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*

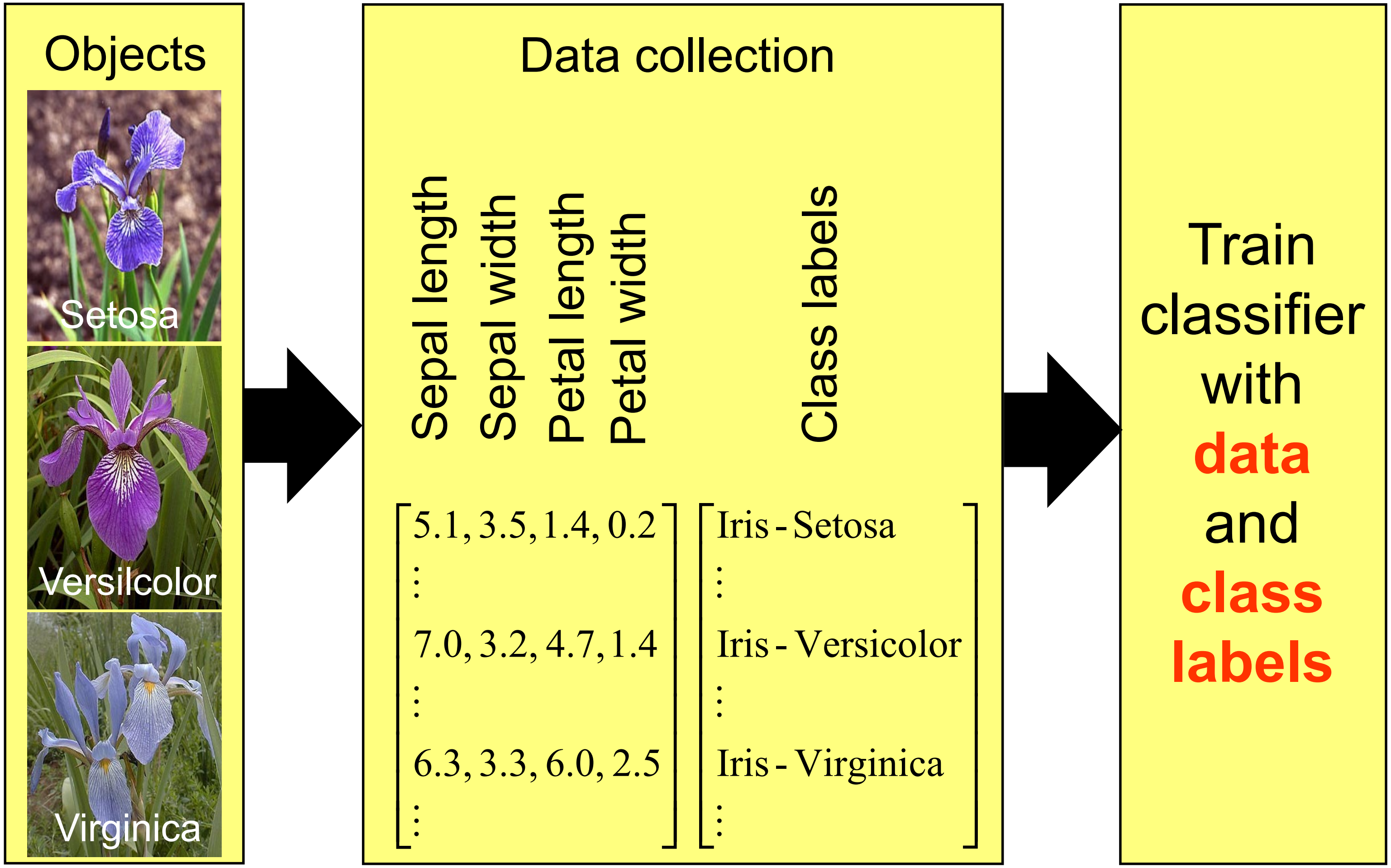
Some material courtesy of Robert Duin and David Tax

Clustering

- Supervised vs. unsupervised learning

- Hierarchical clustering
- Sum-of-squares clustering (k-means)
- Cluster validation
- Mixtures-of-Gaussians clustering (EM algorithm)

Supervised learning



Objects

Data collection

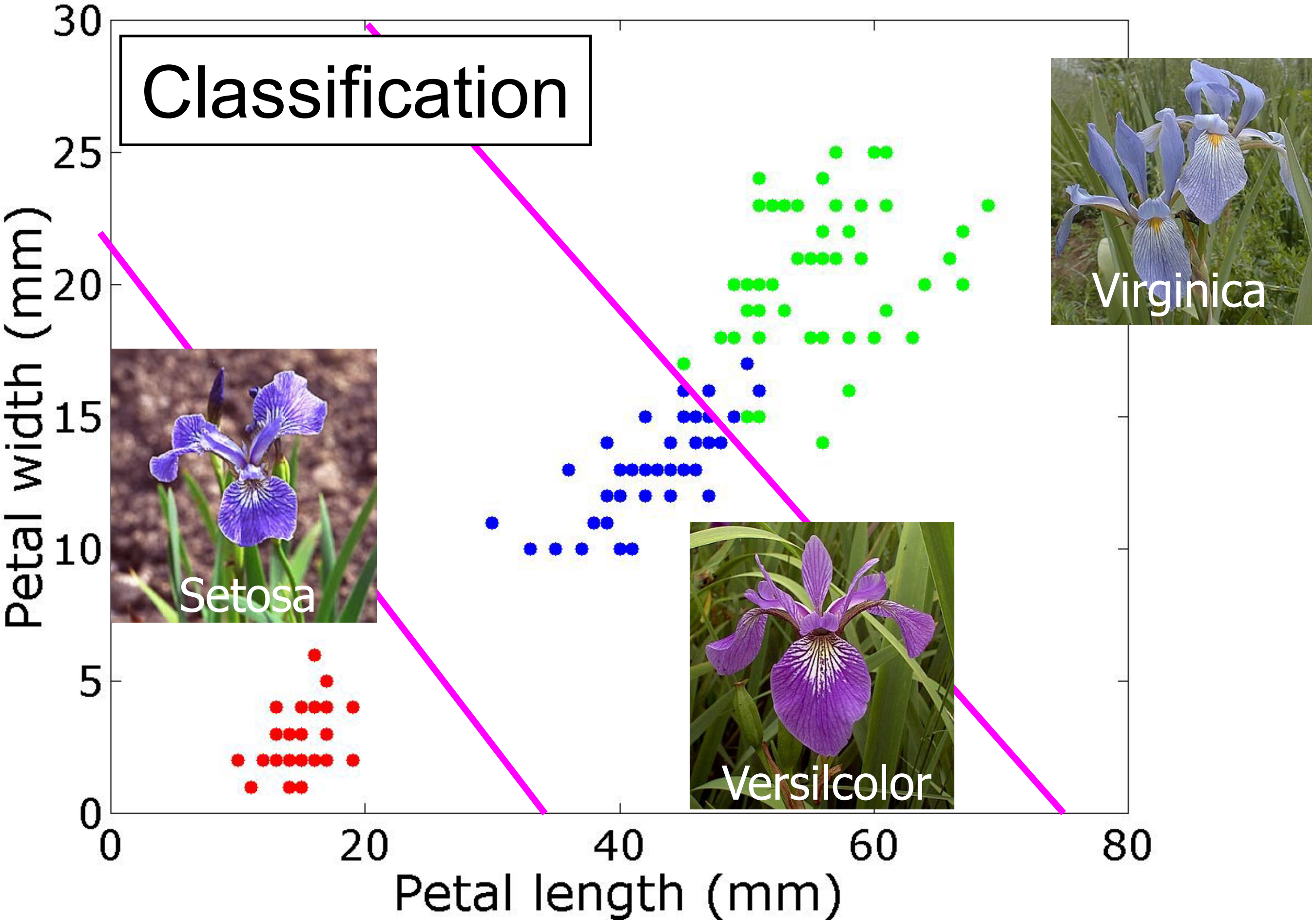
Sepal length, Sepal width, Petal length, Petal width

Class labels

$$\begin{bmatrix} 5.1, 3.5, 1.4, 0.2 \\ \vdots \\ 7.0, 3.2, 4.7, 1.4 \\ \vdots \\ 6.3, 3.3, 6.0, 2.5 \\ \vdots \end{bmatrix} \quad \begin{bmatrix} \text{Iris - Setosa} \\ \vdots \\ \text{Iris - Versicolor} \\ \vdots \\ \text{Iris - Virginica} \\ \vdots \end{bmatrix}$$

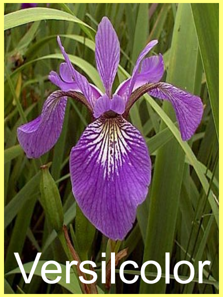
Train classifier with **data** and **class labels**

Supervised learning (2)



Unsupervised learning

Objects



Data collection

Sepal length Sepal width Petal length Petal width

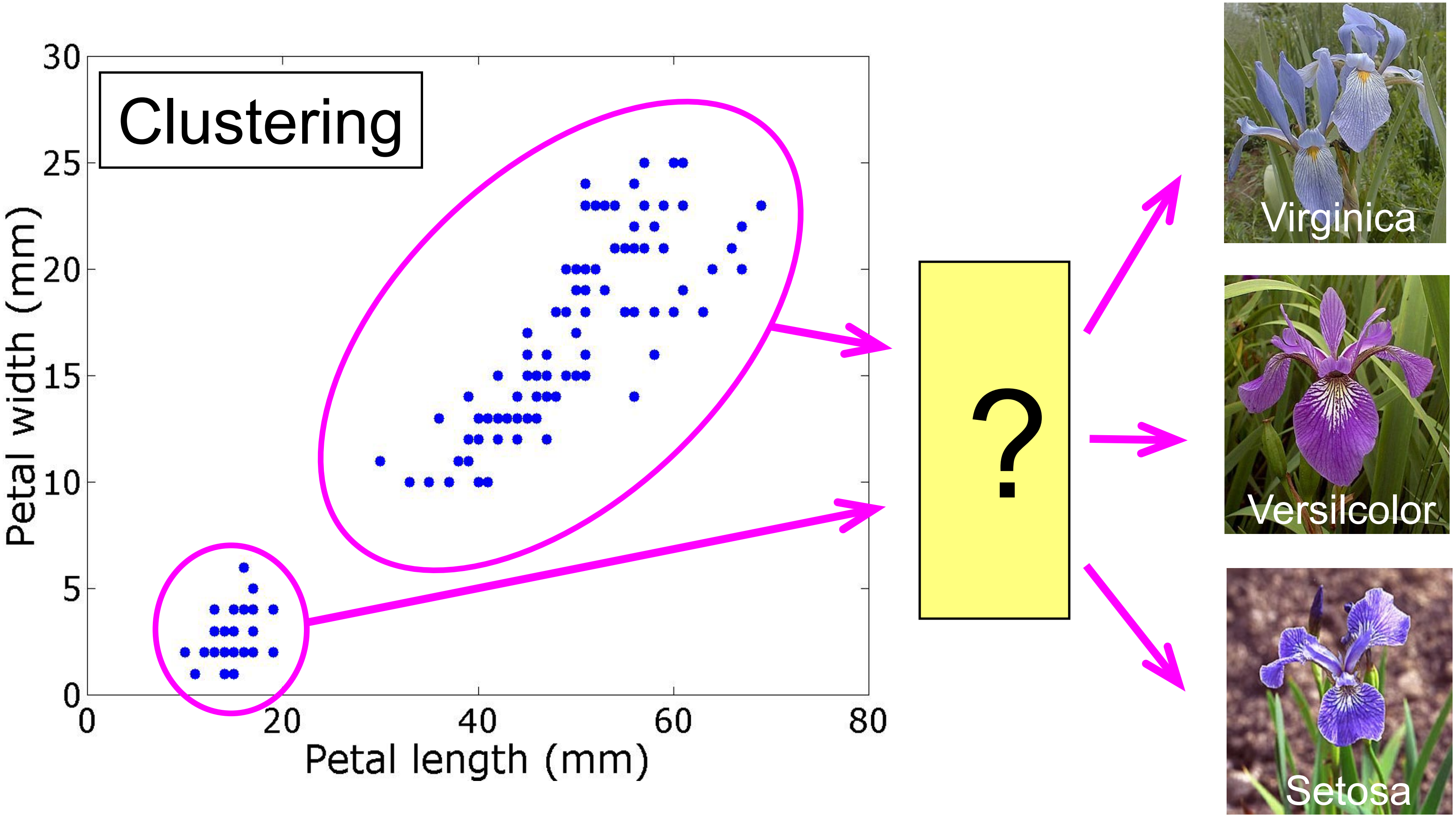
~~Class labels~~

$$\begin{bmatrix} 5.1, 3.5, 1.4, 0.2 \\ \vdots \\ 7.0, 3.2, 4.7, 1.4 \\ \vdots \\ 6.3, 3.3, 6.0, 2.5 \\ \vdots \end{bmatrix} \cancel{\begin{bmatrix} \text{Iris - Setosa} \\ \vdots \\ \text{Iris - Versicolor} \\ \vdots \\ \text{Iris - Virginica} \\ \vdots \end{bmatrix}}$$

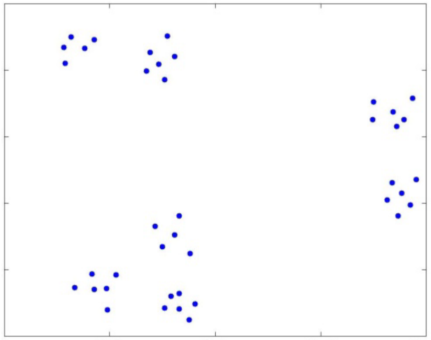
Explore data **without class labels:**

PCA, MDS, clustering

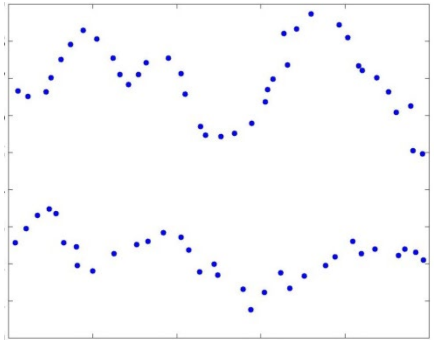
Unsupervised learning (2)



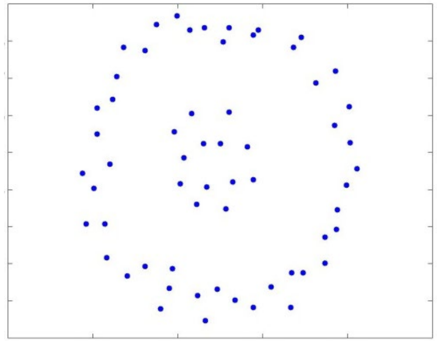
What is a cluster?



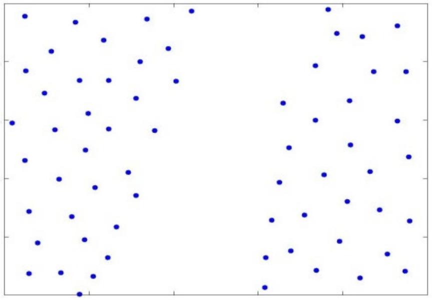
Shape: compact, convex
Separation: large



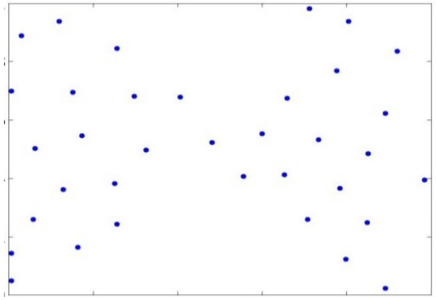
Shape: strings
Separation: large?



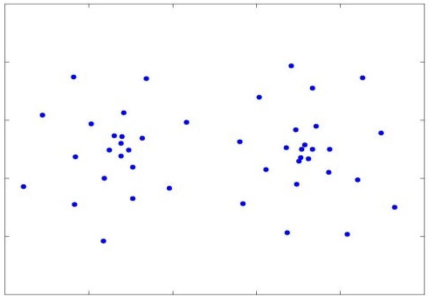
Shape: convex and circular
Separation: large?



Shape: ?
Separation: large?



Shape: loose, convex
Separation: small



Shape: loose, convex
Separation: small

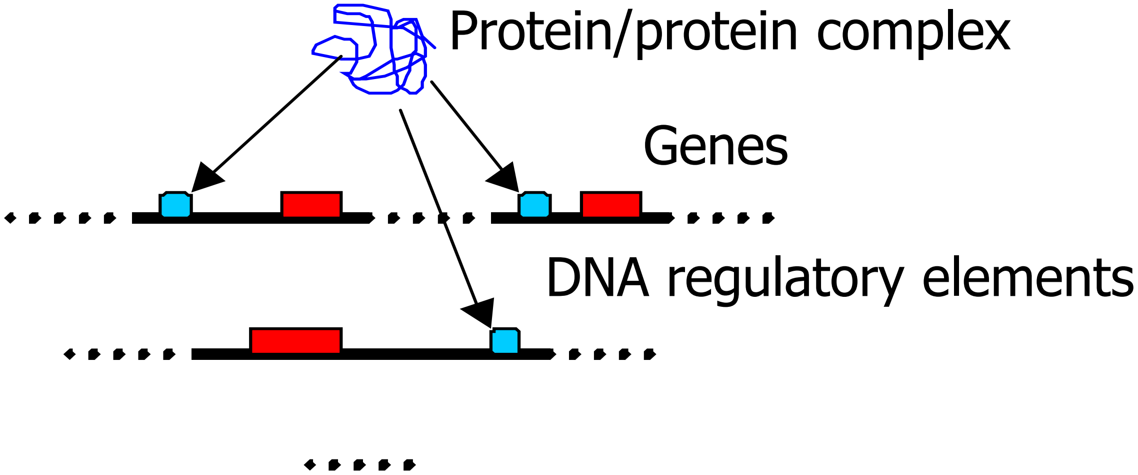
What is a cluster? (2)

- 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?

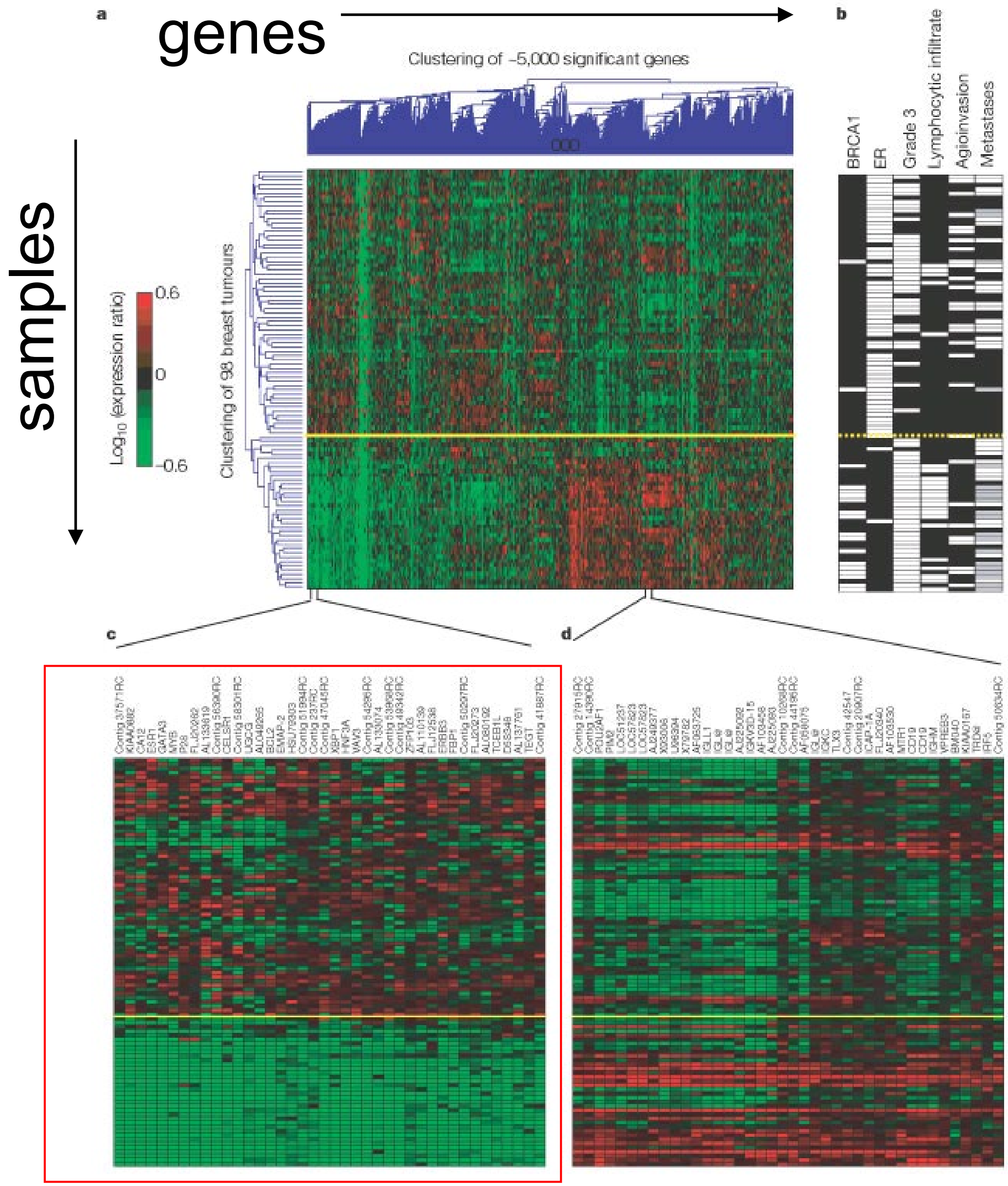
- Clustering gene expression data:
- Genes: similar ~ co-expression ~ co-regulation ~

  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

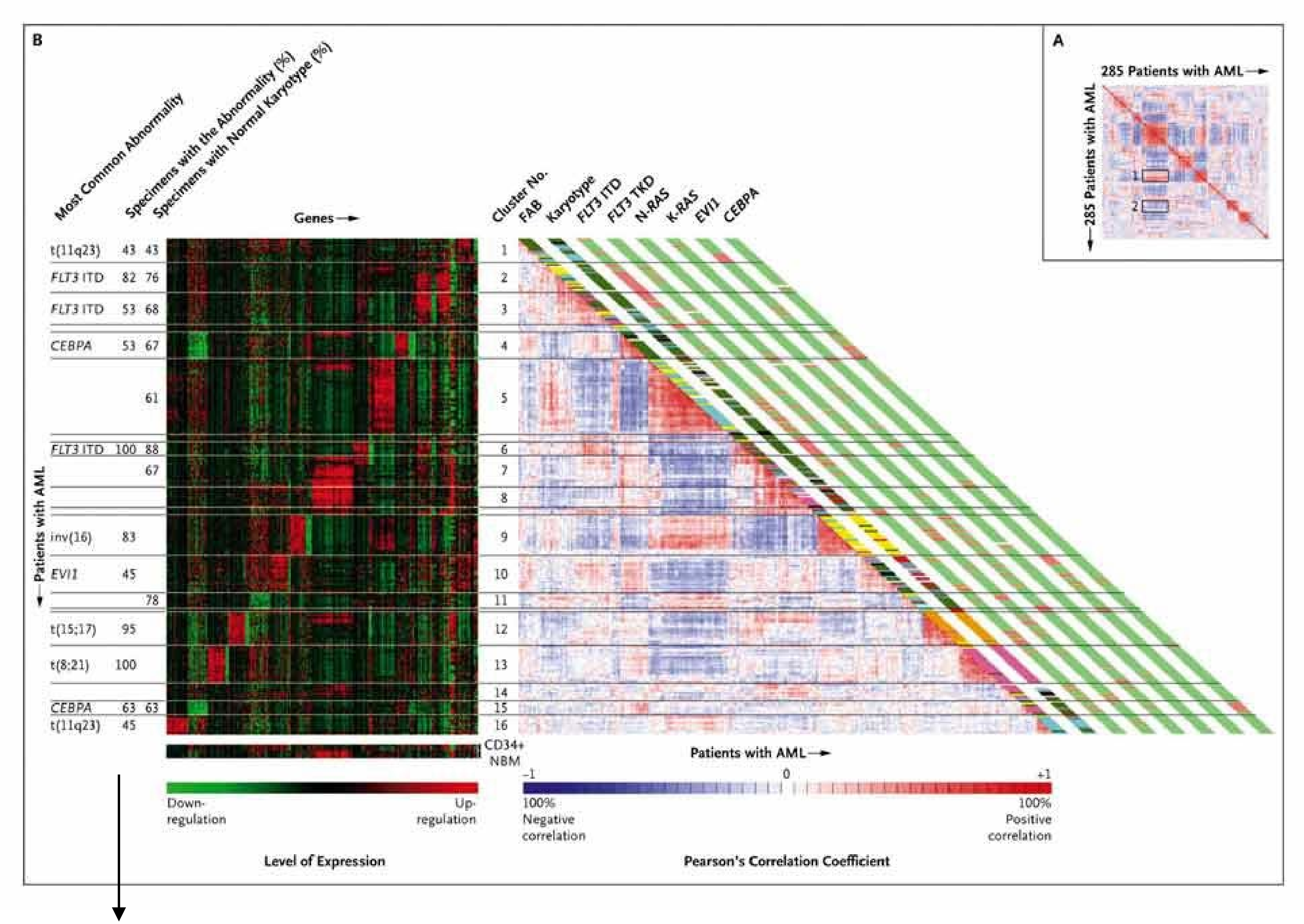
histopathological data

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)

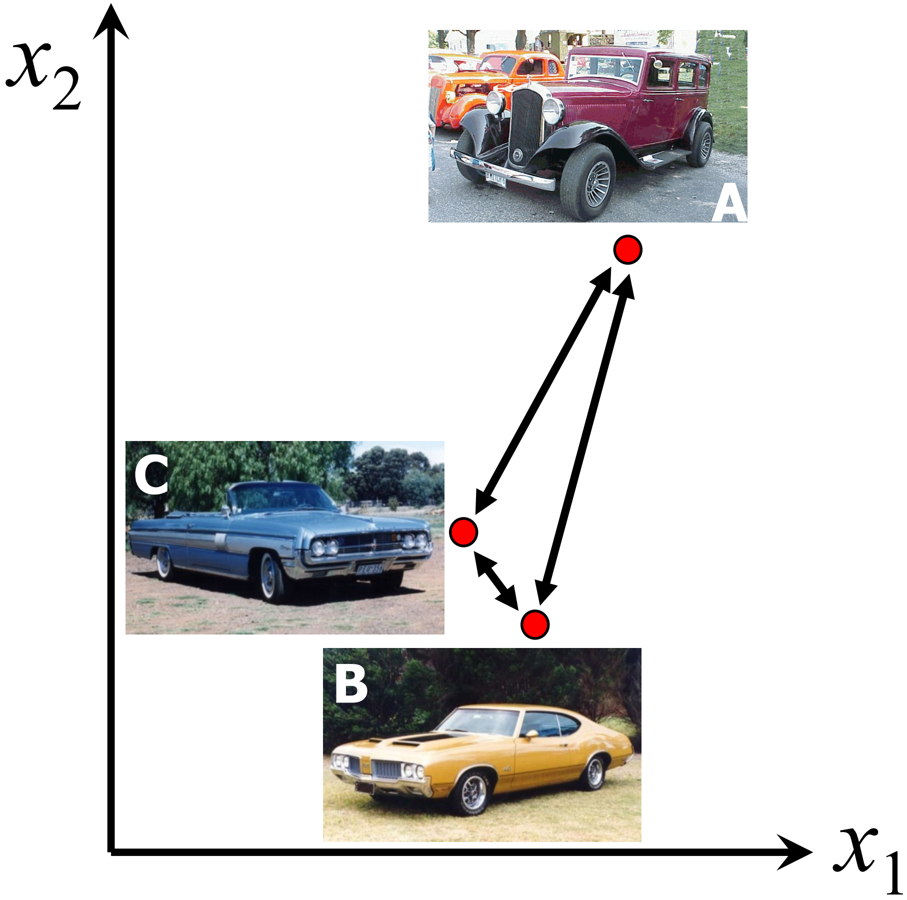
Example: samples

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



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

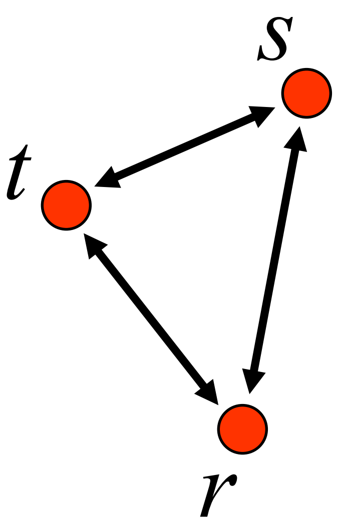
Dissimilarity measures (2)

- 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*

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

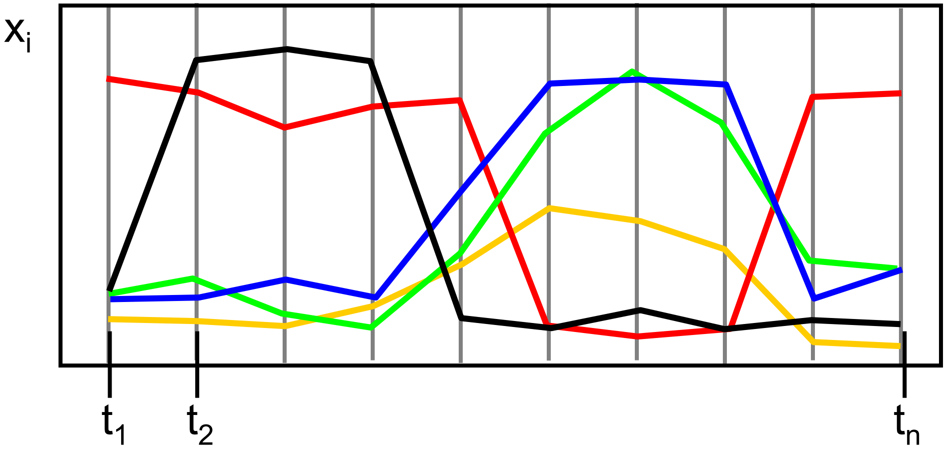
- Most often used: Euclidean distance (metric)

Dissimilarity measures (3)

- Example: time series data

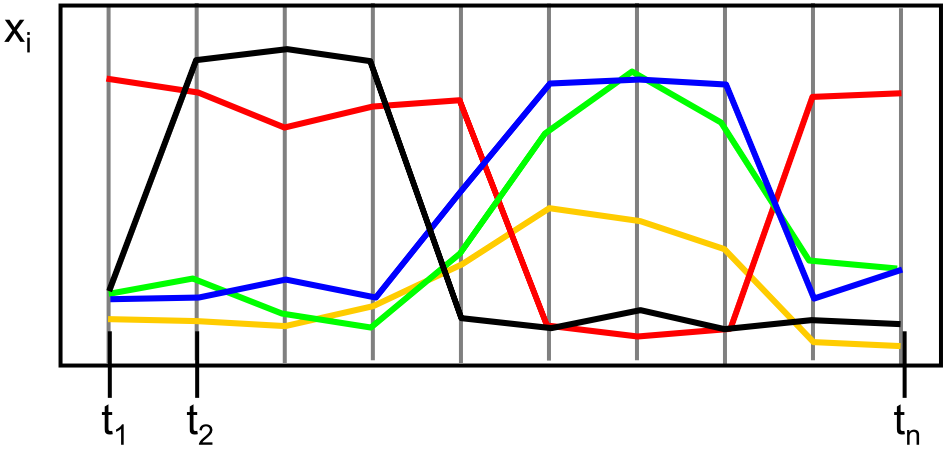
Euclidean distance

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



Dissimilarity measures (3)

- Example:
  time series data



Euclidean distance
match exact shape

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

d(🔵,🟢) < d(🔵,🟡)

d(🔵,🟢) << d(🔵,🔴)

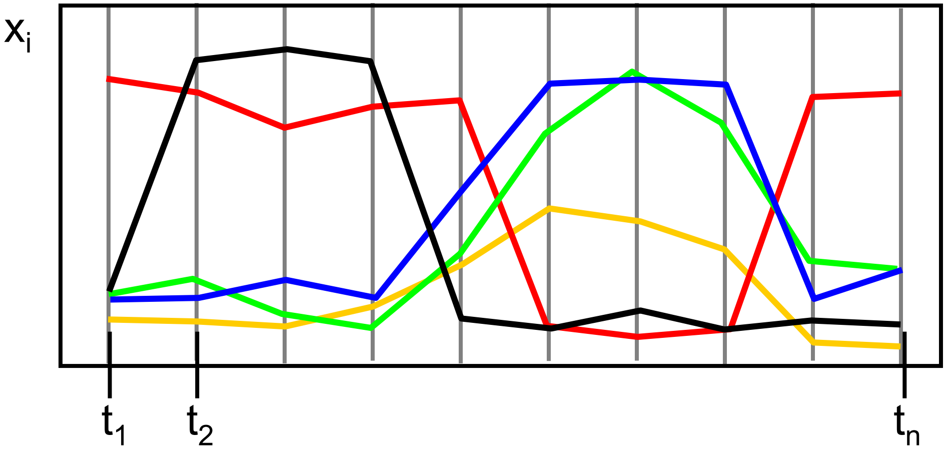
d(🔵,🟢) << d(🔵,⚫)

Dissimilarity measures (3)

- Example: time series data

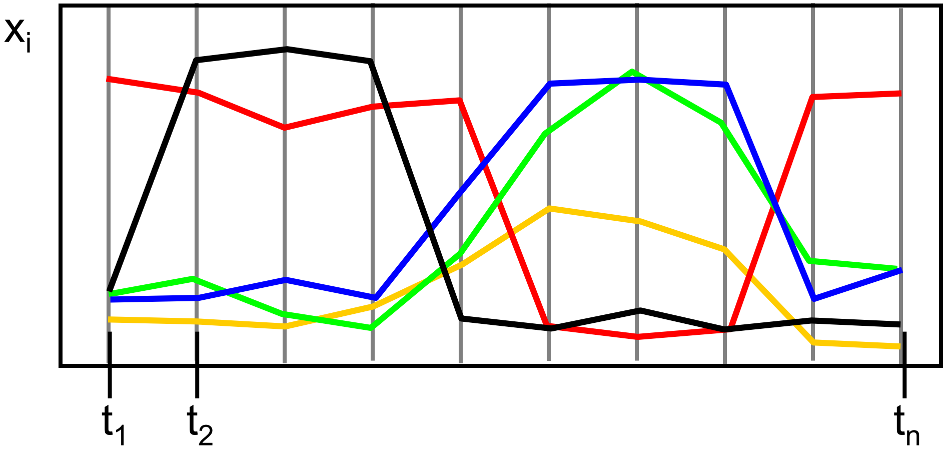
Euclidean distance
match exact shape

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



Dissimilarity measures (3)

- Example: time series data



Euclidean distance
match exact shape

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

Pearson correlation
ignore amplitude

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

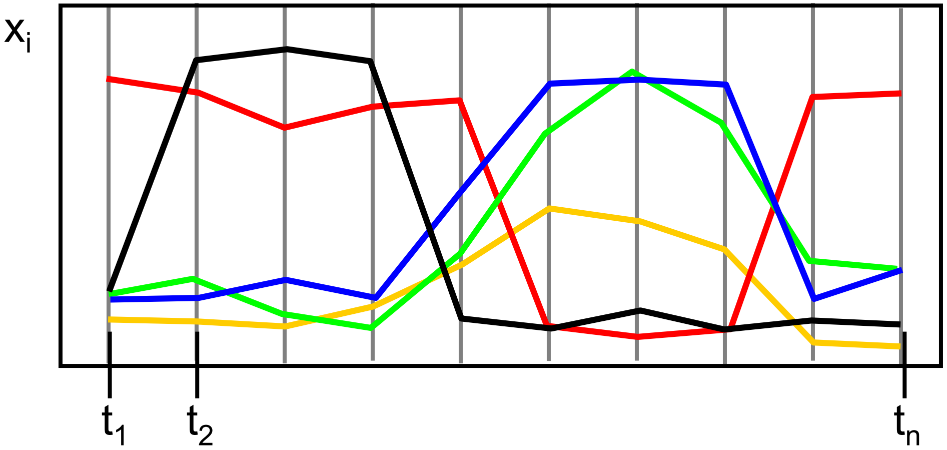
$$1 - \rho_{ij}$$

d(🔵,🟢) < d(🔵,🟡)
d(🔵,🟢) << d(🔵,🔴)
d(🔵,🟢) << d(🔵,⚫)

d(🔵,🟢) ≈ d(🔵,🟡)
d(🔵,🟢) << d(🔵,🔴)
d(🔵,🟢) << d(🔵,⚫)

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(🔵,🟢) < d(🔵,🟡)
d(🔵,🟢) << d(🔵,🔴)
d(🔵,🟢) << d(🔵,⚫)

Pearson correlation
ignore amplitude

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

$$1 - \rho_{ij}$$

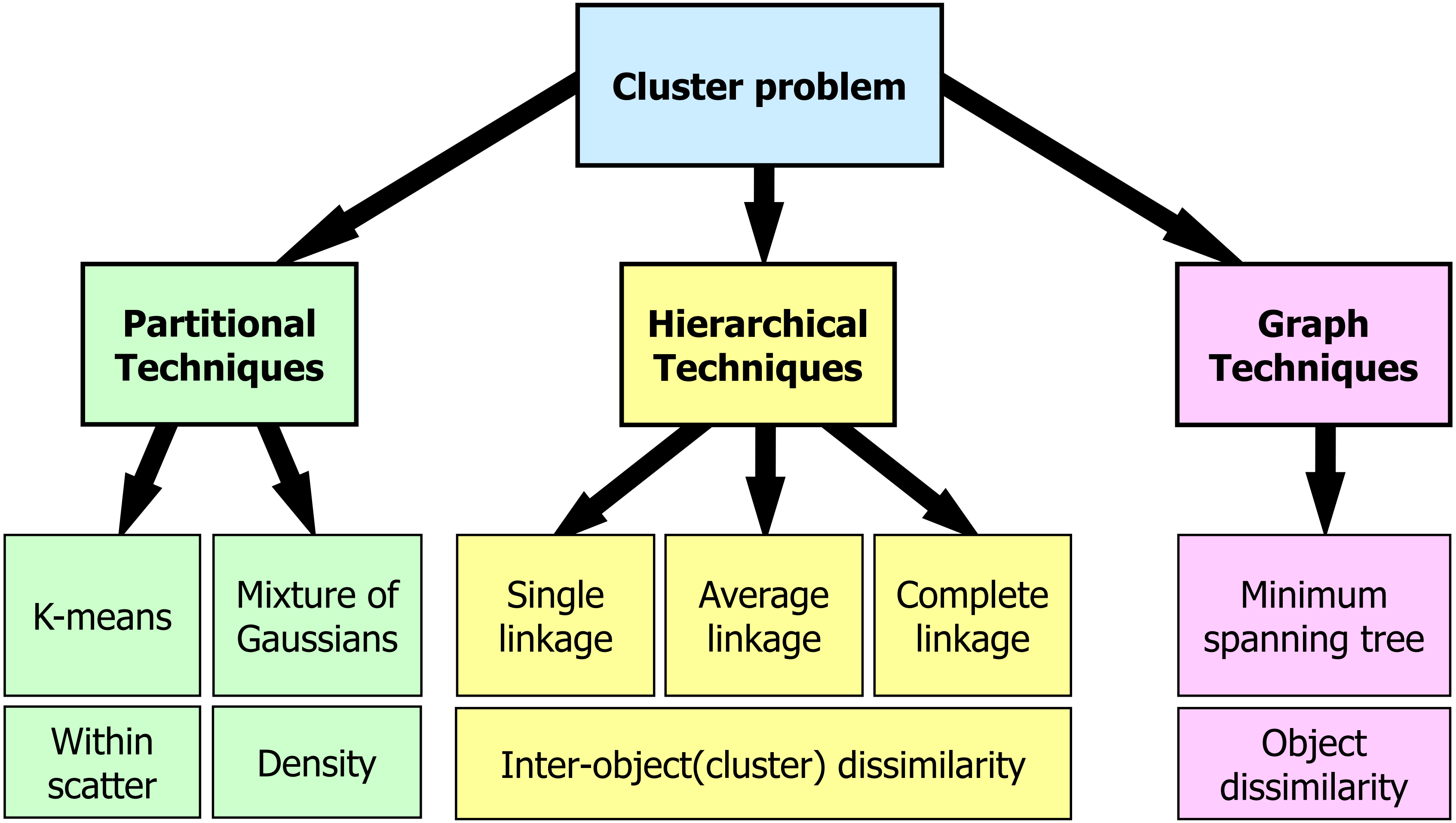
d(🔵,🟢) ≈ d(🔵,🟡)
d(🔵,🟢) << d(🔵,🔴)
d(🔵,🟢) << d(🔵,⚫)

Absolute correlation
ignore amplitude & sign

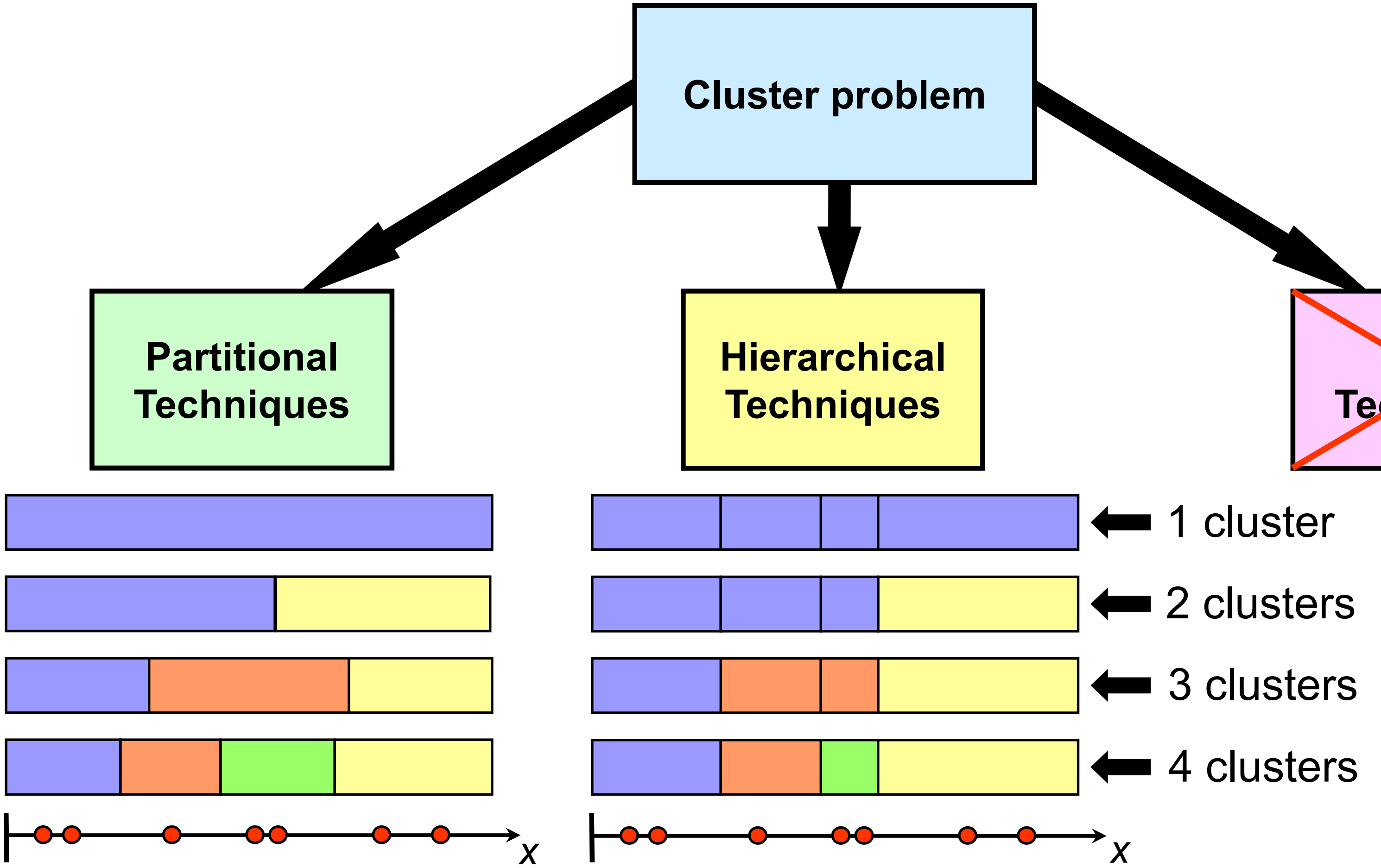
$$1 - |\rho_{ij}|$$

d(🔵,🟢) ≈ d(🔵,🟡)
d(🔵,🟢) ≈ d(🔵,🔴)
d(🔵,🟢) << d(🔵,⚫)

Clustering techniques



Clustering techniques (2)



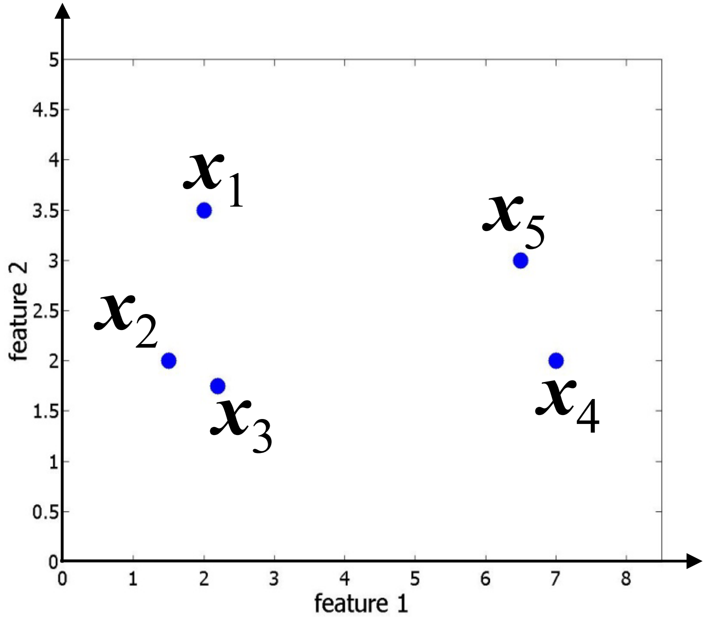
Hierarchical clustering

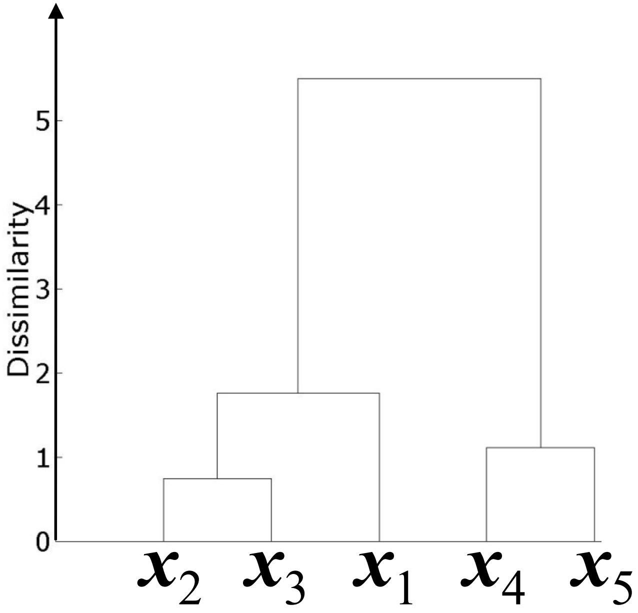
Input:

- dataset, ***X***: [*n* x *p*], or directly:
- dissimilarity matrix, ***D***: [*n* x *n*]
- linkage type

Output:

- dendrogram



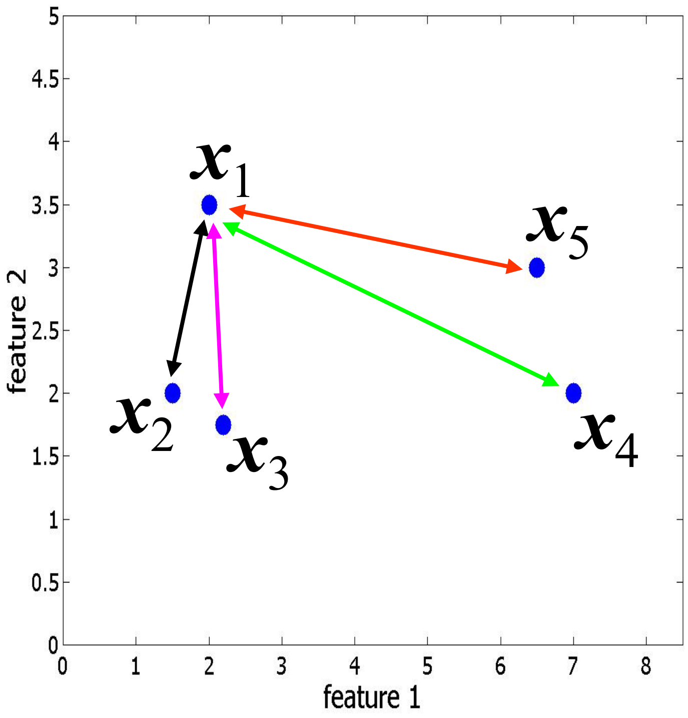


Hierarchical clustering (2)

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

Hierarchical clustering (3)

Dataset

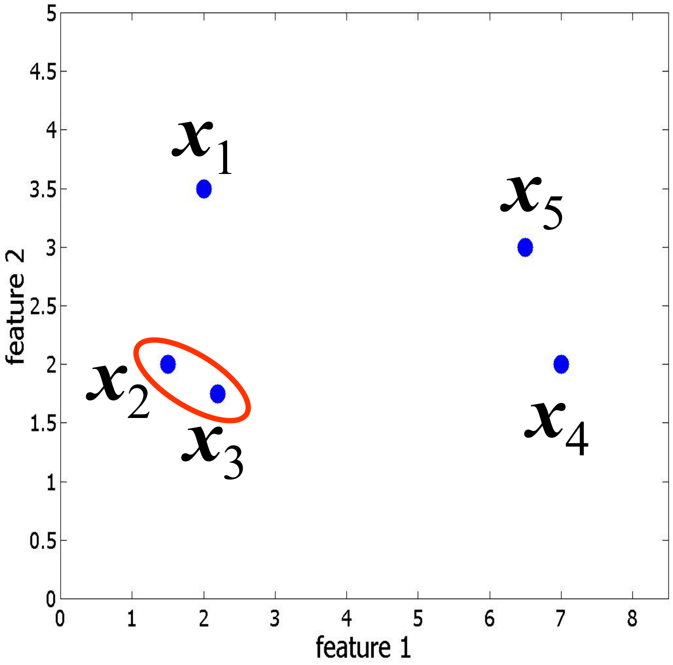


Euclidean distance matrix, ***D***

	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)

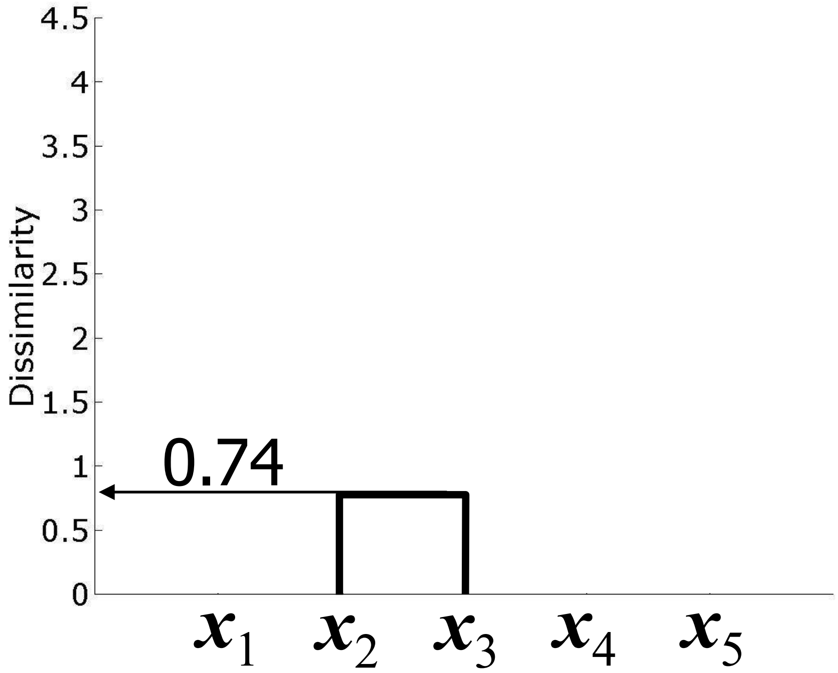
- **Step 1:**
  Find the most similar pair of objects: $\min_{(i,j)}\{d(i,j)\} = d(2,3)$



	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

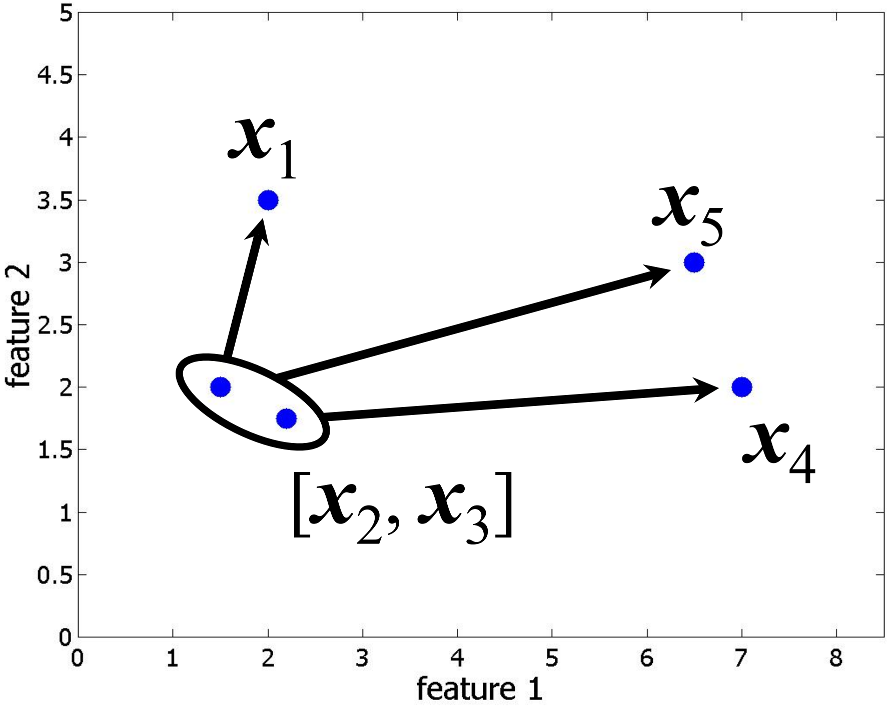
Hierarchical clustering (5)

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



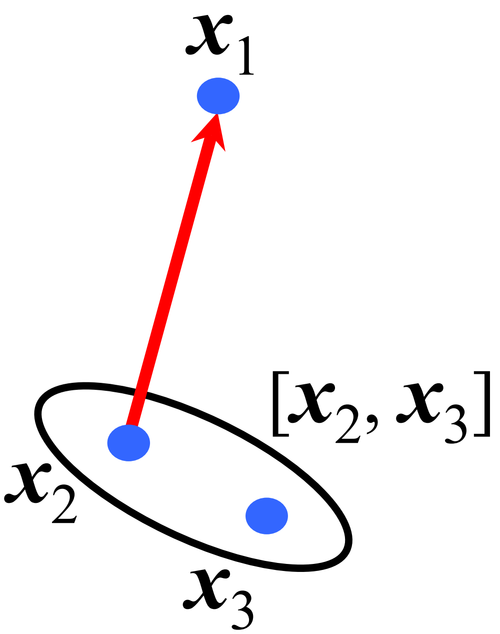
Hierarchical clustering (6)

- **Step 3:**
  Recompute $\boldsymbol{D}$ –
  what is the distance between [$\boldsymbol{x}_2$, $\boldsymbol{x}_3$] and the rest?



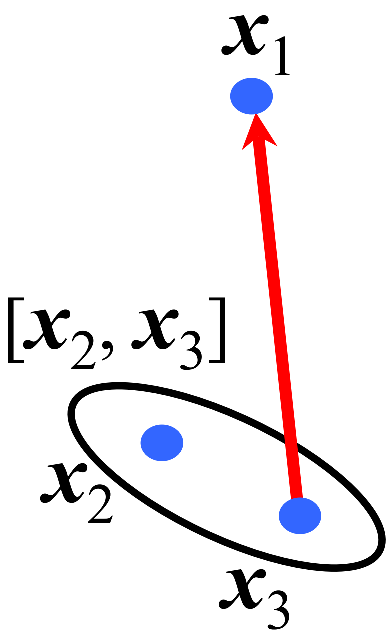
Hierarchical clustering (7)

- **Step 3:**
  Recompute $\boldsymbol{D}$ –
  single linkage: $d([\boldsymbol{x}_2,\boldsymbol{x}_3],\boldsymbol{x}_1) = \min(d(\boldsymbol{x}_1,\boldsymbol{x}_2),d(\boldsymbol{x}_1,\boldsymbol{x}_3))$



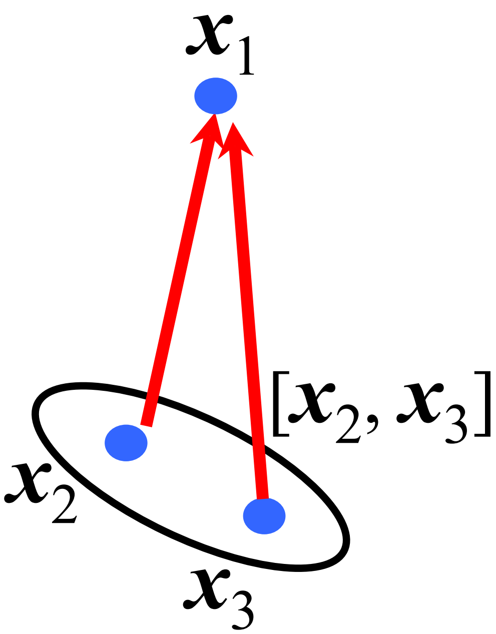
Hierarchical clustering (8)

- **Step 3:**
  Recompute $\boldsymbol{D}$ –
  complete linkage: $d([\boldsymbol{x}_2,\boldsymbol{x}_3],\boldsymbol{x}_1) = \max(d(\boldsymbol{x}_1,\boldsymbol{x}_2),d(\boldsymbol{x}_1,\boldsymbol{x}_3))$



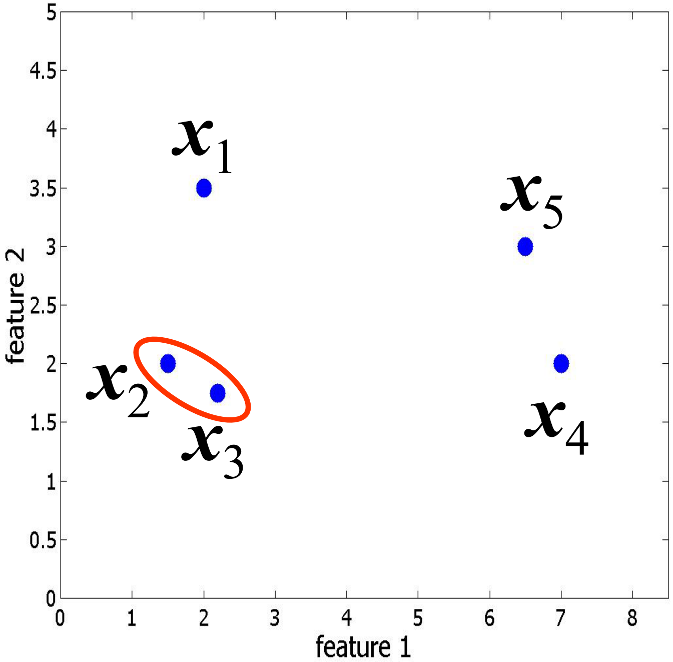
Hierarchical clustering (9)

- **Step 3:**
  Recompute $\boldsymbol{D}$ –
  average linkage: $d([\boldsymbol{x}_2,\boldsymbol{x}_3],\boldsymbol{x}_1) = \mathrm{mean}(d(\boldsymbol{x}_1,\boldsymbol{x}_2),d(\boldsymbol{x}_1,\boldsymbol{x}_3))$



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

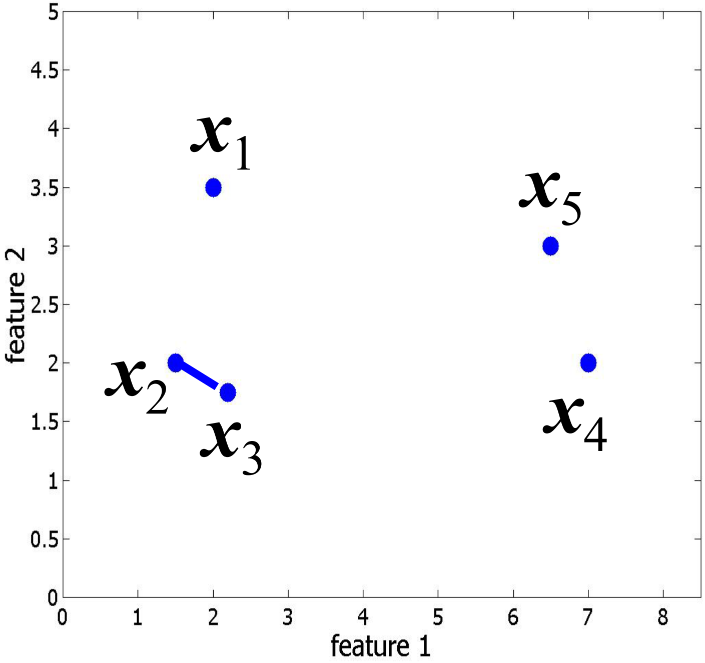
Hierarchical clustering (10b)

- **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)

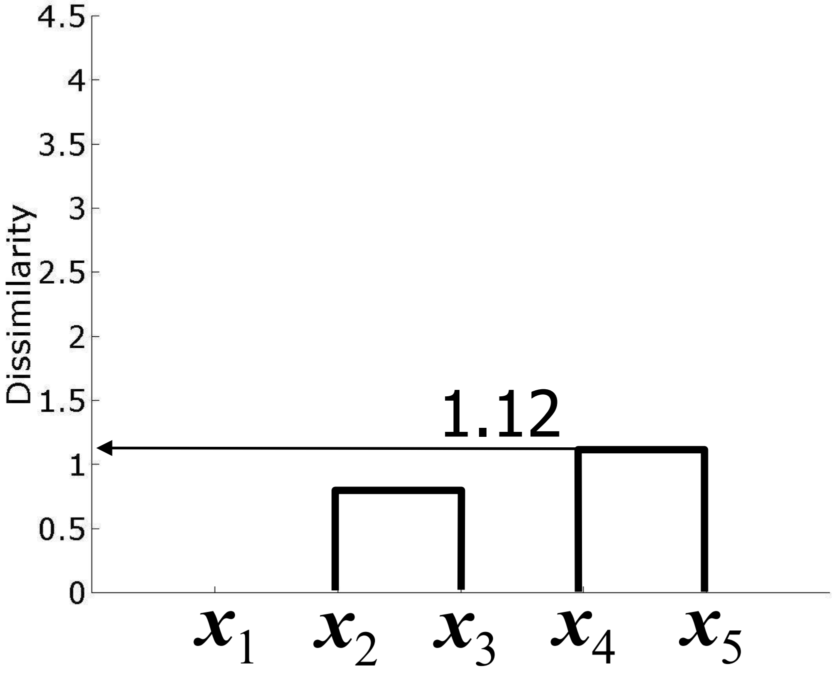
- **Repeat, step 1:**
  Find the most similar pair of objects: $\min_{(i,j)}\{d(i,j)\} = d(4,5)$



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

Hierarchical clustering (12)

- **Repeat, step 2:**
  Merge $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$ into a single object, $[\boldsymbol{x}_4, \boldsymbol{x}_5]$;



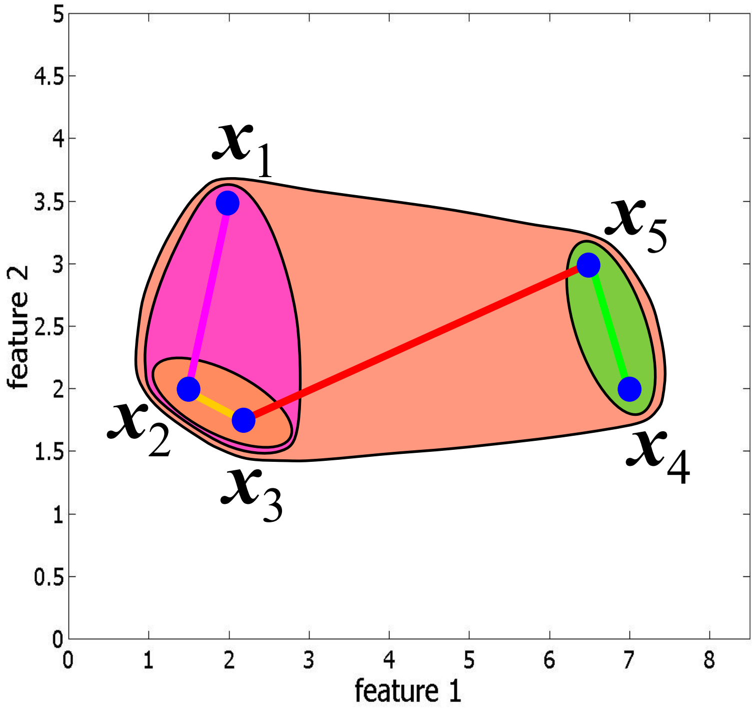
Hierarchical clustering (13)

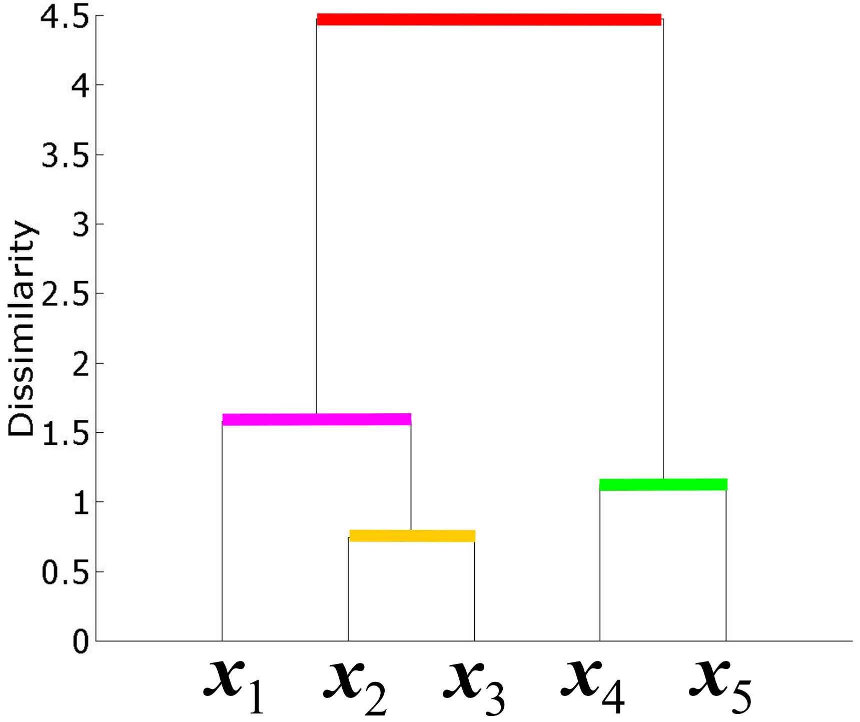
- **Repeat, step 3:**
  Recompute $\boldsymbol{D}$ (single linkage):

	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

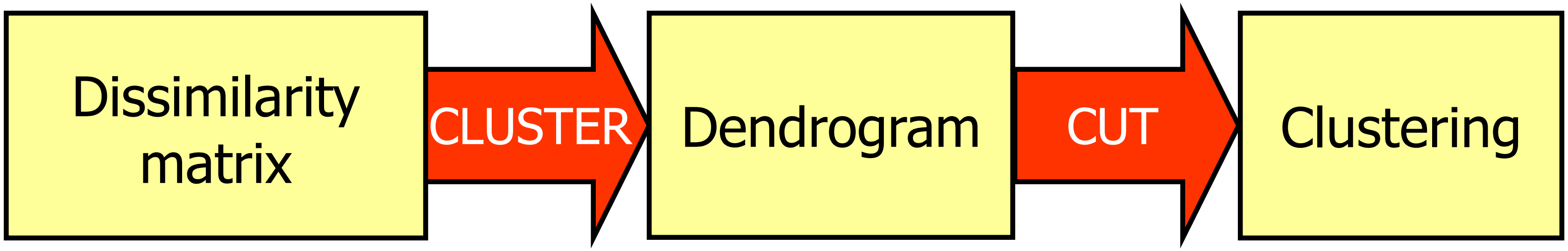
Hierarchical clustering (14)

- Repeat steps 1-3 until a single cluster remains





Hierarchical clustering (15)



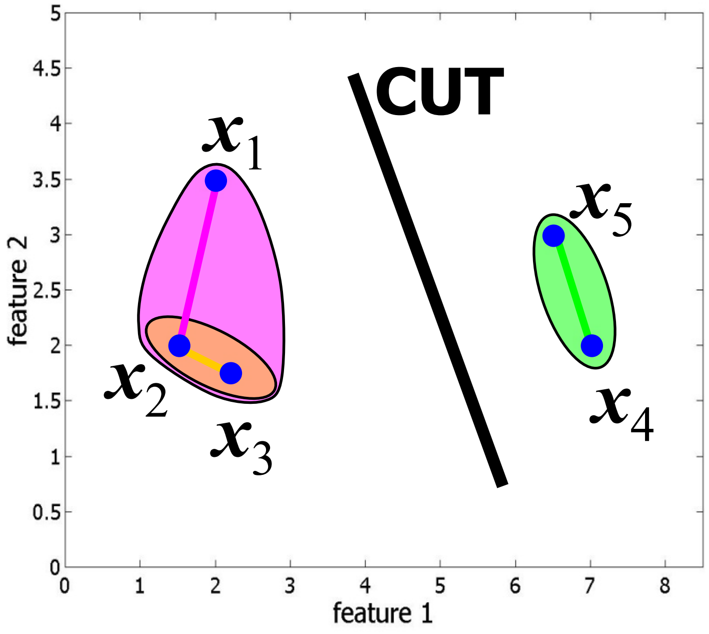
Dissimilarity matrix

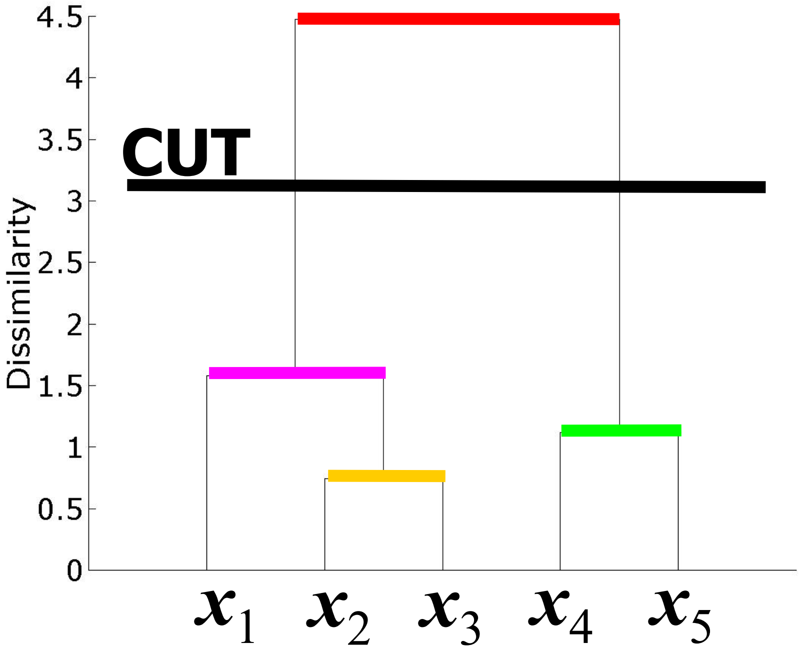
CLUSTER

Dendrogram

CUT

Clustering

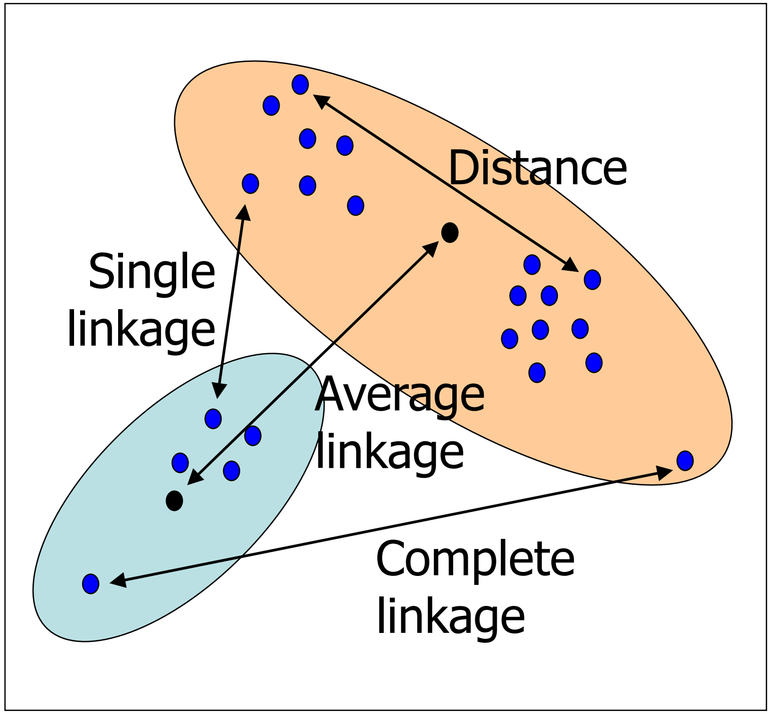




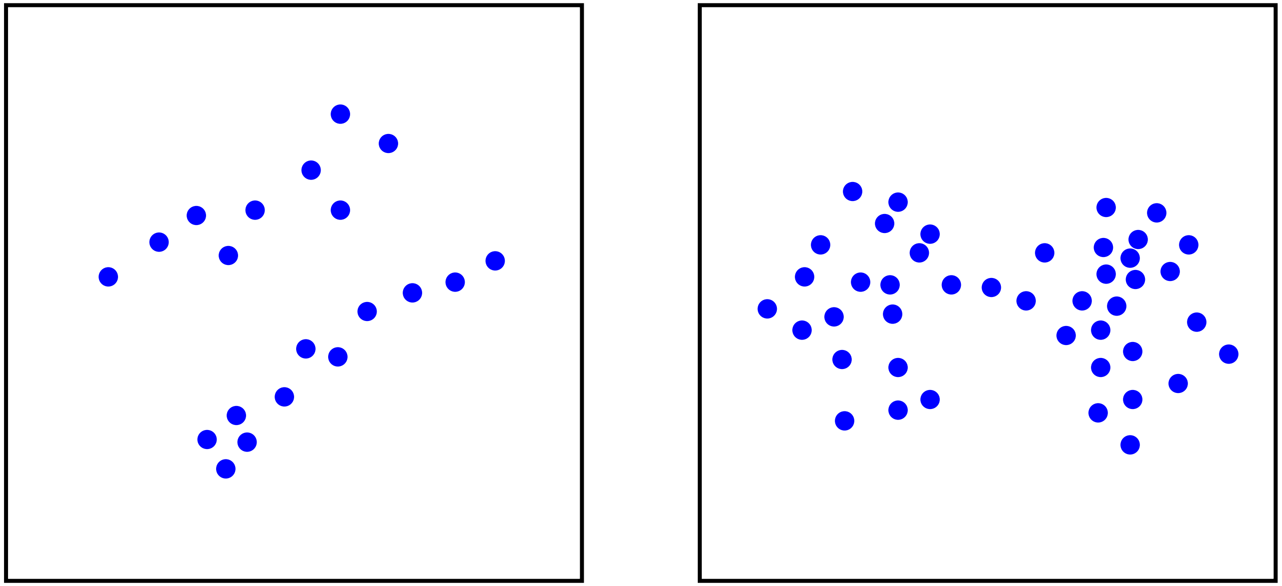
Hierarchical clustering (16)

- Hierarchical clustering: repeatedly group closest clusters

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



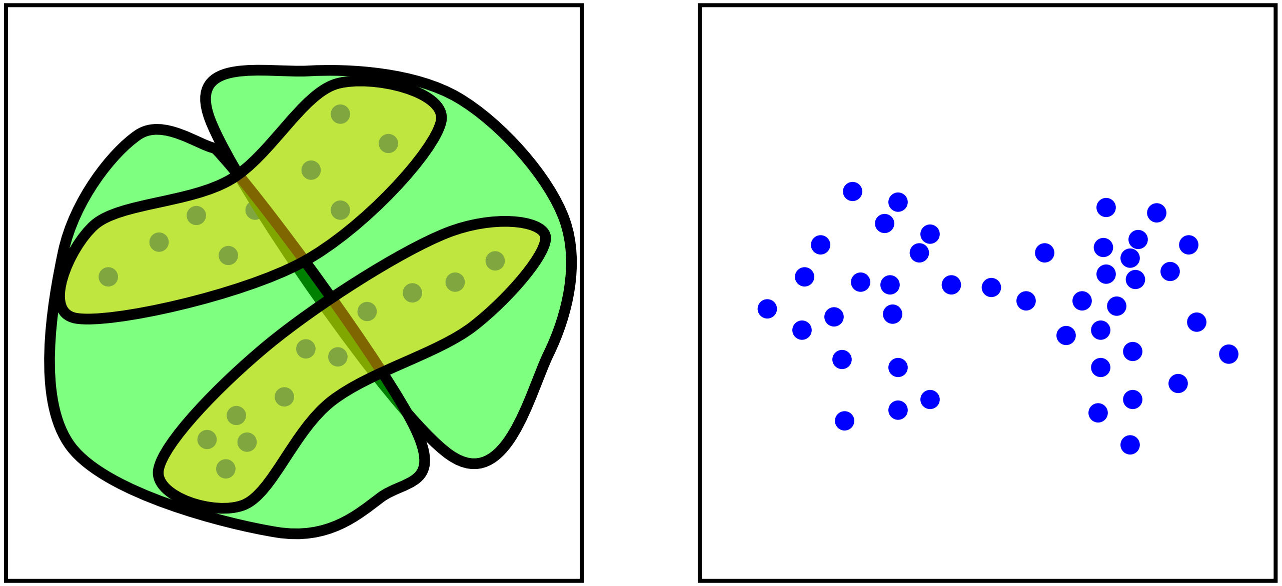
Linkage and cluster shape



🟩 Complete linkage

🟨 Single linkage

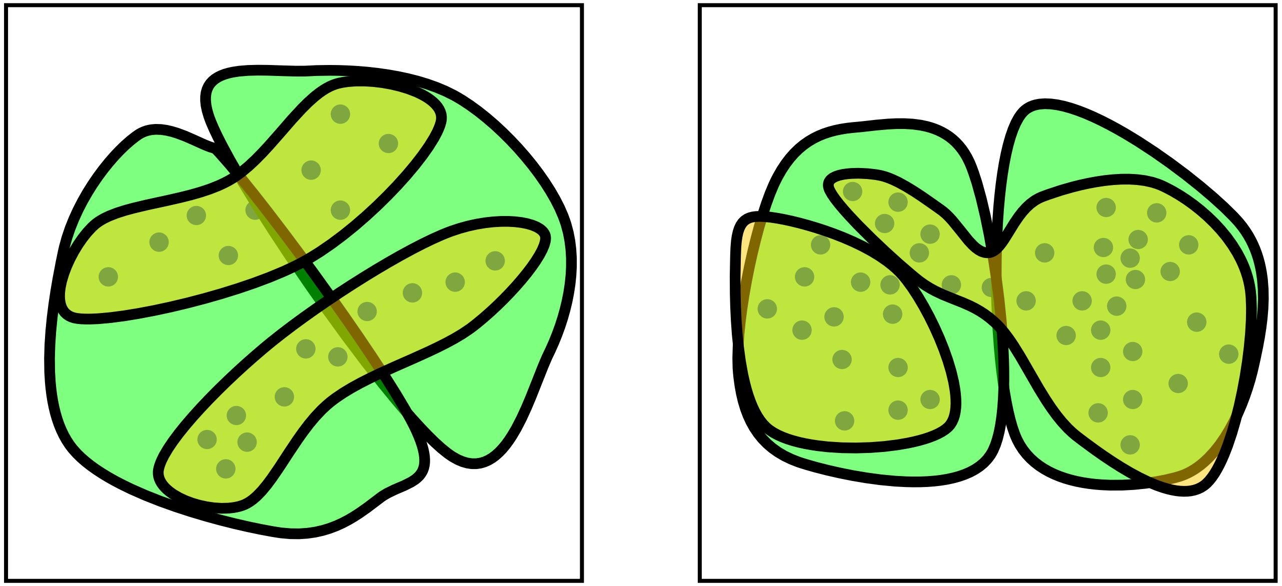
Linkage and cluster shape (2)



🟩 Complete linkage

🟨 Single linkage

Linkage and cluster shape (3)



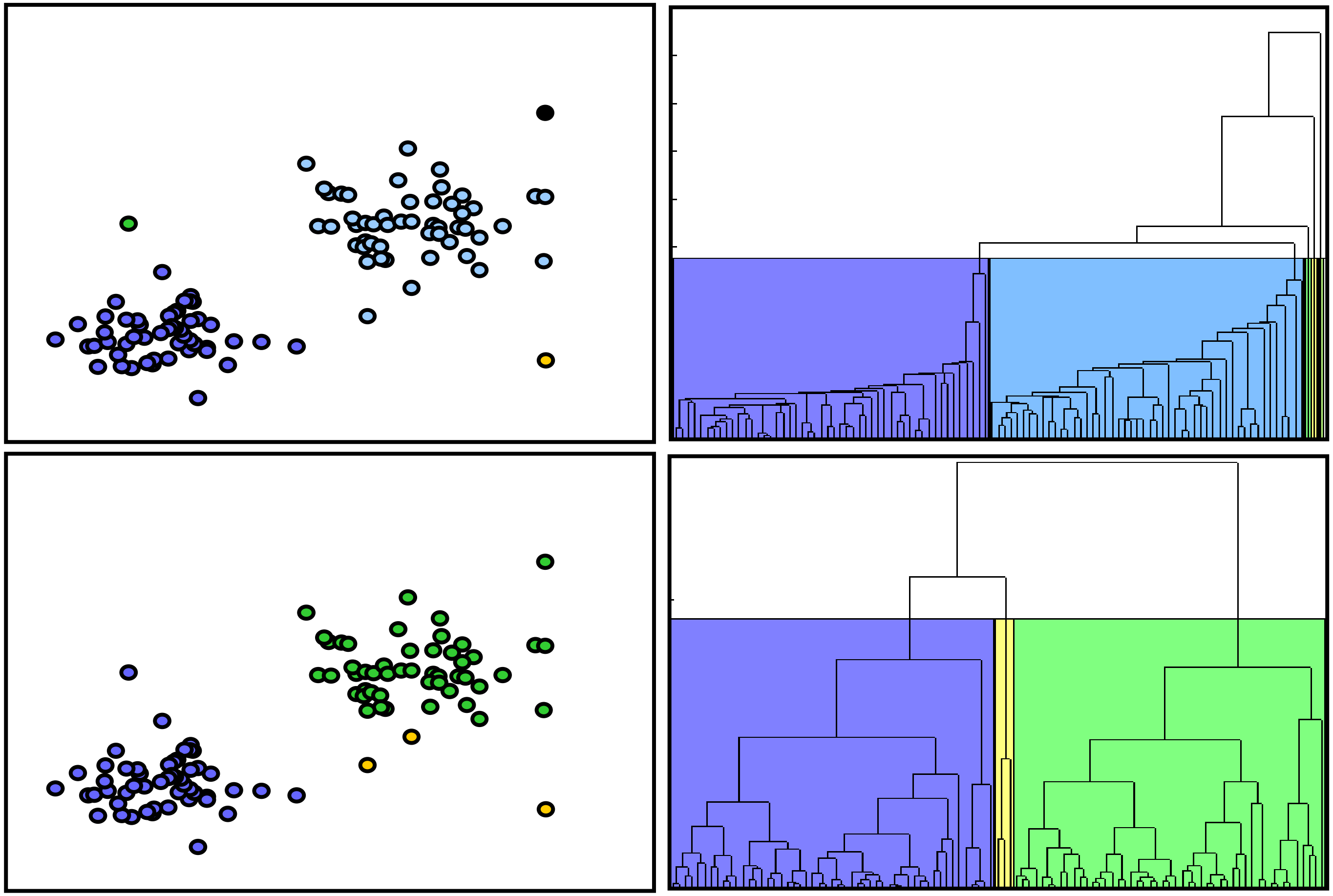
🟩 Complete linkage

🟨 Single linkage

Linkage and outliers

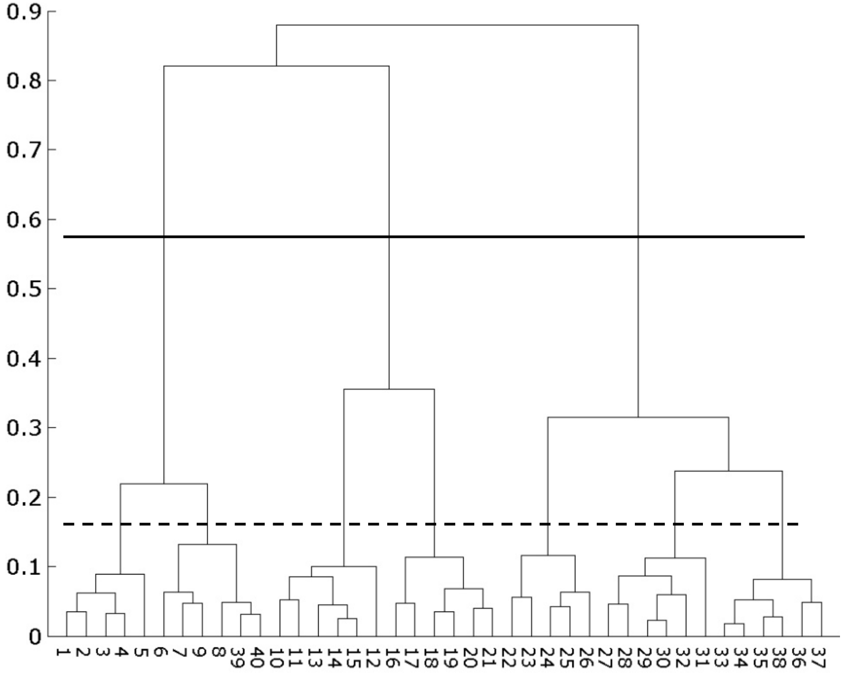
Single linkage

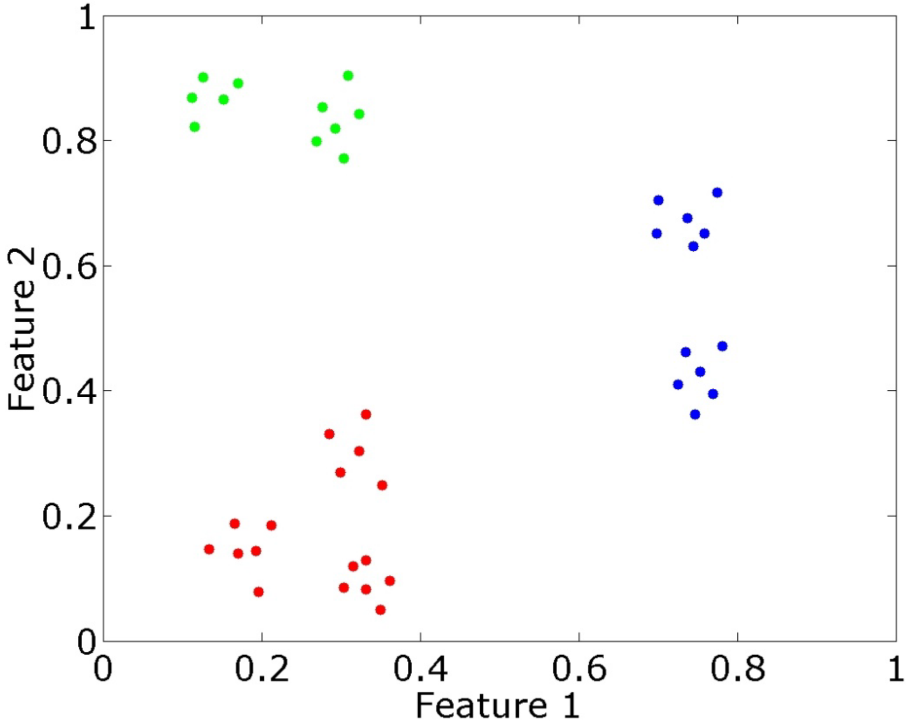
Complete linkage



Hierarchical clustering examples

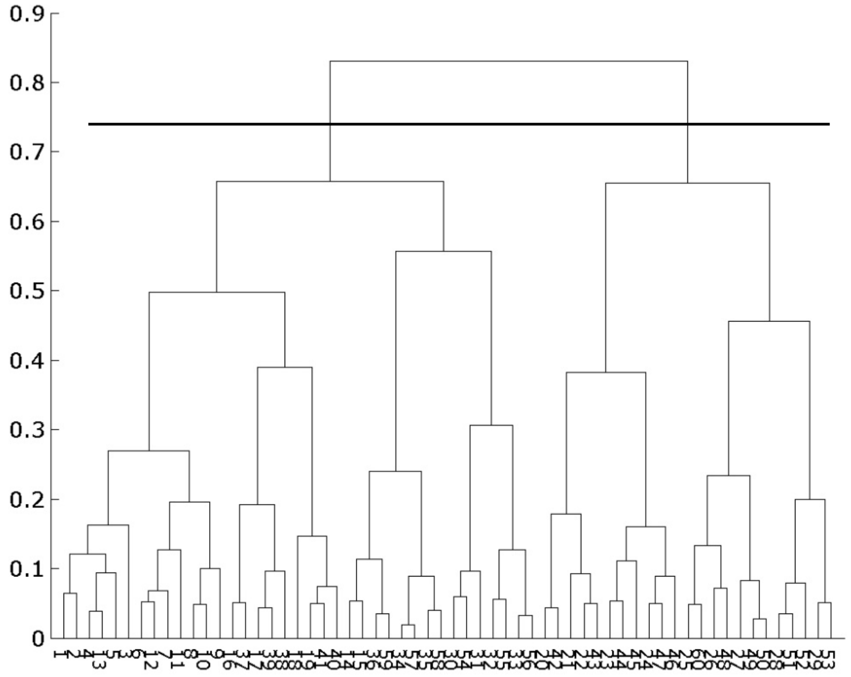
Euclidean, complete linkage

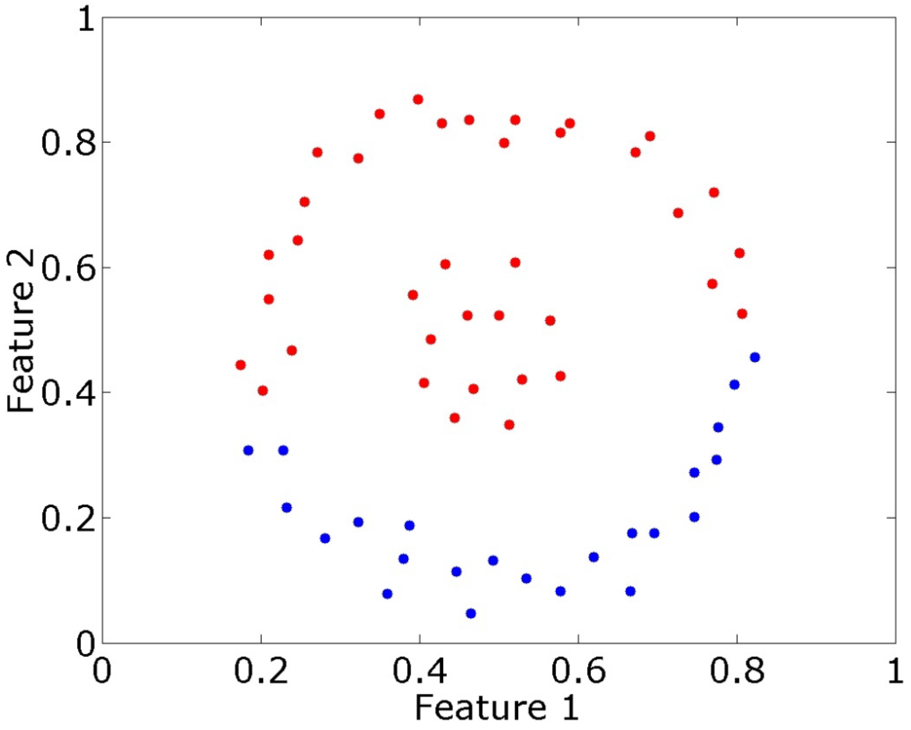




Hierarchical clustering examples (2)

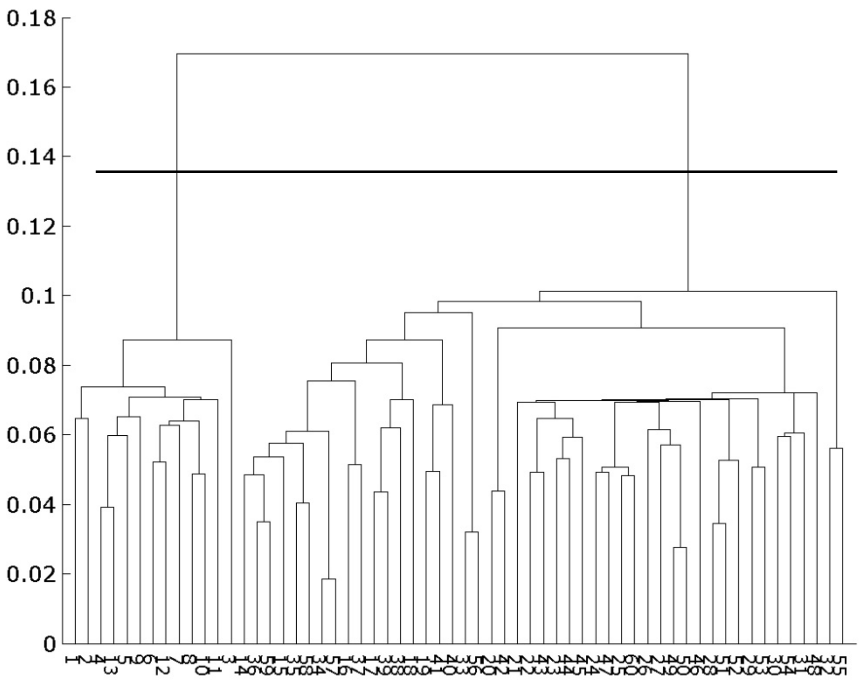
Euclidean, complete linkage

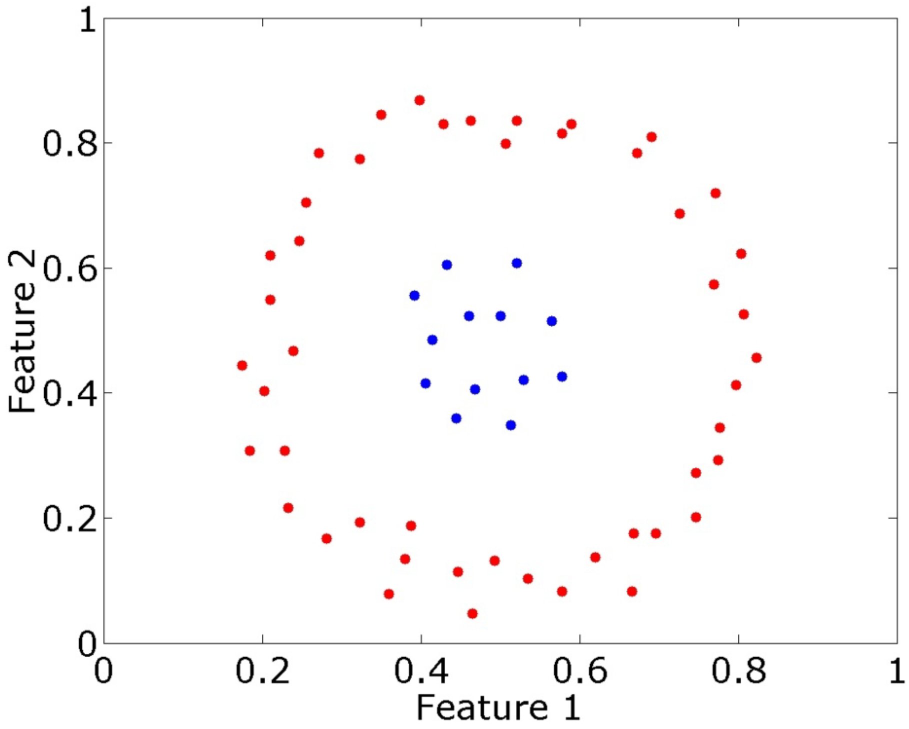




Hierarchical clustering examples (3)

Euclidean, single linkage





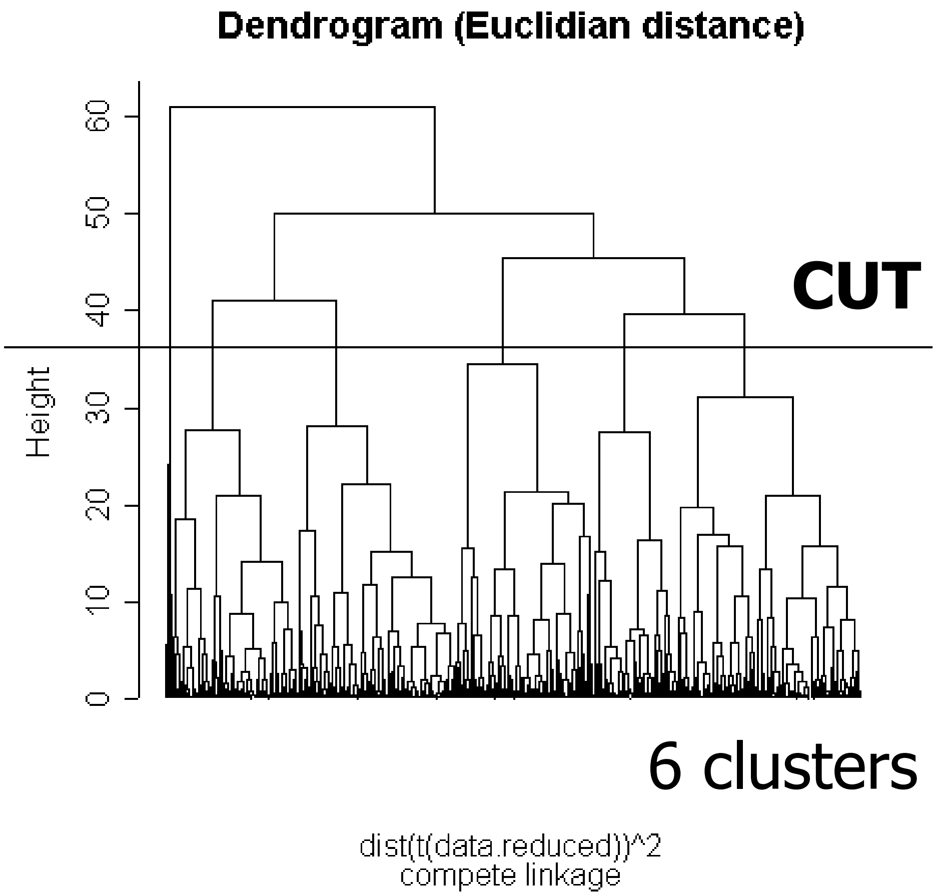
Hierarchical clustering (17)

- 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

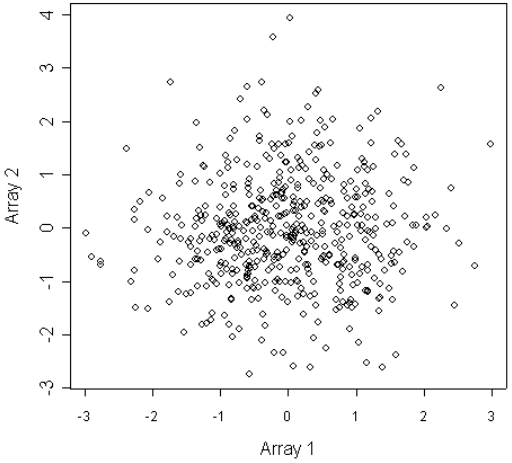
- Cluster 500 genes, 5 arrays:



CUT

6 clusters

Data were random …

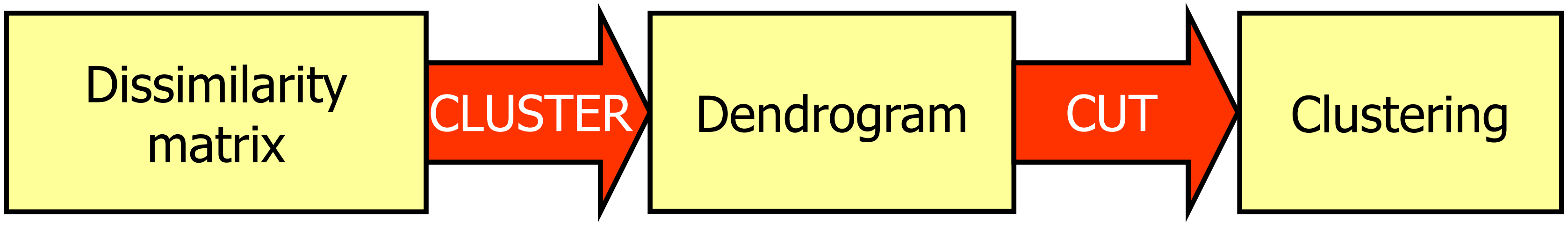


Validation is needed

10min break
Exercise 4.1-4.7

Sum-of-squares clustering

- Hierarchical:



Dissimilarity matrix

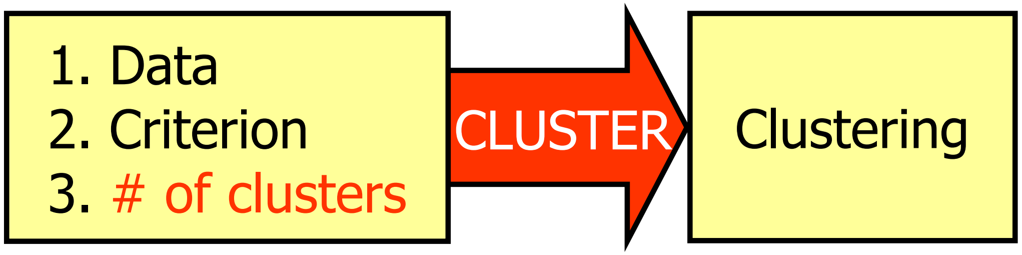
CLUSTER

Dendrogram

CUT

Clustering

- Sum-of-squares:



1. Data
2. Criterion
3. # of clusters

CLUSTER

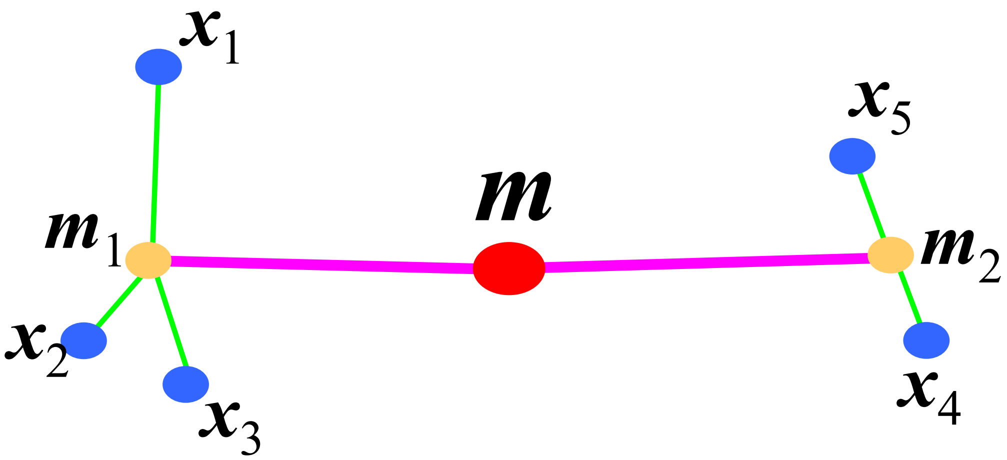
Clustering

Sum-of-squares clustering (2)

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

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

$$\boldsymbol{S}_B = \sum_{i=1}^{C} \frac{n_i}{n} (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T, \ \boldsymbol{m} = \sum_{i=1}^{C} \frac{n_i}{n} \boldsymbol{m}_i$$

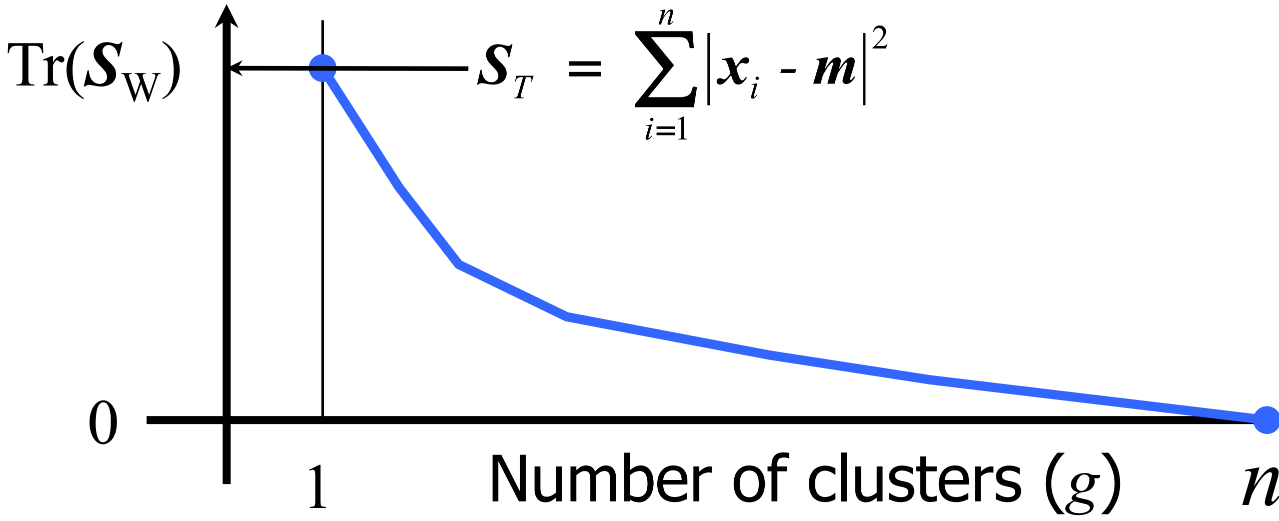


K-means

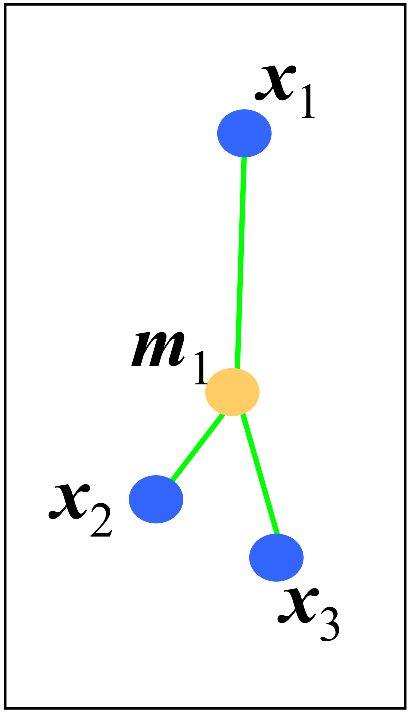
- Minimize: $$\mathrm{Tr}(\mathbf{S}_W) = \frac{1}{n}\sum_{j=1}^{g}\mathbf{S}_j$$

$$\boldsymbol{S}_j = \sum_{i=1}^{n_j}\left|\boldsymbol{x}_i - \boldsymbol{m}_j\right|^2$$

(sum of per cluster variances)



$$\boldsymbol{S}_T = \sum_{i=1}^{n}\left|\boldsymbol{x}_i - \boldsymbol{m}\right|^2$$



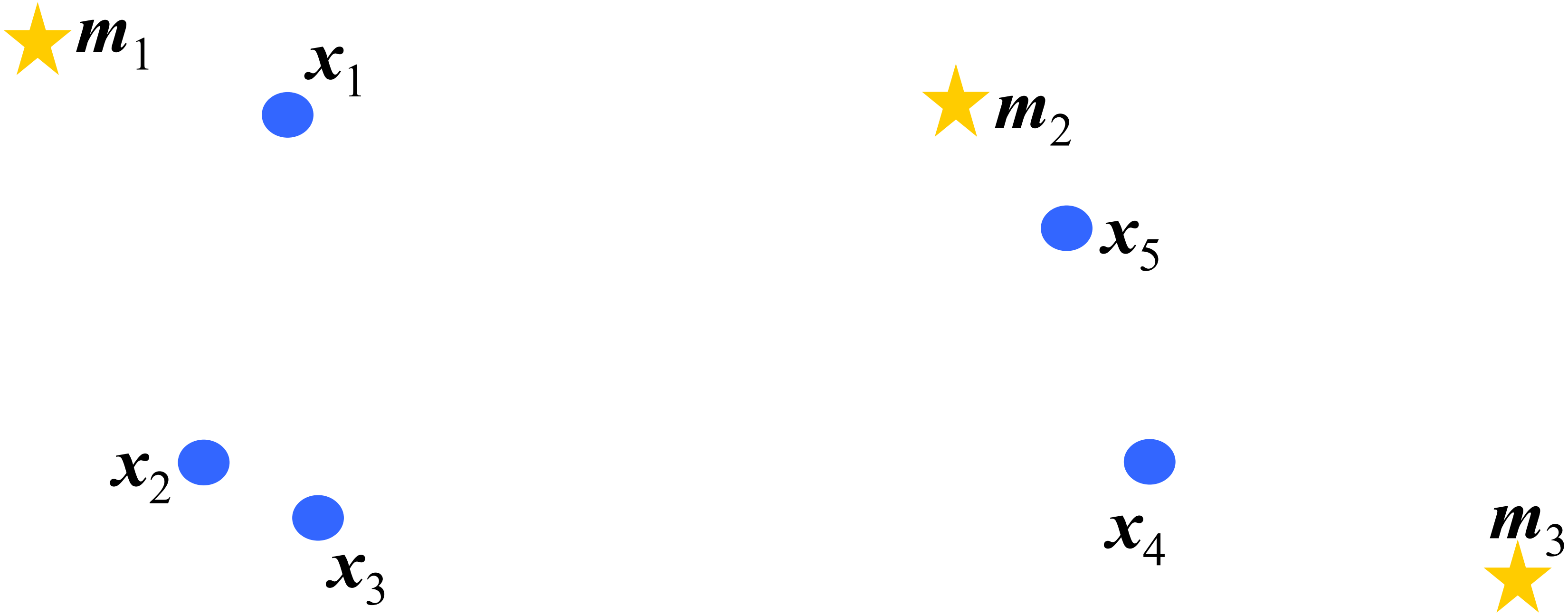
K-means (2)

- Iterative procedure to search for $\min(\mathrm{Tr}(\boldsymbol{S}_\mathrm{W}))$:
  1. choose number of clusters (g)
  2. position prototypes (m_j, $j=1,\ldots,g$) 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

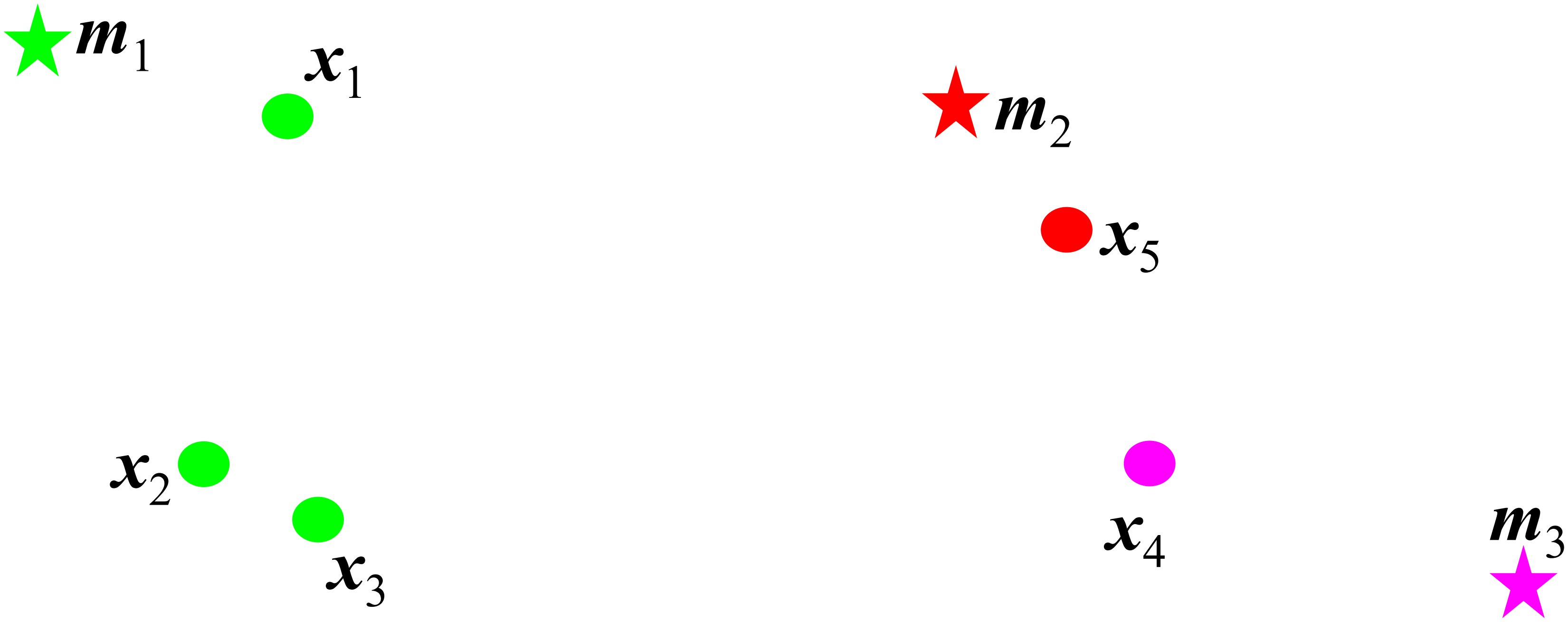
K-means (3)

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



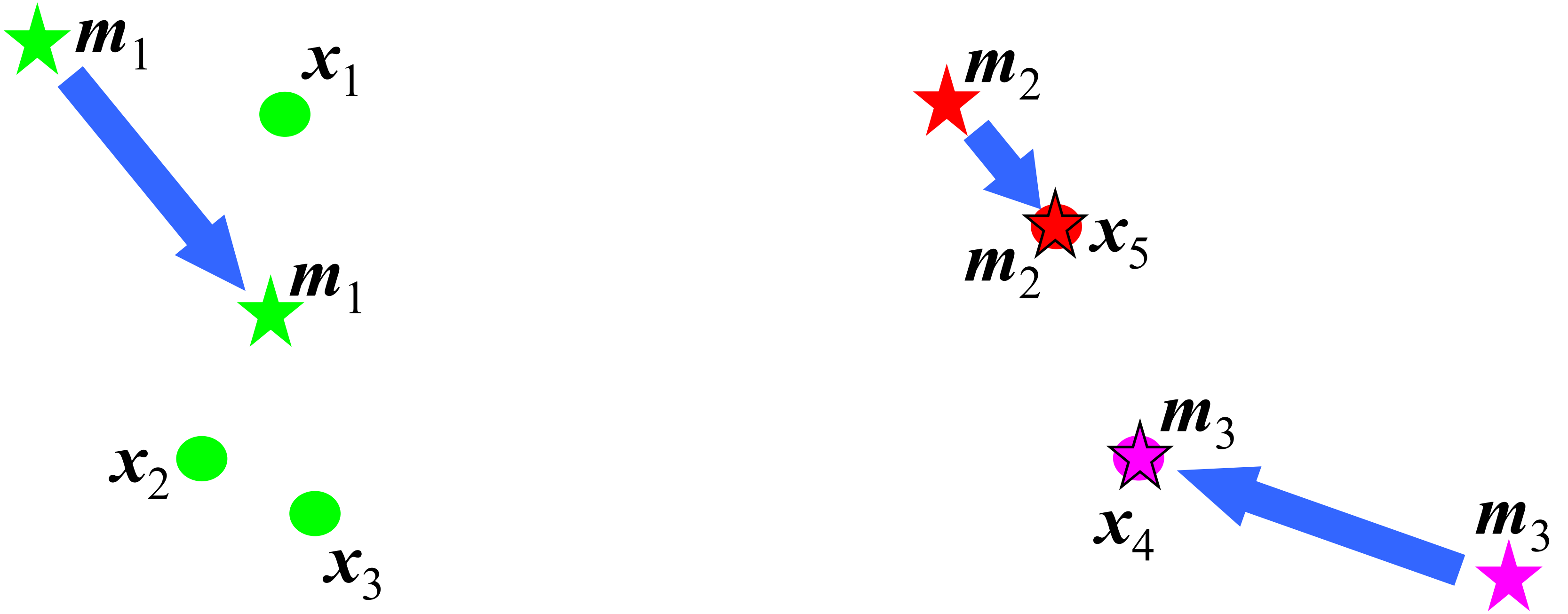
K-means (4)

- **Step 3:** Assign samples to closest prototype



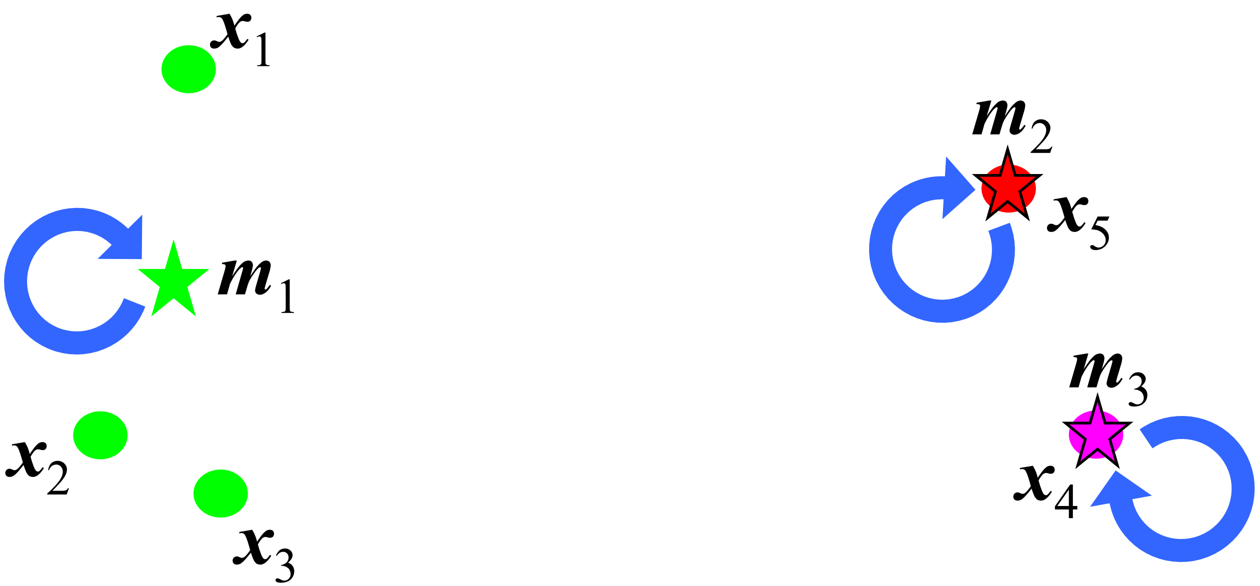
K-means (5)

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



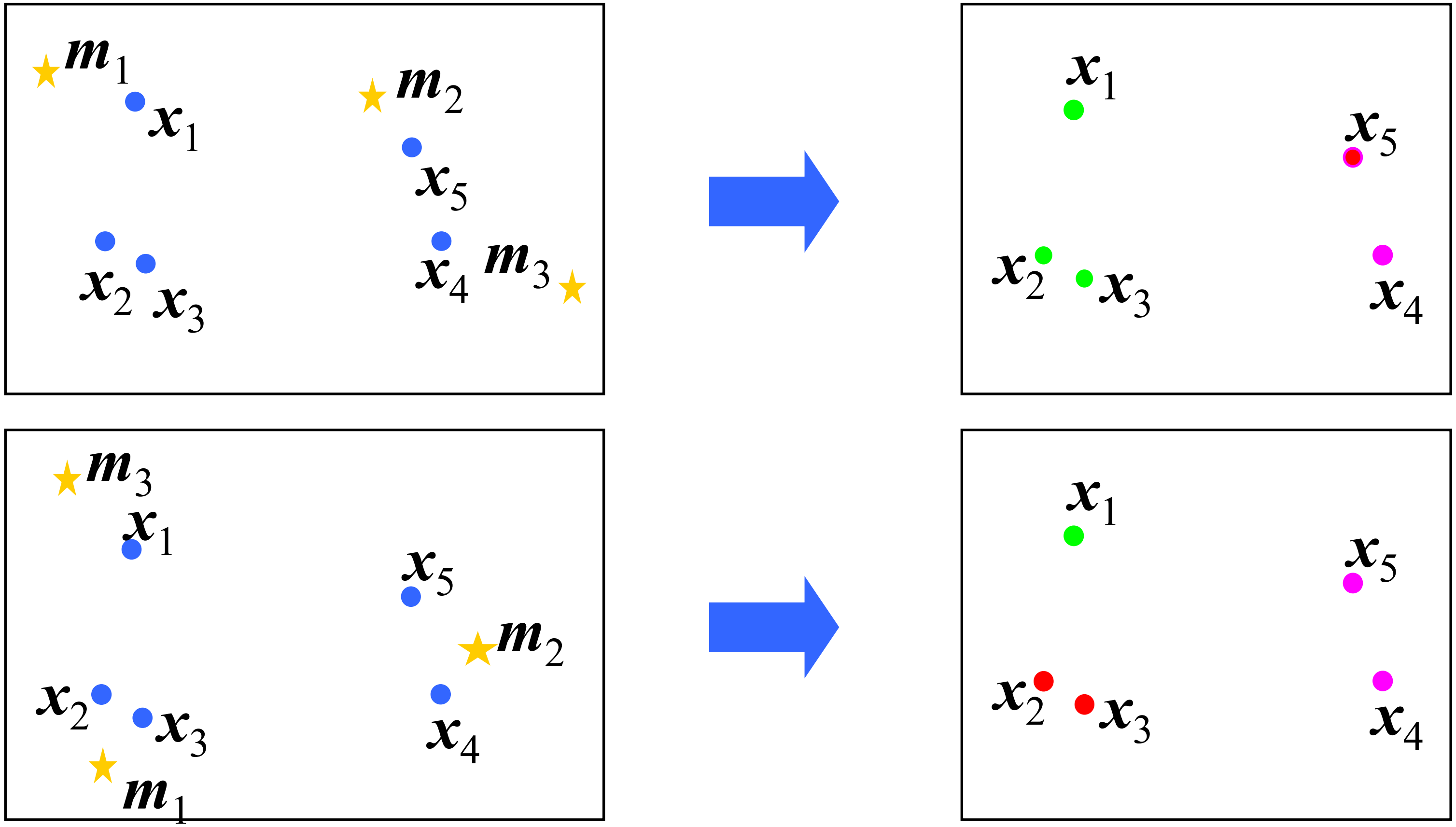
K-means (6)

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



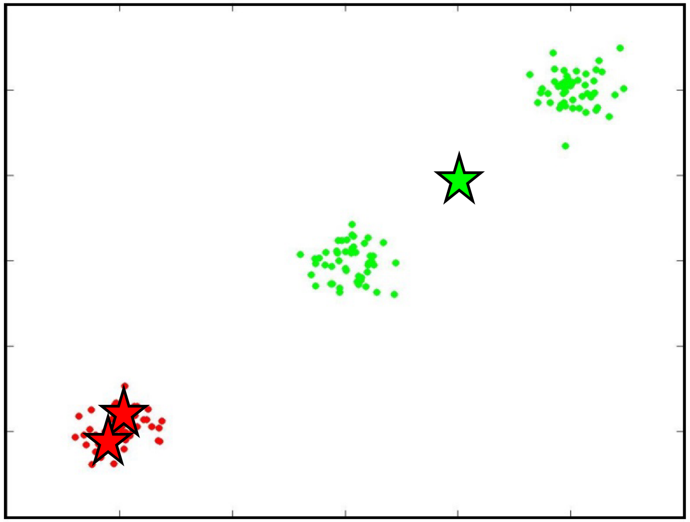
K-means problems

- Clustering depends on initialization



K-means problems (2)

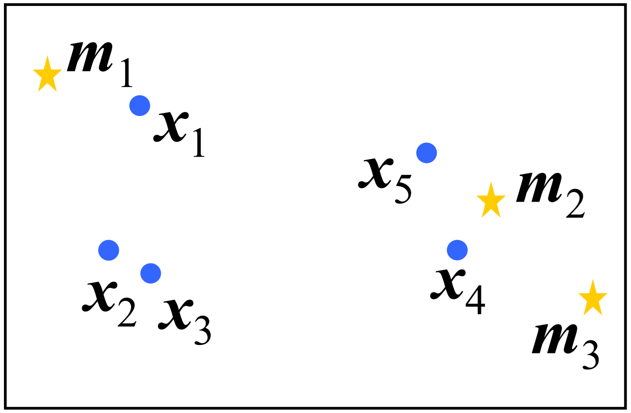
- Algorithm can get stuck in local minima



- Solution:
  - start from I different random initialisations
  - keep the best clustering (lowest $\mathrm{Tr}(\boldsymbol{S}_{\mathrm{W}})$)
  - For high-dimensional data, many restarts can be necessary (e.g. I = 100)

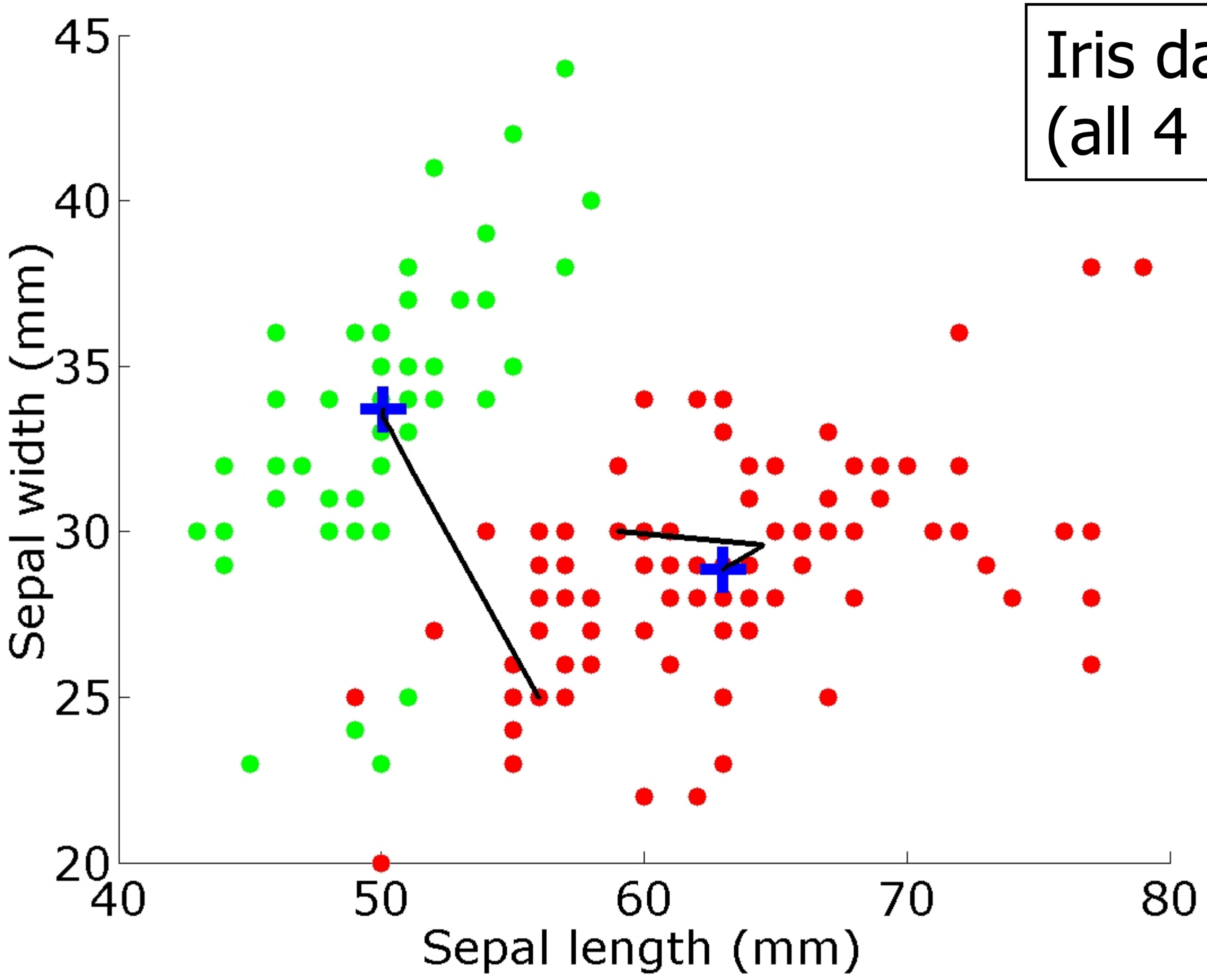
K-means problems (3)

- 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



Iris dataset
(all 4 features)

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

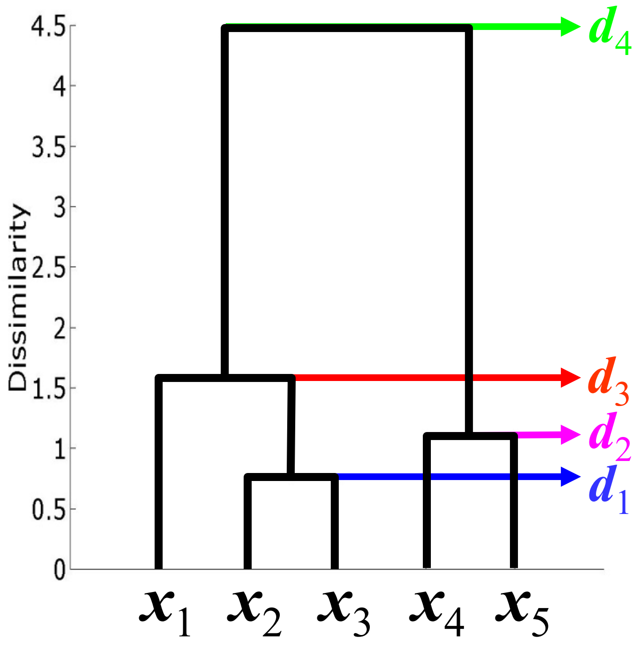
- Clustering is way to detect *natural* groups in data
- What is natural is partly subjective
- We looked at:
  - *Hierarchical clustering*
  - *Sum of squares* (k-means) clustering
- Hierarchical clustering:
  - *dendrogram* shows a complete hierarchy of possible clusterings
  - computionally intensive
- *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****

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

d

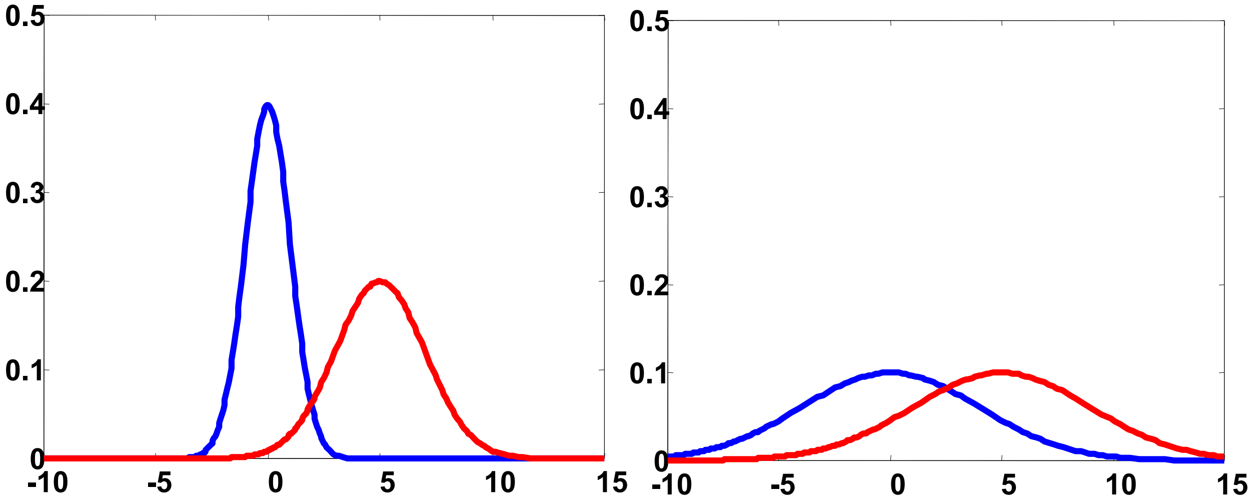
	x_1	x_2	x_3	x_4	x_5
x_1	0.00	1.58	1.76	5.22	4.53
x_2		0.00	0.74	5.50	5.10
x_3			0.00	4.81	4.48
x_4				0.00	1.12
x_5					0.00

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

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

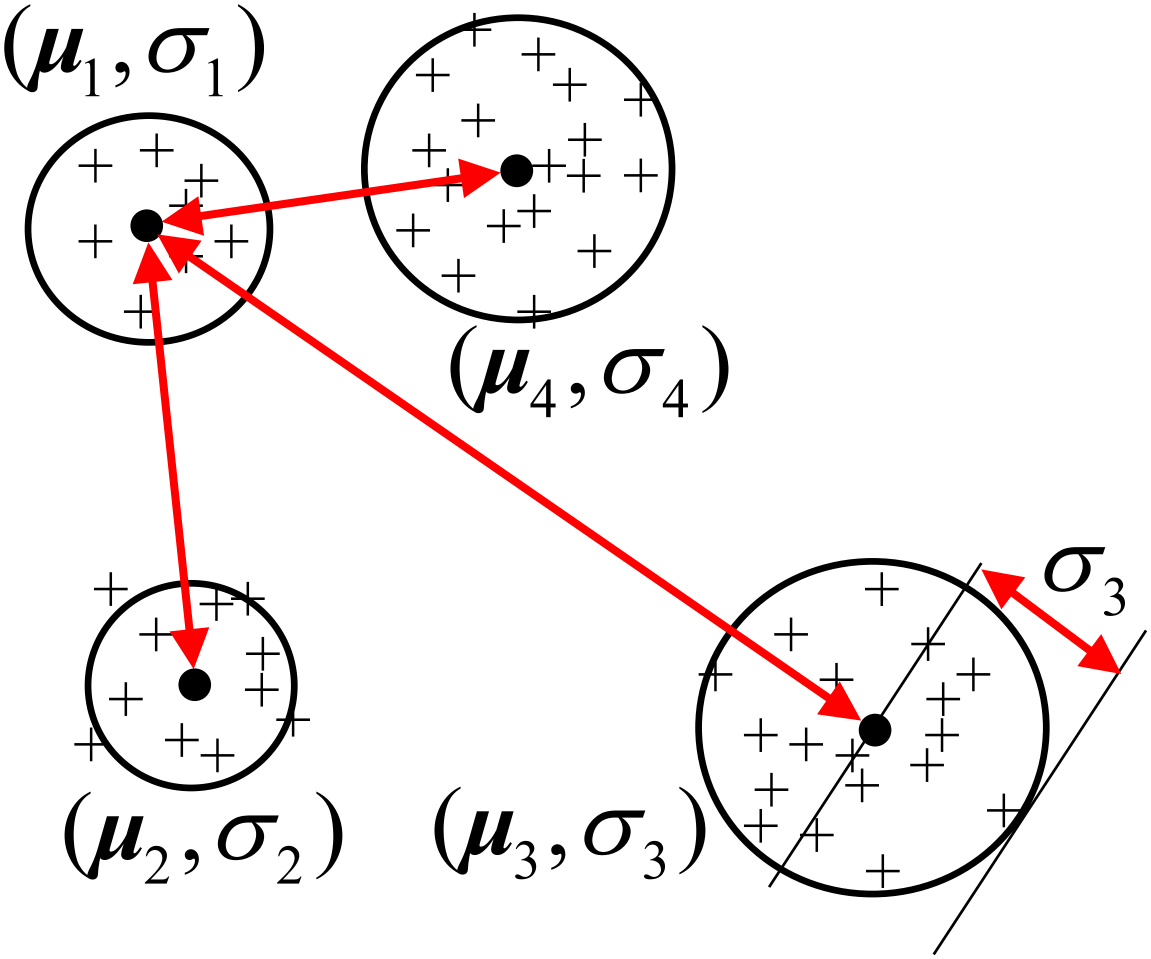


Davies-Bouldin index

- 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

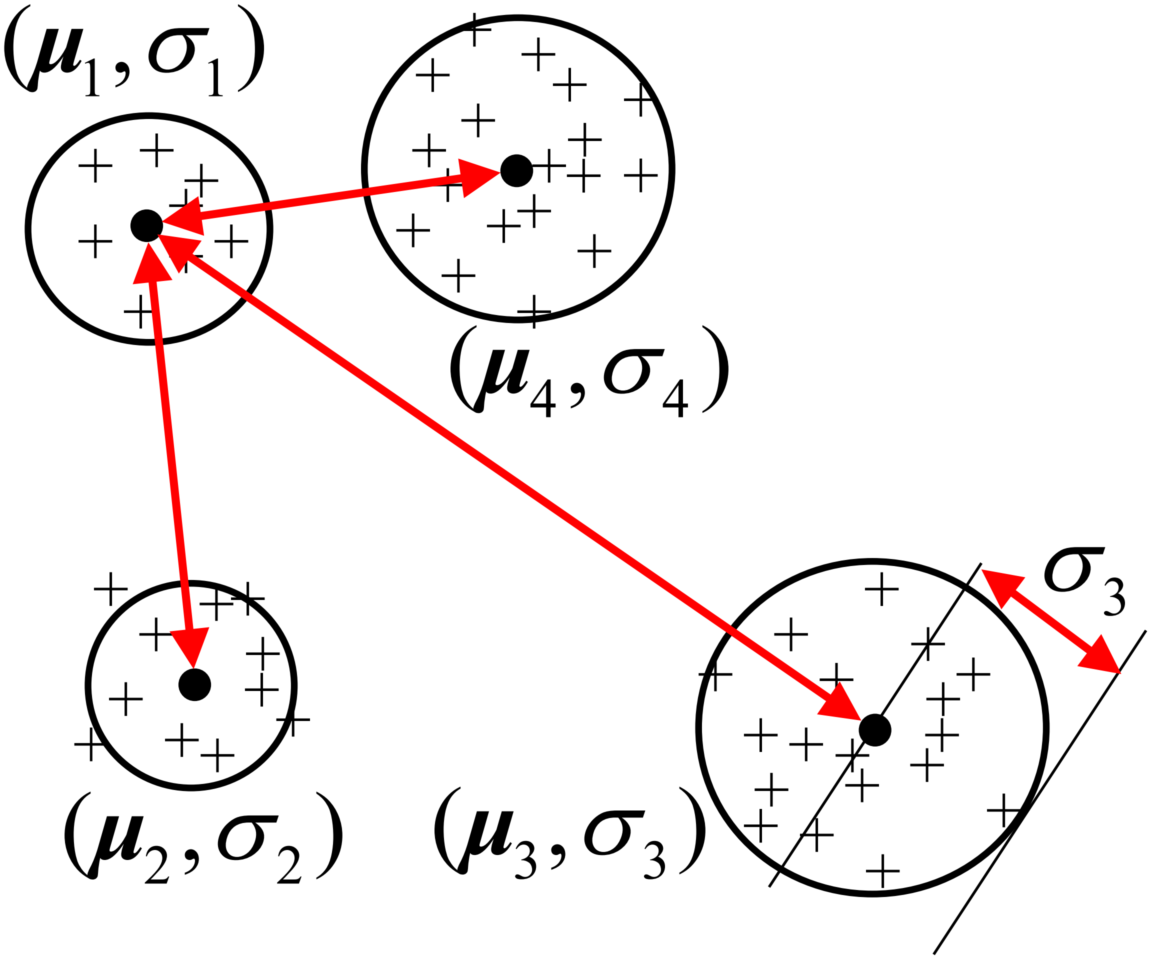
Davies-Bouldin index (2)


$(\boldsymbol{\mu}_1, \sigma_1)$
$(\boldsymbol{\mu}_4, \sigma_4)$
σ_3
$(\boldsymbol{\mu}_2, \sigma_2)$
$(\boldsymbol{\mu}_3, \sigma_3)$

$$\sigma_j = \sqrt{\tfrac{1}{n_j} \sum_{\boldsymbol{x}_i \in C_j} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|^2}$$

$$\boldsymbol{\mu}_j = \tfrac{1}{n_j} \sum_{\boldsymbol{x}_i \in C_j} \boldsymbol{x}_i$$

Davies-Bouldin index (3)

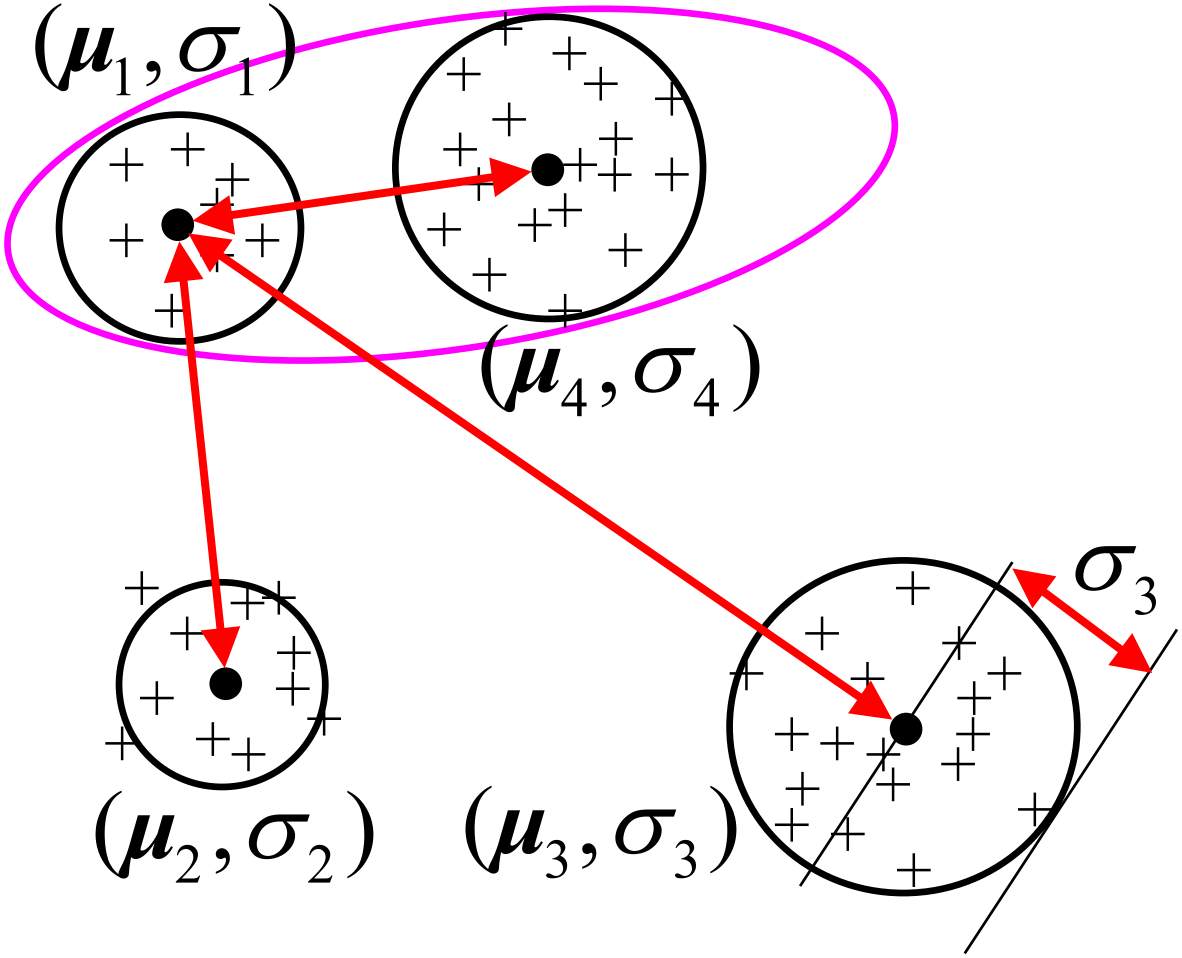

$(\boldsymbol{\mu}_1,\sigma_1)$
$(\boldsymbol{\mu}_4,\sigma_4)$
σ_3
$(\boldsymbol{\mu}_2,\sigma_2)$
$(\boldsymbol{\mu}_3,\sigma_3)$

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

$$\sigma_j = \sqrt{\frac{1}{n_j} \sum_{\boldsymbol{x}_i \in C_j} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|^2}$$

$$\boldsymbol{\mu}_j = \frac{1}{n_j} \sum_{\boldsymbol{x}_i \in C_j} \boldsymbol{x}_i$$

Davies-Bouldin index (4)



$(\boldsymbol{\mu}_1, \sigma_1)$
$(\boldsymbol{\mu}_4, \sigma_4)$
σ_3
$(\boldsymbol{\mu}_2, \sigma_2)$
$(\boldsymbol{\mu}_3, \sigma_3)$

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

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

Davies-Bouldin index (5)

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

Paired cluster criterion

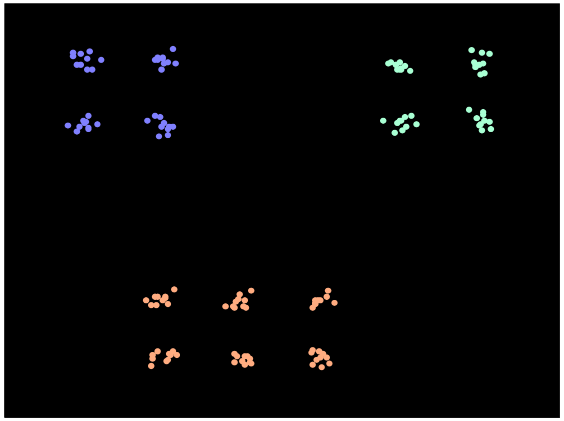
$$R_j = \max_{k=1,..g;k \neq j} R_{jk}$$

Worst-case value per cluster

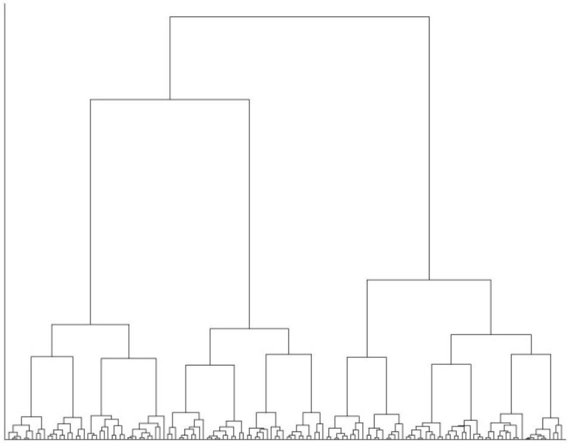
$$I_{DB} = \frac{1}{g} \sum_{j=1}^{g} R_j$$

Average worst-case

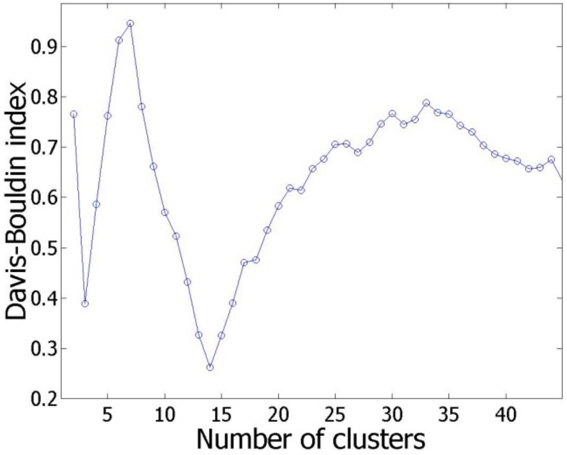
Davies-Bouldin index (5)



Dataset



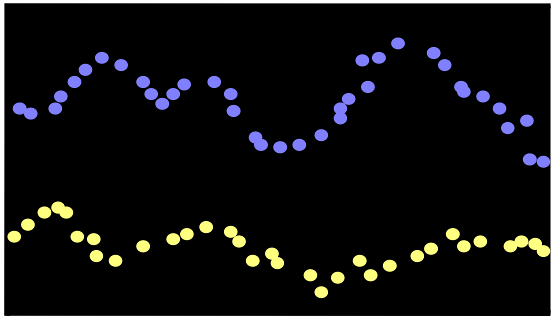
Complete link

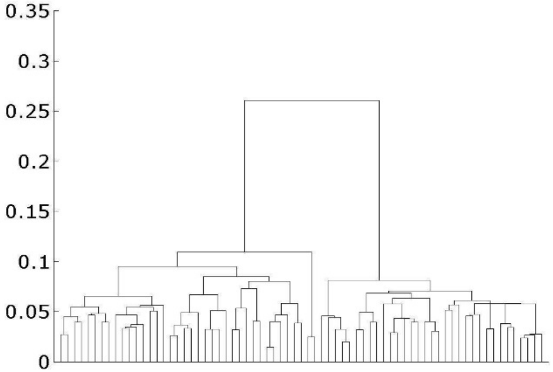


Davies-Bouldin:
3 or 14 clusters

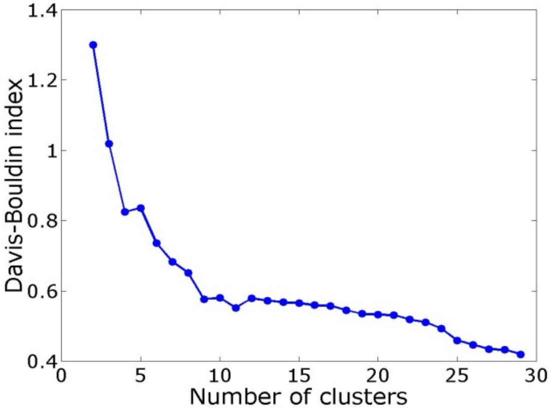
Davies-Bouldin index (7)

Single link



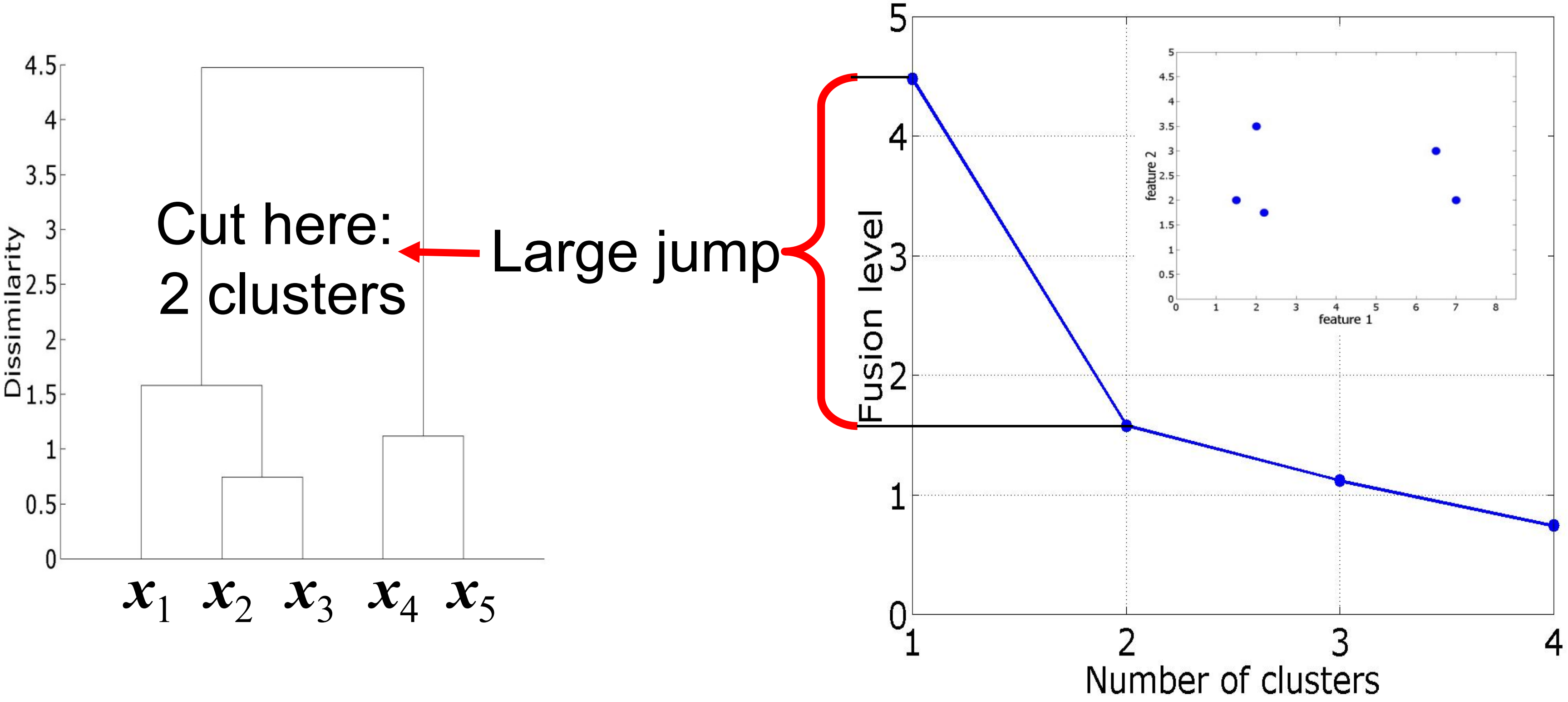


Davies-Bouldin:



Fusion graph

- Heuristic approach: fusion level

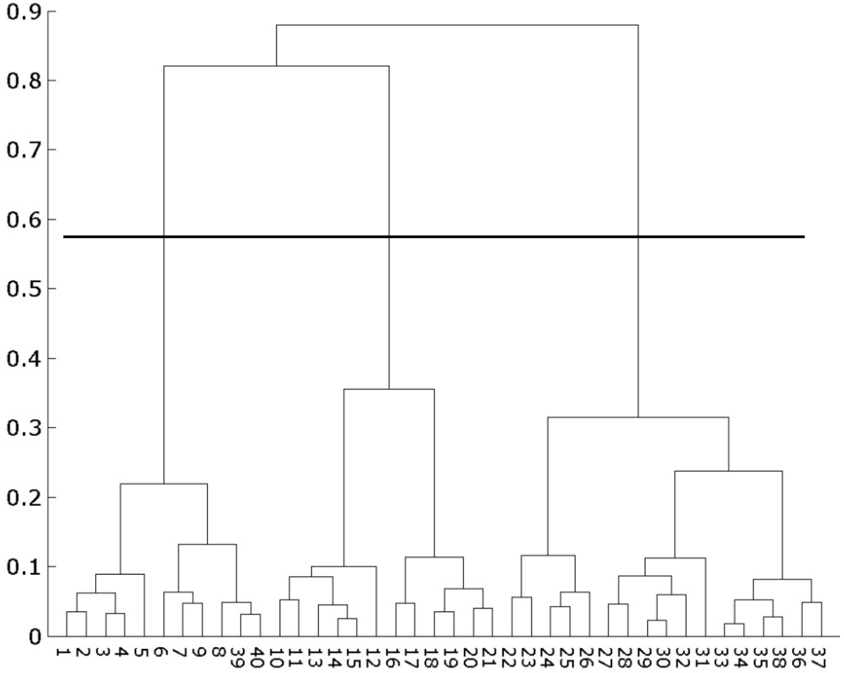


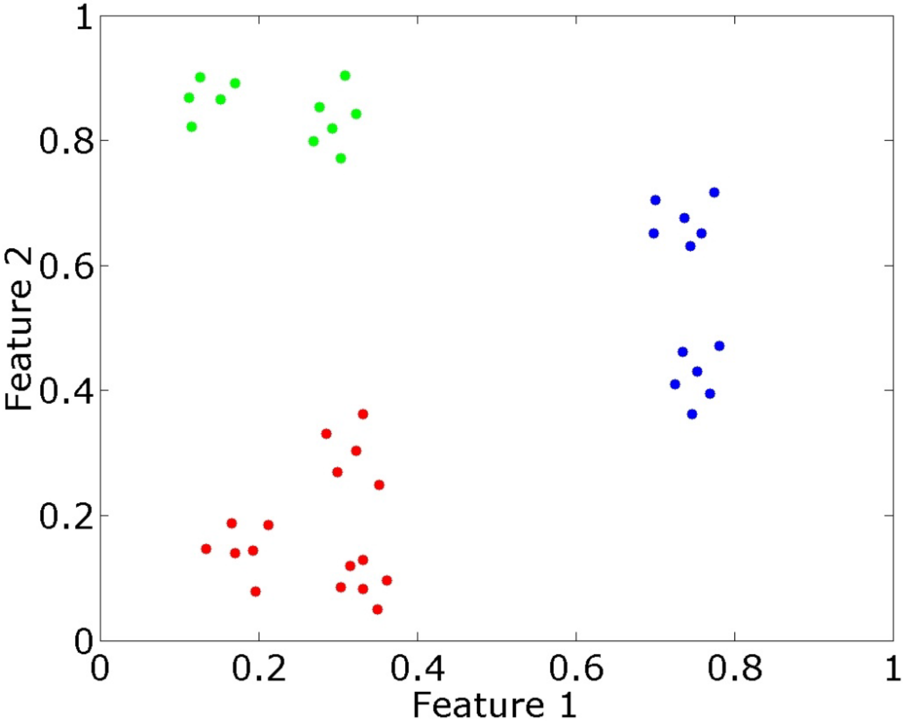
Cut here: 2 clusters

Large jump

Fusion graph (2)

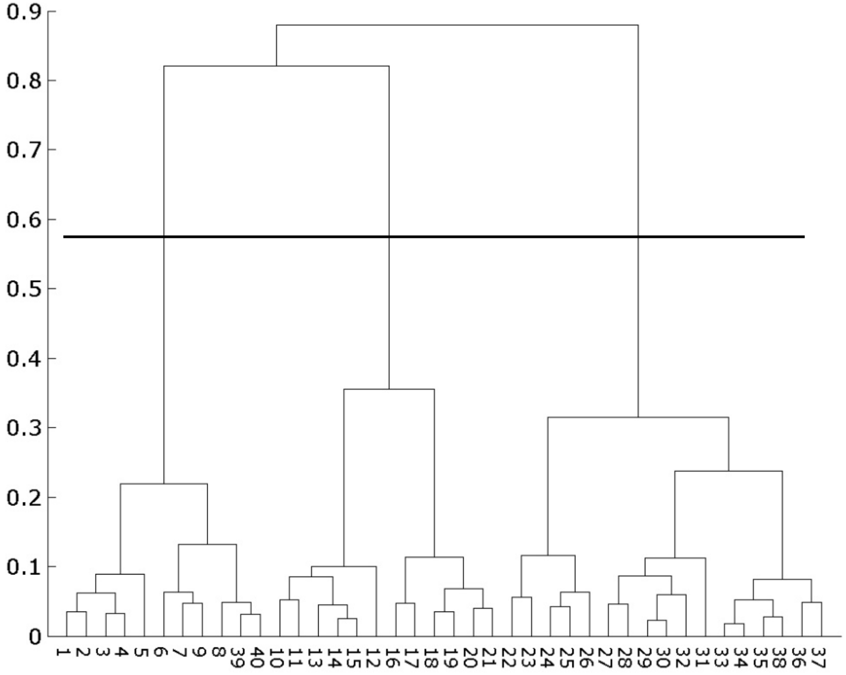
(Euclidean; complete linkage)

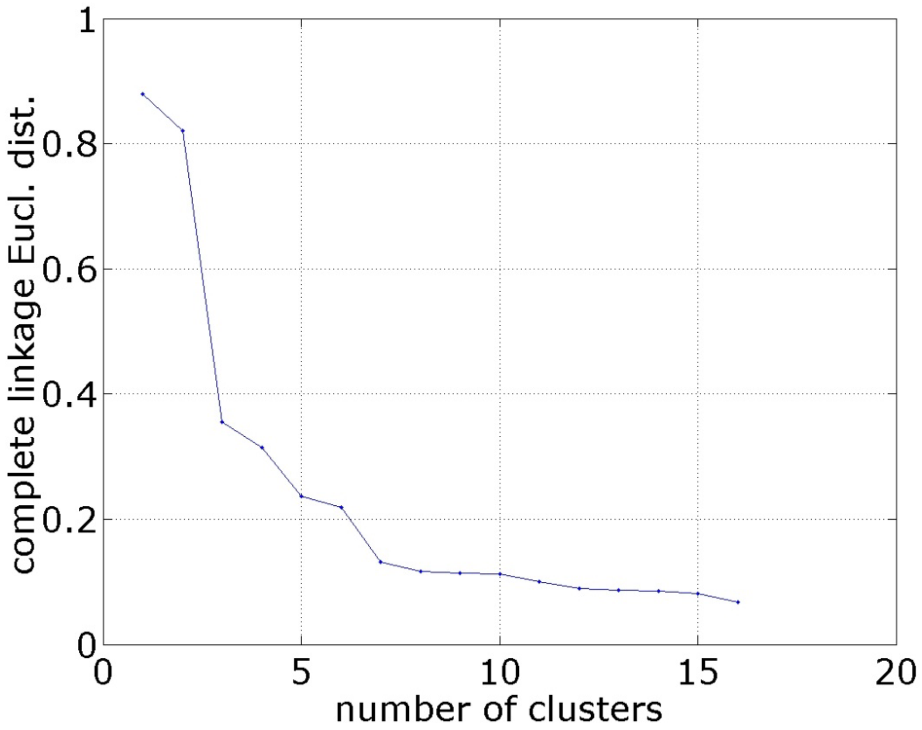




Fusion graph (3)

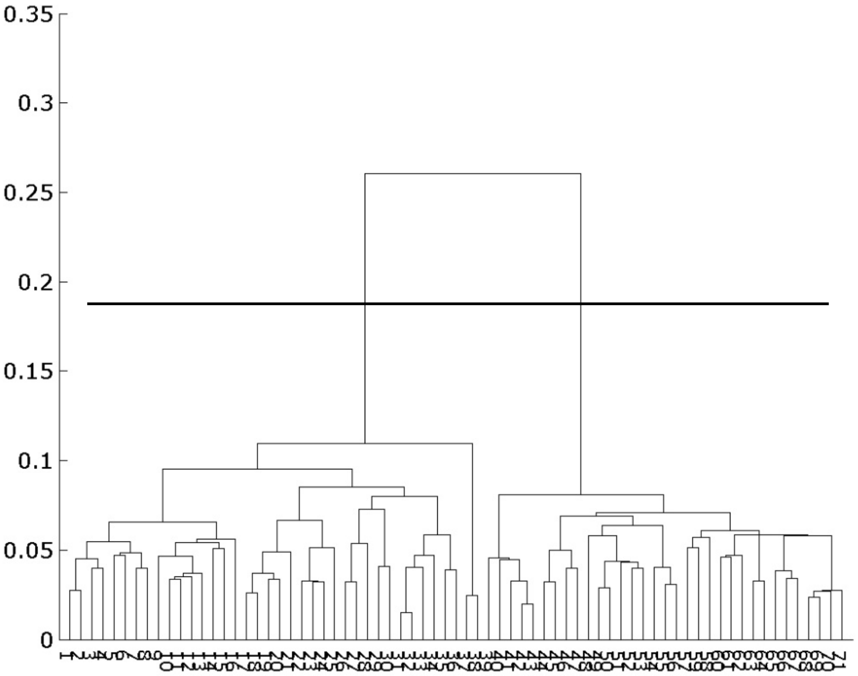
(Euclidean; complete linkage)

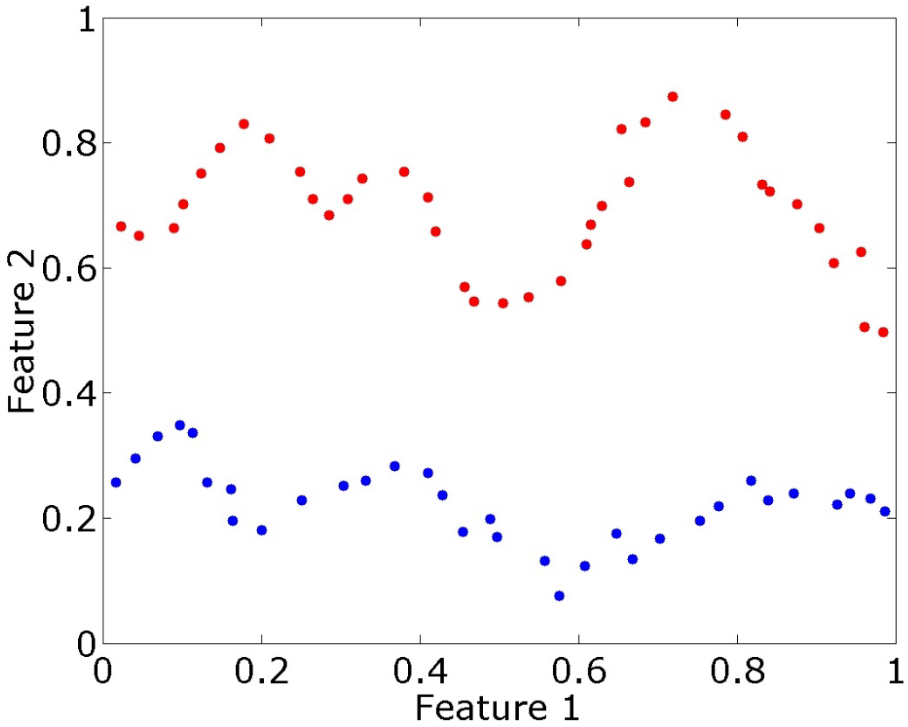




Fusion graph (4)

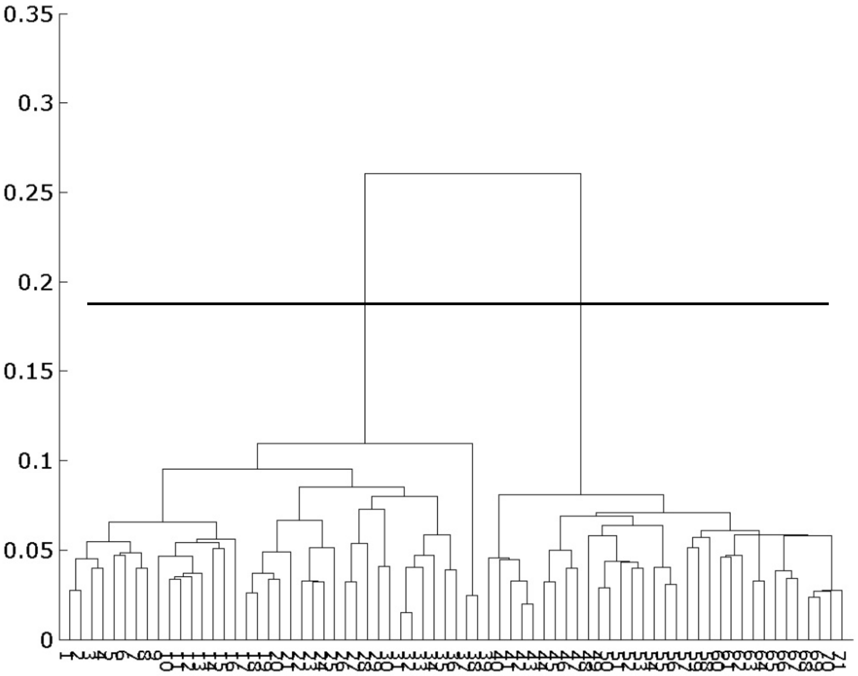
(Euclidean; single linkage)

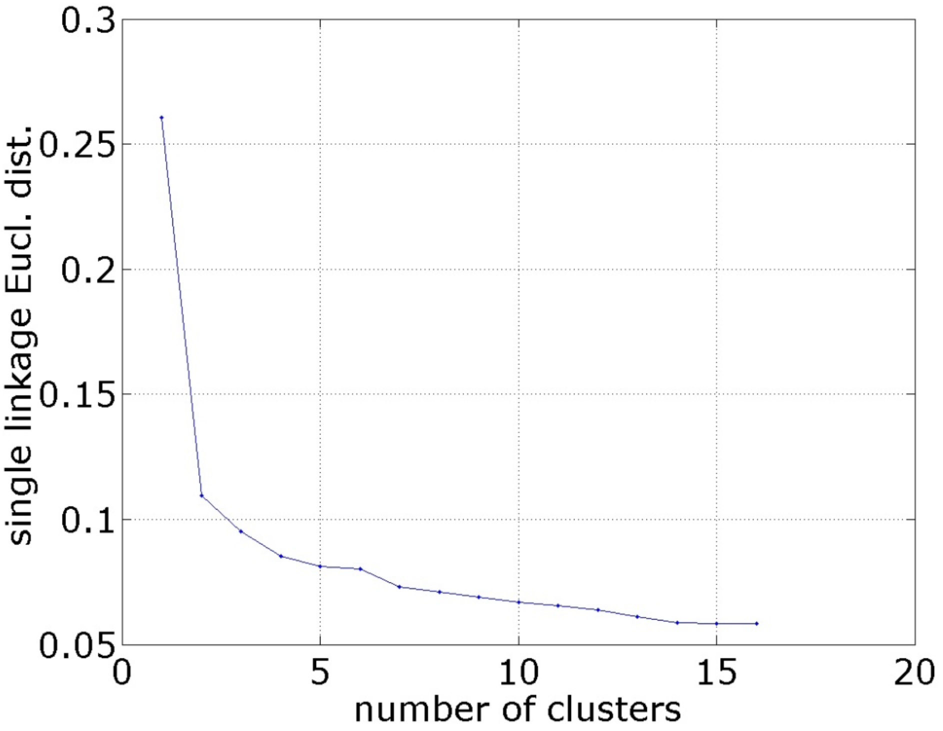




Fusion graph (5)

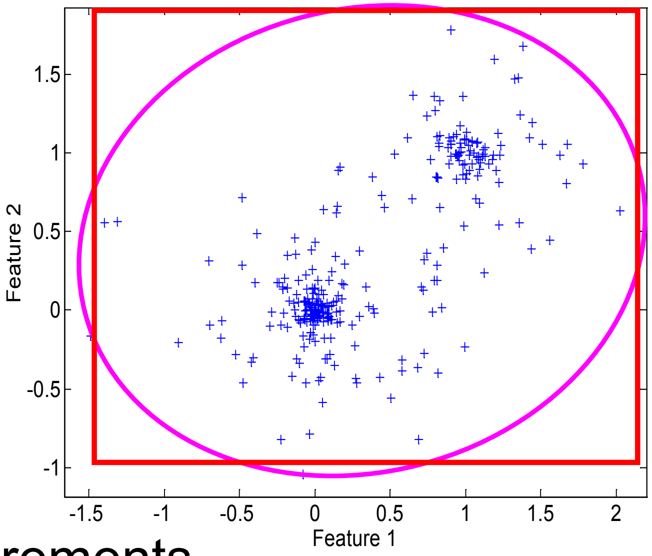
(Euclidean; single linkage)





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



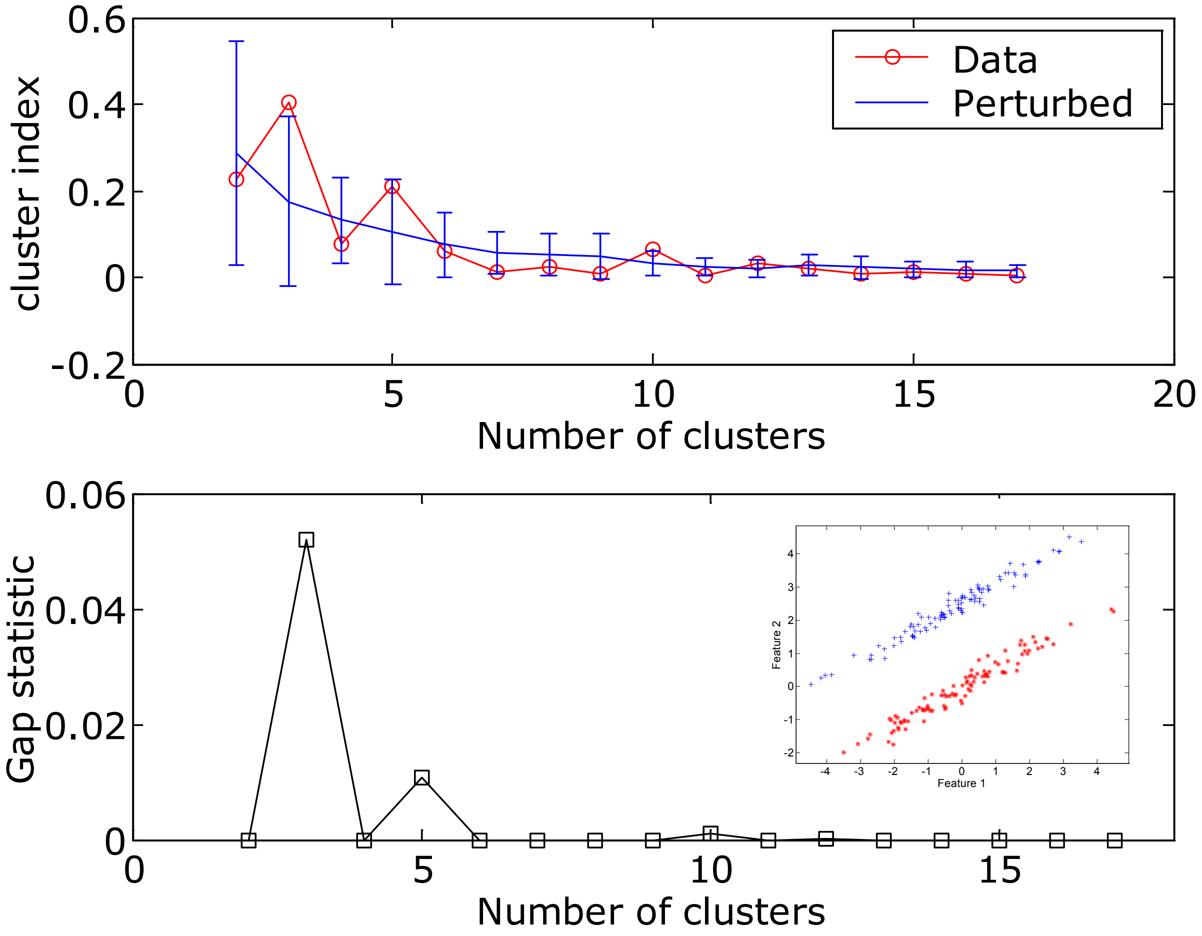
The gap statistic

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 μ^*_j and standard deviation σ^*_j of these perturbed graphs
4. Compute the difference between the data fusion graph and the average perturbed fusion graph (*gap statistic*):

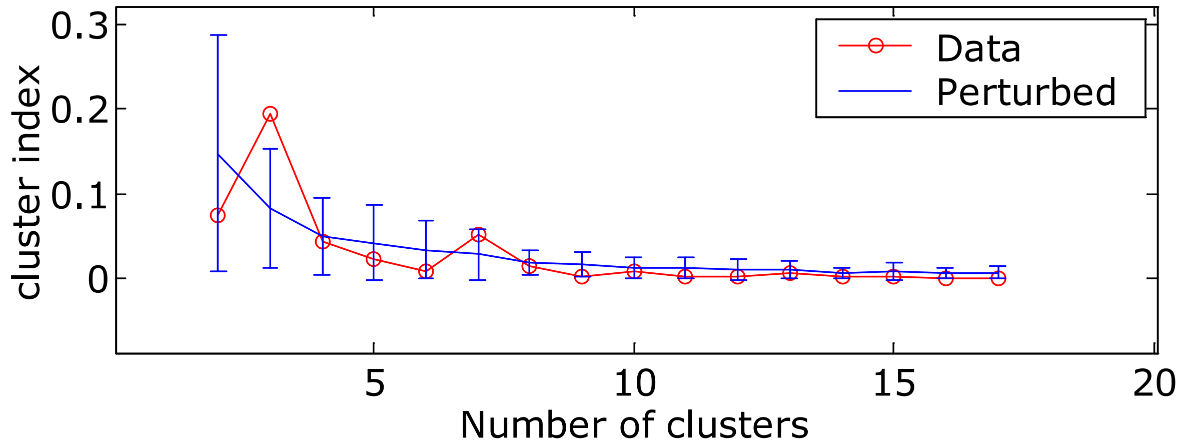
$$g_j^{gap} = \max\left\{ f_j - \mu^*_j, 0 \right\}, \; j = 1, 2, \ldots, g$$

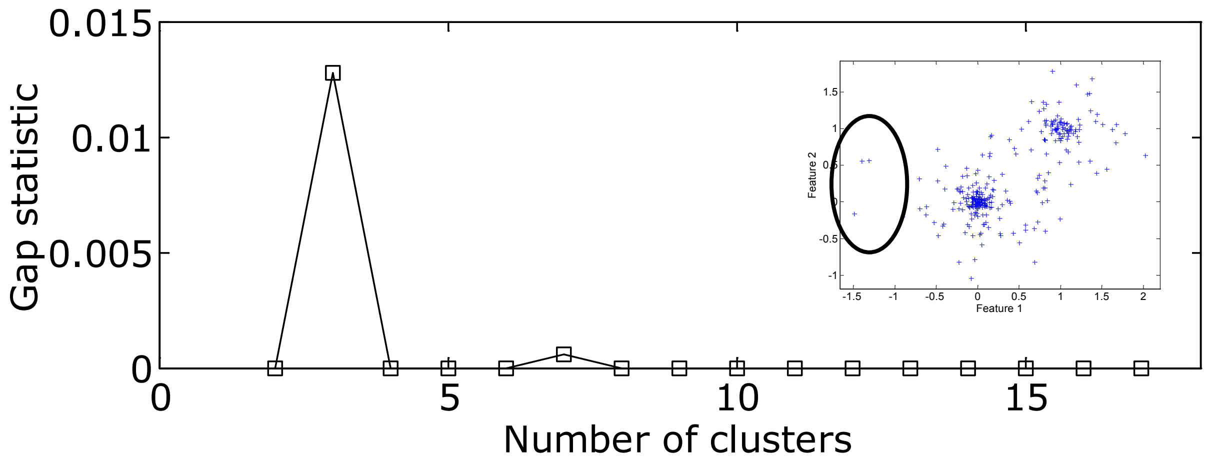
5. Look for large values of gap statistic $g_j^{gap} = f_j$

Gap fusion graph (single linkage)

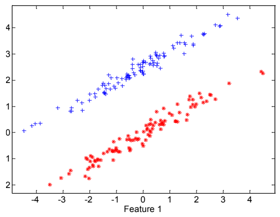
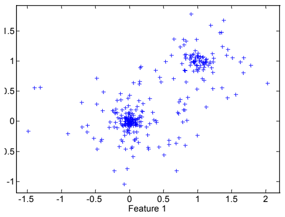
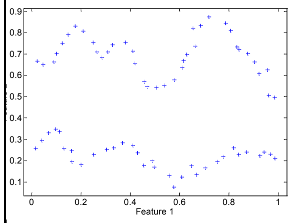
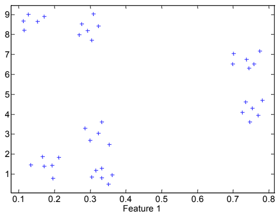
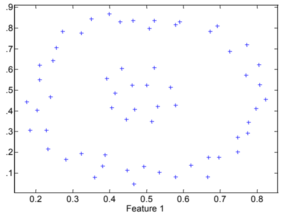


Gap fusion graph (single linkage) (2)





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*

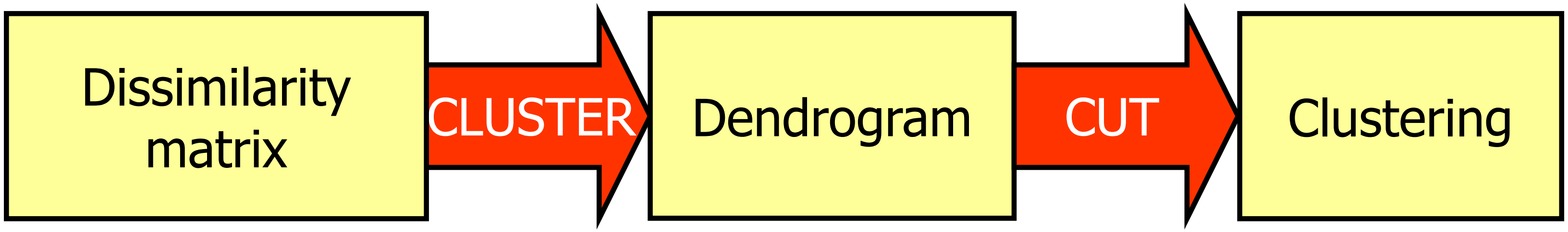
- When applying cluster validation, one also needs to define what a good cluster is – like in clustering itself.
  There's no free lunch...

BioSB research school

Lunch break
Exercise 4.8-4.16

Clustering overview

1. Hierarchical:



Dissimilarity matrix

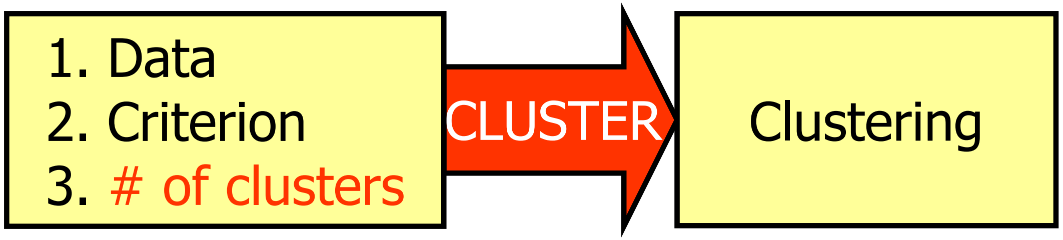
CLUSTER

Dendrogram

CUT

Clustering

2. K-means:

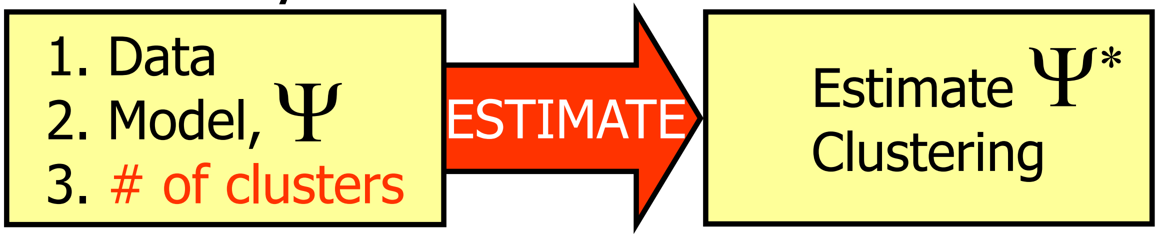


1. Data
2. Criterion
3. # of clusters

CLUSTER

Clustering

3. Density-based:



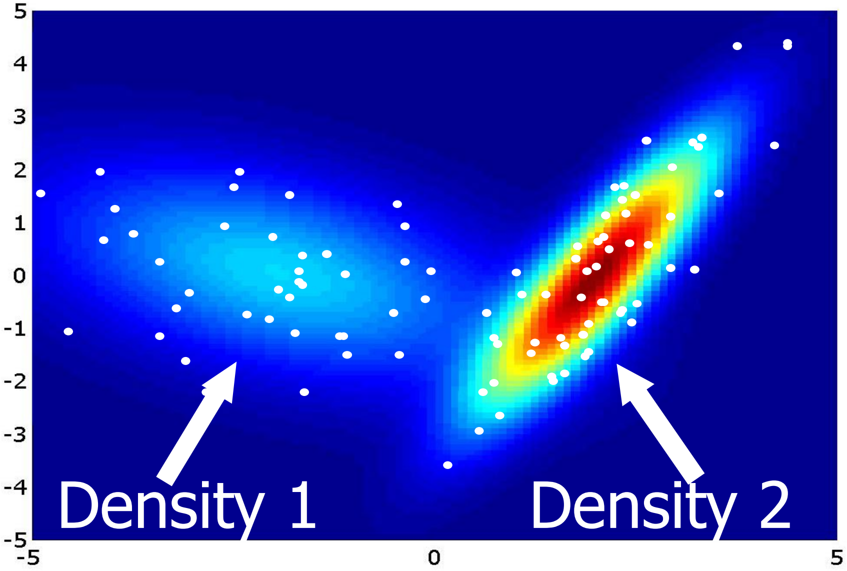
1. Data
2. Model, Ψ
3. # of clusters

ESTIMATE

Estimate Ψ^*
Clustering

Density-based clustering

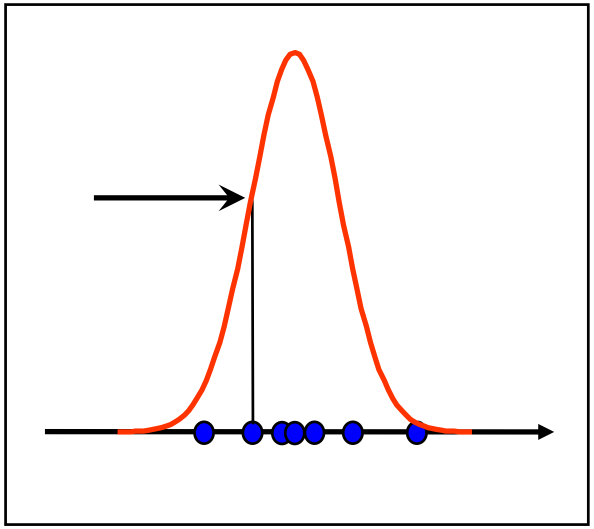
- Each cluster is described by a probability density function
- Total dataset described by a *mixture* of density functions
- Clustering = maximizing the mixture fit
- Clusters are based on *a posteriori probabilities*



Density-based clustering (2)

- Given:
  - n independent objects: $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$
  - probability density function model:

$$p(\boldsymbol{x} \mid \theta) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$



- 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 *θ* of probability distribution from data $D = \{x_1, \ldots, x_n\}$. How?
- Maximize joint probability of the data

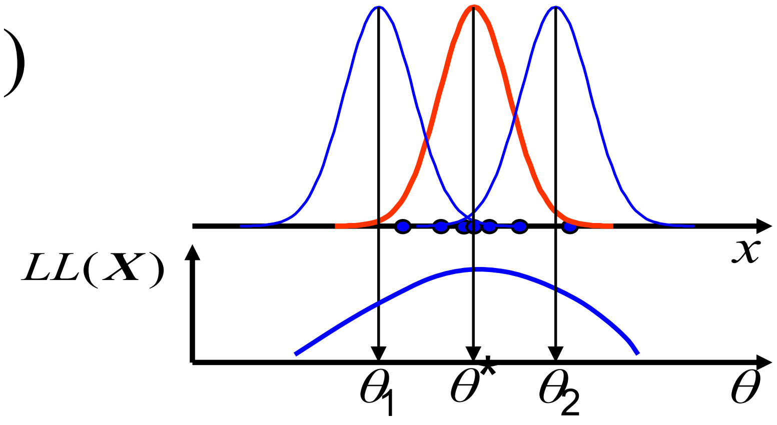
independence

likelihood:

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

log-likelihood:

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



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: $$P(X_1 = x_1, \ldots, X_n = x_n \mid p) = p^{x_1}(1-p)^{1-x_1} \ldots p^{x_n}(1-p)^{1-x_n}$$
$$= p^{n_1}(1-p)^{n-n_1}$$

# of successes

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

Maximum at $p = n_1/n$

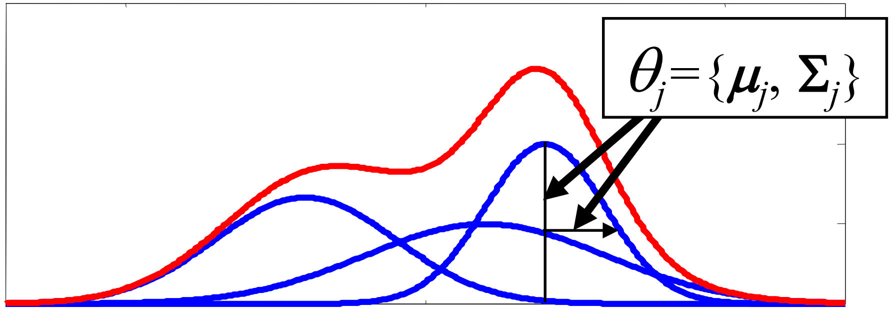
Mixture-of-Gaussians

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

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

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

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

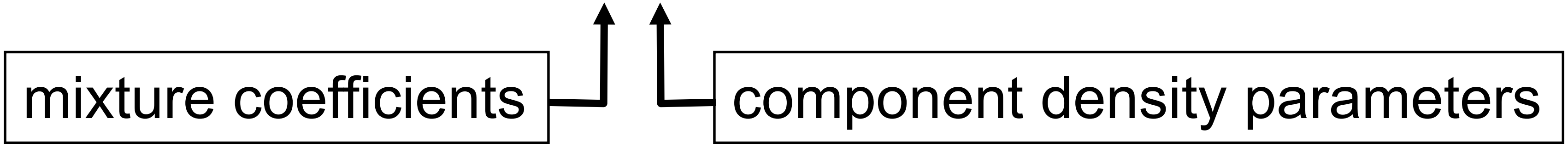


Mixture-of-Gaussians (2)

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

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

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



mixture coefficients

component density parameters

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

EM algorithm

- **Problem:** need to simultaneously estimate two interdependent things...
  - Cluster membership of each object
  - Density parameters of each cluster: $\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j$

- **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: $P(x, y) = P(x \mid y)P(y) = P(y \mid x)P(x)$

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

Intermezzo: Bayes' theorem

From product rule

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

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

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

EM algorithm (2)

arbitrary distribution   hidden variable

$$\log p(D) = \sum_x \log p(x) = \sum_x \sum_z q(z) \log p(x)$$

$$= \sum_{x,z} q(z) \log \frac{p(x,z)}{p(z \mid x)} = \sum_{x,z} q(z) \log \left(\frac{p(x,z)}{p(z \mid x)} \times \frac{q(z)}{q(z)} \right)$$

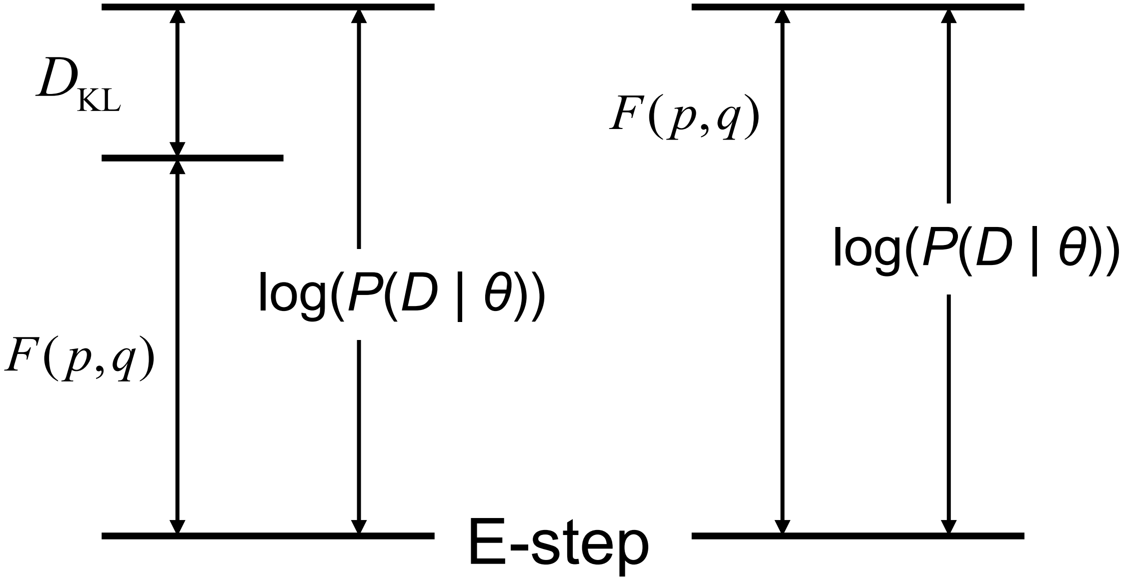
$$= \sum_{x,z} q(z) \log \left(\frac{p(x,z)}{q(z)} \right) + \sum_{x,z} q(z) \log \left(\frac{q(z)}{p(z \mid x)} \right)$$

$$= F(p_{\text{joint}}, q) + D_{KL}(q \,\|\, p_{\text{post}})$$

free energy   relative entropy (≥ 0)

EM algorithm: E-step

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



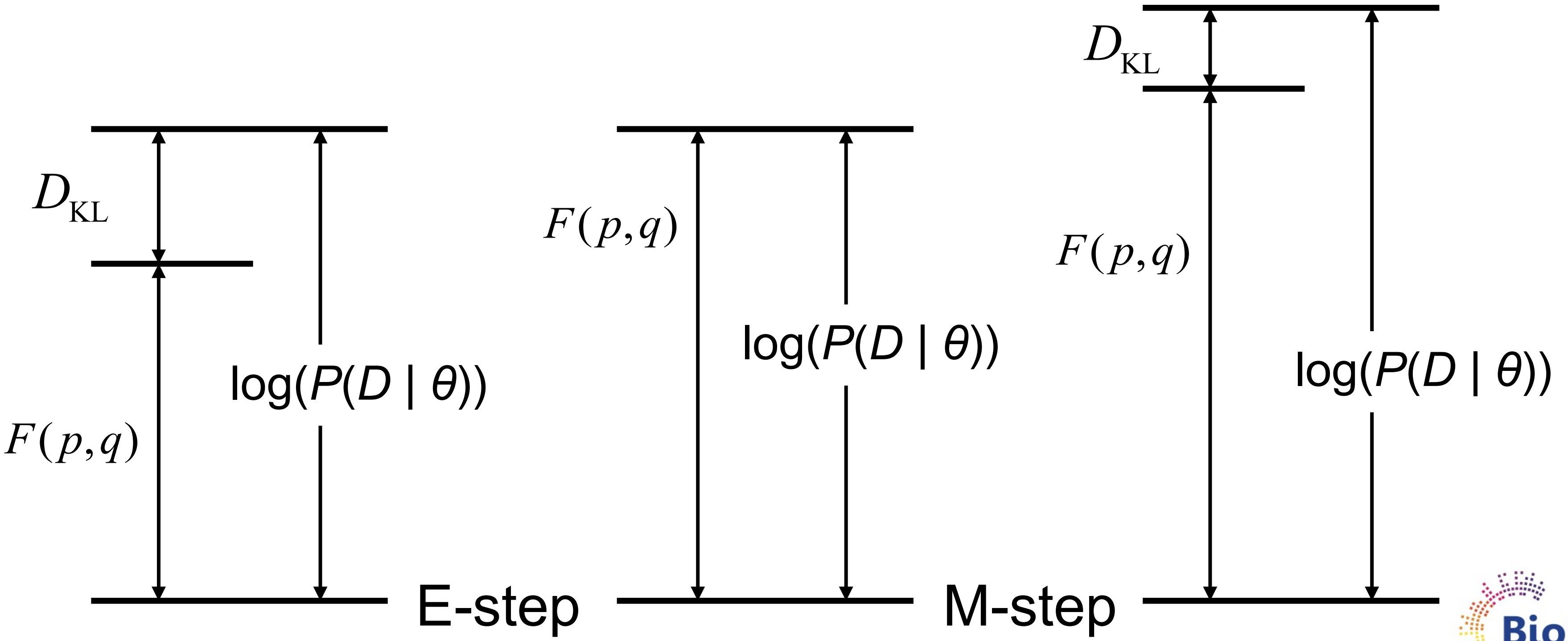
=0 (Gibb's inequality)

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

EM algorithm: M-step

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

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



EM algorithm (3)

Iterate to maximize likelihood:

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

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

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

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

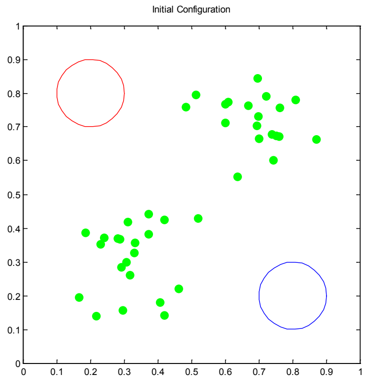
Compare M-step with MoG log-likelihood: $\sum_{i=1}^{n} \log \sum_{j=1}^{g} \pi_j p(\mathbf{x}_i; \theta_j)$

M-step is easier: log within sum

EM: mixture model

Very simple example of a model with hidden variables:

2-component mixture model



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

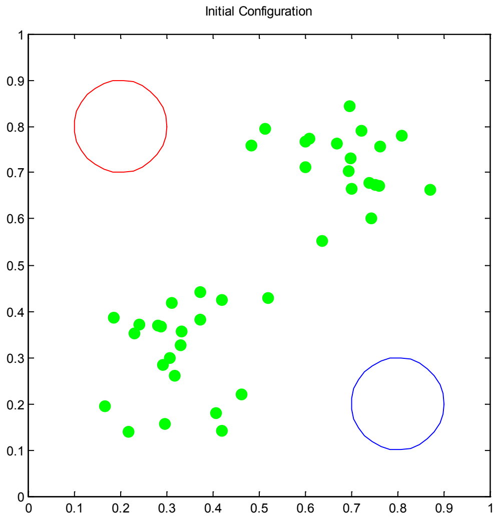
hidden variable *z* = 1,2   - component label

E-step: $p(z = j \mid x, \theta) = \dfrac{p(z = j \mid \theta)\, p(x \mid z = j, \theta)}{p(x \mid \theta)} = \dfrac{\pi_j p_j(x \mid \theta)}{p(x)}$

responsibility

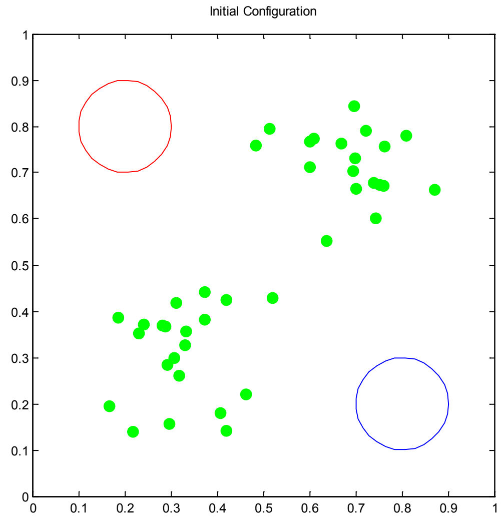
M-step: maximize $\sum_{x, z \in \{1,2\}} p(z \mid x) \log p(x, z \mid \theta)$

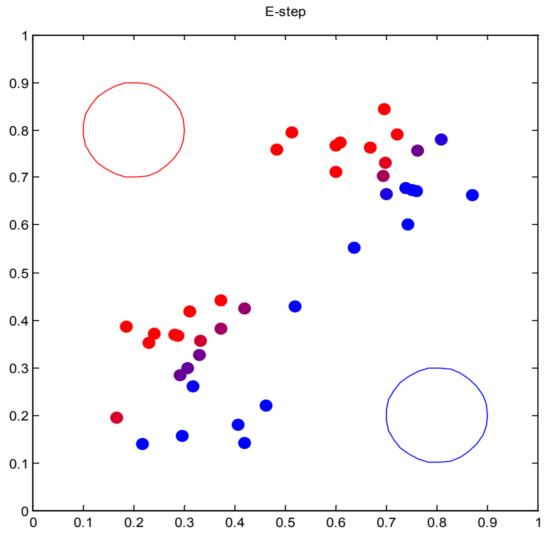
EM: mixture model (2)



Initialization

EM: mixture model (3)





Initialization

E-step

EM: mixture model (4)

- **M-step**: Maximization

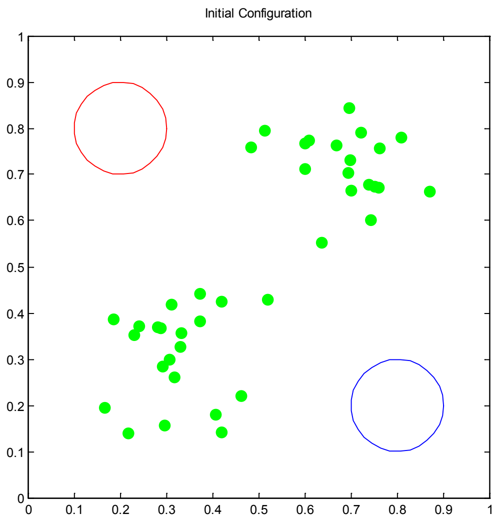
  Maximize the expected complete LL by updating

  - mixture coefficients π_j
  - cluster means and covariances $\theta_j=\{\boldsymbol{\mu}_j, \Sigma_j\}$, $j=1,\dots,g$:

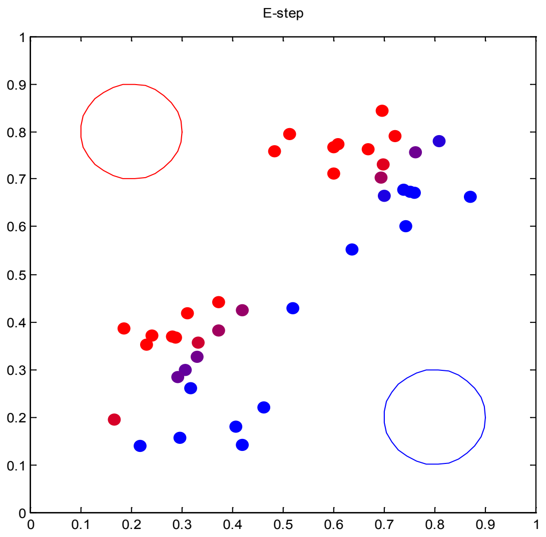
$$\left.\hat{\pi}_j = \frac{1}{n}\sum\nolimits_{i=1}^{n} p(z=j \mid x_i) = \frac{1}{n}\sum\nolimits_{i=1}^{n} w_{ij}\right\} \text{“total membership”}$$

$$\left.\begin{aligned}
\hat{\mu}_j &= \frac{\sum_{i=1}^{n} w_{ij}\boldsymbol{x}_i}{\sum_{i=1}^{n} w_{ij}} \\
\hat{\Sigma}_j &= \frac{\sum_{i=1}^{n} w_{ij}(\boldsymbol{x}_i-\hat{\mu}_j)(\boldsymbol{x}_i-\hat{\mu}_j)^T}{\sum_{i=1}^{n} w_{ij}}
\end{aligned}\right\} \text{weighted sums}$$

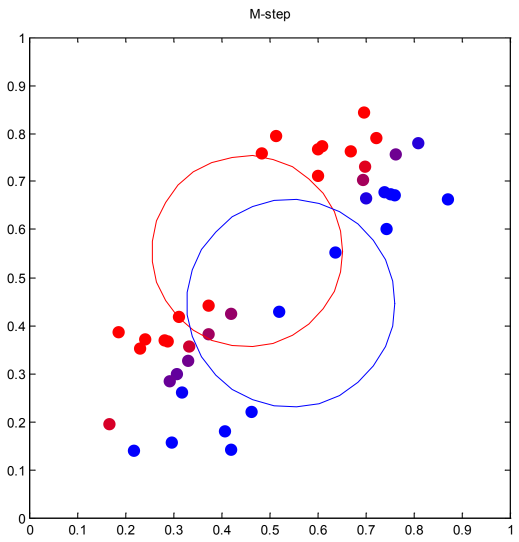
EM: mixture model (5)



Initialization

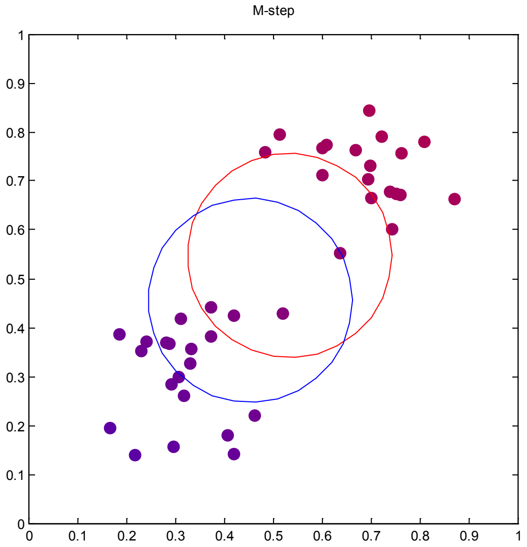


E-step



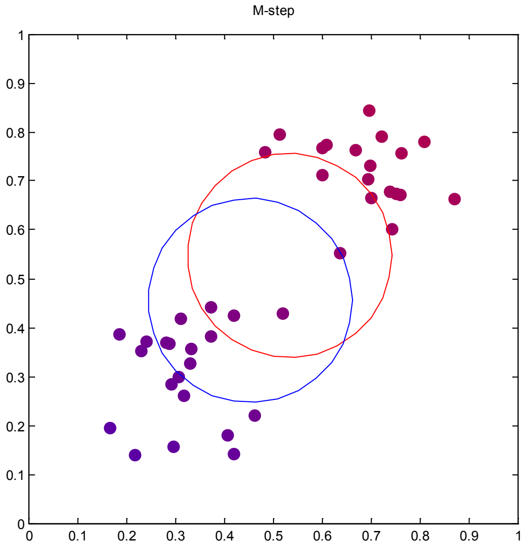
M-step

EM: mixture model (6)

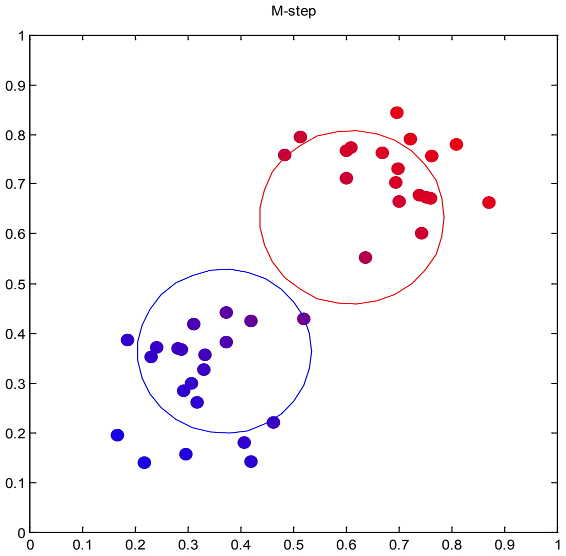


M-step: 3

EM: mixture model (7)

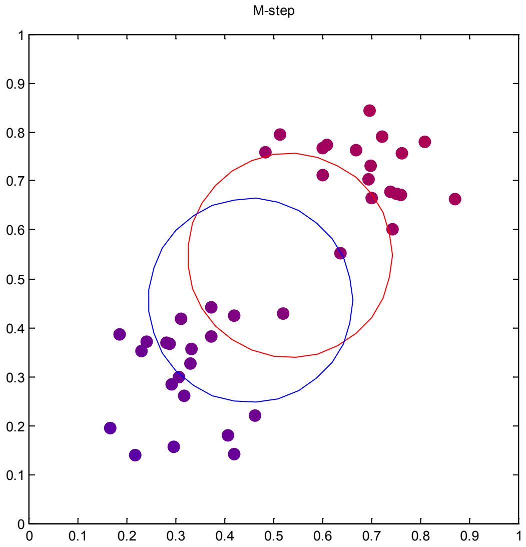


M-step: 3

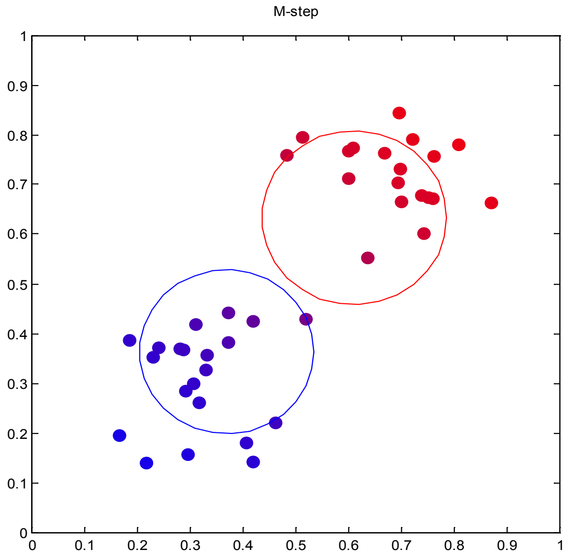


M-step: 5

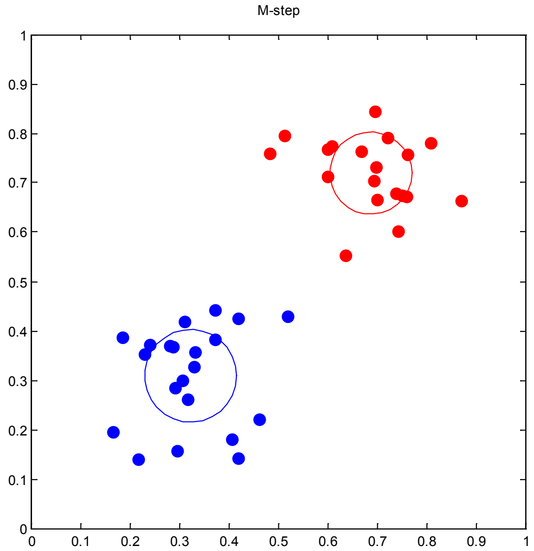
EM: mixture model (8)



M-step: 3



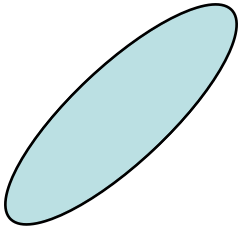
M-step: 5



M-step: 9

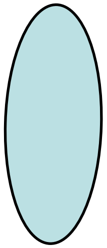
Mixture-of-Gaussians (3)

- ‘Gauss’:



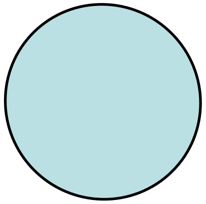
$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$$

- ‘Aligned’:



$$\Sigma = \begin{bmatrix} \sigma_{11} & 0 \\ 0 & \sigma_{22} \end{bmatrix}$$

- ‘Circular’:



$$\Sigma = \begin{bmatrix} \sigma & 0 \\ 0 & \sigma \end{bmatrix}$$

EM: mixture model (9)

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

$$\Sigma = \begin{bmatrix} \varepsilon^2 & 0 & 0 \\ 0 & \varepsilon^2 & 0 \\ 0 & 0 & \varepsilon^2 \end{bmatrix}, \qquad \varepsilon \to 0$$

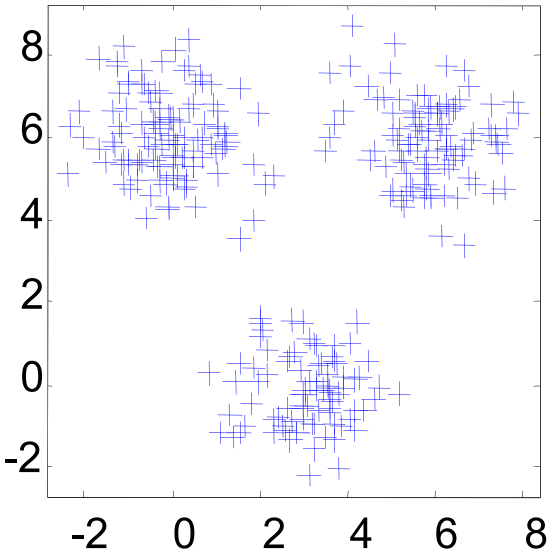
then the EM algorithm simplifies to the *K*-means algorithm (samples are always assigned to the closest cluster!)

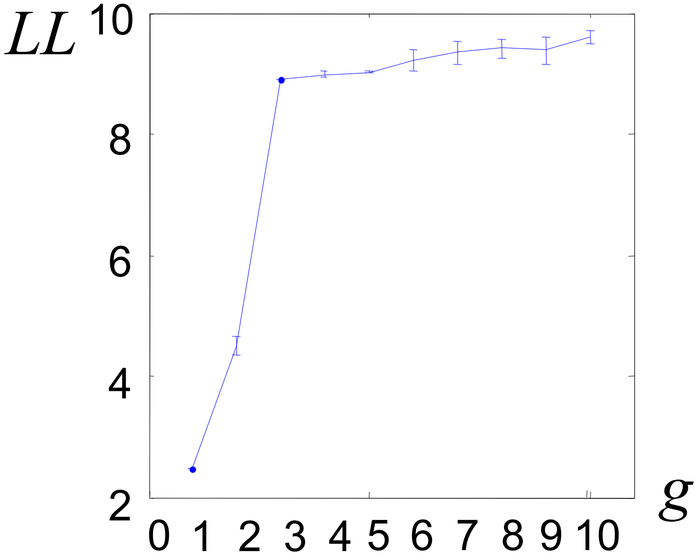
EM algorithm (4)

- 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*

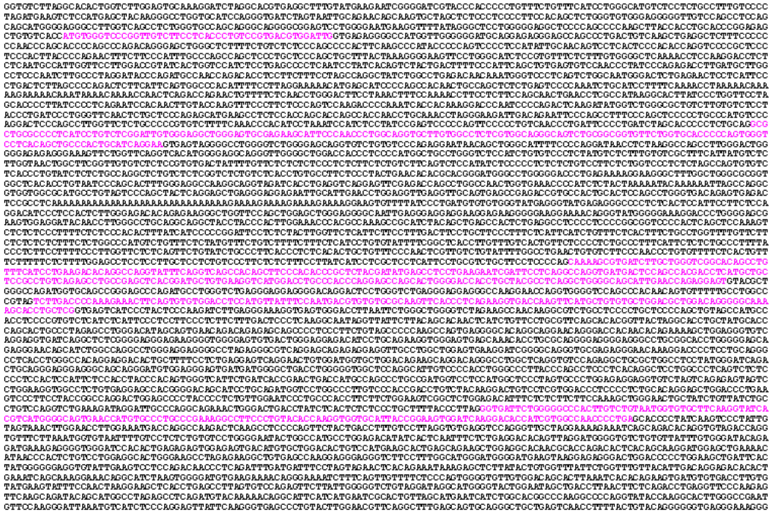
BioSB research school

15min break
Exercise 4.17

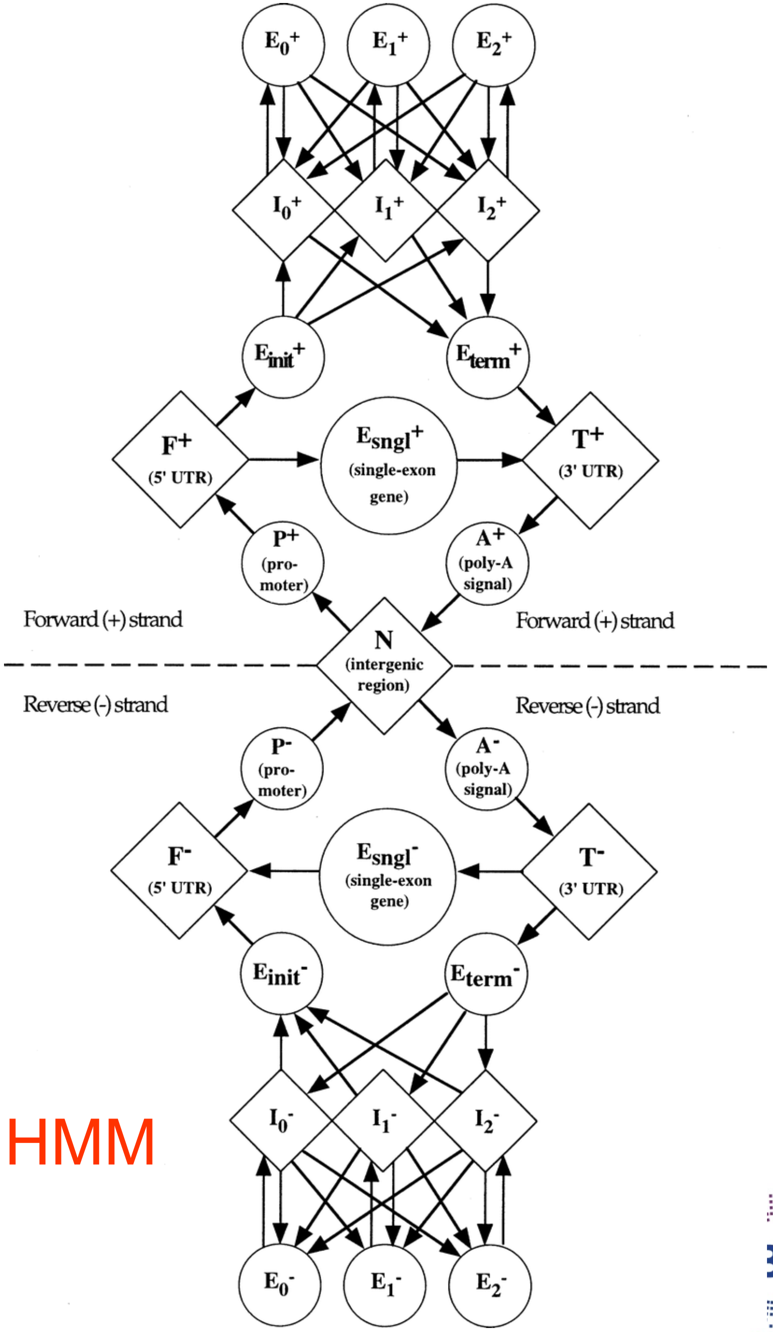
Hidden Markov models

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

Application: genefinding

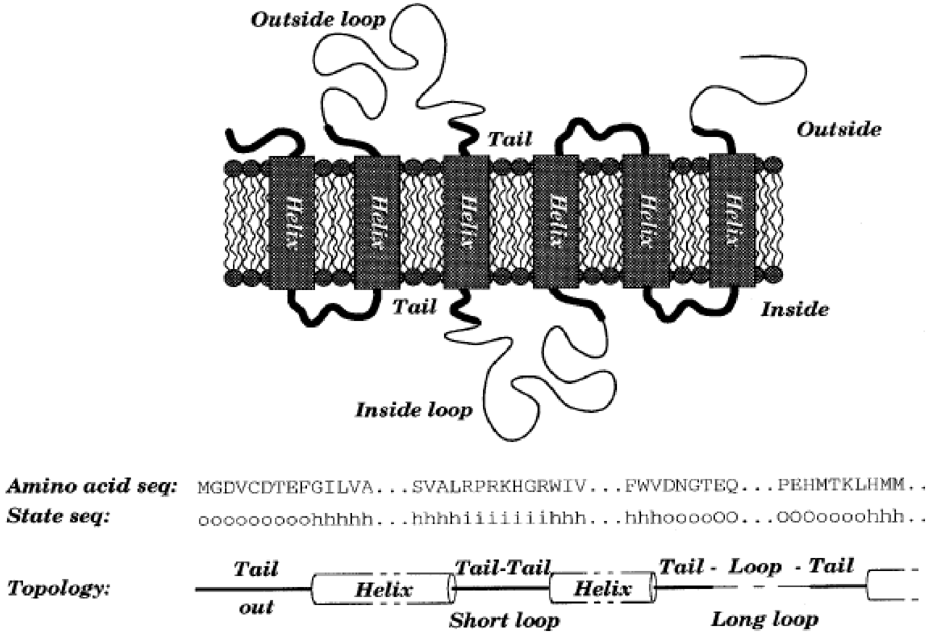


```
GGTGTCTTAGGCACACTGGTCTTGGAGTGCAAAGGATCTAGGCACGTGAGGCTTTGTATGAAGAATCGGGGATCGTACCCACCCCCTGTTTCTGTTTCATCCTGGGCATGTCTCCTCTGCCTTTGTCCCC
TAGATGAAGTCTCCATGAGCTACAAGGGCCTGGTGCATCCAGGGTGATCTAGTAATTGCAGAACAGCAAGTGCTAGCTCTCCCTCCCCTTCCACAGCTCTGGGTGTGGGAGGGGGTTGTCCAGCCTCCAG
CAGCATGGGGAGGGCCTTGGTCAGCCTCTGGGTGCCAGCAGGGCAGGGGCGGAGTCCTGGGGAATGAAGGTTTTATAGGGCTCCTGGGGGAGGCTCCCCAGCCCCAAGCTTACCACCTGCACCCGGAGAG
CTGTGTCACCATGTGGGTCCCGGTTGTCTTCCTCACCCTGTCCGTGACGTGGATTGGTGAGAGGGGCCATGGTTGGGGGATGCAGGAGAGGGAGCCAGCCCTGACTGTCAAGCTGAGGCTCTTTCCCCC
CCAACCCAGCACCCCAGCCCAGACAGGGAGCTGGGCTCTTTTCTGTCTCTCCCAGCCCCACTTCAAGCCCATACCCCCAGTCCCCTCCATATTGCAACAGTCCTCACTCCCACACCAGGTCCCCGCTCCC
TCCCACTTACCCCAGAACTTTCTTCCCATTTGCCCAGCCAGCTCCCTGCTCCCAGCTGCTTTACTAAAGGGGAAGTTCCTGGGCATCTCCGTGTTTCTCTTTGTGGGGCTCAAAACCTCCAAGGACCTCT
CTCAATGCCATTGGTTCCTTGGACCGTATCACTGGTCCATCTCCTGAGCCCCTCAATCCTATCACAGTCTACTGACTTTTCCCATTCAGCTGTGAGTGTCCAACCCTATCCCAGAGACCTTGATGCTTGG
CCTCCCAATCTTGCCCTAGGATACCCAGATGCCAACCAGACACCTCCTTCTTTCCTAGCCAGGCTATCTGGCCTGAGACAACAAATGGGTCCCTCAGTCTGGCAATGGGACTCTGAGAACTCCTCATTCC
CTGACTCTTAGCCCCAGACTCTTCATTCAGTGGCCCACATTTTCCTTAGGAAAAACATGAGCATCCCCAGCCACAACTGCCAGCTCTCTGAGTCCCCAAATCTGCATCCTTTTCAAAACCTAAAAACAAA
AAGAAAAACAAATAAAACAAAACCAACTCAGACCAGAACTGTTTTCTCAACCTGGGACTTCCTAAACTTTCCAAAACCTTCCTCTTCCAGCAACTGAACCTCGCCATAAGGCACTTATCCCTGGTTCCTA
GCACCCCTTATCCCCTCAGAATCCACAACTTGTACCAAGTTTCCCTTCTCCCAGTCCAAGACCCCAAATCACCACAAAGGACCCAATCCCCAGACTCAAGATATGGTCTGGGCGCTGTCTTGTGTCTCCT
ACCCTGATCCCTGGGTTCAACTCTGCTCCCAGAGCATGAAGCCTCTCCACCAGCACCAGCCACCAACCTGCAAACCTAGGGAAGATTGACAGAATTCCCAGCCTTTCCCAGCTCCCCCTGCCCATGTCCC
AGGACTCCCAGCCTTGGTTCTCTGCCCCCGTGTCTTTTCAAACCCACATCCTAAATCCATCTCCTATCCGAGTCCCCCAGTTCCCCCTGTCAACCCTGATTCCCCTGATCTAGCACCCCCTCTGCAGGCG
CTGCGCCCCTCATCCTGTCTCGGATTGTGGGAGGCTGGGAGTGCGAGAAGCATTCCCAACCCTGGCAGGTGCTTGTGGCCTCTCGTGGCAGGGCAGTCTGCGGCGGTGTTCTGGTGCACCCCCAGTGGGT
CCTCACAGCTGCCCACTGCATCAGGAAGTGAGTAGGGGCCTGGGGTCTGGGGAGCAGGTGTCTGTGTCCCAGAGGAATAACAGCTGGGCATTTTCCCCAGGATAACCTCTAAGGCCAGCCTTGGGACTGG
GGGAGAGAGGGAAAGTTCTGGTTCAGGTCACATGGGGAGGCAGGGTTGGGGCTGGACCACCCTCCCCATGGCTGCCTGGGTCTCCATCTGTGTCCCTCTATGTCTCTTTGTGTCGCTTTCATTATGTCTC
TTGGTAACTGGCTTCGGTTGTGTCTCTCCGTGTGACTATTTGTTCTCTCTCTCCCTCTCTTCTCTGTCTTCAGTCTCCATATCTCCCCCTCTCTCTGTCCTTCTCTGGTCCCTCTCTAGCCAGTGTGTC
TCACCCTGTATCTCTCTGCCAGGCTCTGTCTCTCGGTCTCTGTCTCACCTGTGCCTTCTCCCTACTGAACACACGCACGGGATGGGCCTGGGGGACCCTGAGAAAAGGAAGGGCTTTGGCTGGGCGCGGT
GGCTCACACCTGTAATCCCAGCACTTTGGGAGGCCAAGGCAGGTAGATCACCTGAGGTCAGGAGTTCGAGACCAGCCTGGCCAACTGGTGAAACCCCATCTCTACTAAAAATACAAAAAATTAGCCAGGC
GTGGTGGCGCATGCCTGTAGTCCCAGCTACTCAGGAGCTGAGGGAGGAGAATTGCATTGAACCTGGAGGTTGAGGTTGCAGTGAGCCGAGACCGTGCCACTGCACTCCAGCCTGGGTGACAGAGTGAGAC
TCCGCCTCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAAAAGAAAAGAAAAGAAAAGGAAGTGTTTTATCCCTGATGTGTGTGGGTATGAGGGTATGAGAGGGCCCCTCTCACTCCATTCCTTCTCCA
GGACATCCCTCCACTCTTGGGAGACACAGAGAAGGGCTGGTTCCAGCTGGAGCTGGGAGGGGCAATTGAGGGAGGAGGAAGGAGAAGGGGGAAGGAAAACAGGGTATGGGGAAAGGACCCTGGGGAGCG
AAGTGGAGGATACAACCTTGGGCCTGCAGGCAGGCTACCTACCCACTTGGAAACCCACGCCAAAGCCGCATCTACAGCTGAGCCACTCTGAGGCCTCCCCTCCCCGGCGGTCCCCACTCAGCTCCAAAGT
CTCTCTCCCTTTTCTCTCCCACACTTTATCATCCCCCGGATTCCTCTCTACTTGGTTCTCATTCTTCCTTTGACTTCCTGCTTCCCTTTCTCATTCATCTGTTTCTCACTTTCTGCCTGGTTTTGTTCTT
CTCTCTCTCTTTCTCTGGCCCATGTCTGTTTCTCTATGTTTCTGTCTTTTCTTTCTCATCCTGTGTATTTTCGGCTCACCTTGTTTGTCACTGTTCTCCCCTCTGCCCTTTCATTCTCTCTGCCCTTTTA
CCCTCTTCCTTTTCCCTTGGTTCTCTCAGTTCTGTATCTGCCCTTCACCCTCTCACACTGCTGTTTCCCAACTCGTTGTCTGTATTTTGGCCTGAACTGTGTCTTCCCAACCCTGTGTTTTCTCACTGTT
TCTTTTTCTCTTTTGGAGCCTCCTCCTTGCTCCTCTGTCCCTTCTCTCTTTCCTTATCATCCTCGCTCCTCATTCCTGCGTCTGCTTCCTCCCCAGCAAAAGCGTGATCTTGCTGGGTCGGCACAGCCTG
TTTCATCCTGAAGACACAGGCCAGGTATTTCAGGTCAGCCACAGCTTCCCACACCCGCTCTACGATATGAGCCTCCTGAAGAATCGATTCCTCAGGCCAGGTGATGACTCCAGCCACGACCTCATGCTGC
TCCGCCTGTCAGAGCCTGCCGAGCTCACGGATGCTGTGAAGGTCATGGACCTGCCCACCCAGGAGCCAGCACTGGGGACCACCTGCTACGCCTCAGGCTGGGCAGCATTGAACCAGAGGAGTGTACGCT
GGGCCAGATGGTGCAGCCGGGAGCCCAGATGCCTGGGTCTGAGGGAGGAGGGGACAGGACTCCTGGGTCTGAGGGAGGAGGGCCAAGGAACCAGGTGGGGTCCAGCCCACAACAGTGTTTTTGCCTGGCC
CGTAGTCTTGACCCCAAAGAAACTTCAGTGTGTGGACCTCCATGTTATTTCCAATGACGTGTGTGCGCAAGTTCACCCTCAGAAGGTGACCAAGTTCATGCTGTGTGCTGGACGCTGGACAGGGGGCAAA
AGCACCTGCTCGGTGAGTCATCCCTACTCCCAAGATCTTGAGGGAAAGGTGAGTGGACCTTAATTCTGGGCTGGGGTCTAGAAGCCAACAAGGCGTCTGCCTCCCCTGCTCCCCAGCTGTAGCCATGCC
ACCTCCCCGTGTCTCATCTCATTCCCTCCTTCCCTCTTCTTTGACTCCCTCAAGGCAATAGGTTATTCTTACAGCACAACTCATCTGTTCCTGCGTTCAGCACACGGTTACTAGGCACCTGCTATGCACC
CAGCACTGCCCTAGAGCCTGGGACATAGCAGTGAACAGACAGAGAGCAGCCCCTCCCTTCTGTAGCCCCCAAGCCAGTGAGGGGCACAGGCAGGAACAGGGACCACAACACAGAAAAGCTGGAGGGTGTC
AGGAGGTGATCAGGCTCTCGGGGAGGGAGAAGGGGTGGGGAGTGTGACTGGGAGGAGACATCCTGCAGAAGGTGGGAGTGAGCAAACACCTGCGCAGGGGAGGGGAGGGCCTGCGGCACCTGGGGGAGCA
GAGGGAACAGCATCTGGCCAGGCCTGGGAGGAGGGGCCTAGAGGGCGTCAGGAGCAGAGAGGAGGTTGCCTGGCTGGAGTGAAGGATCGGGGCAGGGTGCGAGAGGGAACAAAGGACCCCTCCTGCAGGG
CCTCACCTGGGCCACAGGAGGACACTGCTTTTCCTCTGAGGAGTCAGGAACTGTGGATGGTGCTGGACAGAAGCAGGACAGGGCCTGGCTCAGGTGTCCAGAGGCTGCGCTGGCCTCCTATGGGATCAGA
CTGCAGGGAGGGAGGGCAGCAGGGATGTGGAGGGAGTGATGATGGGGCTGACCTGGGGGTGGCTCCAGGCATTGTCCCCACCTGGGCCCTTACCCAGCCTCCCTCACAGGCTCCTGGCCCTCAGTCTCTC
CCCTCCACTCCATTCTCCACCTACCCACAGTGGGTCATTCTGATCACCGAACTGACCATGCCAGCCCTGCCGATGGTCCTCCATGGCTCCCTAGTGCCCTGGAGAGGAGGTGTCTAGTCAGAGAGTAGTC
CTGGAAGGTGGCCTCTGTGAGGAGCCACGGGGACAGCATCCTGCAGATGGTCCTGGCCCTTGTCCCACCGACCTGTCTACAAGGACTGTCCTCGTGGACCCTCCCCTCTGCACAGGAGCTGGACCCTGAA
GTCCCTTCCTACCGGCCAGGACTGGAGCCCCTACCCCTCTGTTGGAATCCCTGCCCACCTTCTTCTGGAAGTCGGCTCTGGAGACATTTCTCTCTTCTTCCAAAGCTGGGAACTGCTATCTGTTATCTGC
CTGTCCAGGTCTGAAAGATAGGATTGCCCAGGCAGAAACTGGGACTGACCTATCTCACTCTCTCCCTGCTTTTACCCTTAGGGTGATTCTGGGGGCCCACTTGTCTGTAATGGTGTGCTTCAAGGTATCA
CGTCATGGGGCAGTGAACCATGTGCCCTGCCCGAAAGGCCTTCCCTGTACACCAAGGTGGTGCATTACCGGAAGTGGATCAAGGACACCATCGTGGCCAACCCCTGAGCACCCCTATCAAGTCCCTATTG
TAGTAAACTTGGAACCTTGGAAATGACCAGGCCAAGACTCAAGCCTCCCCAGTTCTACTGACCTTTGTCCTTAGGTGTGAGGTCCAGGGTTGCTAGGAAAAGAAATCAGCAGACACAGGTGTAGACCAGG
TGTTTCTTAAATGGTGTAATTTTGTCCTCTCTGTGTCCTGGGGAATACTGGCCATGCCTGGAGACATATCACTCAATTTCTCTGAGGACACAGTTAGGATGGGGTGTCTGTGTTATTTGTGGGATACAGA
GATGAAAGAGGGGTGGGATCCACACTGAGAGAGTGGAGAGTGACATGTGCTGGACACTGTCCATGAAGCACTGAGCAGAAGCTGGAGGCACAACGCACCAGACACTCACAGCAAGGATGGAGCTGAAAAC
ATAACCCACTCTGTCCTGGAGGCACTGGGAAGCCTAGAGAAGGCTGTGAGCCAAGGAGGGAGGGTCTTCCTTTGGCATGGGATGGGGATGAAGTAAGGAGAGGGACTGGACCCCCTGGAAGCTGATTCAC
TATGGGGGGAGGTGTATTGAAGTCCTCCAGACAACCCTCAGATTTGATGATTTCCTAGTAGAACTCACAGAAATAAAGAGCTCTTATACTGTGGTTTATTCTGGTTTGTTACATTGACAGGAGACACACT
GAAATCAGCAAAGGAAACAGGCATCTAAGTGGGGATGTGAAGAAAACAGGGAAAATCTTTCAGTTGTTTTCTCCCAGTGGGGTGTTGTGGACAGCACTTAAATCACACAGAAGTGATGTGTGACCTTGTG
TATGAAGTATTTCCAACTAAGGAAGCTCACCTGAGCCTTAGTGTCCAGAGTTCTTATTGGGGTCTGTAGGATAGGCATGGGGTACTGGAATAGCTGACCTTAACTTCTCAGACCTGAGGTTCCCAAGAG
TTCAAGCAGATACAGCATGGCCTAGAGCCTCAGATGTACAAAAACAGGCATTCATCATGAATCGCACTGTTAGCATGAATCATCTGGCACGGCCCAAGGCCCCAGGTATACCAAGGCACTTGGCCGAAT
GTTCCAAGGGATTAAATGTCATCTCCCAGGAGTTATTCAAGGGTGAGCCCTGTACTTGGAACGTTCAGGCTTTGAGCAGTGCAGGGCTGCTGAGTCAACCTTTTACTGTACAGGGGGGTGAGGAAAGGG
```



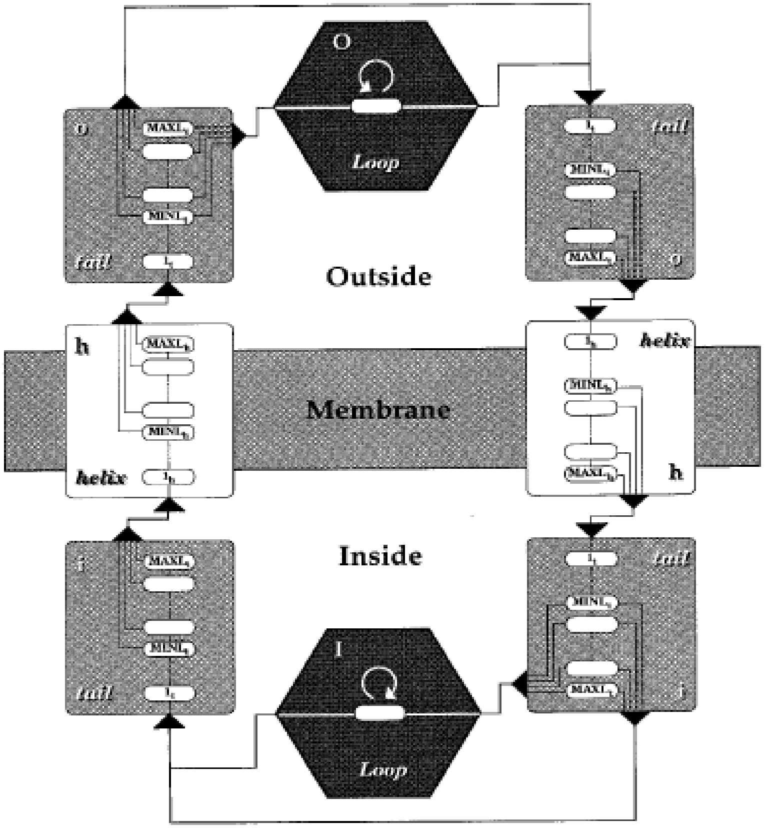
generalized HMM

Application: transmembrane proteins



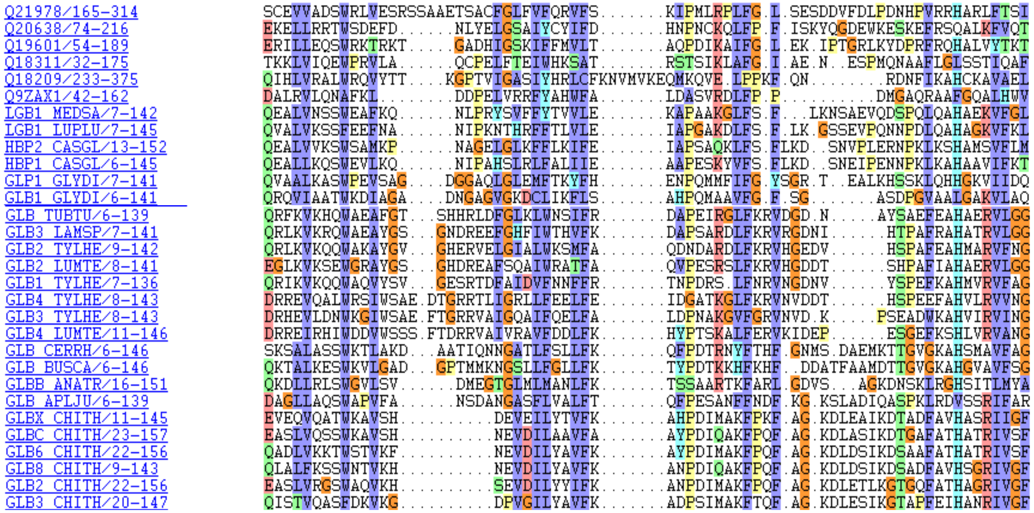
Amino acid seq: MGDVCDTEFGILVA...SVALRPRKHGRWIV...FWVDNGTEQ...PEHMTKLHMM..

State seq: oooooooooohhhhh...hhhhiiiiiiiihhh...hhhoooooO...OOOooooohhh..



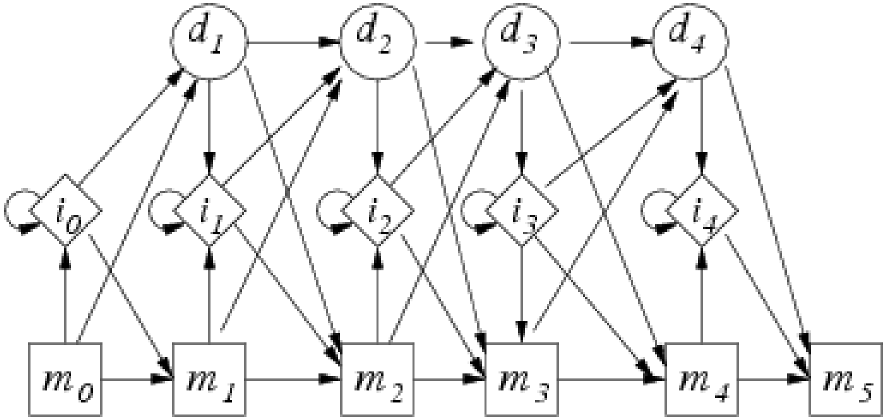
HMM

Application: protein domains



```
Q21978/165-314      SCEVVADSWRLVESRSSAAETSACFGLFVFQRVFS.......KIPMLRPLFG.L.SESDDVFDLPDNHPVRRHARLFTSI
Q20638/74-216       EKELLRRTWSDEFD......NLYELGSAIYCYIFD.......HNPNCKQLFP.F.ISKYQGDEWKESKEFRSQALKFVQT
Q19601/54-189       ERILLEQSWRKTRKT.....GADHIGSKIFFMVLT.......AQPDIKAIFG.L.EK.IPTGRLKYDPRFRQHALVYTKT
Q18311/32-175       TKKLVIQEWPRVLA......QCPELFTEIWHKSAT.......RSTSIKLAFG.I.AE.N..ESPMQNAAFLGLSSTIQAF
Q18209/233-375      QIHLVRALWRQVYTT....KGPTVIGASIYHRLCFKNVMVKEQMKQVE.LPPKF.QN........RDNFIKAHCKAVAEL
Q9ZAX1/42-162       DALRVLQNAFKL........DDPELVRRFYAHWFA.......LDASVRDLFP.P..........DMGAQRAAFGQALHWV
LGB1_MEDSA/7-142    QEALVNSSWEAFKQ......NLPRYSVFFYTVVLE.......KAPAAKGLFS.F...LKNSAEVQDSPQLQAHAEKVFGL
LGB1_LUPLU/7-145    QVALVKSSFEEFNA......NIPKNTHRFFTLVIE.......IAPGAKDLFS.F.LK.GSSEVPQNNPDLQAHAGKVFKL
HBP2_CASGL/13-152   QEALVVKSWSAMKP......NAGELGLKFFLKIFE.......IAPSAQKLFS.FLKD..SNVPLERNPKLKSHAMSVFLM
HBP1_CASGL/6-145    QEALLKQSWEVLKQ......NIPAHSLRLFALIIE.......AAPESKYVFS.FLKD..SNEIPENNPKLKAHAAVIFKT
GLP1_GLYDI/7-141    QVAALKASWPEVSAG....DGGAQLGLEMFTKYFH.......ENPQMMFIFG.YSGR.T..EALKHSSKLQHHGKVIIDQ
GLB1_GLYDI/6-141    QRQVIAATWKDIAGA....DNGAGVGKDCLIKFLS.......AHPQMAAVFG.F.SG.......ASDPGVAALGAKVLAQ
GLB_TUBTU/6-139     QRFKVKHQWAEAFGT...SHHRLDFGLKLWNSIFR.......DAPEIRGLFKRVDGD.N.....AYSAEFEAHAERVLGG
GLB3_LAMSP/7-141    QRLKVKRQWAEAYGS...GNDREEFGHFIWTHVFK.......DAPSARDLFKRVRGDNI......HTPAFRAHATRVLGG
GLB2_TYLHE/9-142    QRLKVKQQWAKAYGV...GHERVELGIALWKSMFA.......QDNDARDLFKRVHGEDV......HSPAFEAHMARVFNG
GLB2_LUMTE/8-141    EGLKVKSEWGRAYGS...GHDREAFSQAIWRATFA.......QVPESRSLFKRVHGDDT......SHPAFIAHAERVLGG
GLB1_TYLHE/7-136    QRIKVKQQWAQVYSV...GESRTDFAIDVFNNFFR.......TNPDRS.LFNRVNGDNV......YSPEFKAHMVRVFAG
GLB4_TYLHE/8-143    DRREVQALWRSIWSAE.DTGRRTLIGRLLFEELFE.......IDGATKGLFKRVNVDDT......HSPEEFAHVLRVVNG
GLB3_TYLHE/8-143    DRHEVLDNWKGIWSAE.FTGRRVAIGQAIFQELFA.......LDPNAKGVFGRVNVD.K.....PSEADWKAHVIRVING
GLB4_LUMTE/11-146   DRREIRHIWDDVWSSS.FTDRRVAIVRAVFDDLFK.......HYPTSKALFERVKIDEP......ESGEFKSHLVRVANG
GLB_CERRH/6-146     SKSALASSWKTLAKD...AATIQNNGATLFSLLFK.......QFPDTRNYFTHF.GNMS.DAEMKTTGVGKAHSMAVFAG
GLB_BUSCA/6-139     QKTALKESWKVLGAD...GPTMMKNGSLLFGLLFK.......TYPDTKKHFKHF..DDATFAAMDTTGVGKAHGVAVFSG
GLBB_ANATR/16-151   QKDLIRLSWGVLSV......DMEGTGLMLMANLFK.......TSSAARTKFARL.GDVS...AGKDNSKLRGHSITLMYA
GLB_APLJU/6-139     DAGLLAQSWAPVFA......NSDANGASFLVALFT.......QFPESANFFNDF.KG.KSLADIQASPKLRDVSSRIFAR
GLBX_CHITH/11-145   EVEQVQATWKAVSH..........DEVEILYTVFK.......AHPDIMAKFPKF.AG.KDLEAIKDTADFAVHASRIIGF
GLBC_CHITH/23-157   EASLVQSSWKAVSH..........NEVDILAAVFA.......AYPDIQAKFPQF.AG.KDLASIKDTGAFATHATRIVSF
GLB6_CHITH/22-156   QADLVKKTWSTVKF..........NEVDILYAVFK.......AYPDIMAKFPQF.AG.KDLDSIKDSAAFATHATRIVSF
GLB8_CHITH/9-143    QLALFKSSWNTVKH..........NEVDILYAVFK.......ANPDIQAKFPQF.AG.KDLDSIKDSADFAVHSGRIVGF
GLB2_CHITH/22-156   EASLVRGSWAQVKH..........SEVDILYYIFK.......ANPDIMAKFPQF.AG.KDLETLKGTGQFATHAGRIVGF
GLB3_CHITH/20-147   QISTVQASFDKVKG..........DPVGILYAVFK.......ADPSIMAKFTQF.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

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

majority vote:

A C A 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

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

[ab]	: union	{a,b}
ab	: concatenation	{ab}
ε	: empty string	
a*	: Kleene star	{ε,a,aa,aaa, …}

[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

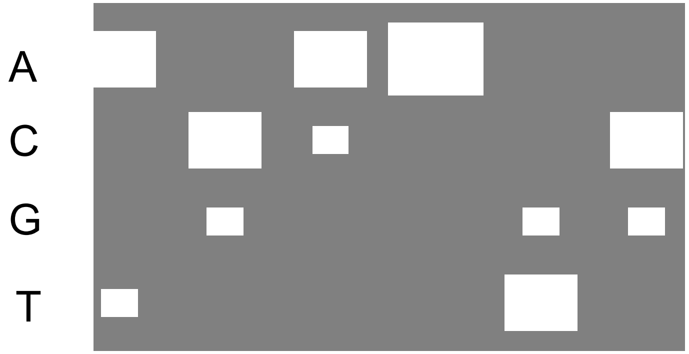
$$\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline A & 4 & 0 & 4 & 5 & 0 & 0 \\ C & 0 & 4 & 1 & 0 & 0 & 4 \\ G & 0 & 1 & 0 & 0 & 1 & 1 \\ T & 1 & 0 & 0 & 0 & 4 & 0 \end{array}$$

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

$$W = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline A & 0.8 & 0.0 & 0.8 & 1.0 & 0.0 & 0.0 \\ C & 0.0 & 0.8 & 0.2 & 0.0 & 0.0 & 0.8 \\ G & 0.0 & 0.2 & 0.0 & 0.0 & 0.2 & 0.2 \\ T & 0.2 & 0.0 & 0.0 & 0.0 & 0.8 & 0.0 \end{array}$$



aka position specific score matrix

Weight matrices (2)

Sequence: $x = x_1 x_2 \ldots x_N$

independence

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

$$P(\mathtt{ACAATC} \mid W) = P_1(\mathtt{A}) P_2(\mathtt{C}) P_3(\mathtt{A}) P_4(\mathtt{A}) P_5(\mathtt{T}) P_6(\mathtt{C})$$

$$= 0.8 \times 0.8 \times 0.8 \times 1 \times 0.8 \times 0.8 = 0.33$$

$$\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline A & \boxed{0.8} & 0.0 & \boxed{0.8} & \boxed{1.0} & 0.0 & 0.0 \\ C & 0.0 & \boxed{0.8} & 0.2 & 0.0 & 0.0 & \boxed{0.8} \\ G & 0.0 & 0.2 & 0.0 & 0.0 & 0.2 & 0.2 \\ T & 0.2 & 0.0 & 0.0 & 0.0 & \boxed{0.8} & 0.0 \end{array}$$

Weight matrices (3)

Sequence: $x = x_1 x_2 \ldots x_N$

independence

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

$$P(\texttt{CCAATC} \mid W) = P_1(\texttt{C}) P_2(\texttt{C}) P_3(\texttt{A}) P_4(\texttt{A}) P_5(\texttt{T}) P_6(\texttt{C})$$

$$= 0 \times 0.8 \times 0.8 \times 1 \times 0.8 \times 0.8 = 0$$

$$\begin{array}{c|cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
A & 0.8 & 0.0 & \boxed{0.8} & \boxed{1.0} & 0.0 & 0.0 \\
C & \boxed{0.0} & \boxed{0.8} & 0.2 & 0.0 & 0.0 & \boxed{0.8} \\
G & 0.0 & 0.2 & 0.0 & 0.0 & 0.2 & 0.2 \\
T & 0.2 & 0.0 & 0.0 & 0.0 & \boxed{0.8} & 0.0
\end{array}$$

Weight matrices: pseudocounts

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

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

$$\begin{array}{c} A \\ C \\ G \\ T \end{array} \begin{pmatrix} 5 & 1 & 5 & 6 & 1 & 1 \\ 1 & 5 & 2 & 1 & 1 & 5 \\ 1 & 2 & 1 & 1 & 2 & 2 \\ 2 & 1 & 1 & 1 & 5 & 1 \end{pmatrix}$$

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

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

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

Bayes' rule: odds

class A: sites   class B: non-sites

$$x \text{ is assigned to class } A \quad \Leftrightarrow \quad \frac{P(x \mid \text{class } A)P(A)}{P(x)} > \frac{P(x \mid \text{class } B)P(B)}{P(x)}$$

$$\Leftrightarrow \quad \frac{P(x \mid \text{class } A)}{P(x \mid \text{class } B)} > \frac{P(B)}{P(A)}$$

priors

equal priors: $$\underbrace{\frac{P(x \mid \text{class } A)}{P(x \mid \text{class } B)}}_{\text{odds}} > 1 \quad \Leftrightarrow \quad \underbrace{\log\left(\frac{P(x \mid \text{class } A)}{P(x \mid \text{class } B)}\right)}_{\text{log-odds}} > 0$$

odds   log-odds

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

Weight matrices: odds

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

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

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

log-odds

Weight matrices: log-odds

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

$$\begin{array}{c} A \\ C \\ G \\ T \end{array} \begin{pmatrix} \boxed{1.16} & -1.17 & \boxed{1.16} & \boxed{1.42} & -1.17 & -1.17 \\ -1.17 & \boxed{1.16} & -0.17 & -1.17 & -1.17 & \boxed{1.16} \\ -1.17 & -0.17 & -1.17 & -1.17 & -0.17 & -0.17 \\ -0.17 & -1.17 & -1.17 & -1.17 & \boxed{1.16} & -1.17 \end{pmatrix} \longrightarrow \log(0.56/0.25)$$

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

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

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

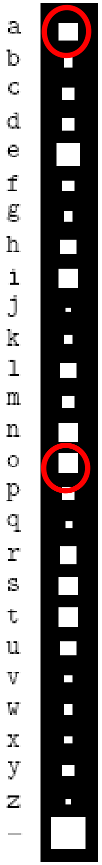
Outline

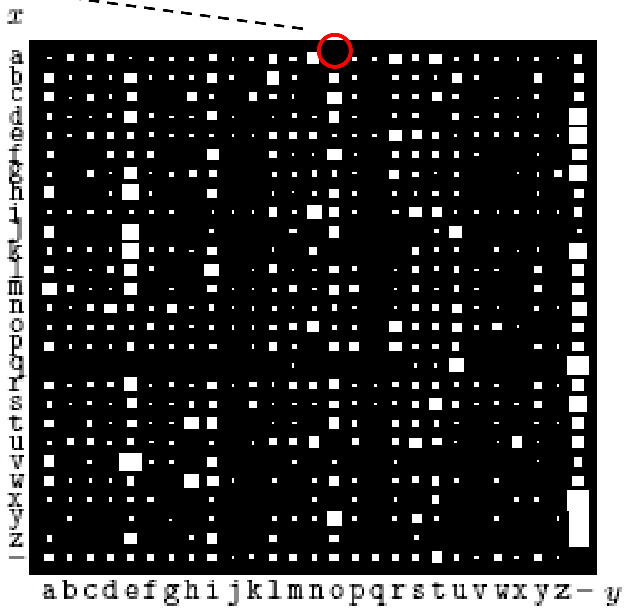
- Regular expressions & weight matrices
- Dependencies & Markov chains
- Hidden Markov models
- HMMs & EM
- Profile HMMs
- Genefinding

Dependencies: language

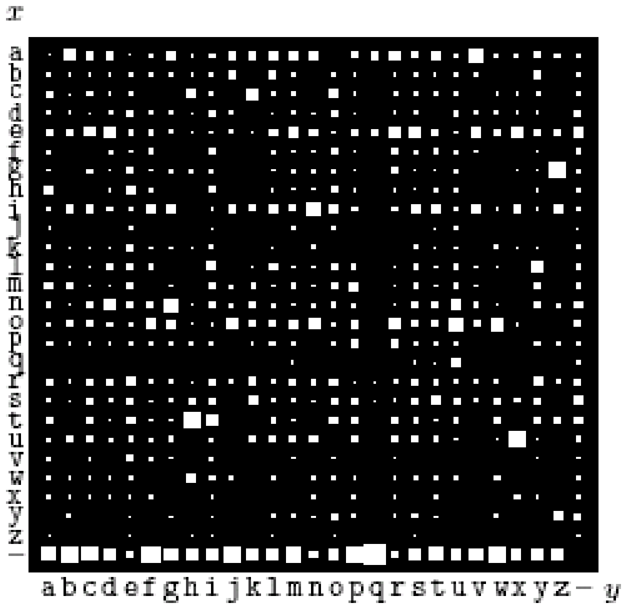
Probability (in English) of “o” given that previous letter is “a”

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





(a) $P(y\,|\,x)$



(b) $P(x\,|\,y)$

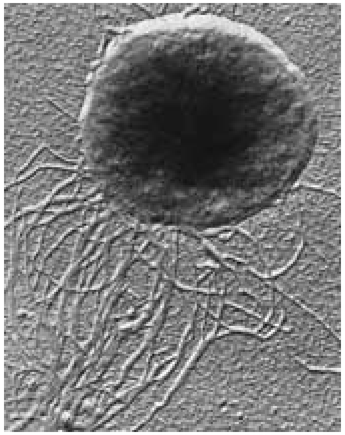
Dependencies: biology

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

$$S = \begin{array}{c|cccc|} & A & C & G & T \\ A & 1.13 & \boxed{0.73} & 1.10 & 0.94 \\ C & 1.03 & \boxed{1.37} & \boxed{0.32} & 1.11 \\ G & 1.05 & 1.12 & \boxed{1.39} & \boxed{0.71} \\ T & 0.83 & 1.05 & 1.13 & 1.14 \end{array}$$

Markov chains

Sequence: $q = q_1 q_2 \ldots q_N$

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

Only dependent on previous symbol:

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

First-order Markov chain

state: value of q_j

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

Markov chains: language

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

```
OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI
ALHENHTTPA OOBTTVA NAH BRL.
```

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

```
ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY
ACHIN D ILONASIVE TUCOOWE 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)

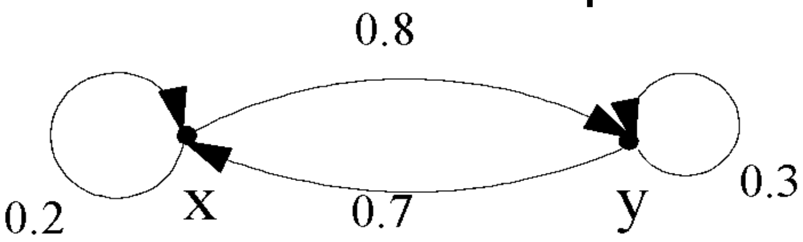
Markov chain: graphical representation

Two states: *x* and *y*

Matrix

$$A = \begin{array}{c|cc} & x & y \\ \hline x & 0.2 & 0.8 \\ y & 0.7 & 0.3 \end{array}$$

Graph: arrows for transition probability



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

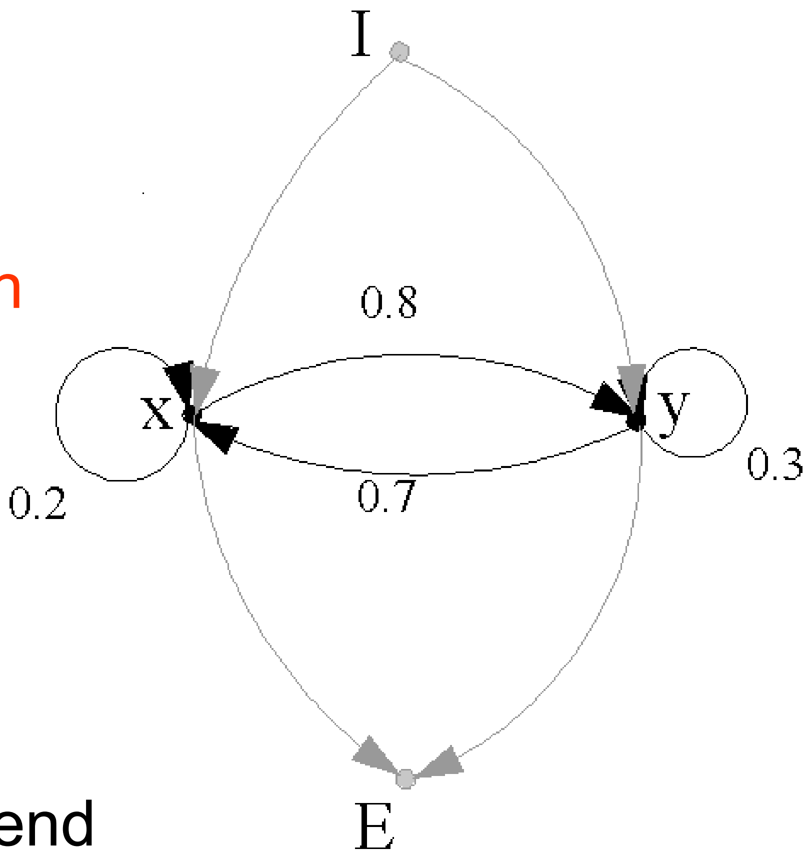
Generative model (example): *xyyxyxyyxxyxyxx*...

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

Markov chain: graphical representation (2)

Two states: *x* and *y*

Graph



Add begin state q_0

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

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

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: $x = x_1 x_2 \ldots x_N$

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

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

$$= \sum_{t=1}^{N} \log\left(\frac{a^{A}_{x_{t-1},x_t}}{a^{B}_{x_{t-1},x_t}}\right)$$

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

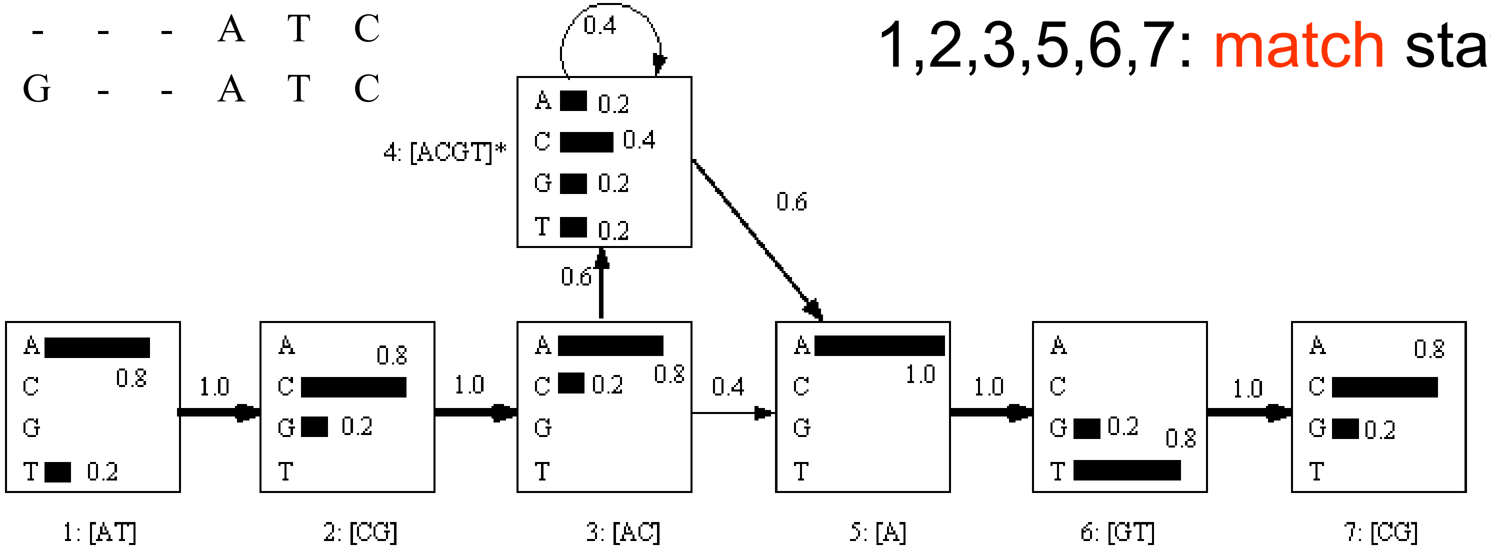
A different representation

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

boxes: states

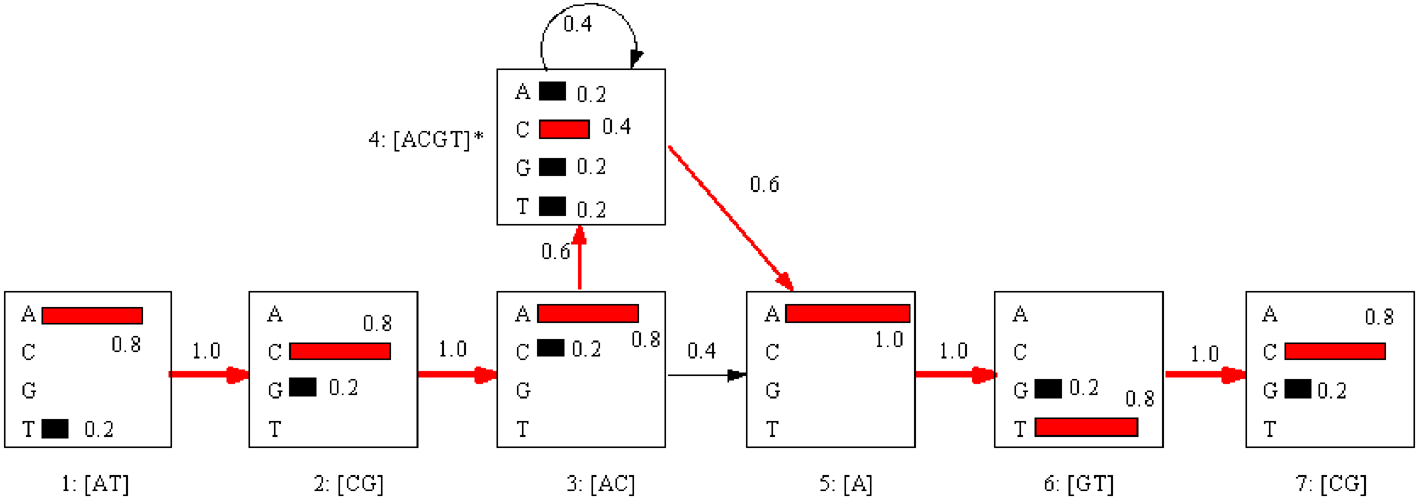
4: insert state

1,2,3,5,6,7: match state



mix of weight matrices and Markov chains

Probability of consensus sequence



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

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, …, S} 0: begin state (non-emitting)

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

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

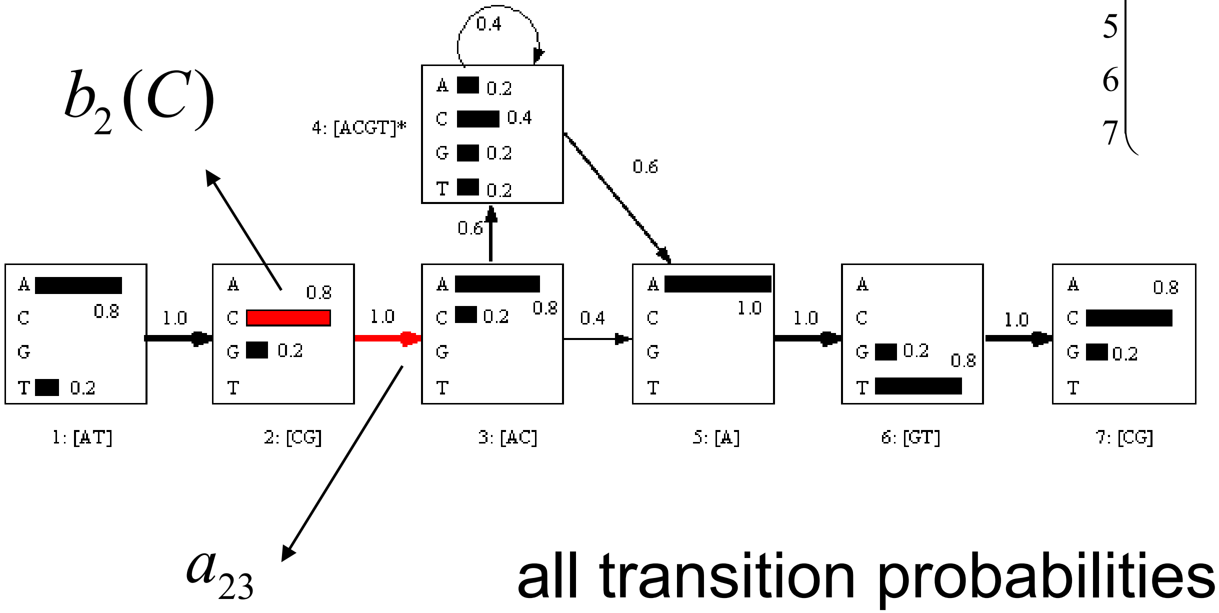
→ probability of emitting symbol *x* in state *i*

Hidden Markov models (2)

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

Q = {1,2, …, 7}

$$A = \begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & & 1.0 & & & & & \\ 2 & & & 1.0 & & & & \\ 3 & & & & 0.6 & 0.4 & & \\ 4 & & & & 0.4 & 0.6 & & \\ 5 & & & & & & 1.0 & \\ 6 & & & & & & & 1.0 \\ 7 & & & & & & & \end{array}$$



all transition probabilities = 1: weight matrix

HMM: three problems

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

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: $Q = q_0 q_1 q_2 \ldots q_N$

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

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

$$\downarrow$$

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

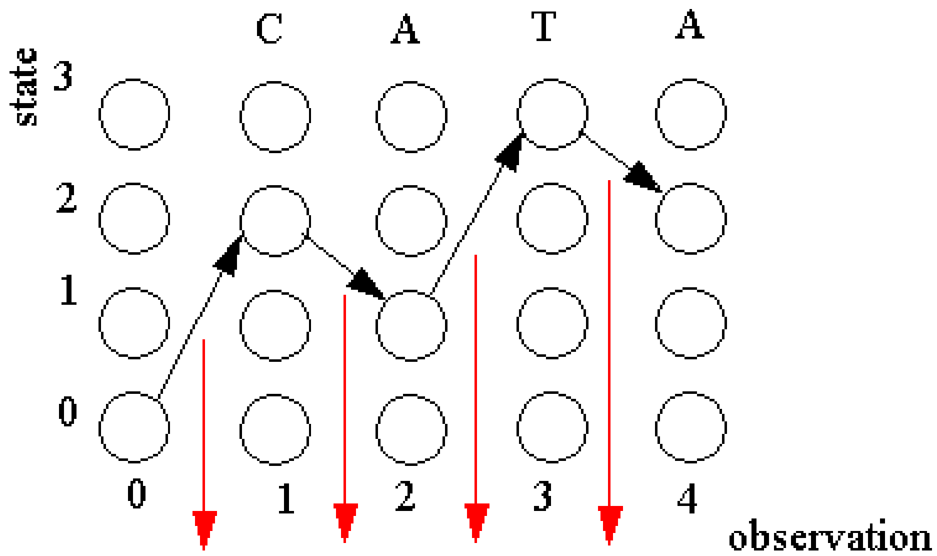
$$P(x,Q) = \prod_{t=1}^{N} P(q_t \mid q_{t-1}) \prod_{t=1}^{N} P(x_t \mid q_t) = \prod_{t=1}^{N} a_{q_{t-1},q_t} \prod_{t=1}^{N} b_{q_t}(x_t)$$

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

HMM evaluation: graphical representation on a trellis

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

state sequence = path



weights: $$\underbrace{a_{02}b_2(C)\quad a_{21}b_1(A)\quad a_{13}b_3(T)\quad a_{32}b_2(A)}_{P(\mathrm{CATA}, 0-2-1-3-2)}$$

HMM evaluation: forward algorithm

State sequence unknown: $P(x) = \sum_{Q} P(x, Q)$

Sum over all paths through trellis: ~ S^{N} state sequences!

forward variable

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

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

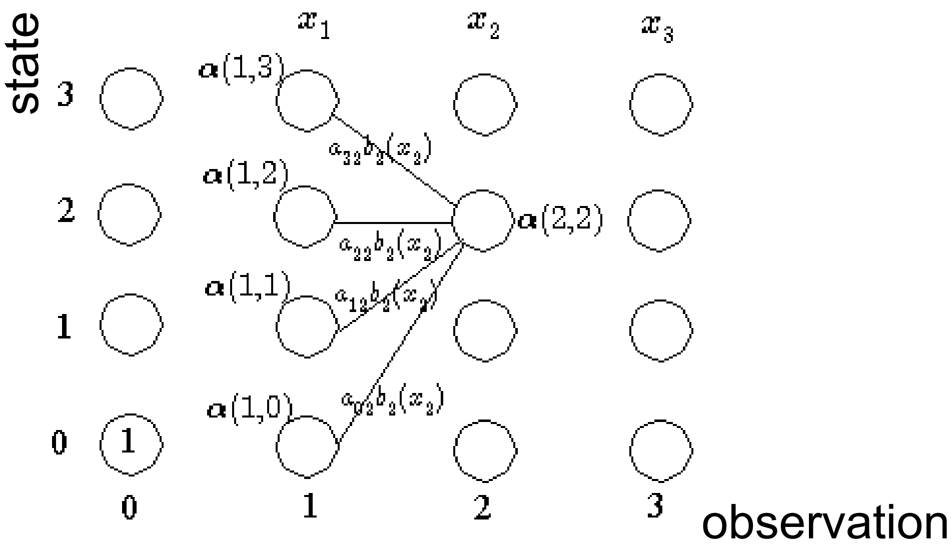
HMM evaluation: forward algorithm (2)

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

initialization: $\alpha(0,0) = 1\ ,\ \alpha(0,j) = 0 \qquad 1 \le j$

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

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



Complexity: *S*×*N*

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

Forward algorithm: proof

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

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

$$\alpha(t,i) = P(x_{1..t}, q_t = i) = \sum_j P(x_{1..t}, q_{t-1} = j, q_t = i)$$

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

.

Observed symbol and the state depend only on previous state:

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

$$= \sum_j \alpha(t-1, j) P(q_t = i \mid q_{t-1} = j) P(x_t \mid q_t = i)$$

$$= \sum_j \alpha(t-1, j) a_{ji} b_i(x_t)$$

recursion

HMM: three problems

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

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

Viterbi variable

$$V(x) = \max_Q P(Q \mid x) = \max_Q \frac{P(x,Q)}{P(x)} = \max_Q P(x,Q)$$

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

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

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

HMM decoding: Viterbi algorithm (2)

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

initialization : $v(0,0) = 1 \;,\; v(0,j) = 0$ $\quad 1 \le j$

recursion :

$v(t,i) = \max_j \; [v(t-1,j)a_{ji}]b_i(x_t)$ $\quad 1 \le t \le N, 0 \le i,j \le S$

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

end :

$V(x) = \max_i \; [v(N,i)]$

$q_N^* = \arg\max_i [v(N,i)]$

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

Dishonest casino: Viterbi

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

transition probabilities

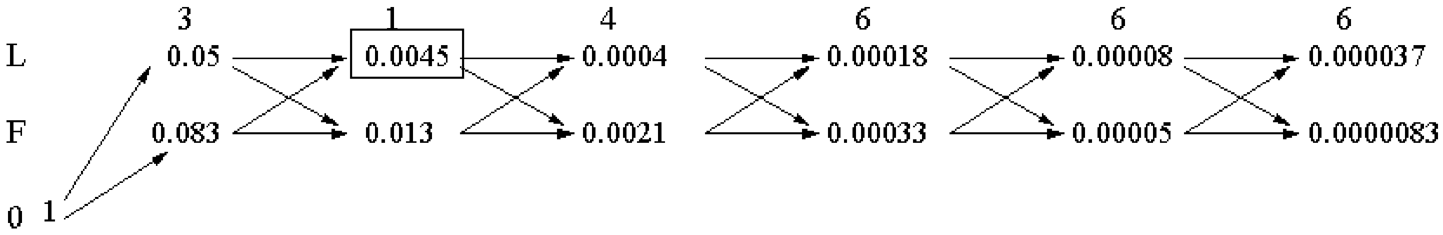
$$A = \begin{array}{c} 0 \\ F \\ L \end{array} \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0 & 0.95 & 0.05 \\ 0 & 0.1 & 0.9 \end{pmatrix}$$

$$\begin{array}{lcccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ F: & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ L: & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \boxed{\frac{1}{2}} \end{array}$$

emission probabilities

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

Viterbi



$v(2,L) = \max[v(1,F)a_{FL}b_L(1)\ ,\ \ v(1,L)a_{LL}b_L(1)]$

$= \max[(0.083\times0.05\times0.1\ ,\ 0.05\times0.9\times0.1] = 0.0045$

Dishonest casino: Viterbi (2)

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

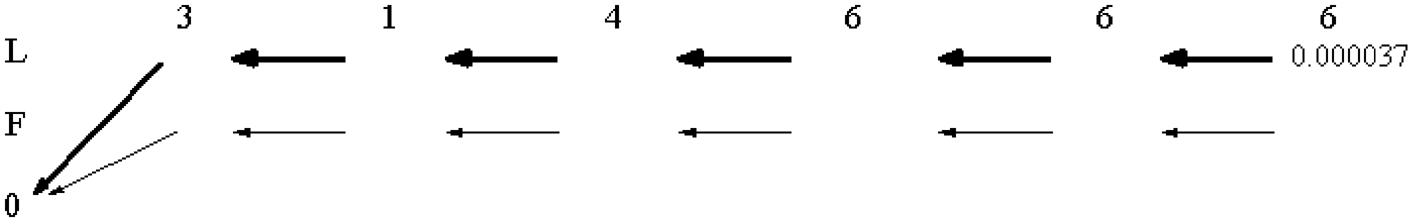
transition probabilities

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

$$\begin{array}{lcccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ F: & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ L: & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \boxed{\frac{1}{2}} \end{array}$$

emission probabilities

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

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, ...,x^n}

Likelihood:

$$P(x^1,\ldots,x^n \mid \theta) = \prod_{i=1}^{n} P(x^i \mid \theta)$$

state sequence

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

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

same solution since log is monotonic

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

HMM estimation: EM

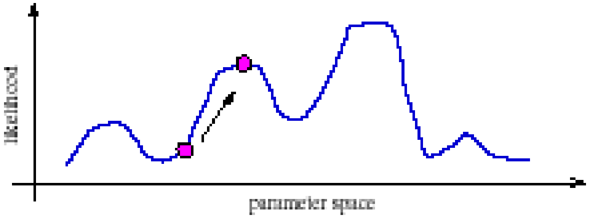
1. If we know the state sequence, parameter estimation is easy: just counting as in Markov chains
2. Can estimate state path using the forward-backward algorithm (not shown)
3. EM: estimate (probability of) states, then estimate parameters, re-estimate the states etc.

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 …

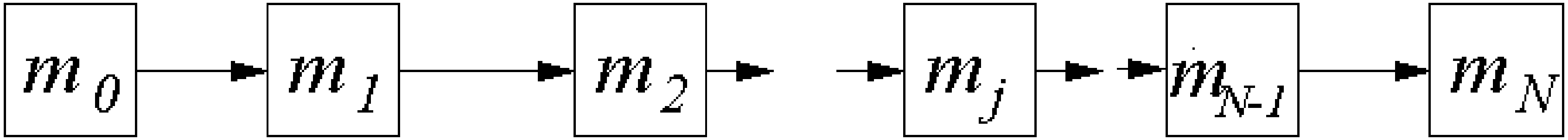
Outline

- 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

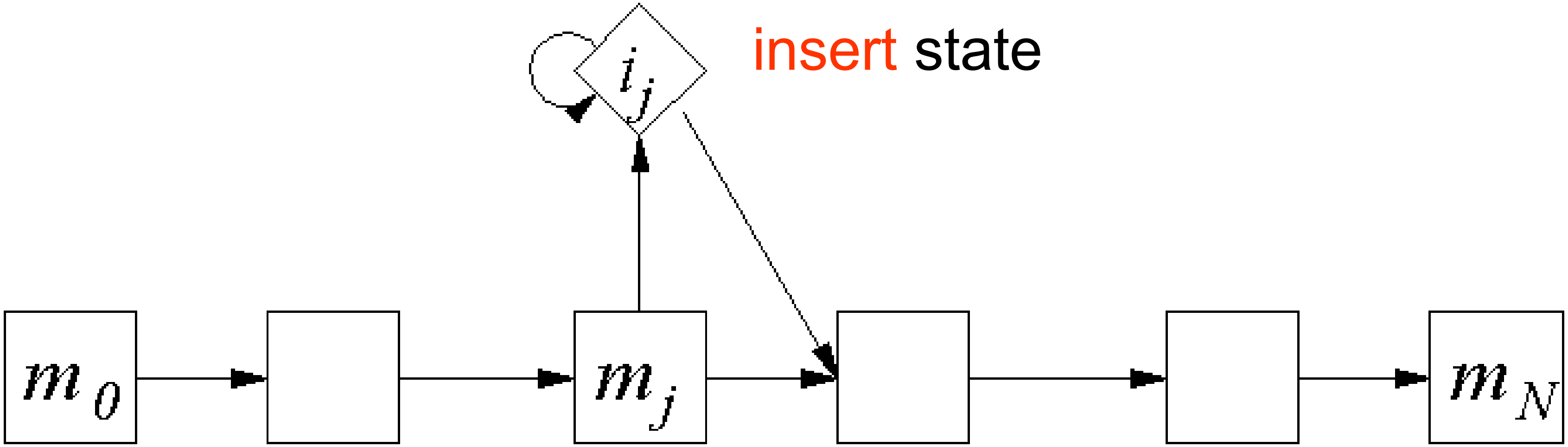
We saw that a weight matrix can be represented as a very simple HMM



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

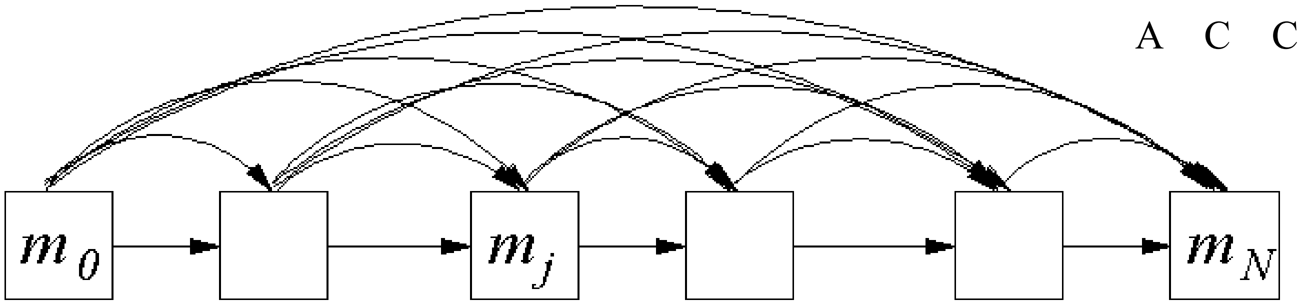


insert state

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

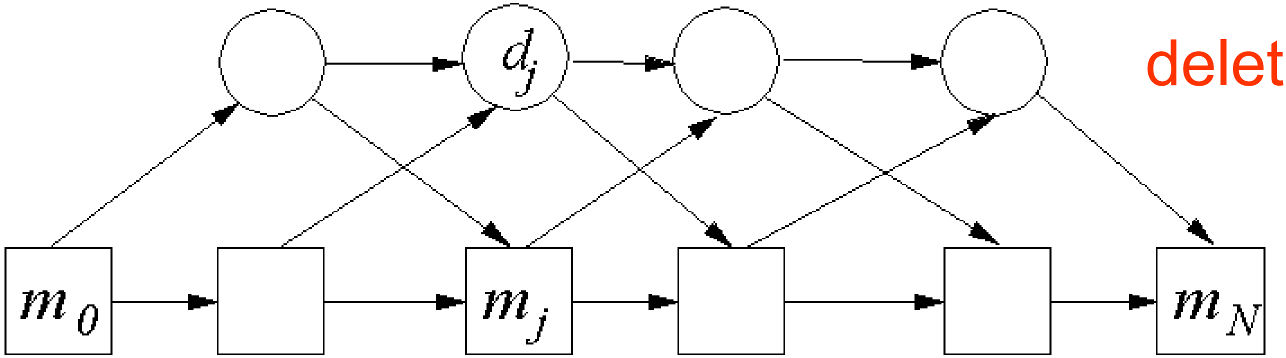
Profile HMMs: deletions

```
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

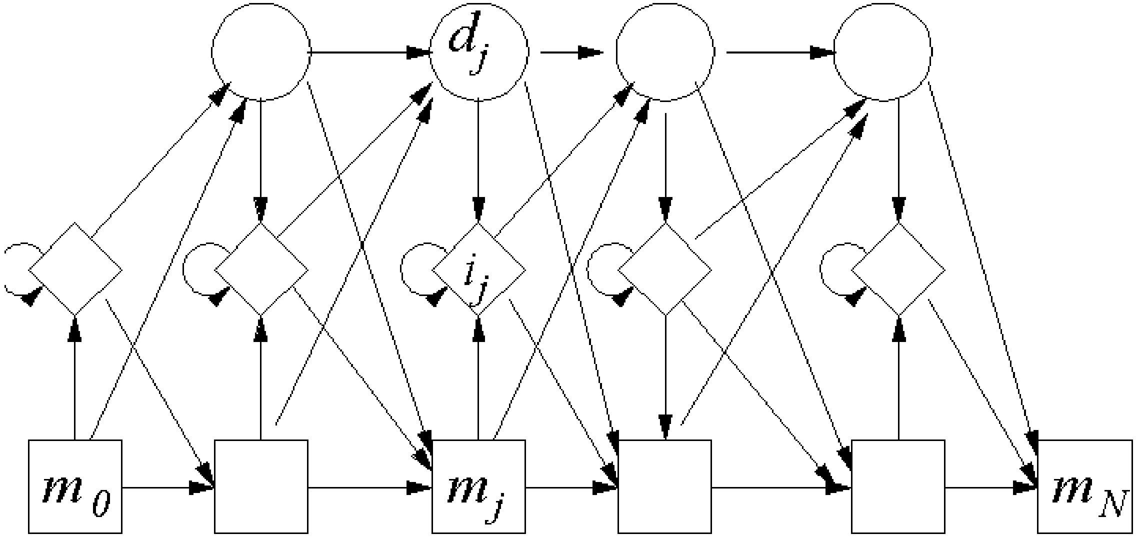


delete states

Profile HMMs (2)

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

Put everything together:



Applications:

http://pfam.xfam.org/

- searching for remote homologs (Forward)

- align a protein to a protein family (Viterbi)

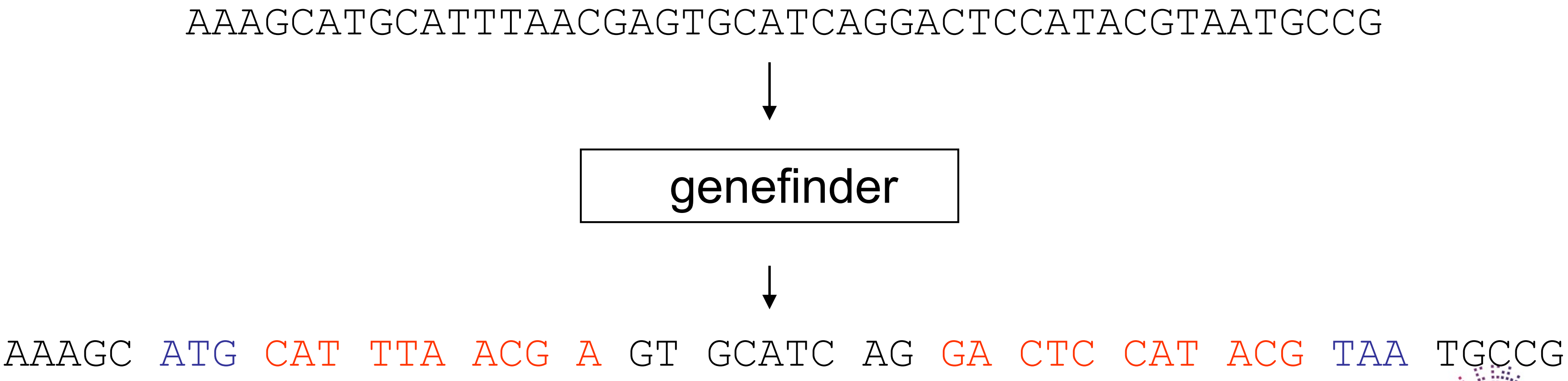
Outline

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

Genefinding

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

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



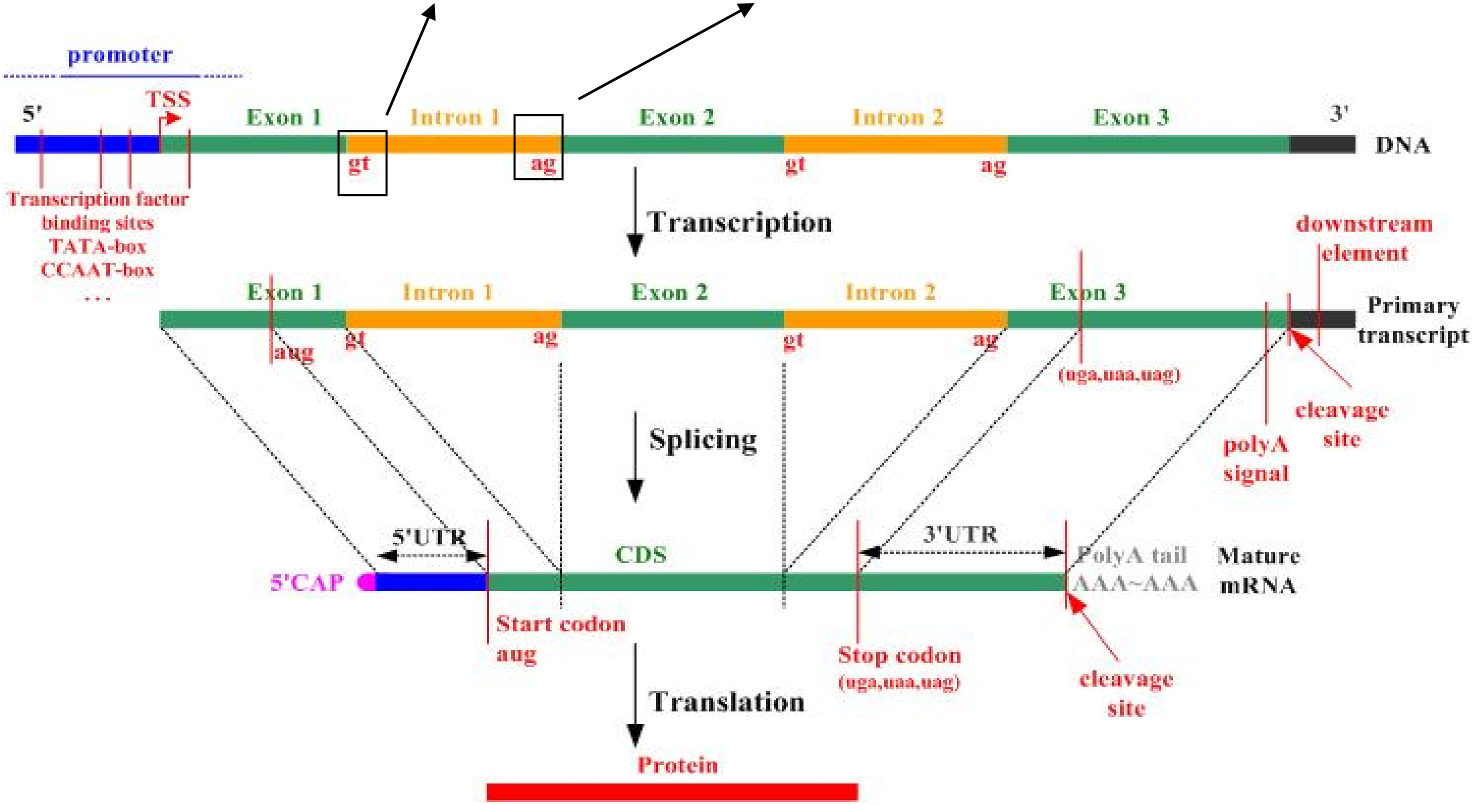
AAAGCATGCATTTAACGAGTGCATCAGGACTCCATACGTAATGCCG

genefinder

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

Genefinding: eukaryotes

splice sites: donor acceptor



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

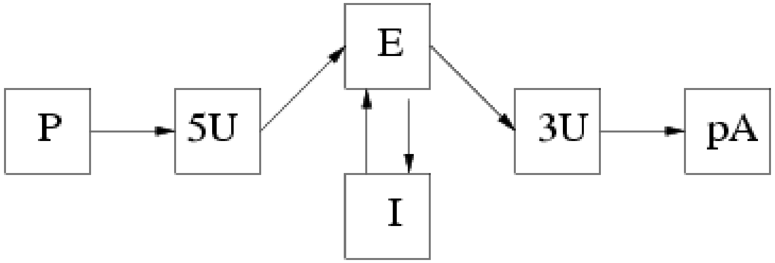
Genes have a certain structure/grammar

… exon – intron – exon – intron – exon …

Regular expression of gene structure:

promoter 5’UTR exon (intron exon)* 3’UTR polyA

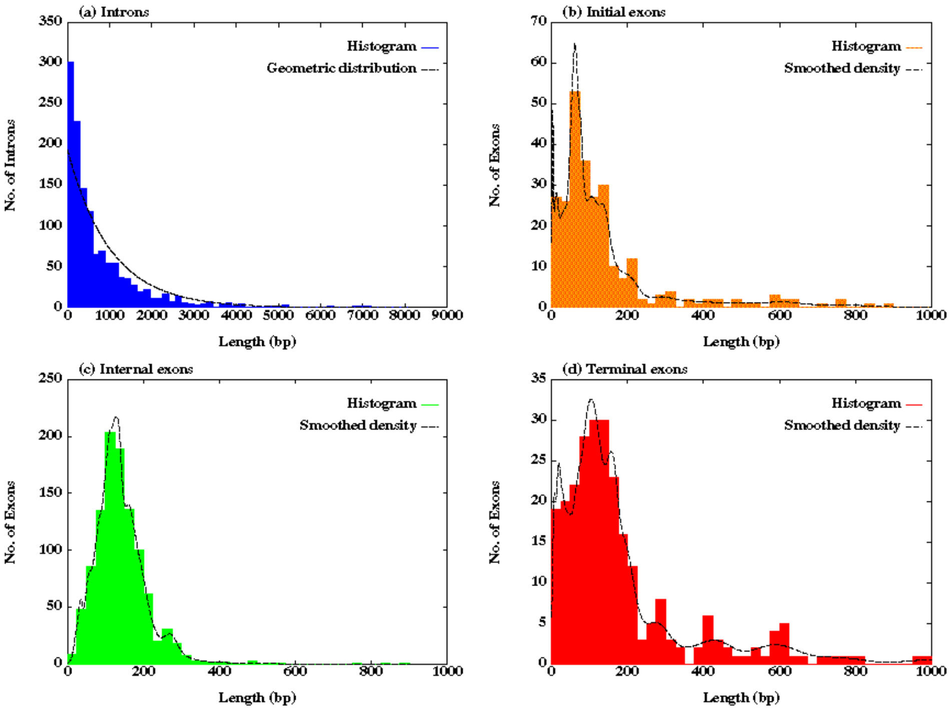
HMM →



Genefinding = annotation with states: Viterbi

Length distributions

Length distributions of human introns and initial, internal and terminal exons



Standard HMM: length ~ geometric distribution

Generalized HMM: states emit sequences + length

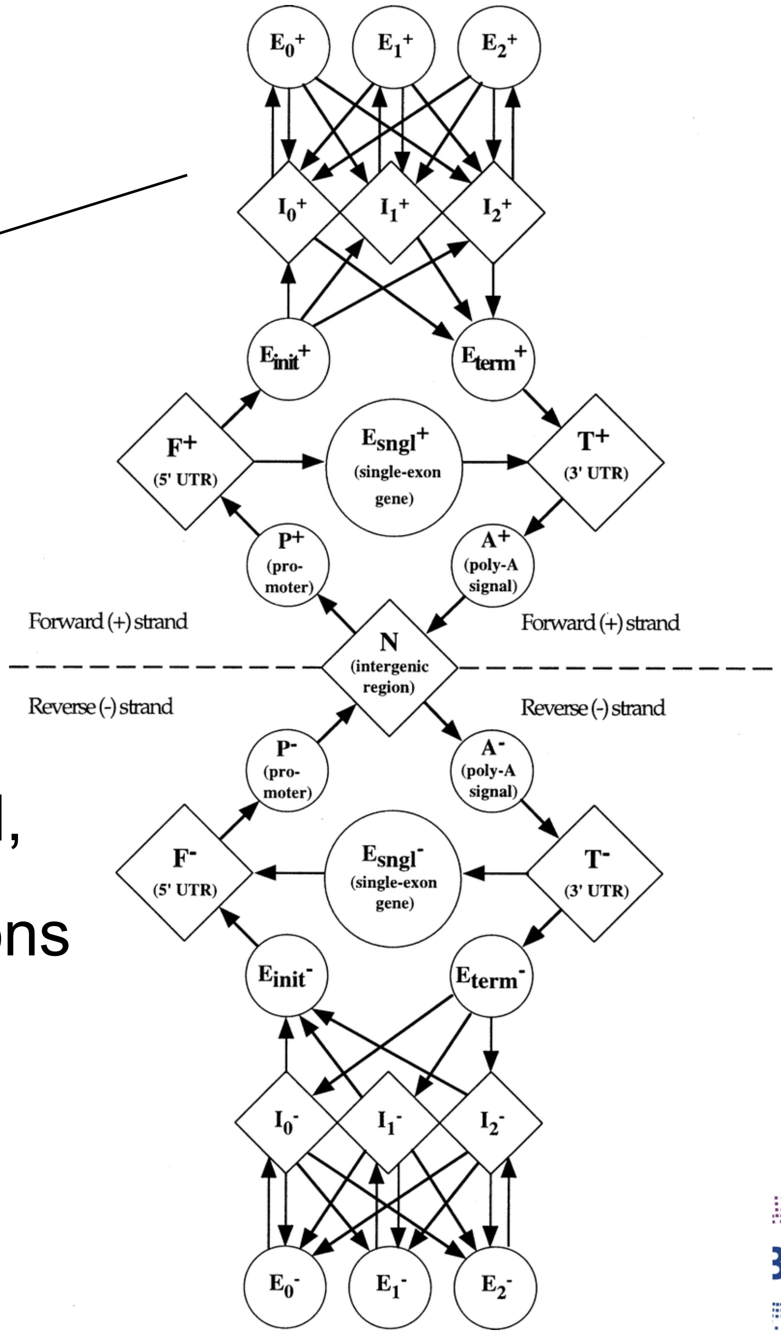
Genefinding: GenScan

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