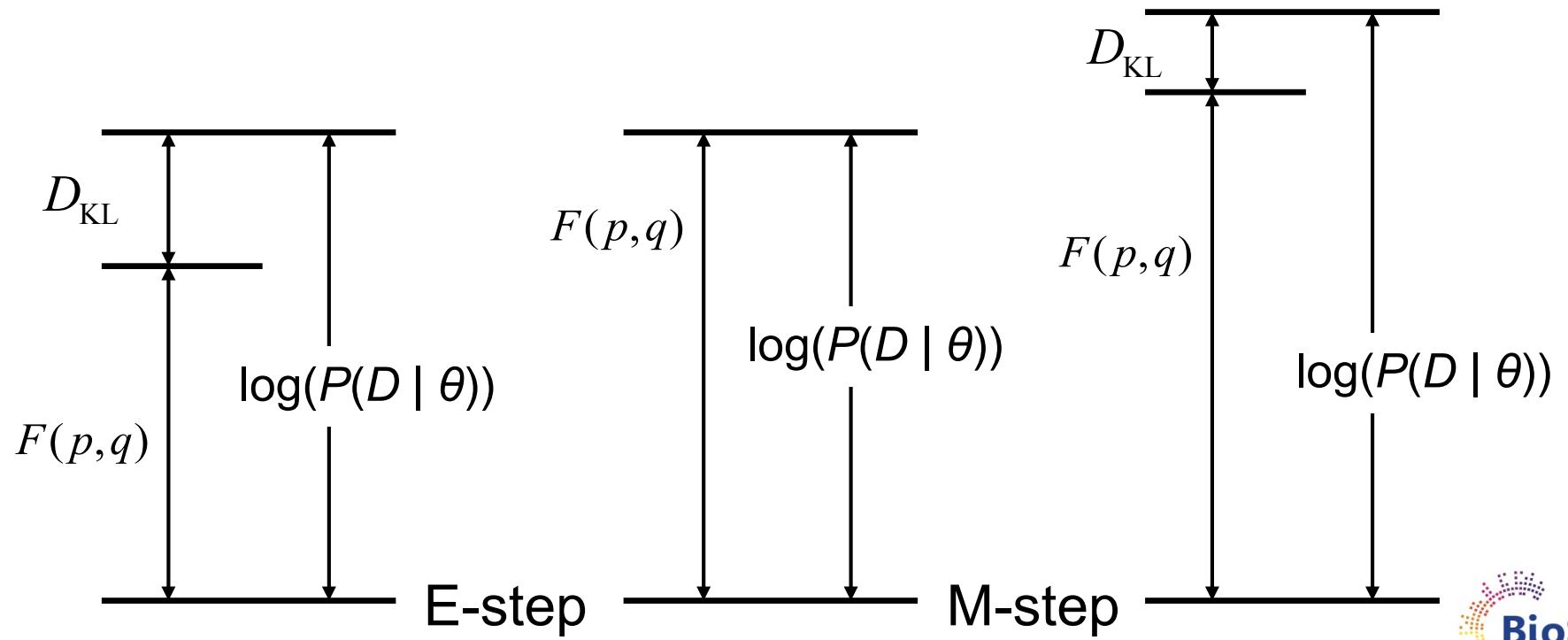


$E\text{-step: } q^{\text{new}}(z | x) = p_{\text{post}} = p(z | x)$

# EM algorithm: M-step

$$\log p(D) = \sum_{x,z} p(z|x) \log \left( \frac{p(x,z)}{p(z|x)} \right)$$

M-step: maximize  $\log[p(D)]$  with respect to the parameters



# EM algorithm (3)

Iterate to maximize likelihood:

**E-step:**  $p_{\text{post}} = p(z | x, \theta)$

Calculate the distribution of the hidden variables given the data and the model parameters

**M-step:**  $\theta^{new} = \arg \max_{\theta} \sum_{x,z} p(z | x) \log p(x, z | \theta)$

Maximize the expected (with respect to hidden variables) log-likelihood of the complete data.

Compare M-step with MoG log-likelihood:

$$\sum_{i=1}^n \log \sum_{j=1}^g \pi_j p(\mathbf{x}_i; \theta_j)$$

M-step is easier: log within sum



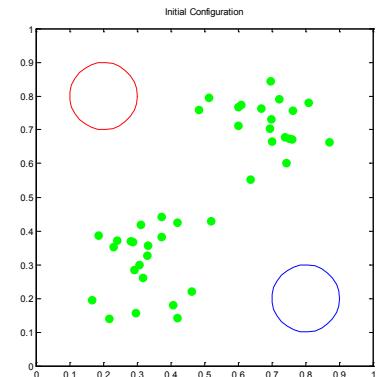
# EM: mixture model

Very simple example of a model with hidden variables:

2-component mixture model

$$p(x) = \pi_1 p_1(x | \theta) + \pi_2 p_2(x | \theta)$$

hidden variable  $z = 1, 2$  - component label

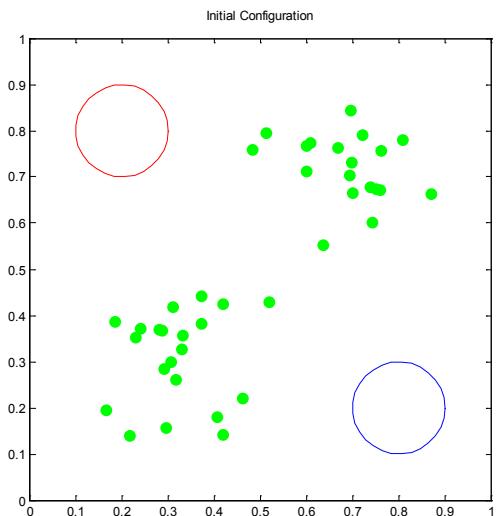


$$\text{E-step: } p(z = j | x, \theta) = \frac{p(z = j | \theta) p(x | z = j, \theta)}{p(x | \theta)} = \frac{\pi_j p_j(x | \theta)}{p(x)}$$

responsibility

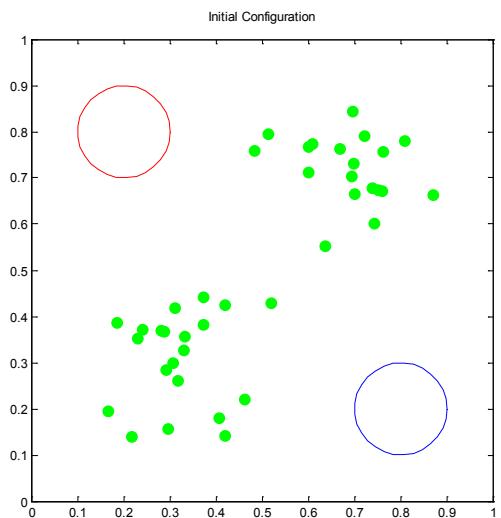
$$\text{M-step: maximize } \sum_{x,z \in \{1,2\}} p(z | x) \log p(x, z | \theta)$$

# EM: mixture model (2)

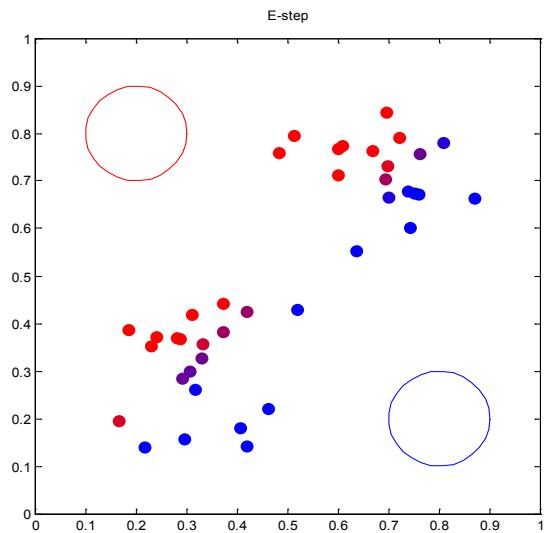


Initialization

# EM: mixture model (3)



Initialization



E-step

# EM: mixture model (4)

- **M-step:** Maximization

Maximize the expected complete LL by updating

- mixture coefficients  $\pi_j$
- cluster means and covariances  $\theta_j = \{\mu_j, \Sigma_j\}, j=1,\dots,g$ :

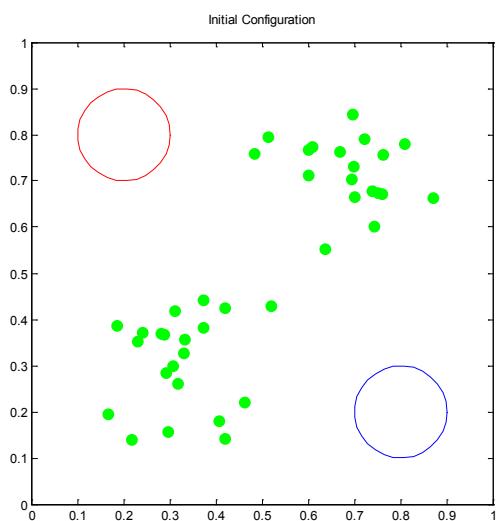
$$\hat{\pi}_j = \frac{1}{n} \sum_{i=1}^n p(z=j | x_i) = \frac{1}{n} \sum_{i=1}^n w_{ij} \quad \left. \right\} \text{“total membership”}$$

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

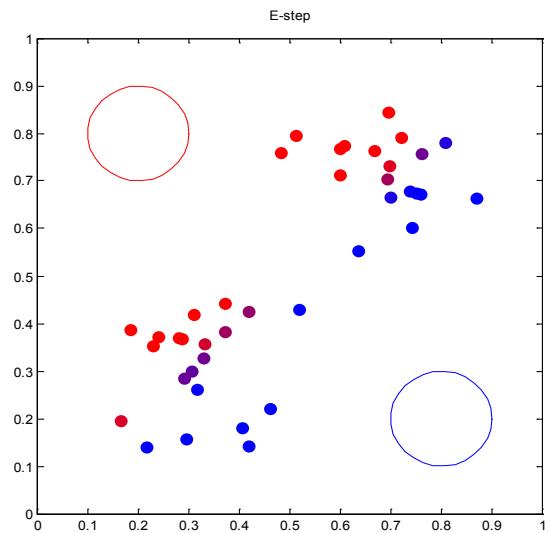
$$\hat{\Sigma}_j = \frac{\sum_{i=1}^n w_{ij} (\mathbf{x}_i - \hat{\mu}_j)(\mathbf{x}_i - \hat{\mu}_j)^T}{\sum_{i=1}^n w_{ij}}$$

} weighted sums

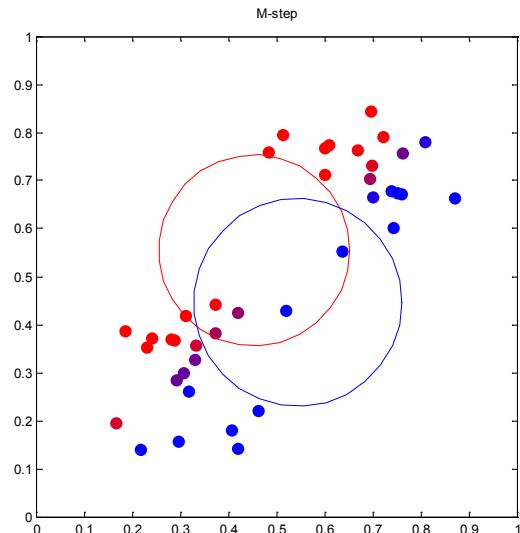
# EM: mixture model (5)



Initialization

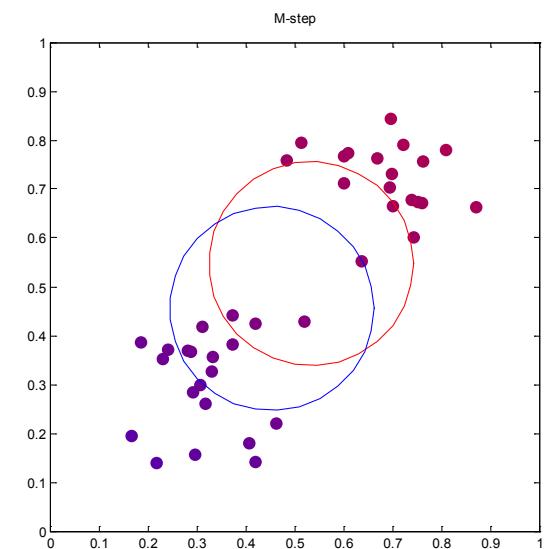


E-step



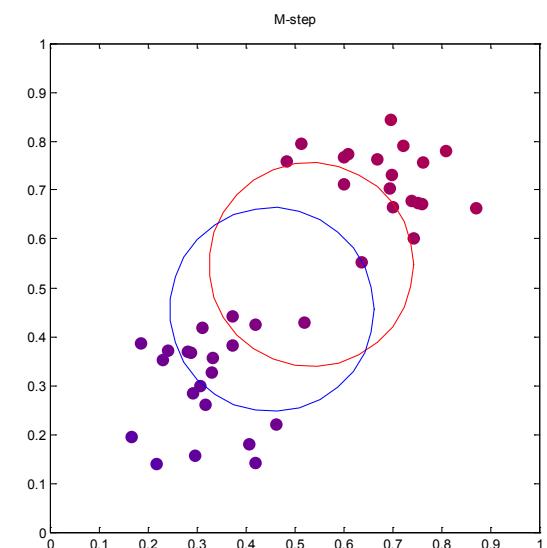
M-step

# EM: mixture model (6)

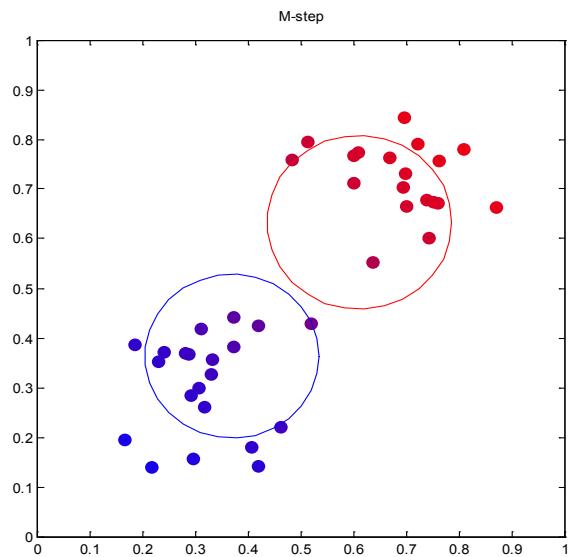


M-step: 3

# EM: mixture model (7)

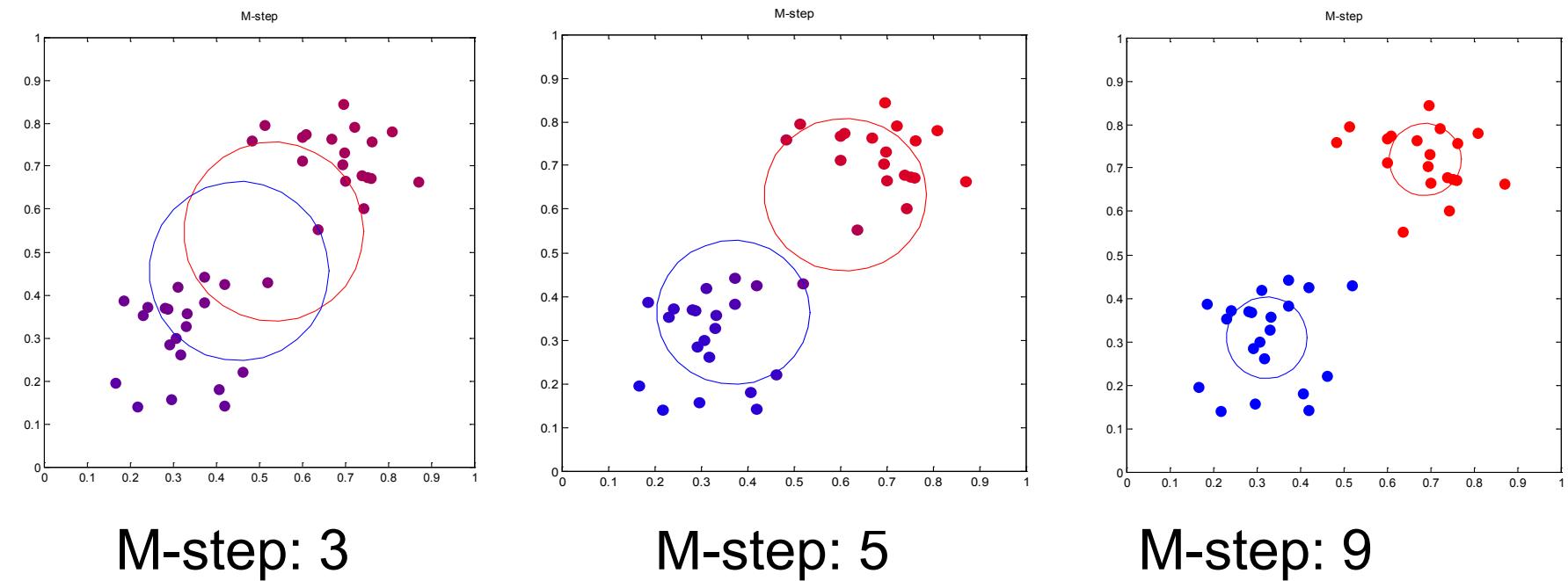


M-step: 3



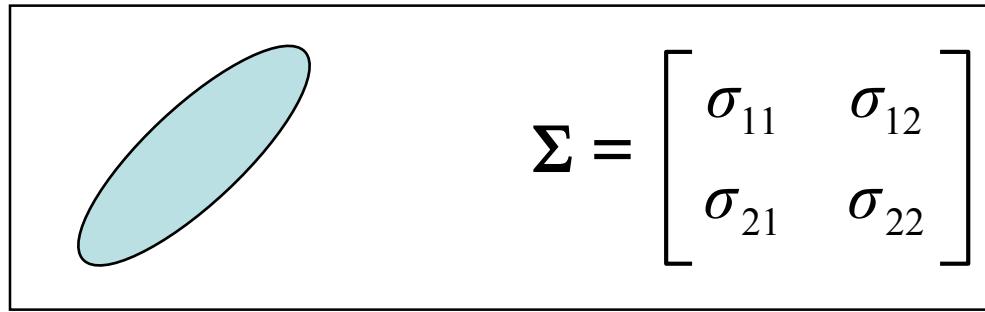
M-step: 5

# EM: mixture model (8)

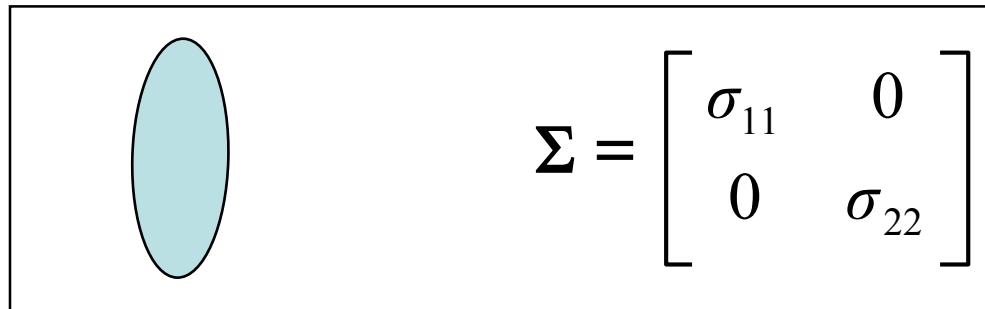


# Mixture-of-Gaussians (3)

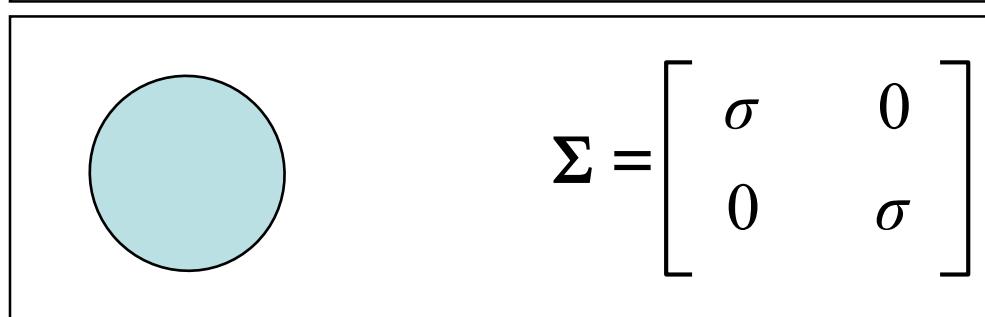
- ‘Gauss’:



- ‘Aligned’:



- ‘Circular’:



# EM: mixture model (9)

- If...
  - all clusters are spherical
  - the variance of each cluster is infinitely small

$$\Sigma = \begin{bmatrix} \varepsilon^2 & 0 & 0 \\ 0 & \varepsilon^2 & 0 \\ 0 & 0 & \varepsilon^2 \end{bmatrix}, \quad \varepsilon \rightarrow 0$$

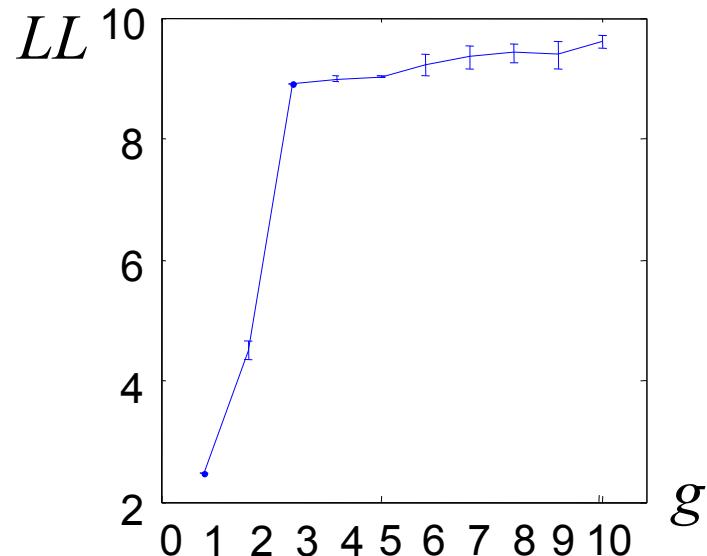
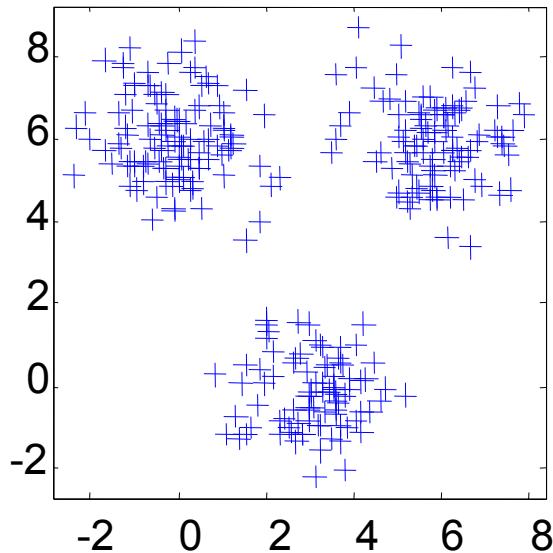
then the EM algorithm simplifies to the  $K$ -means algorithm  
(samples are always assigned to the closest cluster!)

# EM algorithm (4)

- Disadvantages:
  - can get stuck in local minima
  - depends on initial conditions
  - convergence can be slow
  - problems with covariance estimates:  
if too few samples are members of a cluster,  
there will not be enough data to base estimate on
- Advantages:
  - simple to implement

# Cluster validation: log-likelihood

- For probabilistic models (e.g. mixture-of-Gaussians):
  - Log-likelihood will probably not increase anymore when too many clusters are used
  - Look for “plateau” in log-likelihood graph



- Problem: when  $g = n$ , the log-likelihood is infinite;  
Solution: information criteria (Day 5)

# Recapitulation

- Density based clustering:
  - Assume a *probability density function* per cluster
  - Train using the *EM algorithm*
- Example:
  - *Mixture of Gaussians*
  - But many probability densities fit in the same framework  
principal component analysis, factor analysis, ...
- EM algorithm:
  - problem *decomposition*: simple to implement
  - sensitive to *local minima*

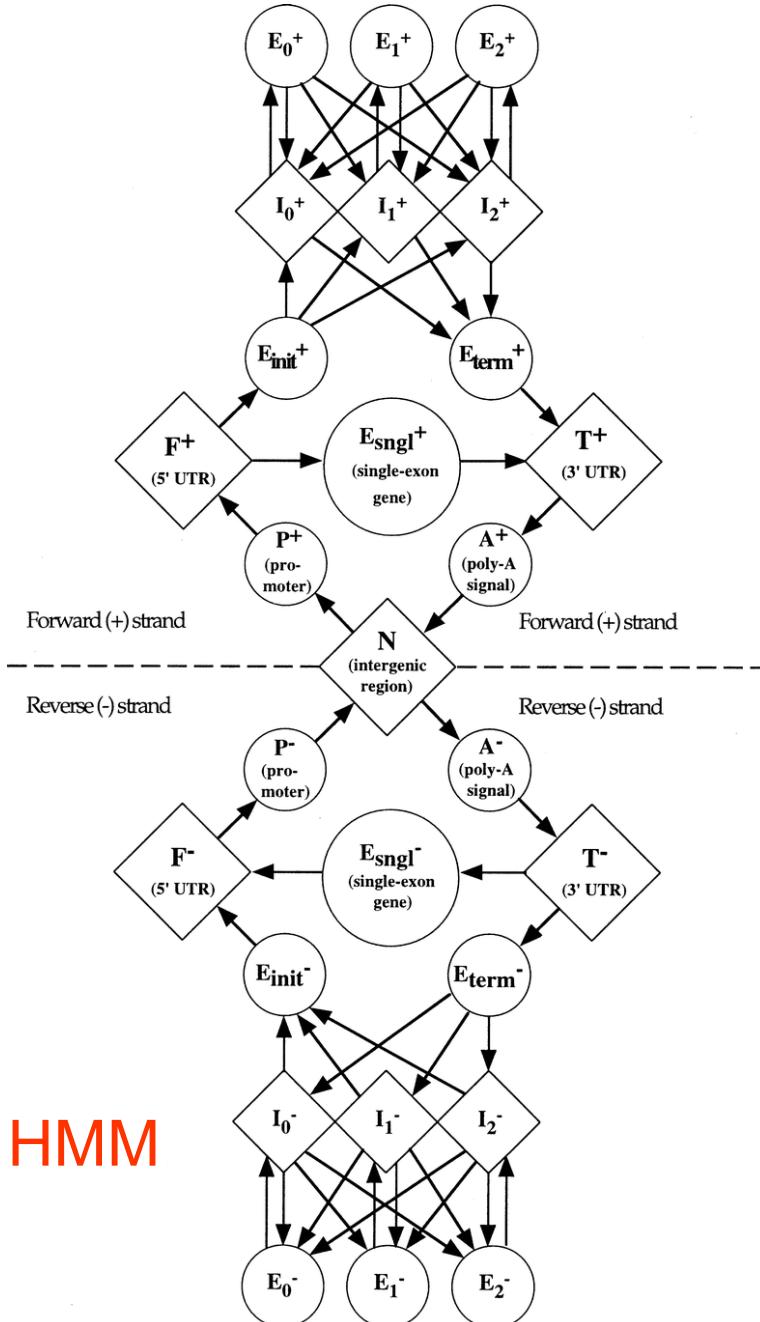


**15min break**  
**Exercise 4.17**

# Hidden Markov models

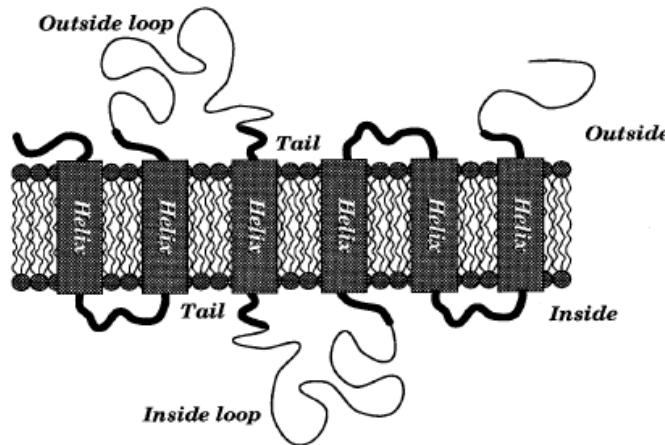
- Regular expressions & weight matrices
- Dependencies & Markov chains
- Hidden Markov models
- HMMs & EM
- Profile HMMs
- Genefinding

# Application: genefinding



# generalized HMM

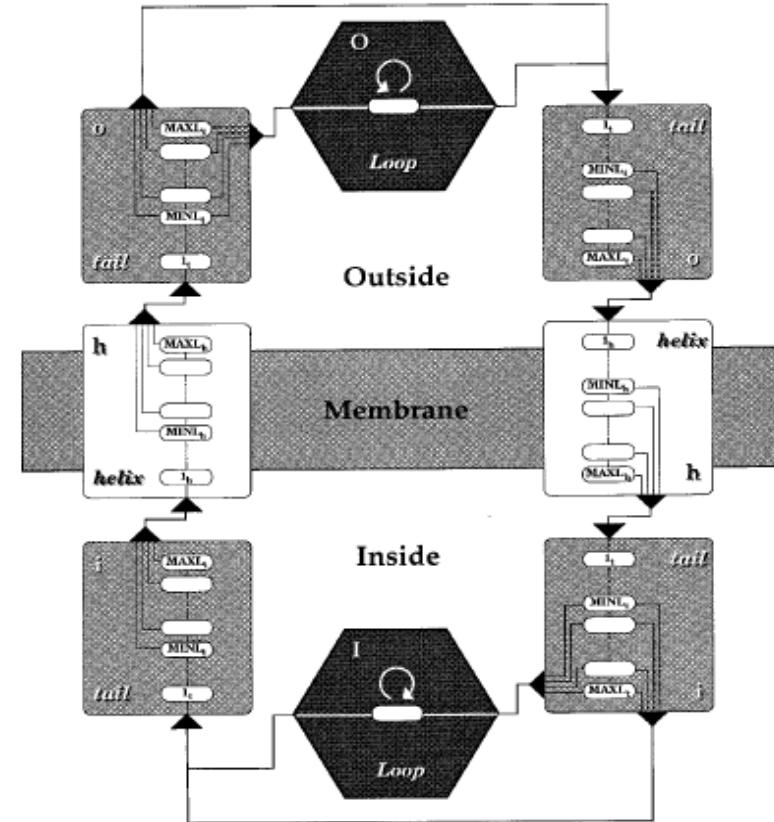
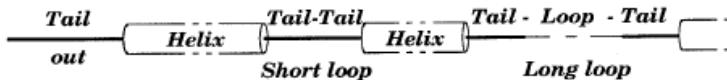
# Application: transmembrane proteins



**Amino acid seq:** MGDVCDTEFGILVA...SVALRPRKHGRWIV...FWVDNGTEQ...PEHMTKLHMM...

**State seq:** oooooooooooooohhhhh...hhhiiiiiiihhh...hhooooooO...OOOooooohhh...

**Topology:**



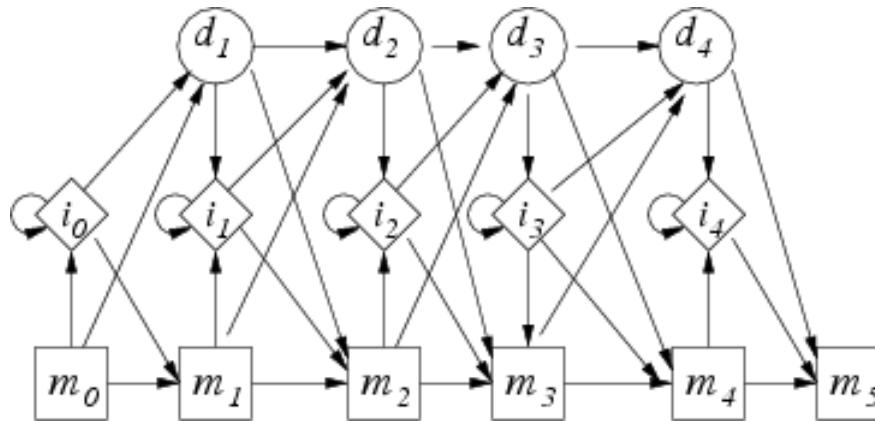
HMM

# Application: protein domains

Q21978/165-314  
 Q20638/74-216  
 Q19601/54-189  
 Q18311/32-175  
 Q18209/233-375  
 Q9ZAX1/42-162  
 IGB1\_MEDEA/7-142  
 IGB1\_IUPLU/7-145  
 HBP2\_CASGL/13-152  
 HBP1\_CASGL/6-145  
 GLP1\_GLYDI/7-141  
 GLB1\_GLYDI/6-141  
 GLB1\_TUBTU/6-139  
 GLB2\_LAMSP/7-141  
 GLB2\_TYLHE/9-142  
 GLB2\_IUMTE/8-141  
 GLB2\_TYLHE/7-136  
 GLB2\_TYLHE/8-143  
 GLB3\_TYLHE/8-143  
 GLB4\_IUMTE/11-146  
 GLB\_CERRH/6-146  
 GLB\_BUSCA/6-146  
 GLB\_ANATR/16-151  
 GLB\_APLJU/6-139  
 GLBx\_CHITH/11-145  
 GLBc\_CHITH/23-157  
 GLB6\_CHITH/22-156  
 GLB5\_CHITH/9-143  
 GLB5\_CHITH/22-156  
 GLB3\_CHITH/20-147

SCEVVADSIRLVESRSSAAETSACFGIFVFQRVFS.	KIFMLRPLEG.I.SESDDWFDLPDNHPWRRHARLFTSI
Q20638/74-216	EKELLRTWNSDEFD.NLYGSAIYCYIFD.HNFNCROLEPF.F.ISKYQGDWEKESKEFRSQAALKFVQI
Q19601/54-189	ERILIEQSWRKTRKT.GADHGSKIFFMVIT.AQPDIAKEIFG.I.EK.IPTGRLYKDPRFRQHALVYTKT
Q18311/32-175	TTKLVIOEPYRVLIA.OCPPELTUEWKSAT.RSTSIIKLAEG.I.AE.N.ESPQMNAAFLGLESSTTQAF
Q18209/233-375	QIHLVRALRIRQWYT.KGPTVIGASIVHRLCFKNVMVKEQMKQWE.IPPKF.QN.RDNFIKAHKVAEL
Q9ZAX1/42-162	DALRVLQNFKL.DDPPELVRFRYAHWFA.LDASVRDLPF.P.DMGQAQRAAFQGAQHWW
IGB1_MEDEA/7-142	IQLPRYSWEEAFKQ.NLPRYSWFFYIWVLE.KAFAAKGIFPS.F.I.LKNSAEVQDQSPQIQAHAEKVFGI
IGB1_IUPLU/7-145	QEAIVVKSSEEFNA.NIPKNTNTHRFITVILE.IAPFQAKDIES.F.II.GSSEVPQNNPDLQOAHAGKVKLI
HBP2_CASGL/13-152	QEAIVVKSISAKP.NAGELCILKEFLIKE.IAPSAQKIES.FLKQ.SNVPLERNPKLKSASHMSVFLM
HBP1_CASGL/6-145	QEALIKOSNEVILQO.NIPAHSLRLFALLIE.IAPESKVWES.FLKQ.SNEIPENNPKLKAHAAVIFKT
GLP1_GLYDI/7-141	QVAALKASQPEVSAG.DCGAQIGLEMFTKYFH.ENFQMMFIFG.YSGR.T.EALKHSSKIQHHGKVIIDQ
GLB1_GLYDI/6-141	QROMVAAITMKDIAKA.DNGACVGKDCLLIFLS.AHFQMAAEVG.E.SG.ASDPGVAALGAKVIAQ
GLB1_TUBTU/6-139	QRFKVKHQAEEAFCG.SHHRDLFGLKLWNSIFR.DAPEIIGLIFKRWDGDN.A.VSYAEEFEAHERVILGG
GLB2_LAMSP/7-141	QRLKVKRQNAEAVGS.GNDREERFGHIFTUHVFK.DAPSARDLIFKRVRGDNQI.HTPAFAHATRVLGG
GLB2_TYLHE/9-142	QRLKVKQWAKAYGV.GHERVELGIALWKSMFA.QNDNDARDLIFKRWHGEDV.HSPAFAHMARVFGN
GLB2_IUMTE/8-141	EG1LKVKSEONGRAYGS.GHDREAFSQAIWRATFA.QNPESRSLIFKRWHGDDT.SHPAFIAHAERVILGG
GLB2_TYLHE/7-136	QRLKVKQWAKAYGS.V.GESRIDEAIDVFNFFR.TNFDRS.LENRWQDNV.YSPERKAHNVRVFAQ
GLB2_TYLHE/8-143	DRREVQALIERSIVSAB.DTGRRTLIGRILLFEELFE.IDGATKGLIFKRVNVDDT.HSPFEEFAHVLRVNG
GLB3_TYLHE/8-143	DRHEVLDNNKGIVSAB.FTGRRVAIQQAFQELFA.IDPNNAKVEGRVNVD.K.PSEADWKAHVIRVING
GLB4_IUMTE/11-146	DRREIHRHINDWUSSS.FTDRVAIIVRAVEDDFLFK.HWPTSKALIFRVKIDEP.ESGERFKSHLVRVANG
GLB_CERRH/6-146	SKSALASSWKTIAKD.AATIQNNCATLFSILFK.QFFDTNRNYFTHE.GNMS.DAEMKTTGVGKAHSMAVFAG
GLB_BUSCA/6-146	QKTAIKESWVKLGAD.CPTTMKNGSILFGILLFK.TYFDITKKHFHFT.DDATFAAMDITGVGKAHQGVAVFSG
GLB_ANATR/16-151	QKDLIRLSQGVLSV.DMEGTGIMLMANLFK.TSSAARTKFARL.GDVS.AGKDMSKLRGHSITLMYA
GLB_APLJU/6-139	DAGLIAQSWAPVFA.NSDANGASPLVALFT.QFFESANFFFNDK.GK.KSLADIQASPKLIRDWSRRIAR
GLBx_CHITH/11-145	EVEQWQATWKAWSH.DEWELLVYTVFK.AHPDIAKEPFKF.AG.KDLEAIKDADFAVHASRILIGF
GLBc_CHITH/23-157	EASLWQSSWKAWSH.NEVDDILAAVFA.AYFDIQAKEPFQF.AG.KDLSAIKDGTGAFATHATRIVSE
GLB6_CHITH/22-156	QADLVKKTNSTWKF.NEVDDILAYAVFK.AYFDIQAKEPFQF.AG.KDLDISKDSAAFATHATRIVSF
GLB5_CHITH/9-143	QALAFKSSWNTWKH.NEVDDILAYAVFK.ANEDTQAKEPFQF.AG.KDLDISKDSADFVHSGRIVGF
GLB5_CHITH/22-156	EASLVRGSWQAWKH.SEVDDILYVIFK.ANPFDIAKEPFQF.AG.KDLETLKGTGOFATHAGRIVGF
GLB3_CHITH/20-147	QISTWQASEDKWKG.DPVGILYAVFK.ADP SIMAKETQF.AG.KDLESIKGTAPFEIHANRIVGF

Profile HMM



# Outline

- Regular expressions & weight matrices
- Dependencies & Markov chains
- Hidden Markov models
- HMMs & EM
- Profile HMMs
- Genefinding

# Sites

Site: short sequence containing  
some signal

A	C	A	A	T	G
T	C	A	A	T	C
A	C	A	A	G	C
A	G	A	A	T	C
A	C	C	A	T	C

Examples: intron splice sites, transcription start site,  
transcription factor binding sites

Goals:

- give a mathematical description (**model**) of a site
- find possible sites in a long sequence

# Consensus sequence

majority vote:

A C A A T C

A	C	A	A	T	G
T	C	A	A	T	C
A	C	A	A	G	C
A	G	A	A	T	C
A	C	C	A	T	C

W S M A K S

from IUPAC code:

M	A/C
R	A/G
W	A/T
S	C/G
Y	C/T
K	G/T
B	C/G/T
D	A/G/T
H	A/C/T
V	A/C/G
N	A/C/G/T



# Regular expressions

[ab] : union {a,b}

ab : concatenation {ab}

$\epsilon$  : empty string

$a^*$  : Kleene star { $\epsilon, a, aa, aaa, \dots$ }

A	C	A	A	T	G
T	C	A	A	T	C
A	C	A	A	G	C
A	G	A	A	T	C
A	C	C	A	T	C

[AT][CG][AC]A[TG][GC]

A C A A T C , but also T G C A G G

See also <http://prosite.expasy.org>



# Weight matrices

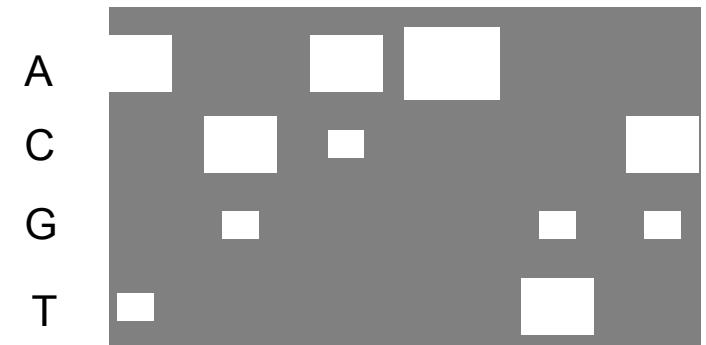
$$\begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 \\ A & \left( \begin{matrix} 4 & 0 & 4 & 5 & 0 & 0 \end{matrix} \right) \\ C & \left( \begin{matrix} 0 & 4 & 1 & 0 & 0 & 4 \end{matrix} \right) \\ G & \left( \begin{matrix} 0 & 1 & 0 & 0 & 1 & 1 \end{matrix} \right) \\ T & \left( \begin{matrix} 1 & 0 & 0 & 0 & 4 & 0 \end{matrix} \right) \end{matrix}$$

counts

A	C	A	A	T	G
T	C	A	A	T	C
A	C	A	A	G	C
A	G	A	A	T	C
A	C	C	A	T	C

probabilities

$$W = \begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 \\ A & \left( \begin{matrix} 0.8 & 0.0 & 0.8 & 1.0 & 0.0 & 0.0 \end{matrix} \right) \\ C & \left( \begin{matrix} 0.0 & 0.8 & 0.2 & 0.0 & 0.0 & 0.8 \end{matrix} \right) \\ G & \left( \begin{matrix} 0.0 & 0.2 & 0.0 & 0.0 & 0.2 & 0.2 \end{matrix} \right) \\ T & \left( \begin{matrix} 0.2 & 0.0 & 0.0 & 0.0 & 0.8 & 0.0 \end{matrix} \right) \end{matrix}$$



aka position specific score matrix

# Weight matrices (2)

Sequence:  $x = x_1 x_2 \dots x_N$

$\xrightarrow{\quad}$  independence

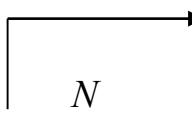
$$P(x_1 x_2 \dots x_N | W) = \prod_{i=1}^N w_{x_i, i} = \prod_{i=1}^N P_i(x_i | W)$$

$$\begin{aligned} P(\text{ACAATC} | W) &= P_1(\text{A})P_2(\text{C})P_3(\text{A})P_4(\text{A})P_5(\text{T})P_6(\text{C}) \\ &= 0.8 \times 0.8 \times 0.8 \times 1 \times 0.8 \times 0.8 = 0.33 \end{aligned}$$

$$\begin{array}{ccccccc} & & 1 & 2 & 3 & 4 & 5 & 6 \\ \begin{matrix} A \\ C \\ G \\ T \end{matrix} & \left( \begin{matrix} 0.8 & 0.0 & 0.8 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.2 & 0.0 & 0.0 & 0.8 \\ 0.0 & 0.2 & 0.0 & 0.0 & 0.2 & 0.2 \\ 0.2 & 0.0 & 0.0 & 0.0 & 0.8 & 0.0 \end{matrix} \right) \end{array}$$

# Weight matrices (3)

Sequence:  $x = x_1 x_2 \dots x_N$

 independence

$$P(x_1 x_2 \dots x_N | W) = \prod_{i=1}^N w_{x_i, i} = \prod_{i=1}^N P_i(x_i | W)$$

$$\begin{aligned} P(\text{CCAATC} | W) &= P_1(\text{C}) P_2(\text{C}) P_3(\text{A}) P_4(\text{A}) P_5(\text{T}) P_6(\text{C}) \\ &= 0 \times 0.8 \times 0.8 \times 1 \times 0.8 \times 0.8 = 0 \end{aligned}$$

$$\begin{array}{ccccccc} & & 1 & 2 & 3 & 4 & 5 & 6 \\ \begin{matrix} A \\ C \\ G \\ T \end{matrix} & \left( \begin{matrix} 0.8 & 0.0 & \boxed{0.8} & \boxed{1.0} & 0.0 & 0.0 \\ \boxed{0.0} & \boxed{0.8} & 0.2 & 0.0 & 0.0 & \boxed{0.8} \\ 0.0 & 0.2 & 0.0 & 0.0 & 0.2 & 0.2 \\ 0.2 & 0.0 & 0.0 & 0.0 & \boxed{0.8} & 0.0 \end{matrix} \right) \end{array}$$

# Weight matrices: pseudocounts

$$P(x) = \frac{\#x + 1}{\sum_i (\#i + 1)}$$

pseudocount (Laplace)

A	C	A	A	T	G
T	C	A	A	T	C
A	C	A	A	G	C
A	G	A	A	T	C
A	C	C	A	T	C

$$A \begin{pmatrix} 5 & 1 & 5 & 6 & 1 & 1 \end{pmatrix}$$

$$C \begin{pmatrix} 1 & 5 & 2 & 1 & 1 & 5 \end{pmatrix}$$

$$G \begin{pmatrix} 1 & 2 & 1 & 1 & 2 & 2 \end{pmatrix}$$

$$T \begin{pmatrix} 2 & 1 & 1 & 1 & 5 & 1 \end{pmatrix}$$

$$W' = \begin{matrix} A \\ C \\ G \\ T \end{matrix} \begin{pmatrix} 0.56 & 0.11 & 0.56 & 0.67 & 0.11 & 0.11 \\ 0.11 & 0.56 & 0.22 & 0.11 & 0.11 & 0.56 \\ 0.11 & 0.22 & 0.11 & 0.11 & 0.22 & 0.22 \\ 0.22 & 0.11 & 0.11 & 0.11 & 0.56 & 0.11 \end{pmatrix}$$

$$P(\text{ACAATC} | W') = P_1(\text{A})P_2(\text{C})P_3(\text{A})P_4(\text{A})P_5(\text{T})P_6(\text{C}) = 0.56^5 \times 0.67 = 0.037$$

$$P(\text{CCAATC} | W') = P_1(\text{C})P_2(\text{C})P_3(\text{A})P_4(\text{A})P_5(\text{T})P_6(\text{C}) = 0.11 \times 0.56^4 \times 0.67 = 0.0072$$

# Bayes' rule: odds ratio

class A: sites

class B: non-sites

$$\begin{aligned} x \text{ is assigned to class } A &\iff \frac{P(x | \text{class } A)P(A)}{P(x)} > \frac{P(x | \text{class } B)P(B)}{P(x)} \\ &\iff \frac{P(x | \text{class } A)}{P(x | \text{class } B)} > \frac{P(B)}{P(A)} \quad \text{priors} \end{aligned}$$

equal priors:

$$\frac{P(x | \text{class } A)}{P(x | \text{class } B)} > 1 \iff \log\left(\frac{P(x | \text{class } A)}{P(x | \text{class } B)}\right) > 0$$

odds ratio

log-odds ratio

$$\text{unequal priors, e.g.: } \log \frac{P(B)}{P(A)} = \log \frac{0.7}{0.3} = 1.22$$

# Weight matrices: odds ratio

$W$ : weight matrix,  $R$ : background model (independent of position)

$$\frac{P(x_1 x_2 \dots x_N | W)}{P(x_1 x_2 \dots x_N | R)} = \frac{\prod_{i=1}^N P_i(x_i | W)}{\prod_{i=1}^N P(x_i | R)}$$

$$\log_2 \left( \frac{P(x_1 x_2 \dots x_N | W)}{P(x_1 x_2 \dots x_N | R)} \right) = \log_2 \left( \frac{\prod_{i=1}^N P_i(x_i | W)}{\prod_{i=1}^N P(x_i | R)} \right) = \sum_{i=1}^N \log_2 \left( \frac{P_i(x_i | W)}{P(x_i | R)} \right)$$

log-odds ratio



# Weight matrices: log-odds ratio

$R$  uniform:  $P(\text{A}|R) = P(\text{C}|R) = P(\text{G}|R) = P(\text{T}|R) = 0.25$

$$\begin{matrix} A \\ C \\ G \\ T \end{matrix} \left( \begin{array}{cccccc} 1.16 & -1.17 & 1.16 & 1.42 & -1.17 & -1.17 \\ -1.17 & 1.16 & -0.17 & -1.17 & -1.17 & 1.16 \\ -1.17 & -0.17 & -1.17 & -1.17 & -0.17 & -0.17 \\ -0.17 & -1.17 & -1.17 & -1.17 & 1.16 & -1.17 \end{array} \right) \rightarrow \log(0.56/0.25)$$

$$\text{log-odds(ACAATC)} = 1.16 + 1.16 + 1.16 + 1.42 + 1.16 + 1.16 = 7.22$$

$$\text{log-odds(TGCAGG)} = -0.17 - 0.17 - 0.17 + 1.42 - 0.17 - 0.17 = 0.57$$

$$\text{log-odds(CTTGAT)} = 6 \times -1.17 = -7.02$$

# Outline

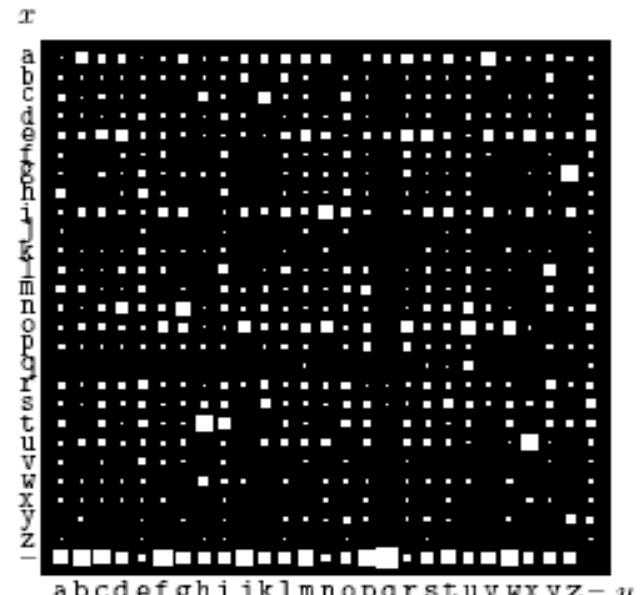
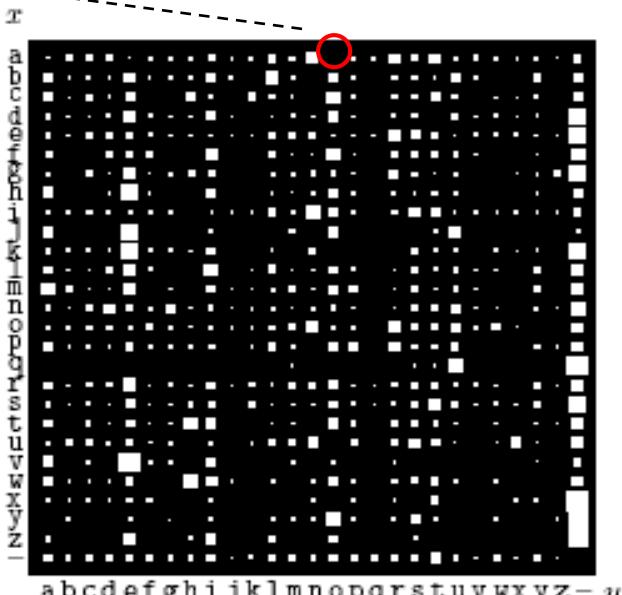
- Regular expressions & weight matrices
- Dependencies & Markov chains
- Hidden Markov models
- HMMs & EM
- Profile HMMs
- Genefinding



# Dependencies: language

Probability (in English) of “o” given that previous letter is “a”

$i$	$a_i$	$p_i$
1	a	0.0575
2	b	0.0128
3	c	0.0263
4	d	0.0285
5	e	0.0913
6	f	0.0173
7	g	0.0133
8	h	0.0313
9	i	0.0599
10	j	0.0006
11	k	0.0084
12	l	0.0335
13	m	0.0235
14	n	0.0596
15	o	0.0689
16	p	0.0192
17	q	0.0008
18	r	0.0508
19	s	0.0567
20	t	0.0706
21	u	0.0334
22	v	0.0069
23	w	0.0119
24	x	0.0073
25	y	0.0164
26	z	0.0007
27	-	0.1928



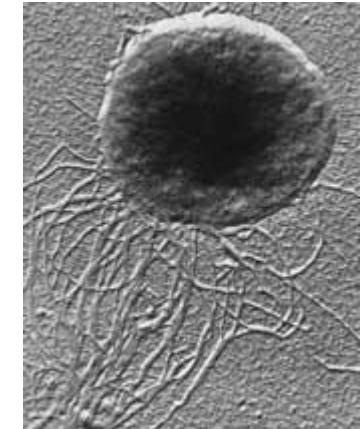
# Dependencies: biology

$P_i$ : probability of nucleotide  $i$

$P_{ij}$ : probability of dinucleotide  $ij$

$$S_{ij} = \frac{P_{ij}}{P_i P_j}$$

independent  $\Leftrightarrow S_{ij} = 1$



*M. jannaschii*

	<i>A</i>	<i>C</i>	<i>G</i>	<i>T</i>
<i>A</i>	1.13	0.73	1.10	0.94
<i>C</i>	1.03	1.37	0.32	1.11
<i>G</i>	1.05	1.12	1.39	0.71
<i>T</i>	0.83	1.05	1.13	1.14

# Markov chains

Sequence:  $q = q_1 q_2 \dots q_N$

$$P(q_N, q_{N-1}, \dots, q_1) = P(q_N | q_{N-1}, \dots, q_1) P(q_{N-1} | q_{N-2}, \dots, q_1) \dots P(q_1) = \prod_{t=2}^N P(q_t | q_{t-1}, \dots, q_1) P(q_1)$$

Only dependent on previous symbol:

$$P(q_N, q_{N-1}, \dots, q_1) = \prod_{t=2}^N P(q_t | q_{t-1}) P(q_1) \quad \text{First-order Markov chain}$$

**state**: value of  $q_j$

**transition probability**:  $P(q_t = j | q_{t-1} = i)$



# Markov chains: language

Zero-order approximation (symbols independent but with frequencies of English text).

OCRO HLI RGWR NMIELWIS EU LL NBNSEBYA TH EEI  
ALHENHTPA OOBTTVA NAH BRL.

First-order Markov (transition probabilities as in English).

ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY  
ACHIN D ILONASIVE TUOOWE AT TEASONARE FUSO  
TIZIN ANDY TOBE SEACE CTISBE.

Second-order Markov (transition probabilities as in English).

IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID  
PONDENOME OF DEMONSTURES OF THE REPTAGIN IS  
REGOACTIONA OF CRE.

C.E. Shannon (1948)



# Markov chains: language

Zero-order word approximation. Words are chosen independently but with their appropriate frequencies.

REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN  
DIFFERENT NATURAL HERE HE THE A IN CAME THE TOOF TO  
EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE  
THESE.

First-order Markov (on words). Word transition probabilities are as in English.

THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER  
THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER  
METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD  
THE PROBLEM FOR AN UNEXPECTED.

C.E. Shannon (1948)



# Markov chain: graphical representation

Two states:  $x$  and  $y$

**Matrix**

$$A = \begin{matrix} & x & y \\ x & \begin{pmatrix} 0.2 & 0.8 \\ 0.7 & 0.3 \end{pmatrix} \\ y & & \end{matrix}$$

**Graph:** arrows for transition probability

```
graph LR; x((x)) -- "0.2" --> x; x -- "0.7" --> y((y)); y -- "0.3" --> y; y -- "0.8" --> x;
```

$a_{ij}$  : transition probability from  $i$  to  $j$

Generative model (example):  $xyyxxyyyxxxyxyxx\dots$

$$P(xyyxy) = P(x)P(y|x)P(y|y)P(x|y)P(y|x) = P(x) \times 0.8 \times 0.3 \times 0.7 \times 0.8$$

# Markov chain: graphical representation (2)

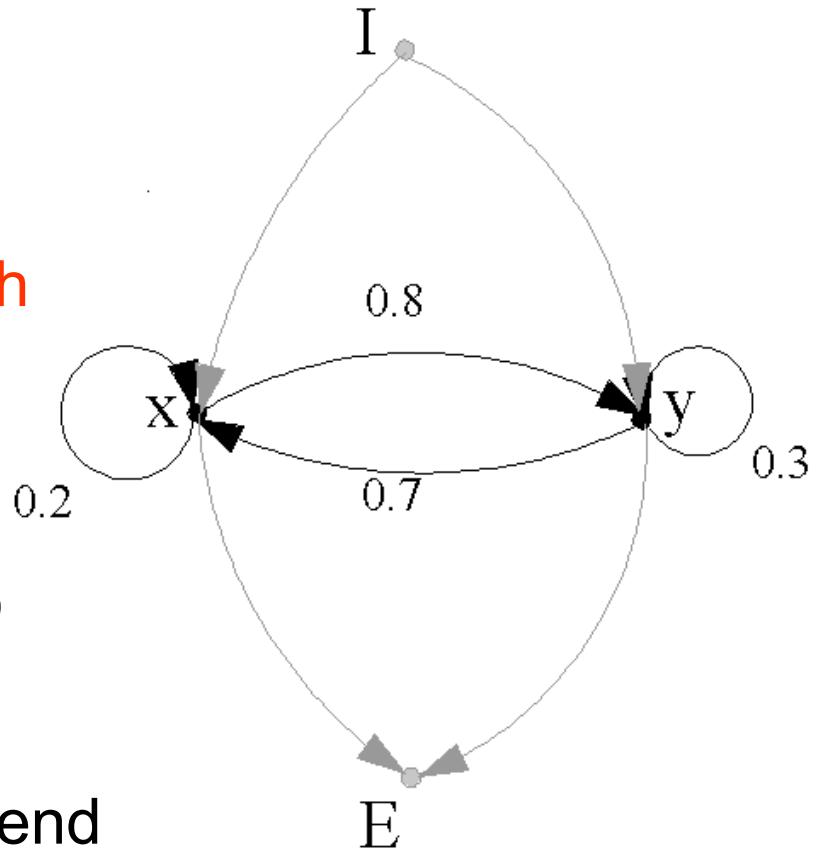
Two states:  $x$  and  $y$

Add **begin** state  $q_0$

$$P(q_N, q_{N-1}, \dots, q_1) = \prod_{t=1}^N P(q_t | q_{t-1})$$

and **end** state  $q_{N+1}$  to model end  
of sequence

Graph



# Markov chain: estimation

$a_{ij}$  : transition probability from  $i$  to  $j$

Estimation: simply by counting

$$a_{ij} = \frac{\# \text{ of } t \text{ such that } q_{t-1} = i, q_t = j}{\# \text{ of } t \text{ such that } q_{t-1} = i}$$

Begin state:

$$a_{0i} = \frac{\# \text{ of } t \text{ such that } q_t = i}{N}$$

# Markov chains: log-odds

Sequence:  $x = x_1 x_2 \dots x_N$

$A, B$  : Markov chains for class  $A$  and  $B$ , respectively

$$\begin{aligned} \log\left(\frac{P(x \mid \text{class } A)}{P(x \mid \text{class } B)}\right) &= \log\left(\frac{\prod_{t=1}^N P_A(x_t \mid x_{t-1})}{\prod_{t=1}^N P_B(x_t \mid x_{t-1})}\right) = \sum_{t=1}^N \log\left(\frac{P_A(x_t \mid x_{t-1})}{P_B(x_t \mid x_{t-1})}\right) \\ &= \sum_{t=1}^N \log\left(\frac{a_{x_{t-1}, x_t}^A}{a_{x_{t-1}, x_t}^B}\right) \end{aligned}$$

# Markov chains: limitations

For biological sequences:

- Mononucleotide repeats (due to polymerase slippage) are more frequent than predicted by Markov chain  
Reason: probability of  $d$  consecutive  $i$ 's is  $(a_{ii})^{d-1}(1-a_{ii})$  (geometric distribution)
- Codon (position) biases are not taken into account

# Outline

- Regular expressions & weight matrices
- Dependencies & Markov chains
- **Hidden Markov models**
- HMMs & EM
- Profile HMMs
- Genefinding

# Multiple alignment

Sequence ensemble as before but now with some insertions and gaps

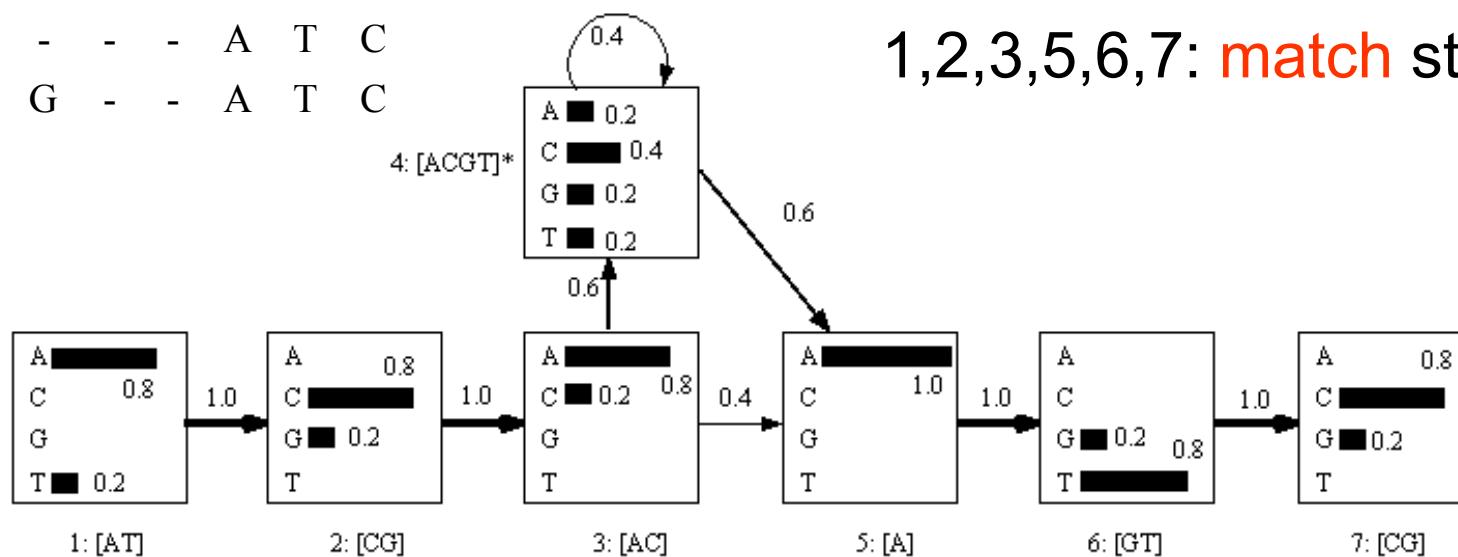
A	C	A	-	-	-	A	T	G
T	C	A	A	C	T	A	T	C
A	C	A	C	-	-	A	G	C
A	G	A	-	-	-	A	T	C
A	C	C	G	-	-	A	T	C

regular expression: [AT][CG][AC][ACGT]\*A[GT][CG]

insertions and gaps ←

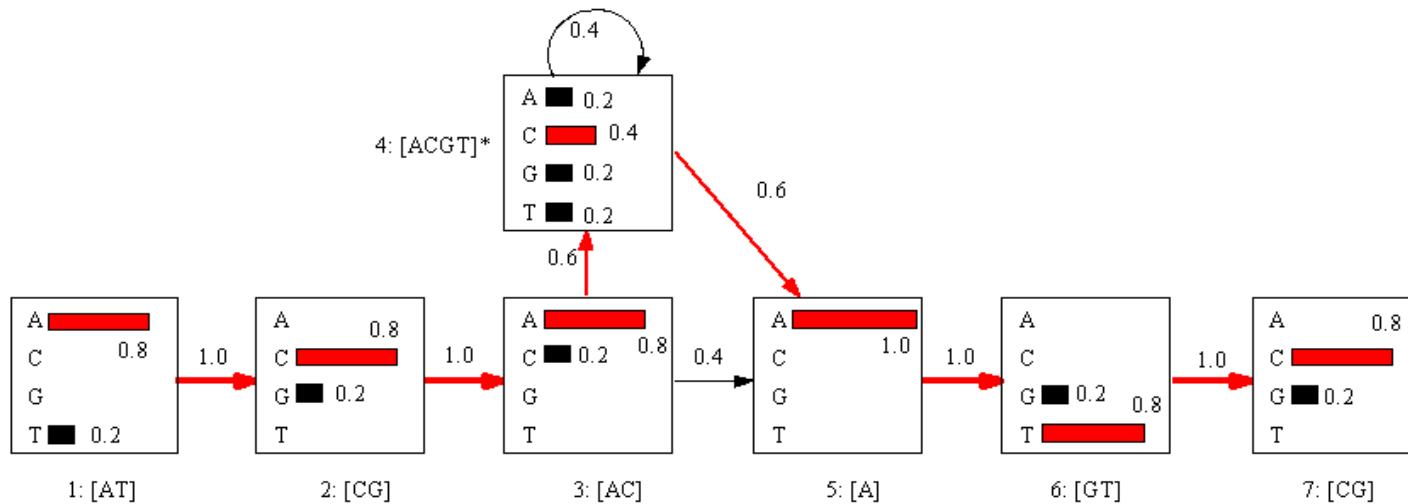
# A different representation

A	C	A	-	-	-	A	T	G
T	C	A	A	C	T	A	T	C
A	C	A	C	-	-	A	G	C
A	G	A	-	-	-	A	T	C
A	C	C	G	-	-	A	T	C



mix of weight matrices and Markov chains

# Probability of consensus sequence



$$P(\text{ACACATC}) = 0.8 \times 1 \times 0.8 \times 1 \times 0.8 \times 0.6 \times 0.4 \times 0.6 \times 1 \times 1 \times 0.8 \times 1 \times 0.8 = 0.047$$

Markov chain: one state = one symbol

Here: C can be generated by states 2,3,4 or 7 – states are **hidden**

# Hidden Markov models

Alphabet  $K$  of (observed) symbols

States:  $Q = \{0, 1, 2, \dots, S\}$       0: begin state (non-emitting)

transition probability:

$$a_{ij} = P(q_t = j \mid q_{t-1} = i) \quad 0 \leq i, j \leq S$$

emission probability:

$$b_i(x) = P(x \mid i) \quad x \in K, 1 \leq i \leq S$$

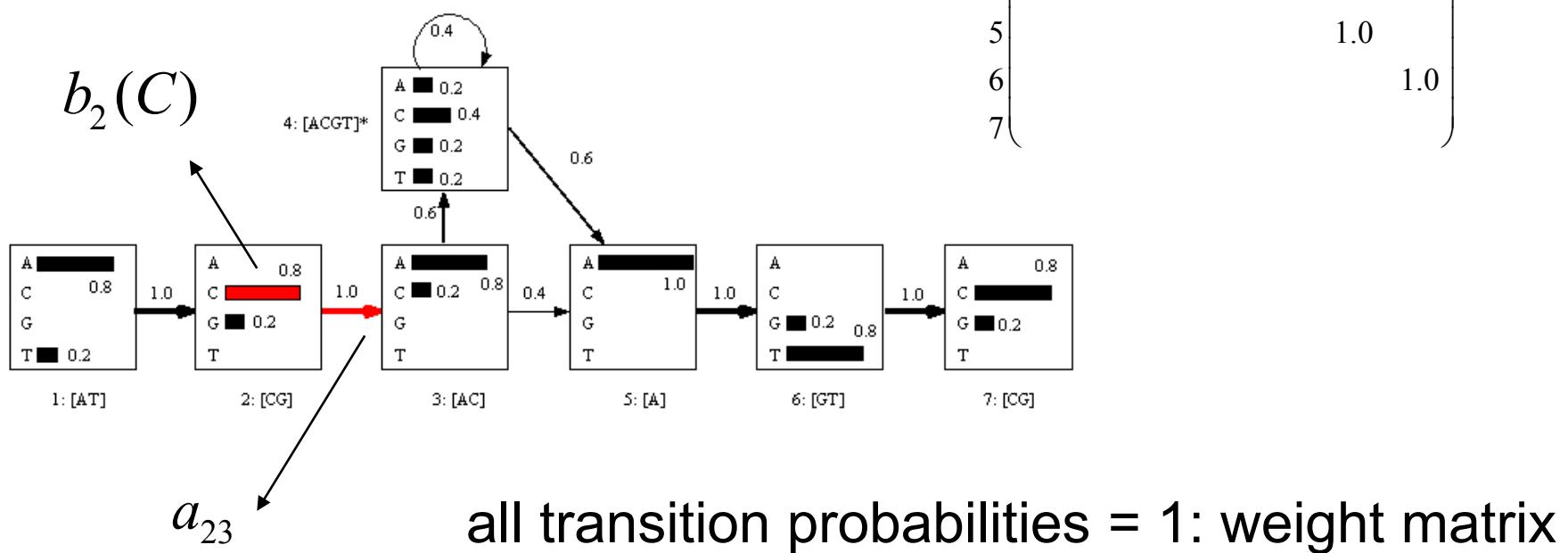


→ probability of emitting symbol  $x$  in state  $i$

# Hidden Markov models (2)

Alphabet of 4 symbols {A,C,G,T}

$$Q = \{1, 2, \dots, 7\}$$



# HMM: three problems

Evaluation: probability of an observed sequence, given the model, e.g., to calculate odds ratio.

Decoding: optimal state sequence for an observed sequence

Estimation: of transition and emission probabilities from a given set of sequences

# HMM evaluation: known state sequence

State sequence:  $Q = q_0 q_1 q_2 \dots q_N$

Observed sequence:  $x = x_1 x_2 \dots x_N$

$$P(x, Q) = P(x | Q)P(Q) \xrightarrow{\text{Markov}} P(Q) = \prod_{t=1}^N P(q_t | q_{t-1})$$

$$\downarrow$$
$$P(x | Q) = P(x_N | x_{N-1}, \dots, x_1, Q)P(x_{N-1} | x_{N-2}, \dots, x_1, Q)\dots P(x_1 | Q) = \prod_{t=1}^N P(x_t | q_t)$$

$$P(x, Q) = \prod_{t=1}^N P(q_t | q_{t-1}) \prod_{t=1}^N P(x_t | q_t) = \prod_{t=1}^N a_{q_{t-1}, q_t} \prod_{t=1}^N b_{q_t}(x_t)$$

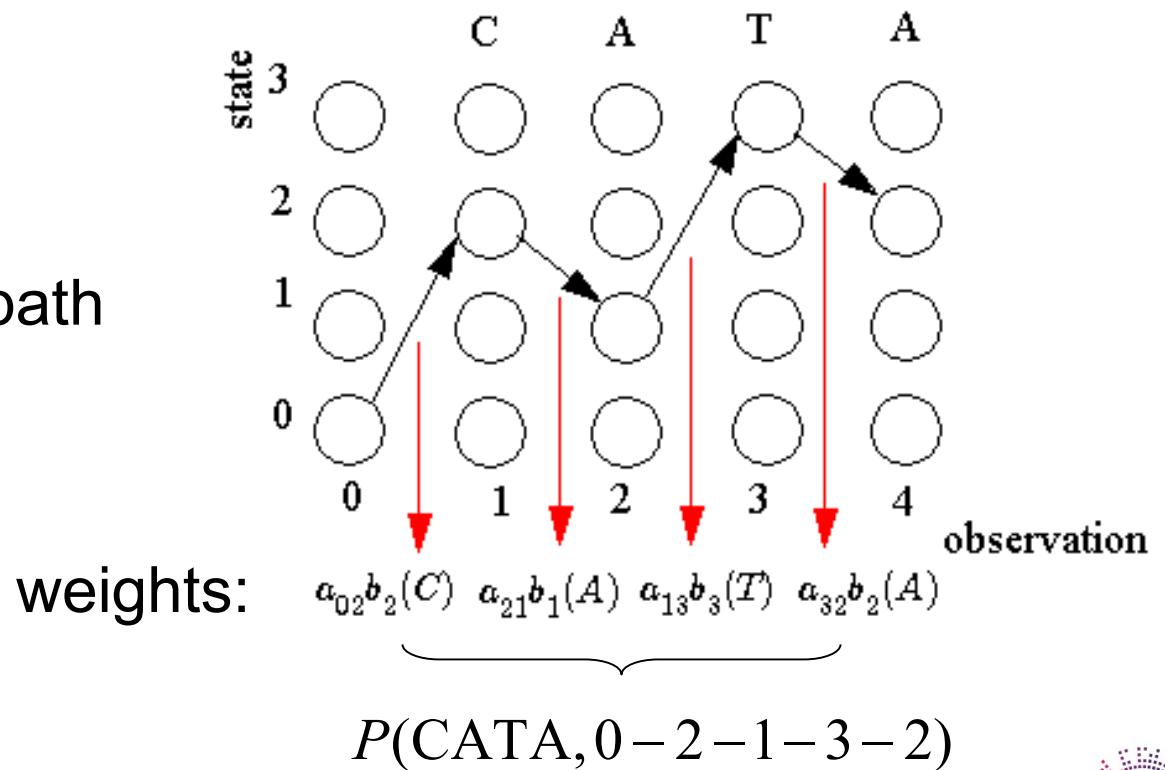
$$P(\text{ACACATC}) = 0.8 \times 1 \times 0.8 \times 1 \times 0.8 \times 0.6 \times 0.4 \times 0.6 \times 1 \times 1 \times 0.8 \times 1 \times 0.8 = 0.047$$



# HMM evaluation: graphical representation on a trellis

$$P(x, Q) = \prod_{t=1}^N P(q_t | q_{t-1}) \prod_{t=1}^N P(x_t | q_t) = \prod_{t=1}^N a_{q_{t-1}, q_t} \prod_{t=1}^N b_{q_t}(x_t)$$

state sequence = path



# HMM evaluation: forward algorithm

State sequence unknown:  $P(x) = \sum_Q P(x, Q)$

Sum over all paths through trellis:  $\sim S^N$  state sequences!

Smarter:  $P(x) = \sum_{i=0}^S P(x, q_N = i) = \sum_{i=0}^S \alpha(N, i)$  forward variable

$\alpha(t, i) = P(x_1 x_2 \dots x_t, q_t = i)$ , that is, probability of having observed  $x_1 x_2 \dots x_t$  and being in state  $i$  at step  $t$

# HMM evaluation: forward algorithm (2)

$$\alpha(t, i) = P(x_1 x_2 \dots x_t, q_t = i)$$

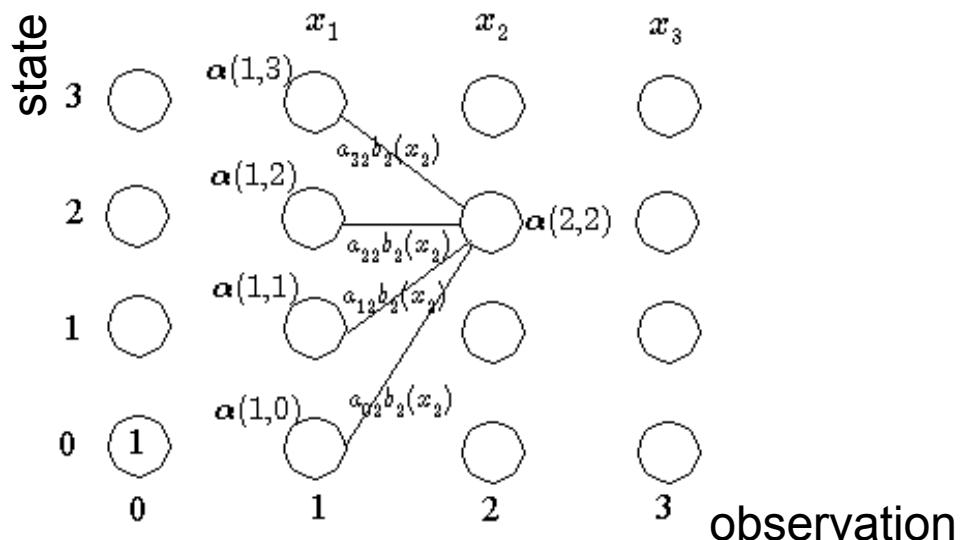
**initialization:**  $\alpha(0, 0) = 1$ ,  $\alpha(0, j) = 0$   $1 \leq j$

**recursion** :  $\alpha(t, i) = \sum_j \alpha(t-1, j) a_{ji} b_i(x_t)$   $1 \leq t \leq N, 0 \leq i, j \leq S$

$$P(x) = \sum_{i=0}^S \alpha(N, i)$$

Complexity:  $S \times N$

$$\alpha(2, 2) = \sum_{j=0}^3 \alpha(1, j) a_{j2} b_2(x_2)$$



# Forward algorithm: proof

$$x_{1..t} = x_1 x_2 \dots x_t$$

$$1 \leq t \leq N, 0 \leq i, j \leq S$$

$$\begin{aligned}\alpha(t, i) &= P(x_{1..t}, q_t = i) = \sum_j P(x_{1..t}, q_{t-1} = j, q_t = i) \\ &= \sum_j P(x_{1..t-1}, q_{t-1} = j) P(x_t, q_t = i \mid x_{1..t-1}, q_{t-1} = j)\end{aligned}$$

Observed symbol and the state depend only on previous state:

$$\begin{aligned}&= \sum_j P(x_{1..t-1}, q_{t-1} = j) P(x_t, q_t = i \mid q_{t-1} = j) \\ &= \sum_j \alpha(t-1, j) P(q_t = i \mid q_{t-1} = j) P(x_t \mid q_t = i) \\ &= \sum_j \alpha(t-1, j) a_{ji} b_i(x_t)\end{aligned}$$

↑                          recursion



# HMM: three problems

Evaluation: probability of an observed sequence, given the model, e.g., to calculate odds ratio.

**Decoding**: optimal state sequence for an observed sequence

Estimation: of transition and emission probabilities from a given set of sequences

# HMM decoding: Viterbi algorithm

Decoding: find state sequence which best explains observed sequence.

Viterbi: best = most probable

$$V(x) = \max_Q P(Q | x) = \max_Q \frac{P(x, Q)}{P(x)} = \max_Q P(x, Q)$$
$$V(x) = \max_Q P(x, Q) = \max_i \left[ \max_{Q_{0..N-1}} P(x, Q_{0..N-1}, q_N = i) \right] = \max_i [v(N, i)]$$
$$v(t, i) = \max_{Q_{0..t-1}} [P(x_{1..t}, Q_{0..t-1}, q_t = i)]$$

probability of having observed  $x_1 x_2 \dots x_t$  along most probable path ending in state  $i$  at step  $t$

# HMM decoding: Viterbi algorithm (2)

$$v(t, i) = \max_{Q_{0..t-1}} [P(x_{1..t}, Q_{0..t-1}, q_t = i)]$$

**initialization :**  $v(0, 0) = 1$  ,  $v(0, j) = 0$   $1 \leq j$

**recursion :**  $v(t, i) = \max_j [v(t-1, j)a_{ji}]b_i(x_t)$   $1 \leq t \leq N, 0 \leq i, j \leq S$

$$p(t, i) = \operatorname{argmax}_j [v(t-1, j)a_{ji}]$$

**end :**  $V(x) = \max_i [v(N, i)]$

$$q_N^* = \operatorname{arg max}_i [v(N, i)]$$

**backtracking :**  $q_t^* = p(t+1, q_{t+1}^*)$   $0 \leq t \leq N-1$



# Dishonest casino: Viterbi

Casino switches between a fair (F) die and a loaded (L) die

transition probabilities

$$A = F \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0 & 0.95 & 0.05 \\ L & 0.1 & 0.9 \end{pmatrix}$$

1 2 3 4 5 6

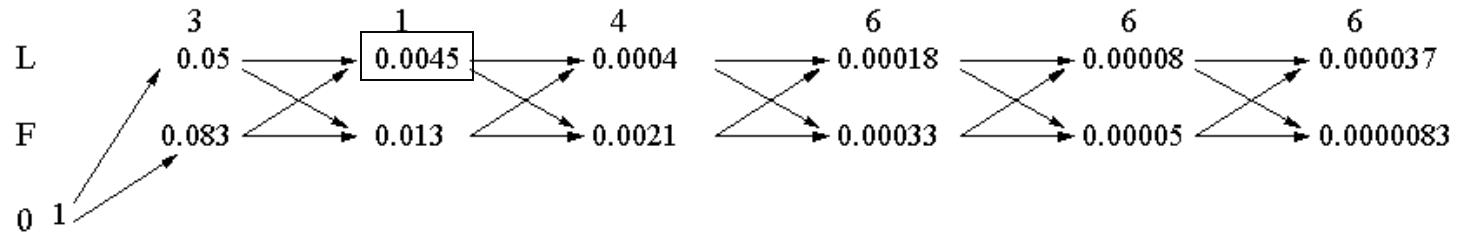
$$F : \frac{1}{6} \quad \frac{1}{6} \quad \frac{1}{6} \quad \frac{1}{6} \quad \frac{1}{6} \quad \frac{1}{6}$$

emission probabilities

$$L : \frac{1}{10} \quad \frac{1}{10} \quad \frac{1}{10} \quad \frac{1}{10} \quad \frac{1}{10} \quad \boxed{\frac{1}{2}}$$

observations: 3-1-4-6-6-6

Viterbi



$$\begin{aligned} v(2, L) &= \max[v(1, F)a_{FL}b_L(1), v(1, L)a_{LL}b_L(1)] \\ &= \max[(0.083 \times 0.05 \times 0.1, 0.05 \times 0.9 \times 0.1)] = 0.0045 \end{aligned}$$

# Dishonest casino: Viterbi (2)

Casino switches between a fair (F) die and a loaded (L) die

transition probabilities

$$A = F \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0 & 0.95 & 0.05 \\ L & 0.1 & 0.9 \end{pmatrix}$$

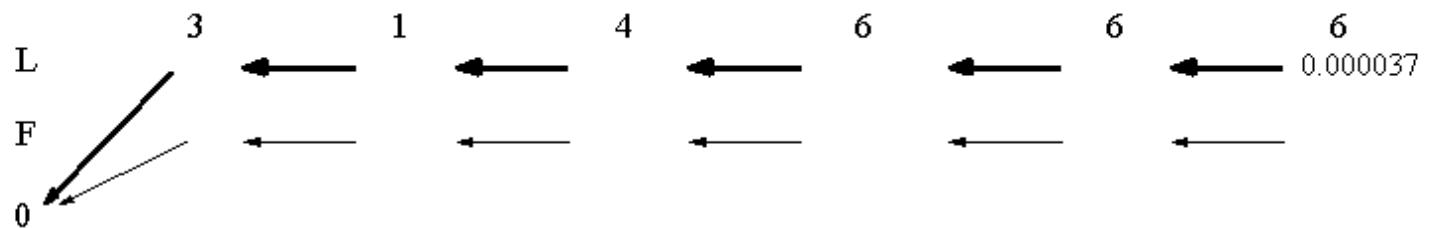
1 2 3 4 5 6

$$F : \frac{1}{6} \quad \frac{1}{6} \quad \frac{1}{6} \quad \frac{1}{6} \quad \frac{1}{6} \quad \frac{1}{6}$$

emission probabilities

$$L : \frac{1}{10} \quad \frac{1}{10} \quad \frac{1}{10} \quad \frac{1}{10} \quad \frac{1}{10} \quad \boxed{\frac{1}{2}}$$

## Backtracking



Optimal state sequence: 0-L-L-L-L-L-L

# Outline

- Regular expressions & weight matrices
- Dependencies & Markov chains
- Hidden Markov models
- HMMs & EM
- Profile HMMs
- Genefinding



# HMM: three problems

Evaluation: probability of an observed sequence, given the model, e.g., to calculate odds ratio.

Decoding: optimal state sequence for an observed sequence

**Estimation:** of transition and emission probabilities from a given set of sequences

# HMM: estimation

Sequences:  $\{x^1, \dots, x^n\}$

Likelihood:  $P(x^1, \dots, x^n | \theta) = \prod_{i=1}^n P(x^i | \theta)$  state sequence

$$= \prod_{i=1}^n \sum_Q P(x^i, Q | \theta)$$

Log-likelihood:  $\sum_{i=1}^n \log \sum_Q P(x^i, Q | \theta)$

same solution since log is  
monotonic

Maximization of this log-likelihood is difficult because of sum over hidden (state) variables

# HMM estimation: EM

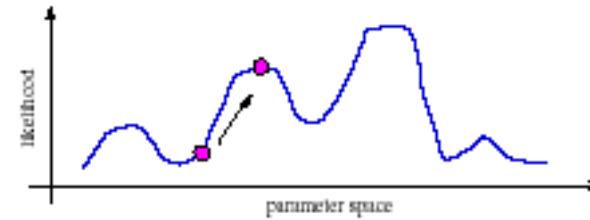
1. If we **know** the state sequence, parameter estimation is easy: just counting as in Markov chains
2. Can estimate state path using the forward-backward algorithm (not shown)
3. EM: estimate (probability of) states, then estimate parameters, re-estimate the states etc.

This maximizes the likelihood (see MoG)

# HMM estimation: remarks

See references in lecture notes for EM for HMM (aka Baum-Welch algorithm) in full detail

EM converges only to a **local** maximum of the likelihood.  
Good initial values are important!



How to choose the structure of an HMM? Black magic ...

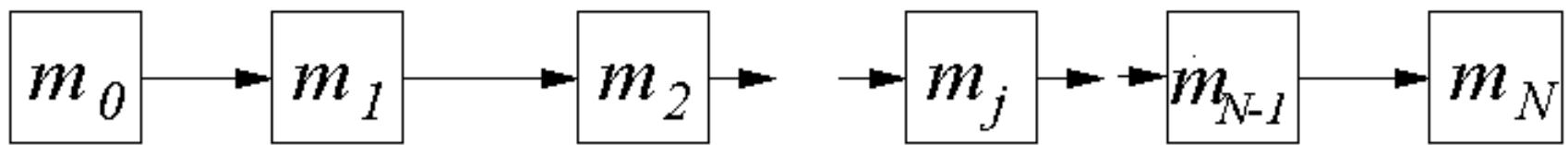
# Outline

- Regular expressions & weight matrices
- Dependencies & Markov chains
- Hidden Markov models
- HMMs & EM
- Profile HMMs
- Genefinding

# Profile HMMs

A	C	A	-	-	-	A	T	G
T	C	A	A	C	T	A	T	C
A	C	A	C	-	-	A	G	C
A	G	A	-	-	-	A	T	C
A	C	C	G	-	-	A	T	C

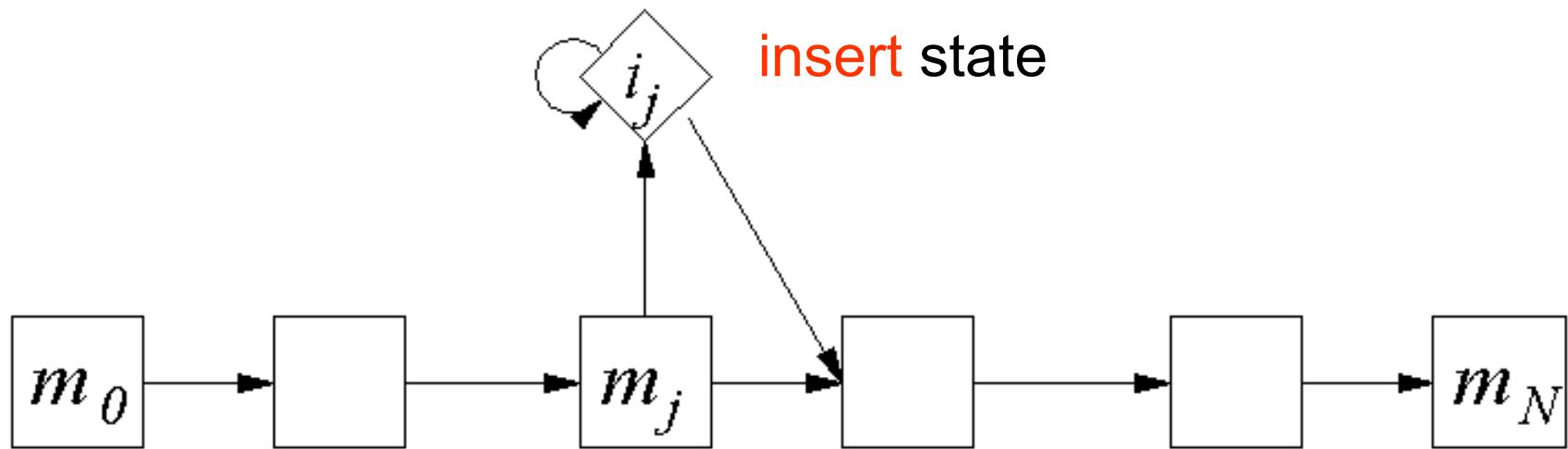
We saw that a weight matrix can be represented as a very simple HMM



transition probabilities = 1

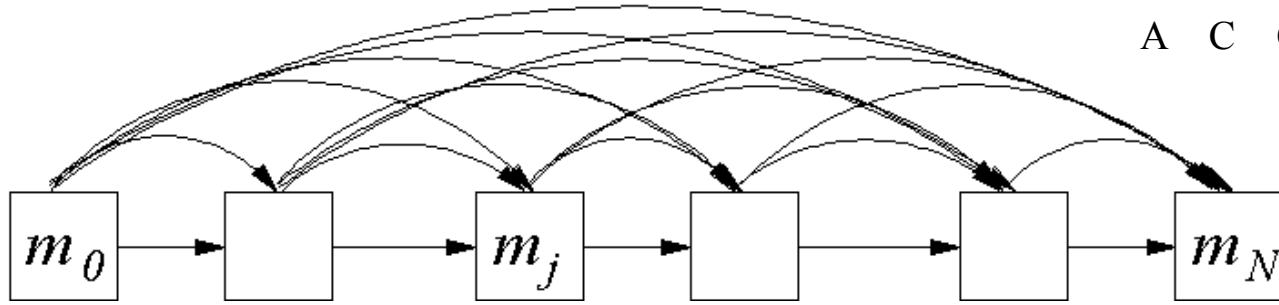
# Profile HMMs: insertions

A	C	A	-	-	-	A	T	G
T	C	A	A	C	T	A	T	C
A	C	A	C	-	-	A	G	C
A	G	A	-	-	-	A	T	C
A	C	C	G	-	-	A	T	C



Model insertion(s) between position  $j$  and  $j+1$

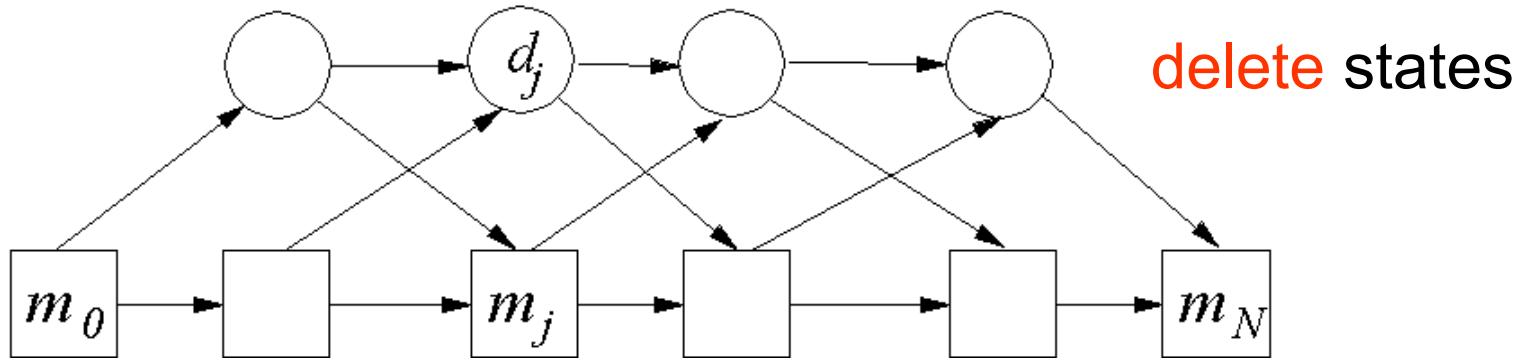
# Profile HMMs: deletions



A	C	A	-	-	-	A	T	G
T	C	A	A	C	T	A	T	C
A	C	A	C	-	-	A	G	C
A	G	A	-	-	-	A	T	C
A	C	C	G	-	-	A	T	C

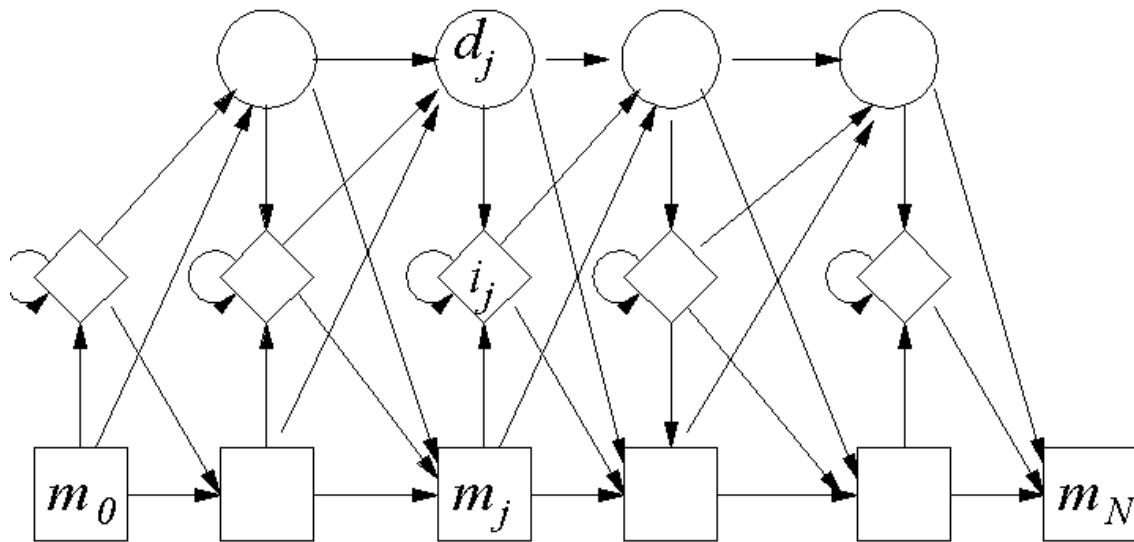
Many transitions = many parameters, but limited data

Solution: introduce **silent** (=non-emitting) delete states



# Profile HMMs (2)

Put everything together:



Applications:

- searching for remote homologs (Forward)
- align a protein to a protein family (Viterbi)

A	C	A	-	-	-	A	T	G
T	C	A	A	C	T	A	T	C
A	C	A	C	-	-	A	G	C
A	G	A	-	-	-	A	T	C
A	C	C	G	-	-	A	T	C

<http://pfam.xfam.org/>

# Outline

- Regular expressions & weight matrices
- Dependencies & Markov chains
- Hidden Markov models
- HMMs & EM
- Profile HMMs
- **Genefinding**



# Genefinding

Input: DNA string  $S \in \{A, C, G, T\}^*$

Output: annotation of string  $S$  showing for each nucleotide whether it is coding or non-coding

AAAGCATGCATTTAACGAGTGCATCAGGACTCCATACGTAATGCCG

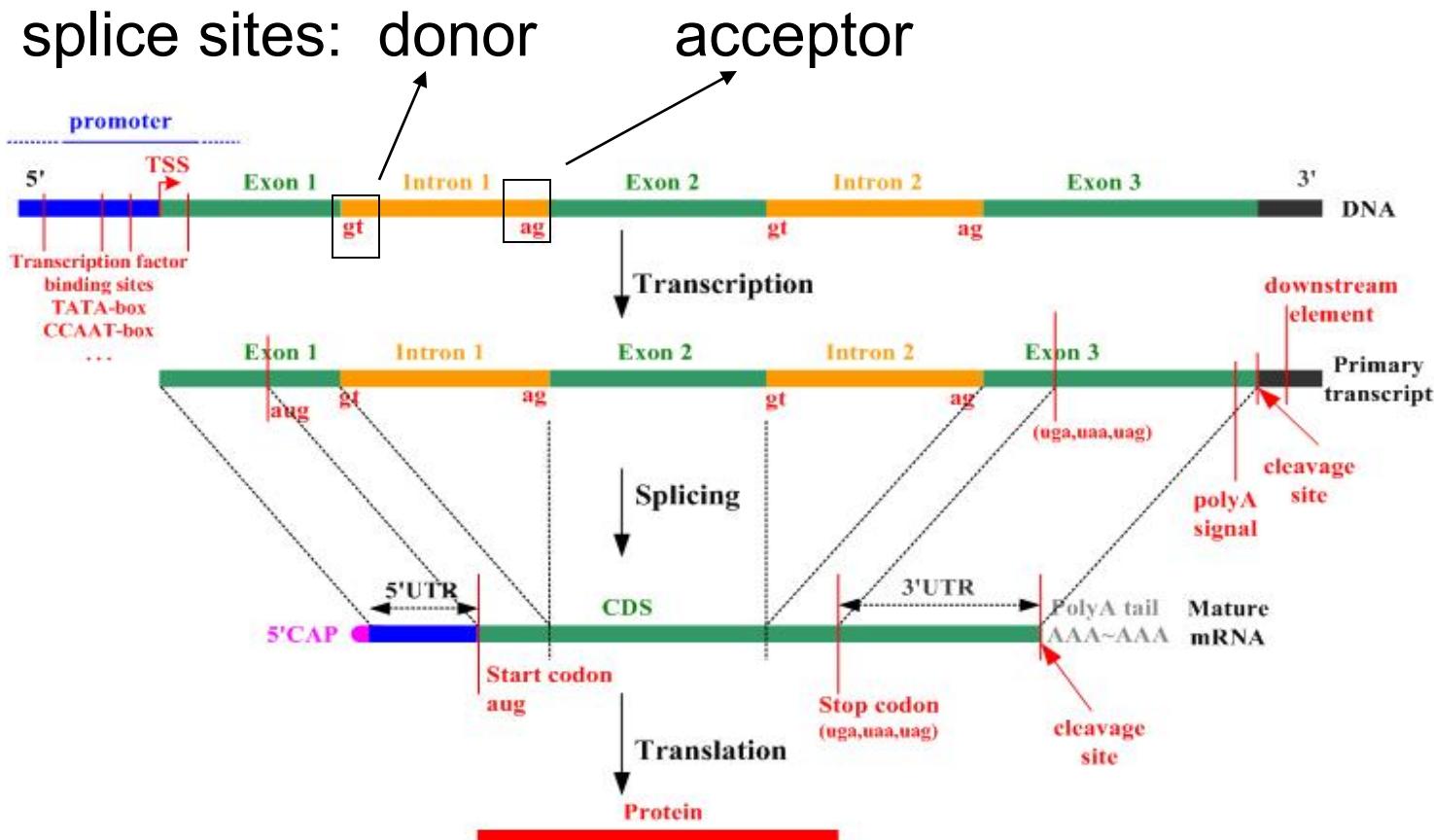


genefinder



AAAGC ATG CAT TTA ACG A GT GCATC AG GA CTC CAT ACG TAA TGCCG

# Genefinding: eukaryotes



More complex than for prokaryotes: lower coding density (<25% instead of >80%), splicing

# Genefinding: many signals

Possible signals: splice sites, promoter, codon bias, polyA site, dinucleotide usage ...

Possible models: everything you've seen before ...

How to **integrate** all these models in one consistent model that can be used for genefinding?

Solution: **HMMs again!**

Building blocks (=states): weight matrices, (inhomogeneous, higher-order, interpolated) Markov chains, ...

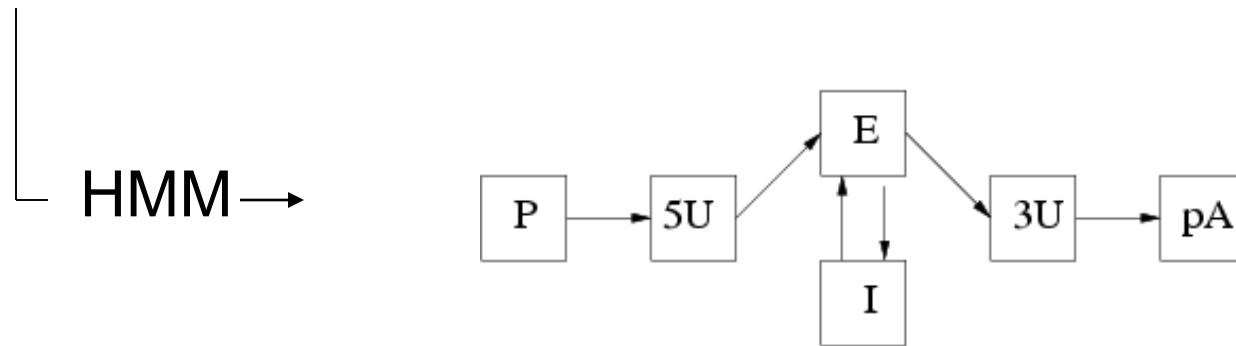
# Genefinding: HMM

Genes have a certain structure/grammar

... exon – intron – exon – intron – exon ...

Regular expression of gene structure:

promoter 5'UTR exon (intron exon)\* 3'UTR polyA



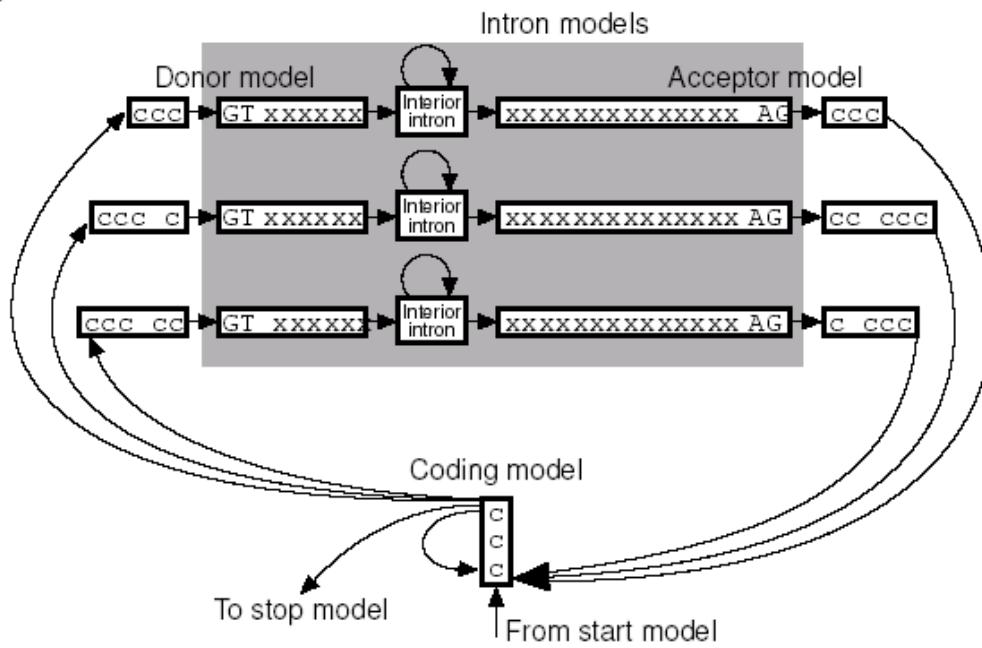
Genefinding = annotation with states: Viterbi

# Genefinding: HMMs & frame consistency

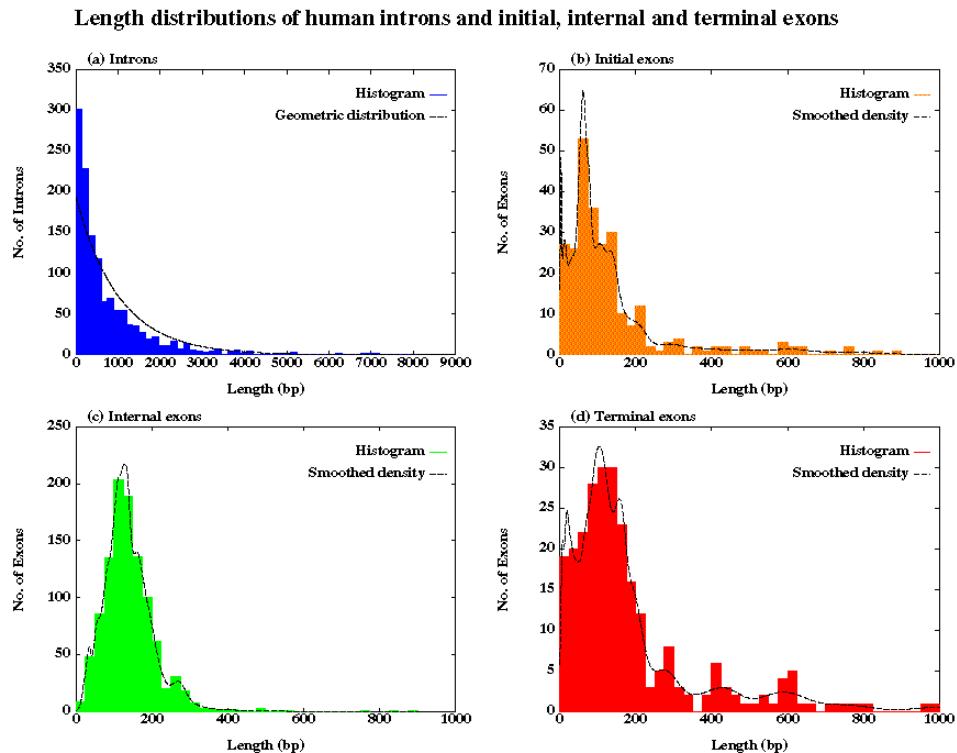
**Frame 1:** AAAGC ATG CAT TTA ACG A GTGCATCAG GA CTC CAT ACG TAA TGCCG

**Frame 2:** AAAGC ATG CAT TTA ACG AG TGCATCAGG A CTC CAT ACG TAA TGCCG

**Frame 3:** AAAGC ATG CAT TTA ACG AGT GCATCAGGA CTC CAT ACG TAA TGCCG



# Length distributions



Standard HMM: length  $\sim$  geometric distribution

Generalized HMM: states emit sequences + **length**

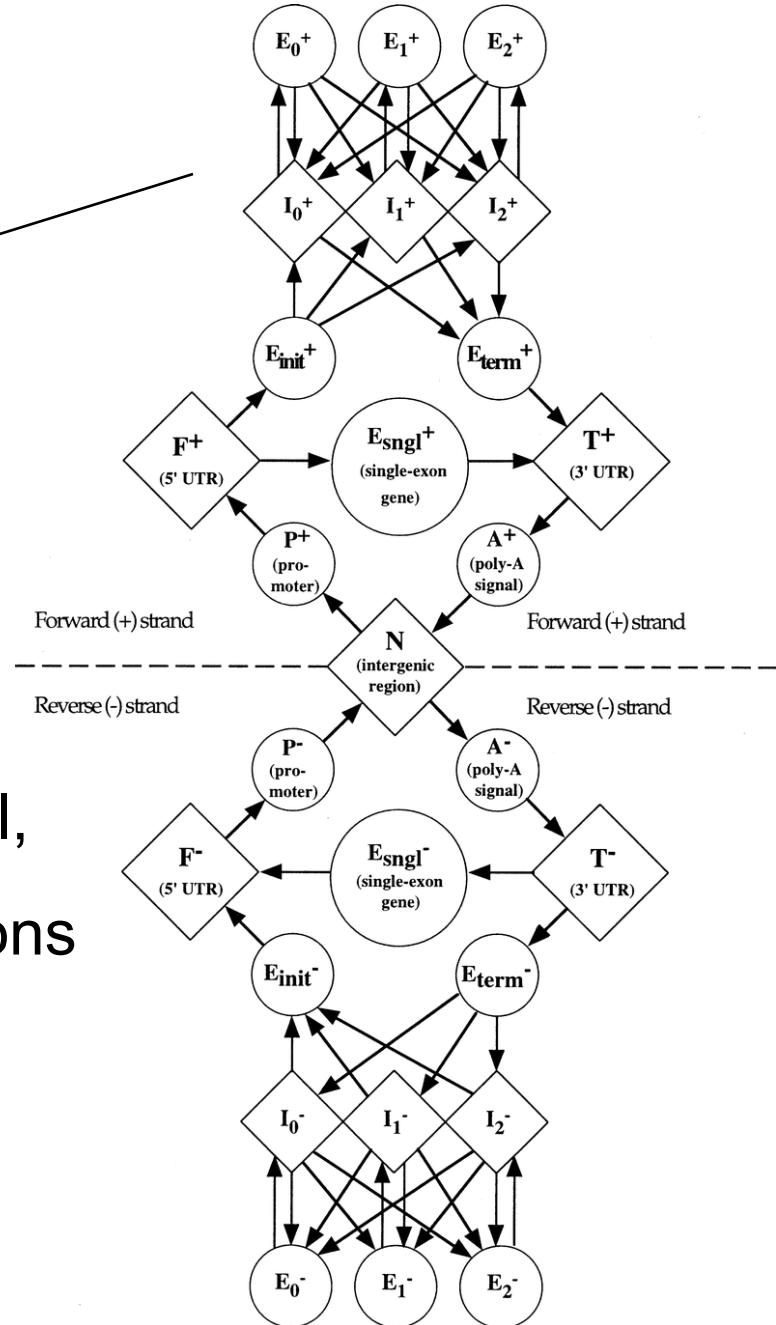
# Genefinding: GenScan

frame-aware

both strands

**Exons:** separate states for initial, terminal, single and internal exons

GenScan: generalized HMM



# Recapitulation

- Hidden Markov models:  
flexible models for modeling sequences
  - *Evaluation*: forward algorithm
  - *Decoding*: Viterbi
  - *Estimation*: EM
- Applications:
  - Genefinding
  - Modeling protein families
  - Segmentation of array CGH data
  - SNP imputation in GWAS
  - Error correction in nanopore sequencing data



**10min break**  
**Exercise 4.18-4.20**