Clustering Gene Expression Data

(Slides thanks to Dr. Mark Craven)

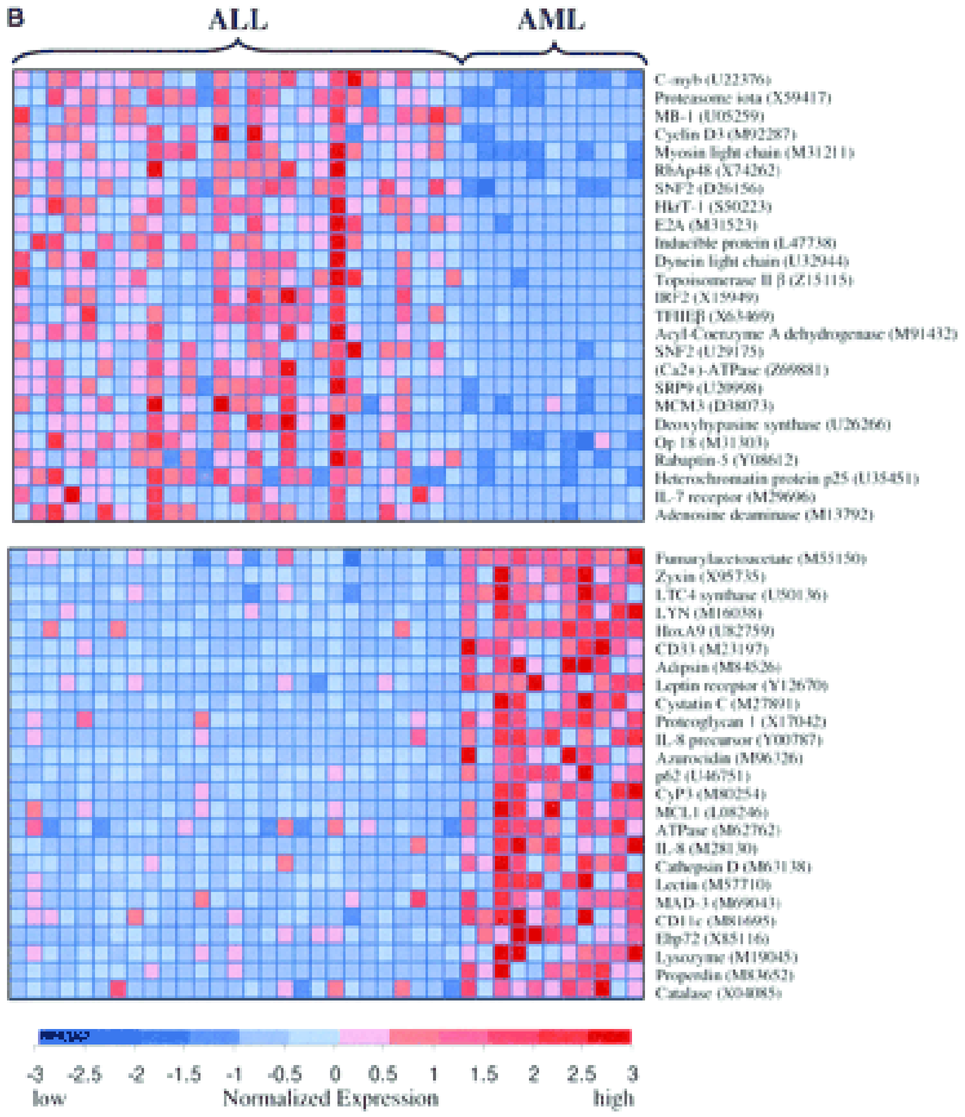
Gene Expression Profiles

- we'll assume we have a 2D matrix of gene expression measurements
  - rows represent genes
  - columns represent different experiments, time points, individuals etc. (what we can measured using one* microarray)
- we'll refer to individual rows or columns as *profiles*
  - a row is a profile for a gene

* Depending on the number of genes being considered, we might actually use several arrays per experiment, time point, individual.

Expression Profile Example

- rows represent genes
- columns represent people with leukemia



Task Definition: Clustering Gene Expression Profiles

- given: expression profiles for a set of genes or experiments/individuals/time points (whatever columns represent)
- do: organize profiles into clusters such that
  - instances in the same cluster are highly similar to each other
  - instances from different clusters have low similarity to each other

Motivation for Clustering

- *exploratory data analysis*
  - understanding general characteristics of data
  - visualizing data
- generalization
  - infer something about an instance (e.g. a gene) based on how it relates to other instances
- everyone else is doing it

The Clustering Landscape

- there are many different clustering algorithms
- they differ along several dimensions
  - hierarchical vs. partitional (flat)
  - hard (no uncertainty about which instances belong to a cluster) vs. soft clusters
  - disjunctive (an instance can belong to multiple clusters) vs. non-disjunctive
  - deterministic (same clusters produced every time for a given data set) vs. stochastic
  - distance (similarity) measure used

Distance/Similarity Measures

- many clustering methods employ a distance (similarity) measure to assess the distance between
  - a pair of instances
  - a cluster and an instance
  - a pair of clusters
- given a distance value, it is straightforward to convert it into a similarity value

$$\mathrm{sim}(x, y) = \frac{1}{1 + \mathrm{dist}(x, y)}$$

- not necessarily straightforward to go the other way
- we’ll describe our algorithms in terms of distances

Distance Metrics

- properties of metrics

$$\text{dist}(x_i, x_j) \geq 0$$

$$\text{dist}(x_i, x_i) = 0$$

$$\text{dist}(x_i, x_j) = \text{dist}(x_j, x_i)$$

$$\text{dist}(x_i, x_j) \leq \text{dist}(x_i, x_k) + \text{dist}(x_k, x_j)$$

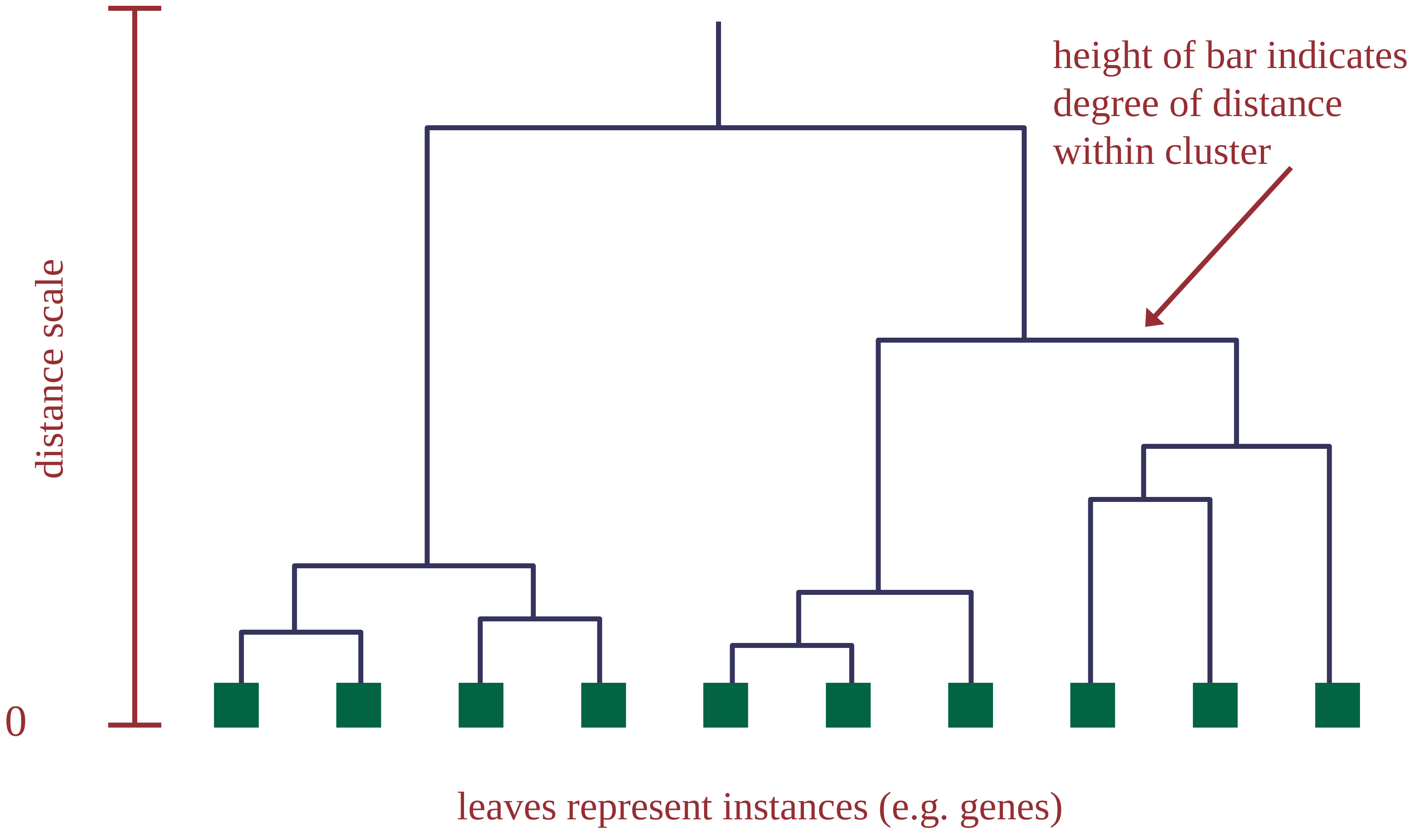
- some distance metrics

Manhattan $$\text{dist}(x_i, x_j) = \sum_e \left| x_{i,e} - x_{j,e} \right|$$

Euclidean $$\text{dist}(x_i, x_j) = \sqrt{\sum_e \left(x_{i,e} - x_{j,e} \right)^2}$$

e ranges over the individual measurements for x_i and x_j

Hierarchical Clustering: A Dendogram


height of bar indicates degree of distance within cluster
distance scale
0
leaves represent instances (e.g. genes)

Hierarchical Clustering

- can do top-down (divisive) or bottom-up (agglomerative)
- in either case, we maintain a matrix of distance (or similarity) scores for all pairs of
  - instances
  - clusters (formed so far)
  - instances and clusters

Distance Between Two Clusters

- the distance between two clusters can be determined in several ways
  - *single link*: distance of two most similar instances

$$\text{dist}(c_u, c_v) = \min\{\text{dist}(a, b) \mid a \in c_u, b \in c_v\}$$

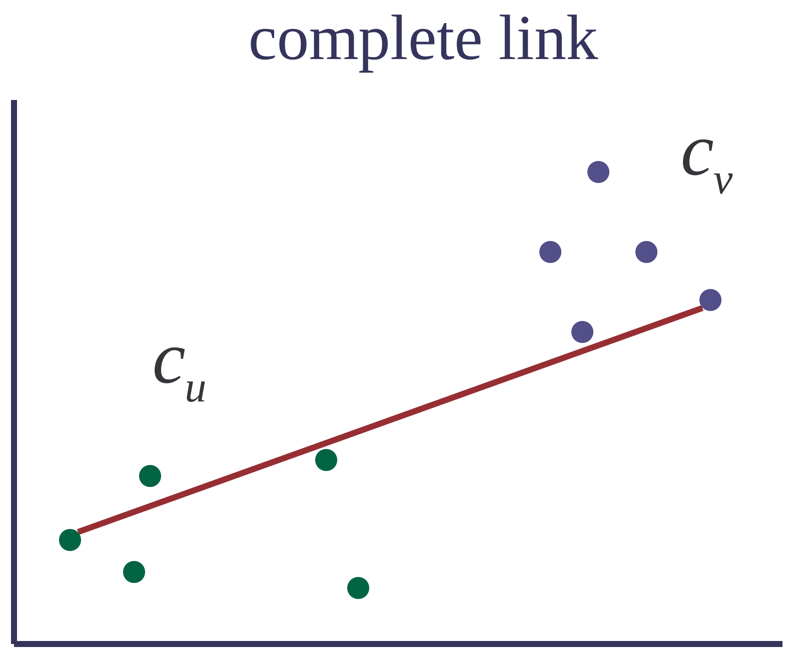
  - *complete link*: distance of two least similar instances

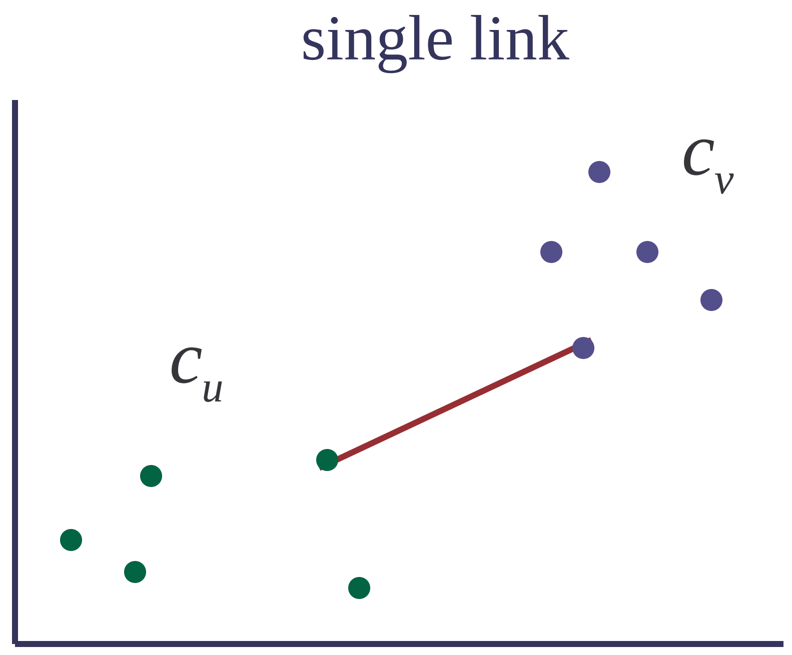
$$\text{dist}(c_u, c_v) = \max\{\text{dist}(a, b) \mid a \in c_u, b \in c_v\}$$

  - *average link*: average distance between instances

$$\text{dist}(c_u, c_v) = \text{avg}\{\text{dist}(a, b) \mid a \in c_u, b \in c_v\}$$

Complete-Link vs. Single-Link Distances





Updating Distances Efficiently

- if we just merged c_u and c_v into c_j, we can determine distance to each other cluster c_k as follows
  - single link:

$$\text{dist}(c_j, c_k) = \min\{\text{dist}(c_u, c_k), \text{dist}(c_v, c_k)\}$$

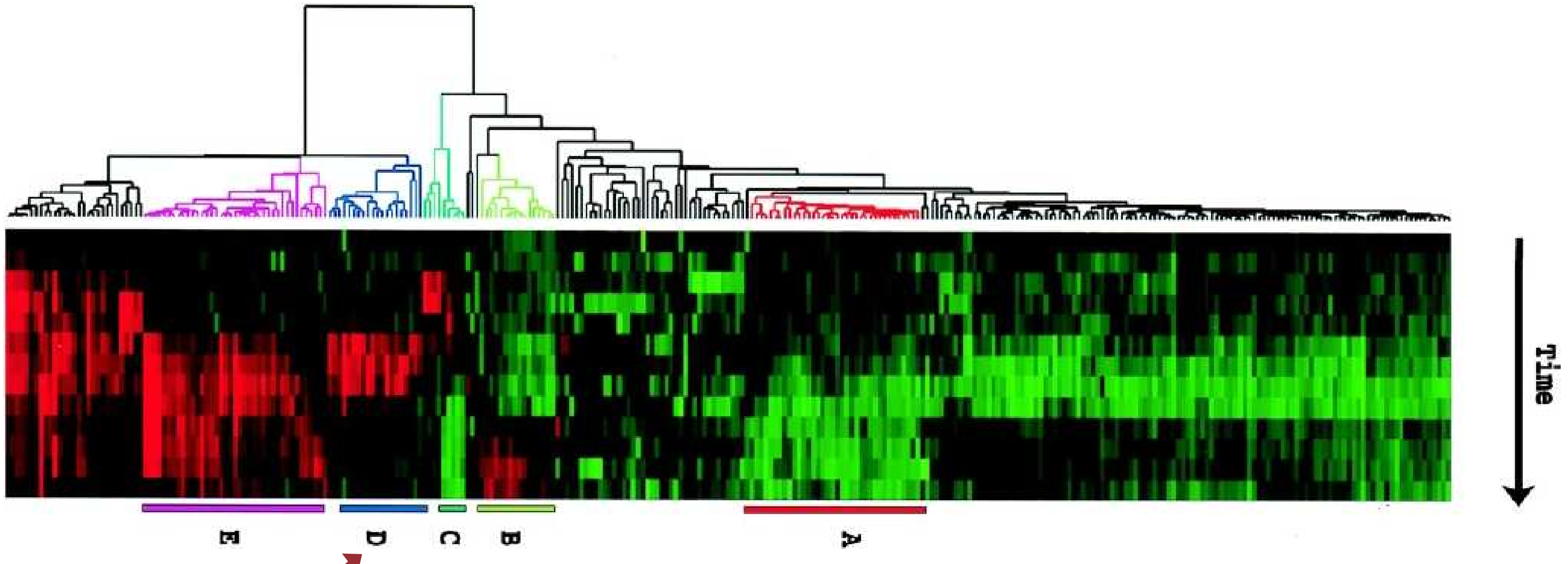
  - complete link:

$$\text{dist}(c_j, c_k) = \max\{\text{dist}(c_u, c_k), \text{dist}(c_v, c_k)\}$$

  - average link:

$$\text{dist}(c_j, c_k) = \frac{|c_u| \times \text{dist}(c_u, c_k) + |c_v| \times \text{dist}(c_v, c_k)}{|c_u| + |c_v|}$$

Dendogram for Serum Stimulation of Fibroblasts



signaling & angiogenesis

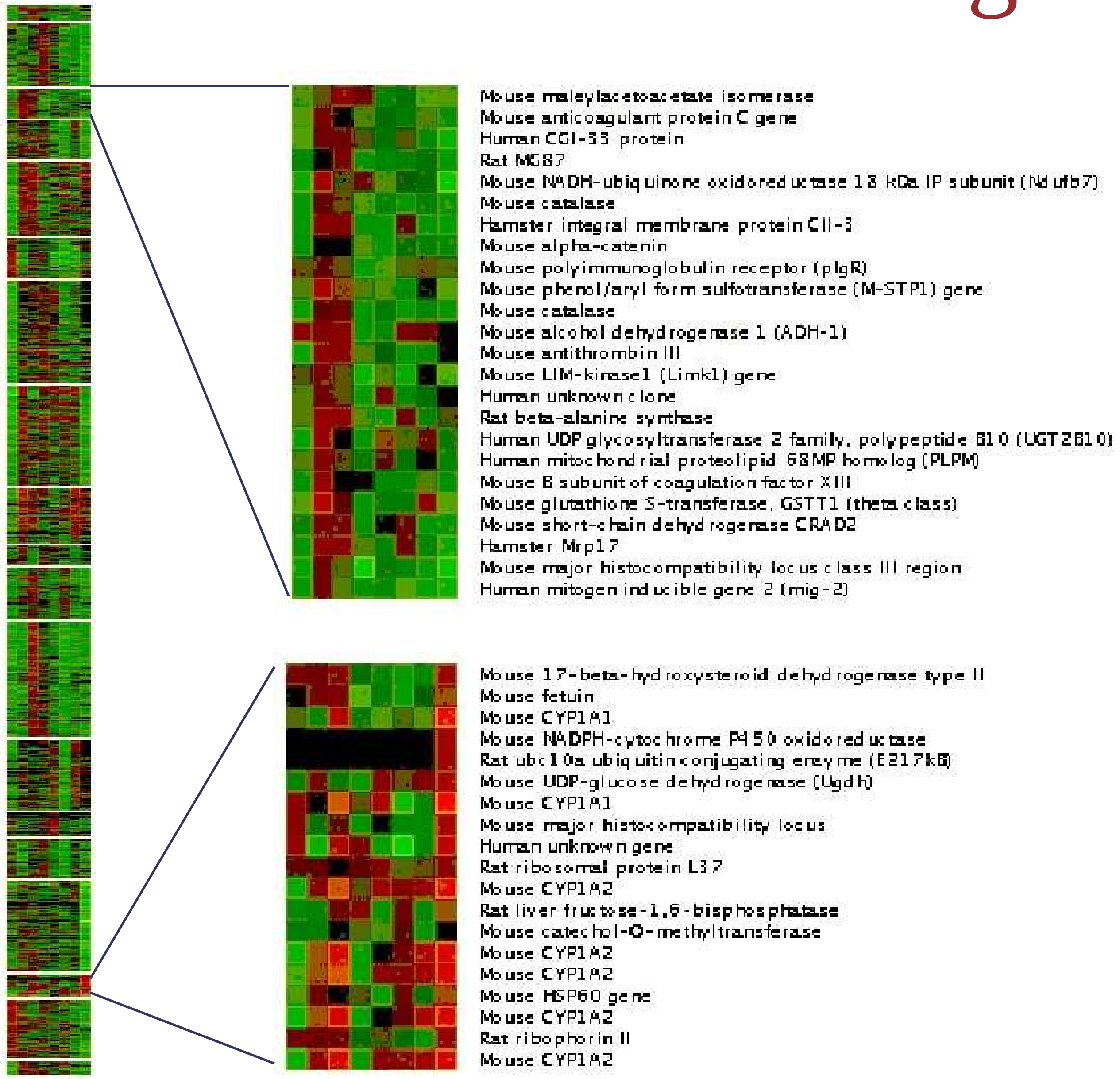
cell cyle

cholesterol biosynthesis

Partitional Clustering

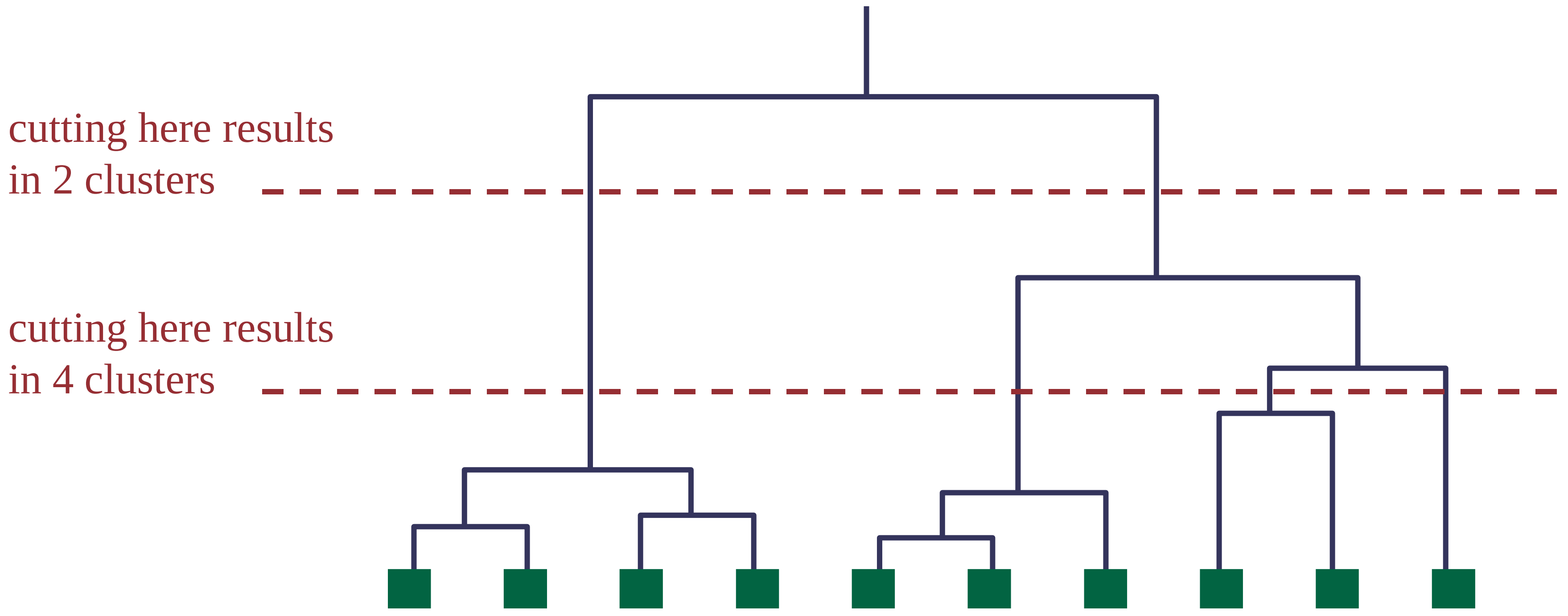
- divide instances into disjoint clusters
  - flat vs. tree structure
- key issues
  - how many clusters should there be?
  - how should clusters be represented?

Partitional Clustering Example



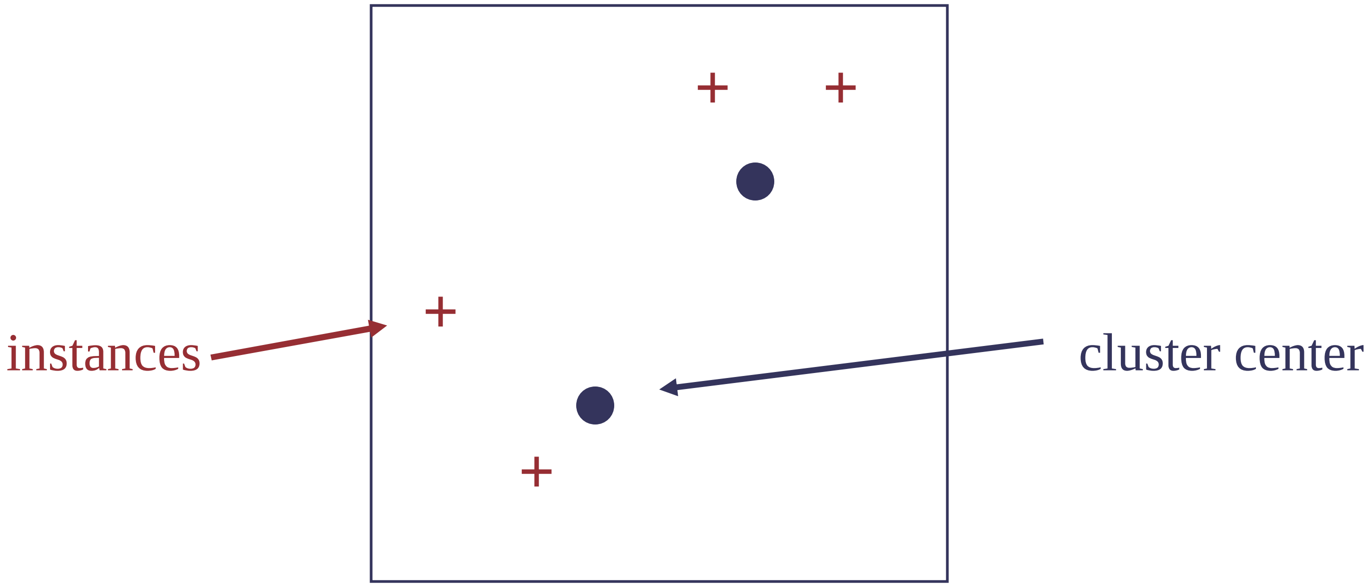
Partitional Clustering from a Hierarchical Clustering

- we can always generate a partitional clustering from a hierarchical clustering by “cutting” the tree at some level


cutting here results in 2 clusters
cutting here results in 4 clusters

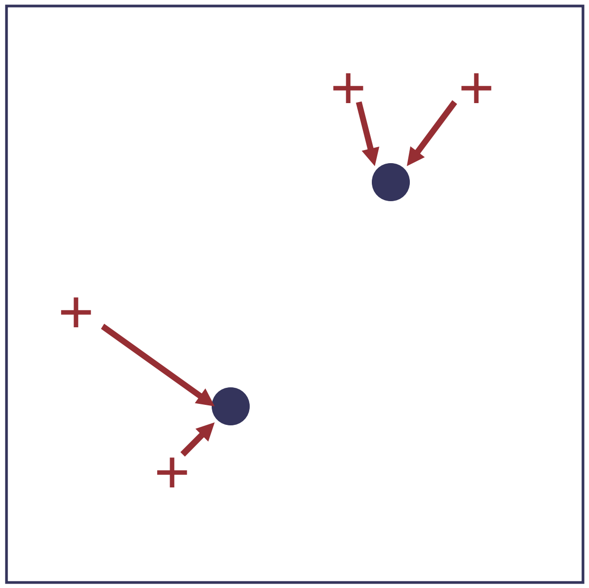
K-Means Clustering

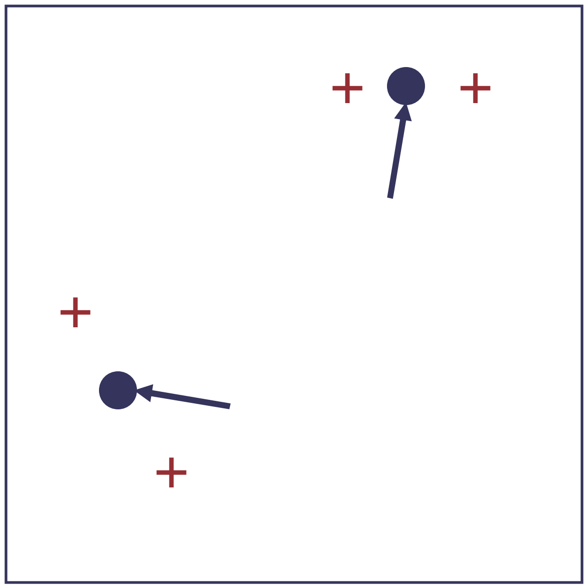
- assume our instances are represented by vectors of real values
- put *k* cluster centers in same space as instances
- each cluster is represented by a vector $\vec{f}_j$
- consider an example in which our vectors have 2 dimensions



K-Means Clustering

- each iteration involves two steps
  - assignment of instances to clusters
  - re-computation of the means


assignment


re-computation of means

K-Means Clustering: Updating the Means

- for a set of instances that have been assigned to a cluster c_j , we re-compute the mean of the cluster as follows

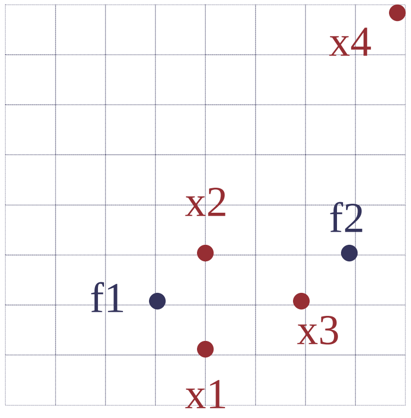
$$\mu(c_j) = \frac{\sum_{\vec{x}_i \in c_j} \vec{x}_i}{|c_j|}$$

K-Means Clustering

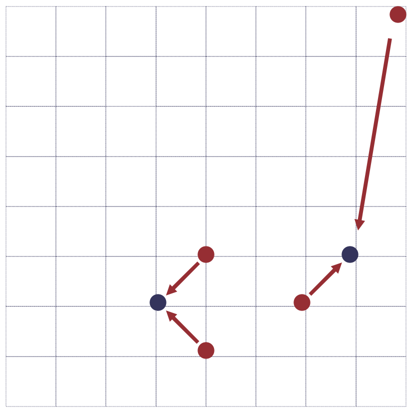
```
given: a set $X = \{\vec{x}_1 \ldots \vec{x}_n\}$ of instances
select $k$ initial cluster centers $\vec{f}_1 \ldots \vec{f}_k$
while stopping criterion not true do
    for all clusters $c_j$ do
        // determine which instances are assigned to this cluster
        $c_j = \left\{ \vec{x}_i \mid \forall f_l \ \mathrm{dist}\left(\vec{x}_i, \vec{f}_j\right) < \mathrm{dist}\left(\vec{x}_i, \vec{f}_l\right) \right\}$
    for all means $\vec{f}_j$ do
        // update the cluster center
        $\vec{f}_j = \mu(c_j)$
```

K-means Clustering Example

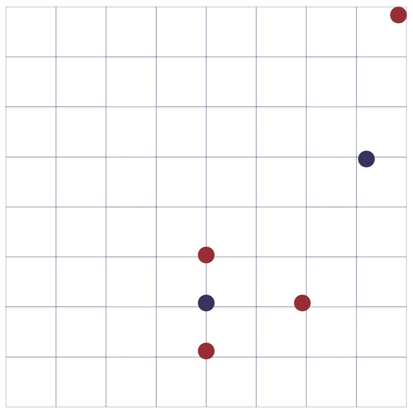
Given the following 4 instances and 2 clusters initialized as shown. Assume the distance function is $\mathrm{dist}(x_i, x_j) = \sum_e |x_{i,e} - x_{j,e}|$



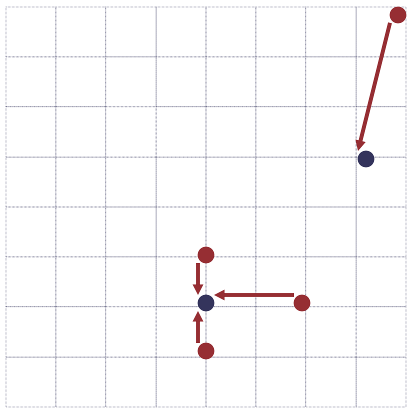
$$dist(x_1, f_1) = 2, \quad dist(x_1, f_2) = 5$$
$$dist(x_2, f_1) = 2, \quad dist(x_2, f_2) = 3$$
$$dist(x_3, f_1) = 3, \quad dist(x_3, f_2) = 2$$
$$dist(x_4, f_1) = 11, \quad dist(x_4, f_2) = 6$$



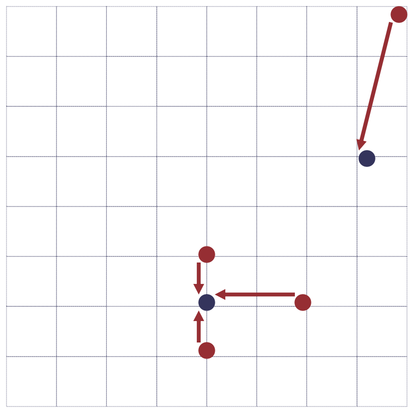
$$f_1 = \left\langle \frac{4+4}{2}, \frac{1+3}{2} \right\rangle = \langle 4,2 \rangle$$
$$f_2 = \left\langle \frac{6+8}{2}, \frac{2+8}{2} \right\rangle = \langle 7,5 \rangle$$



$$dist(x_1, f_1) = 1, \quad dist(x_1, f_2) = 7$$
$$dist(x_2, f_1) = 1, \quad dist(x_2, f_2) = 5$$
$$dist(x_3, f_1) = 2, \quad dist(x_3, f_2) = 4$$
$$dist(x_4, f_1) = 10, \quad dist(x_4, f_2) = 4$$

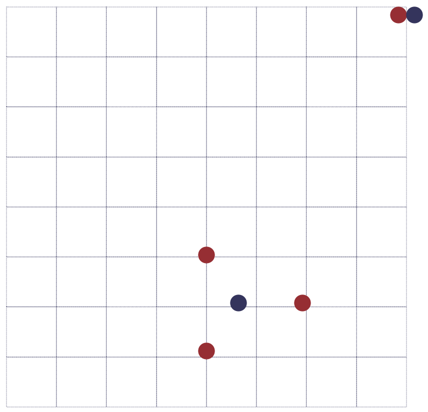


K-means Clustering Example (Continued)



$$f_1 = \left\langle \frac{4+4+6}{3}, \frac{1+3+2}{3} \right\rangle = \langle 4.67, 2 \rangle$$

$$f_2 = \left\langle \frac{8}{1}, \frac{8}{1} \right\rangle = \langle 8, 8 \rangle$$



assignments remain the same,
so the procedure has converged

EM Clustering

- in *k*-means as just described, instances are assigned to one and only one cluster
- we can do “soft” *k*-means clustering via an *Expectation Maximization* (EM) algorithm
  - each cluster represented by a distribution (e.g. a Gaussian)
  - E step: determine how likely is it that each cluster “generated” each instance
  - M step: adjust cluster parameters to maximize likelihood of instances

Representation of Clusters

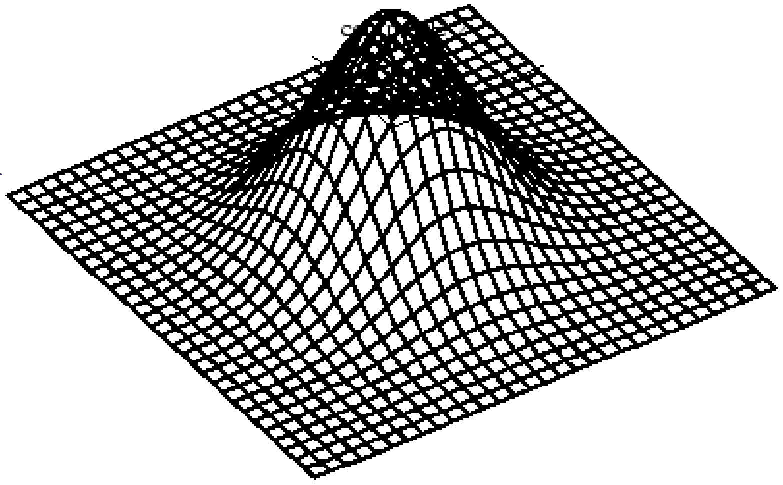
- in the EM approach, we'll represent each cluster using an *m*-dimensional multivariate Gaussian

$$N_j(\vec{x}_i) = \frac{1}{\sqrt{(2\pi)^m |\Sigma_j|}} \exp\left[-\frac{1}{2}(\vec{x}_i - \vec{\mu}_j)^T \Sigma_j^{-1} (\vec{x}_i - \vec{\mu}_j) \right]$$

where

$\vec{\mu}_j$ is the mean of the Gaussian

Σ_j is the covariance matrix



this is a representation of a Gaussian in a 2-D space

EM Clustering

- the EM algorithm will try to set the parameters of the Gaussians, Θ , to maximize the log likelihood of the data, X

$$\begin{aligned}
\text{log likelihood}(X \mid \Theta) &= \log \prod_{i=1}^{n} \Pr(\vec{x}_i) \\
&= \log \prod_{i=1}^{n} \sum_{j=1}^{k} N_j(\vec{x}_i) \\
&= \sum_{i=1}^{n} \log \sum_{j=1}^{k} N_j(\vec{x}_i)
\end{aligned}$$

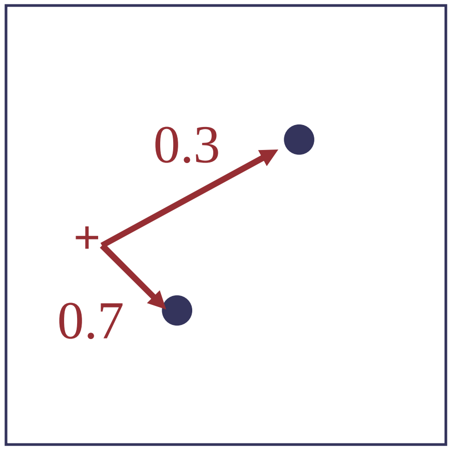
EM Clustering

- the parameters of the model, Θ , include the means, the covariance matrix and sometimes prior weights for each Gaussian
- here, we'll assume that the covariance matrix and the prior weights are fixed; we'll focus just on setting the means

EM Clustering: the E-step

- recall that z_{ij} is a hidden variable which is 1 if N_j generated $\vec{x}_i$ and 0 otherwise
- in the E-step, we compute h_{ij}, the expected value of this hidden variable

$$h_{ij} = E(z_{ij} \mid \vec{x}_i) = \frac{N_j(\vec{x}_i)}{\sum_{l=1}^{k} N_l(\vec{x}_i)}$$



assignment

EM Clustering: the M-step

- given the expected values h_{ij} , we re-estimate the means of the Gaussians

$$\vec{\mu}'_j = \frac{\sum_{i=1}^{n} h_{ij}\vec{x}_i}{\sum_{i=1}^{n} h_{ij}}$$

- can also re-estimate the covariance matrix and prior weights, if we’re varying them

EM and *K*-Means Clustering

- both will converge to a local maximum
- both are sensitive to initial positions (means) of clusters
- have to choose value of *k* for both

Evaluating Clustering Results

- given random data without any “structure”, clustering algorithms will still return clusters
- the gold standard: do clusters correspond to natural categories?
- do clusters correspond to categories we care about? (there are lots of ways to partition the world)

Evaluating Clustering Results

- some approaches
  - external validation
    - E.g. do genes clustered together have some common function?
  - internal validation
    - How well does clustering optimize intra-cluster similarity and inter-cluster dissimilarity?
  - relative validation
    - How does it compare to <u>other</u> clusterings using these criteria?
    - E.g. with a probabilistic method (such as EM) we can ask: how probable does held-aside data look as we vary the number of clusters.

Comments on Clustering

- there many different ways to do clustering; we ‘ve discussed just a few methods
- hierarchical clusters may be more informative, but they’re more expensive to compute
- clusterings are hard to evaluate in many cases