Clustering Gene Expression Data

BMI/CS 576
www.biostat.wisc.edu/bmi576/
Mark Craven
craven@biostat.wisc.edu
Fall 2011

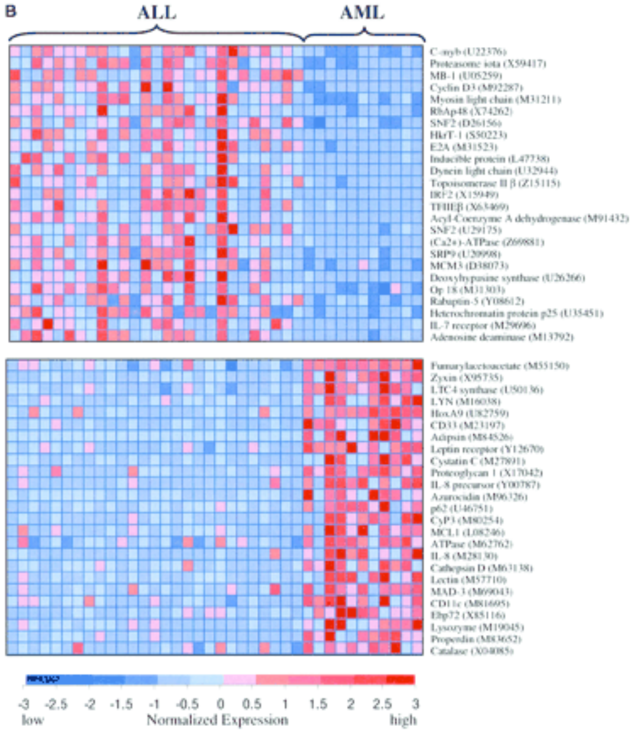
Gene expression profiles

- we'll assume we have a 2D matrix of gene expression measurements
  - rows represent genes
  - columns represent different experiments, time points, individuals etc.

- we'll refer to individual rows or columns as *profiles*
  - a row is a profile for a gene
  - a column is a profile for an experiment, time point, etc.

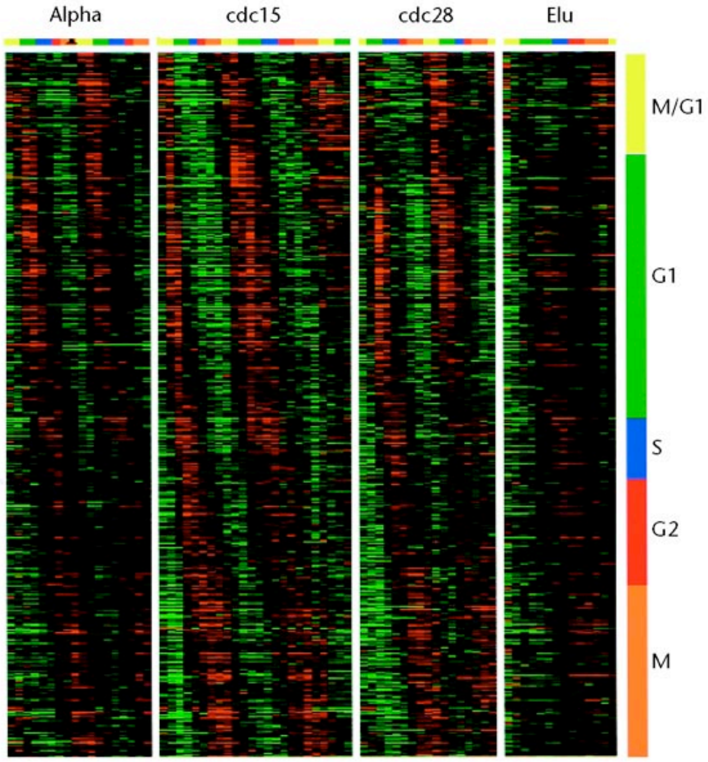
Expression profile example

- rows represent genes
- columns represent people with leukemia



Expression profile example

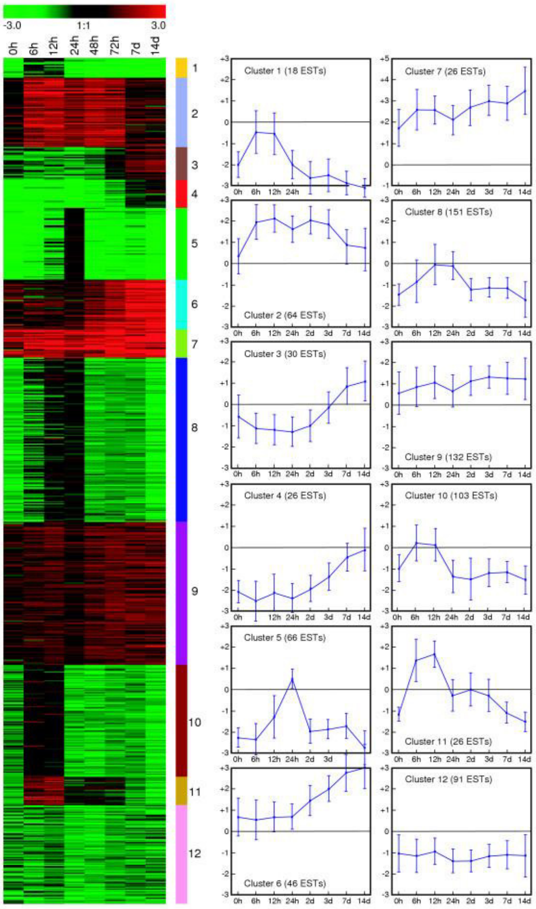
- rows represent yeast genes
- columns represent time points in a given experiment



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
  - profiles in the same cluster are highly similar to each other
  - profiles from different clusters have low similarity to each other



Clustering example

- pre-adipocyte (fat) cell development over 14-day time course
- clustering of 780 genes that are > 2-fold upregulated or downregulated at ≥ 4 time points

figure from: Hack et al. *Genome Biology* 6(13), 2005

Motivation for clustering

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

The clustering landscape

- there are many different clustering algorithms
- they differ along several dimensions
  - hierarchical vs. flat
  - hard (no uncertainty about which profiles belong to a cluster) vs. soft clusters
  - non-partitional (a profile can belong to multiple clusters) vs. partitional
  - 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 profiles
  - a cluster and a profile
  - a pair of clusters

- given a distance value, it is straightforward to convert it into a similarity value

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

- not necessarily straightforward to go the other way

$$\text{dist}(x,y) = \exp(-a \times \text{sim}(x,y))$$

- 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

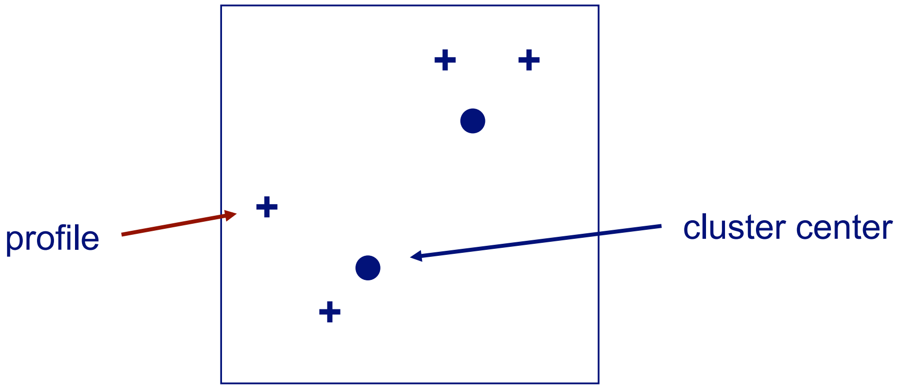
Manhattan $\text{dist}(x_i, x_j) = \sum_e \left| x_{i,e} - x_{j,e} \right|$

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

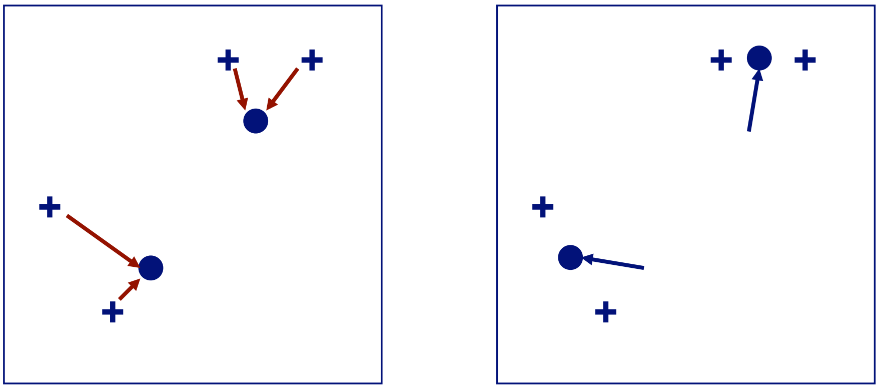
K-means clustering

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



K-means clustering

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



assignment

re-computation of means

K-means clustering: updating the means

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

$$\vec{\mu}_j = \frac{1}{|c_j|} \sum_{\vec{x}_i \in c_j} \vec{x}_i$$

K-means clustering

```
given: $k$, a set $X = \{\vec{x}_1 \ldots \vec{x}_n\}$ of profiles
select $k$ initial cluster means $\vec{\mu}_1 \ldots \vec{\mu}_k$
while stopping criterion not met do
    for all clusters $c_j$ do
        // determine which profiles are assigned to this cluster
        $c_j = \left\{ \vec{x}_i \mid \forall f_l \ \text{dist}(\vec{x}_i, \vec{\mu}_j) < \text{dist}(\vec{x}_i, \vec{\mu}_l) \right\}$
    for all means $\vec{f}_j$ do
        // update the cluster center
        $\vec{\mu}_j = \frac{1}{|c_j|} \sum_{\vec{x}_i \in c_j} \vec{x}_i$
```

K-means objective function

- *residual sum of squares* (RSS): measure of how well cluster means represent their members

$$RSS = \sum_{k} \sum_{\vec{x}_i \in c_k} \left| \vec{x}_i - \vec{\mu}_k \right|^2$$

- when Euclidean distance used, *k*-means locally minimizes this quantity

- local optimum depends on starting positions for cluster means

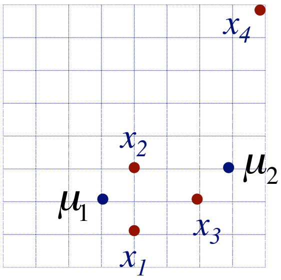
K-means stopping criteria

- standard stopping criterion: assignment of profiles to clusters does not change (equivalently, cluster means do not change)

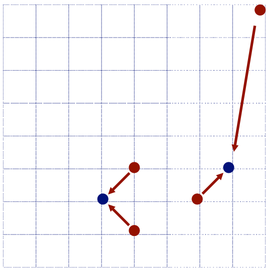
- for faster runtimes, can stop
  - after a fixed number of iterations
  - when RSS (or change in RSS) falls below a threshold

K-means clustering example

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

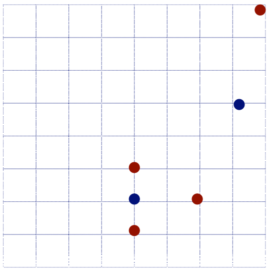


$$\begin{aligned}
&dist(x_1,\mu_1) = 2, \quad dist(x_1,\mu_2) = 5\\
&dist(x_2,\mu_1) = 2, \quad dist(x_2,\mu_2) = 3\\
&dist(x_3,\mu_1) = 3, \quad dist(x_3,\mu_2) = 2\\
&dist(x_4,\mu_1) = 11, \quad dist(x_4,\mu_2) = 6
\end{aligned}$$

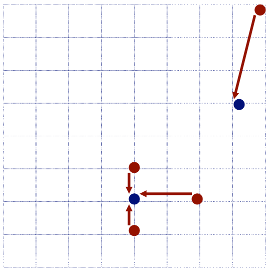


$$\mu_1 = \left\langle \frac{4+4}{2}, \frac{1+3}{2} \right\rangle = \langle 4,2 \rangle$$

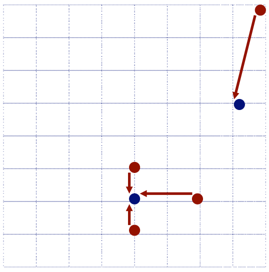
$$\mu_2 = \left\langle \frac{6+8}{2}, \frac{2+8}{2} \right\rangle = \langle 7,5 \rangle$$



$$\begin{aligned}
&dist(x_1,\mu_1) = 1, \quad dist(x_1,\mu_2) = 7\\
&dist(x_2,\mu_1) = 1, \quad dist(x_2,\mu_2) = 5\\
&dist(x_3,\mu_1) = 2, \quad dist(x_3,\mu_2) = 4\\
&dist(x_4,\mu_1) = 10, \quad dist(x_4,\mu_2) = 4
\end{aligned}$$

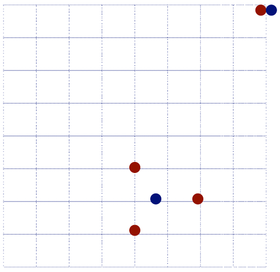


K-means clustering example (continued)



$$\mu_1 = \left\langle \frac{4+4+6}{3}, \frac{1+3+2}{3} \right\rangle = \langle 4.67,2 \rangle$$

$$\mu_2 = \left\langle \frac{8}{1}, \frac{8}{1} \right\rangle = \langle 8,8 \rangle$$



assignments remain the same,
so the procedure has converged

EM clustering

- in *k*-means as just described, profiles 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 profile
  - M step: adjust cluster parameters to maximize likelihood of profiles

Representation of clusters

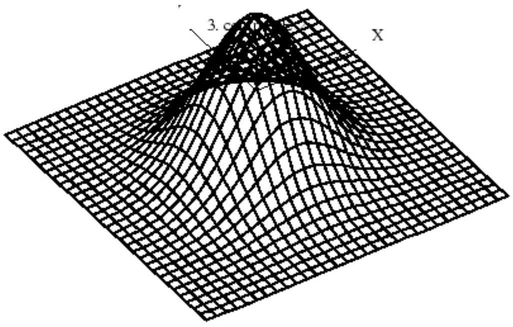
- in the EM approach, we’ll represent each cluster using an m-dimensional multivariate Gaussian

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

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 parameters of the model include the means, the covariance matrix and sometimes prior weights for each Gaussian

$$\Theta = \{\vec{\mu}_1, \ldots, \vec{\mu}_k, \Sigma_1, \ldots, \Sigma_k\}$$

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

$$\Theta = \arg\max_\Theta \log \prod_{i=1}^{n} P(\vec{x}_i \mid \Theta)$$

$$= \arg\max_\Theta \sum_{i=1}^{n} \log P(\vec{x}_i \mid \Theta)$$

$$= \arg\max_\Theta \sum_{i=1}^{n} \log \sum_{j=1}^{k} f_j(\vec{x}_i)$$

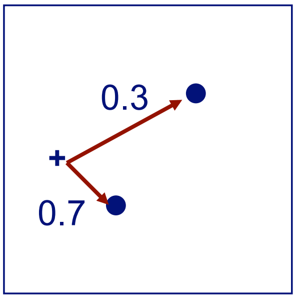
EM clustering: hidden variables

- on each iteration of *k*-means clustering, we had to assign each profile to a cluster
- in the EM approach, we'll use hidden variables to represent this idea
- for each profile $\vec{x}_i$ we have a set of hidden variables $Z_{i1},\ldots,Z_{ik}$
- we can think of Z_{ij} as being 1 if $\vec{x}_i$ is a member of cluster j and 0 otherwise

EM clustering: the E-step

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

$$h_{ij} = P\left(Z_{ij} = 1 \mid \vec{x}_i\right) = \frac{f_j(\vec{x}_i)}{\sum_{l=1}^{k} f_l(\vec{x}_i)}$$



assignment

EM clustering: the M-step

- given the expected values, we re-estimate the means of the Gaussians

$$\vec{\mu}_j = \frac{\sum_i h_{ij}\vec{x}_i}{\sum_i h_{ij}}$$

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

EM clustering example

Consider a one-dimensional clustering problem in which the data given are:

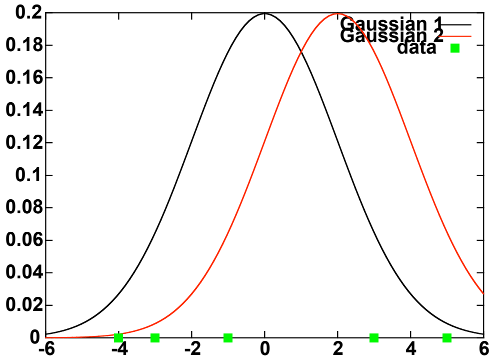
x_1 = -4
x_2 = -3
x_3 = -1
x_4 = 3
x_5 = 5

The initial mean of the first Gaussian is 0 and the initial mean of the second is 2. The Gaussians have fixed width; their density function is:

$$f(x) = \frac{1}{\sqrt{8\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{2}\right)^2}$$

where μ denotes the mean (center) of the Gaussian.

EM clustering example



$$f(x) = \frac{1}{\sqrt{8\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{2}\right)^2}$$

$$f_1(-4) = \frac{1}{\sqrt{8\pi}} e^{-\frac{1}{2}\left(\frac{-4\ -\ 0}{2}\right)^2} = .0269 \qquad f_2(-4) = \frac{1}{\sqrt{8\pi}} e^{-\frac{1}{2}\left(\frac{-4-2}{2}\right)^2} = .0022$$

$$f_1(-3) = .0646 \qquad f_2(-3) = .00874$$

$$f_1(-1) = .176 \qquad f_2(-1) = .0646$$

$$f_1(3) = .0646 \qquad f_2(3) = .176$$

$$f_1(5) = .00874 \qquad f_2(5) = .0646$$

EM clustering example: E-step

$$h_{11} = \frac{f_1(x_1)}{f_1(x_1) + f_2(x_1)} = \frac{.0269}{.0269 + .0022} \qquad h_{12} = \frac{f_2(x_1)}{f_1(x_1) + f_2(x_1)} = \frac{.0022}{.0269 + .0022}$$

$$h_{21} = \frac{f_1(x_2)}{f_1(x_2) + f_2(x_2)} = \frac{.0646}{.0646 + .00874} \qquad h_{22} = \frac{.00874}{.0646 + .00874}$$

$$h_{31} = \frac{.176}{.176 + .0646} \qquad h_{32} = \frac{.0646}{.176 + .0646}$$

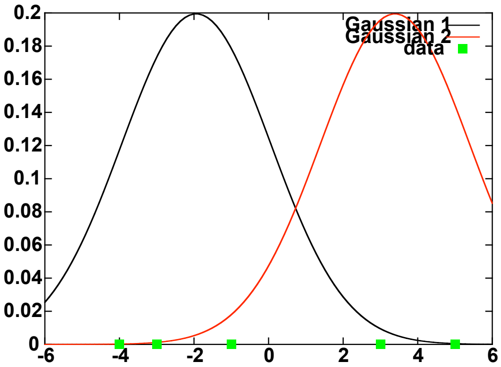
$$h_{41} = \frac{.0646}{.0646 + .176} \qquad h_{42} = \frac{.176}{.0646 + .176}$$

$$h_{51} = \frac{.00874}{.00874 + .0646} \qquad h_{52} = \frac{.0646}{.00874 + .0646}$$

EM clustering example: M-step

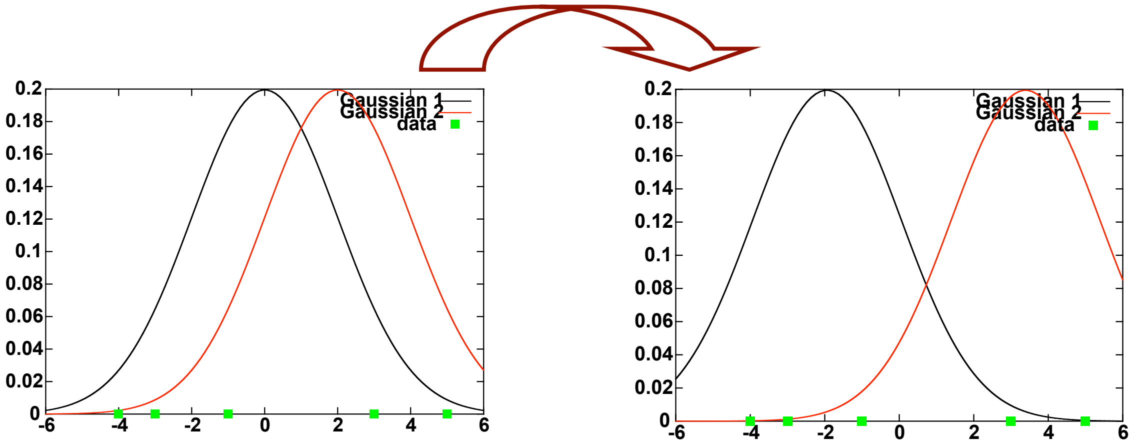
$$\mu_1 = \frac{\sum_i x_i \times h_{i1}}{\sum_i h_{i1}} = \frac{-4 \times .924 + -3 \times .881 + -1 \times .732 + 3 \times .268 + 5 \times .119}{.924 + .881 + .732 + .268 + .119} = -1.94$$

$$\mu_2 = \frac{\sum_i x_i \times h_{i2}}{\sum_i h_{i2}} = \frac{-4 \times .076 + -3 \times .119 + -1 \times .268 + 3 \times .732 + 5 \times .881}{.076 + .119 + .268 + .732 + .881} = 3.39$$



EM clustering example

- here we've shown just one step of the EM procedure



- we would continue the E- and M-steps until convergence

Computational complexity

- k-means and EM have time complexity $O(kn)$ for each iteration
  - reassignment step: compute $k \times n$ distances
  - recomputation step: loop through *n* profiles updating k means

EM and *k*-Means clustering

- both will converge to a local optimum
- both are sensitive to initial positions (means) of clusters, thus it’s often beneficial to run multiple times with different starting positions
- have to choose value of k for both

Choosing the value of k

- we can run *k*-means/EM multiple times with different values of *k*
- Can we pick the best clustering by seeing which run results in the best value of the objective function?

$$k = \arg\max_{k,\Theta} \sum_{i=1}^{n} \log P\left(\vec{x}_i \mid k, \Theta\right)$$

for EM

$$k = \arg\min_{k,\Theta} \sum_{k} \sum_{\vec{x}_i \in c_k} \left|\vec{x}_i - \vec{\mu}_k\right|^2$$

for *k*-means

  - No – the objective function will generally improve as *k* increases. The best value will be with *k* = *n*.

Choosing the value of k

- an alternative is to add a penalty for complexity

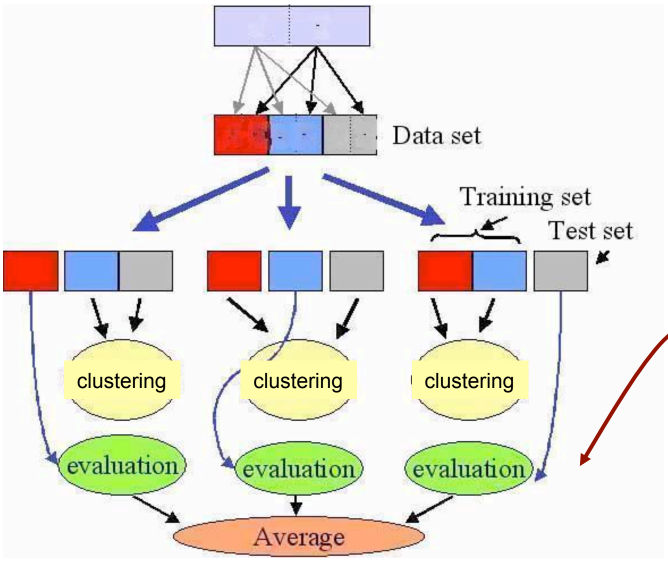
$$k = \arg\min_{k,\Theta} \sum_{k} \sum_{\vec{x}_i \in c_k} \left|\vec{x}_i - \vec{\mu}_k\right|^2 + \lambda \cdot k$$

λ determines how much weight is put on complexity

- e.g. the *Akaike Information Criterion* sets $\lambda = 2M$ where M is the number of elements in each profile

Cross validation to select *k*

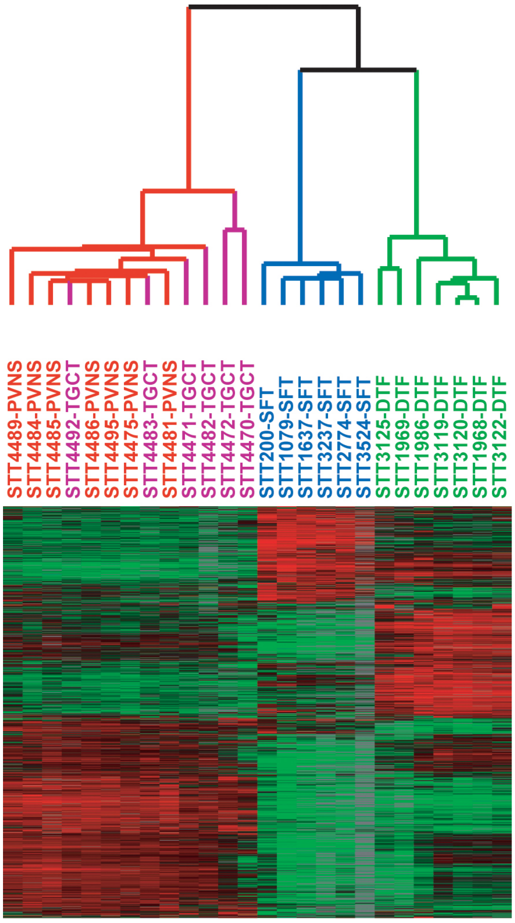
- using cross validation, we can use held-aside data to assess the objective function for different values of *k*



compute objective function on held-aside data to evaluate clustering

- then run method on all data once we've picked *k*

Hierarchical clustering example



- clustering of related cancers and an inflammatory disorder

  TGCT: Tenosynovial giant-cell tumor

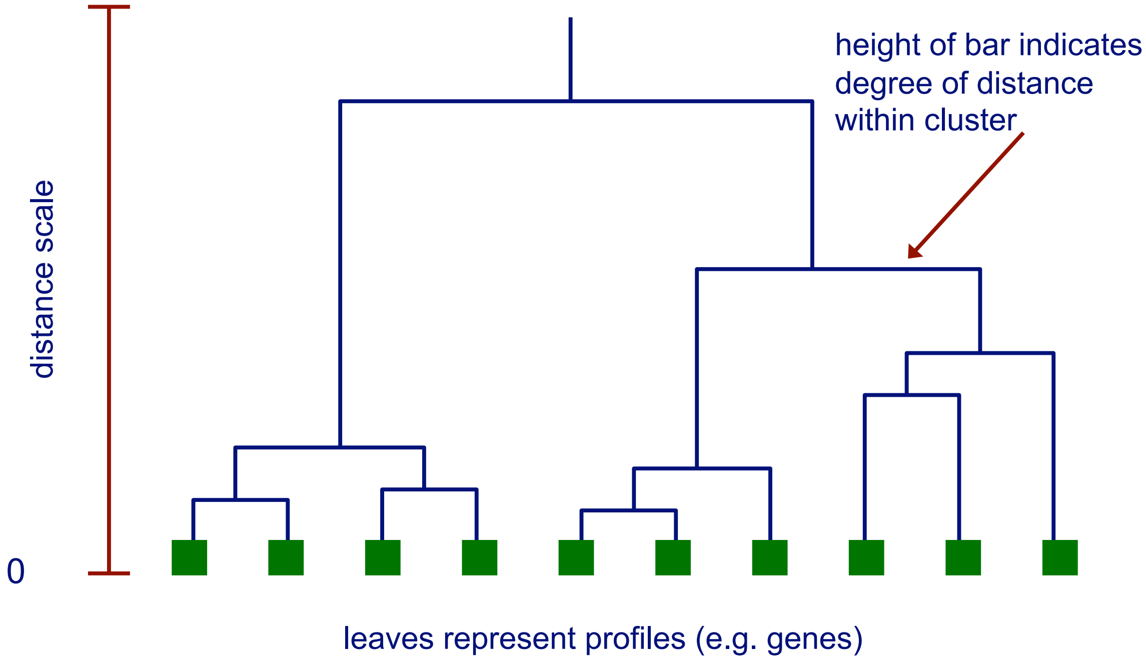
  PVNS: pigmented villonodular synovitis

  SFT: solitary fibrous tumor

  DTF: desmoid-type fibromatosis

figure from: West et al. *PNAS* 103, 2006

Hierarchical clustering: a dendogram



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
  - expression profiles
  - clusters (formed so far)
  - profiles and clusters

Bottom-up hierarchical clustering

```
given:a set $X = \{x_1 \ldots x_n\}$ of instances
for $i := 1$ to $n$ do
    $c_i := \{x_i\}$          // each instance is initially its own cluster, and a leaf in tree
$C := \{c_1 \ldots c_n\}$
$j := n$
while $|C| > 1$
    $j := j + 1$
    $(c_a, c_b) := \underset{(c_u, c_v)}{\operatorname{argmin}} \ \operatorname{dist}(c_u, c_v)$          // find least distant pair in C
    $c_j = c_a \cup c_b$          // create a new cluster for pair
    add a new node $j$ to the tree joining $a$ and $b$
    $C := C - \{c_a, c_b\} \cup \{c_j\}$
return tree with root node $j$
```

Haven’t we seen this already?

- this algorithm is very similar to UPGMA and neighbor joining; there are some differences
- what tree represents
  - phylogenetic inference: tree represents hypothesized sequence of evolutionary events; internal nodes represent hypothetical ancestors
  - clustering: inferred tree represents similarity of instances; internal nodes don’t represent ancestors
- form of tree
  - UPGMA: rooted tree
  - neighbor joining: unrooted
  - hierarchical clustering: rooted tree
- how distances among clusters are calculated
  - UPGMA*: average link*
  - neighbor joining: based on additivity
  - hierarchical clustering: various

Distance between two clusters

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

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

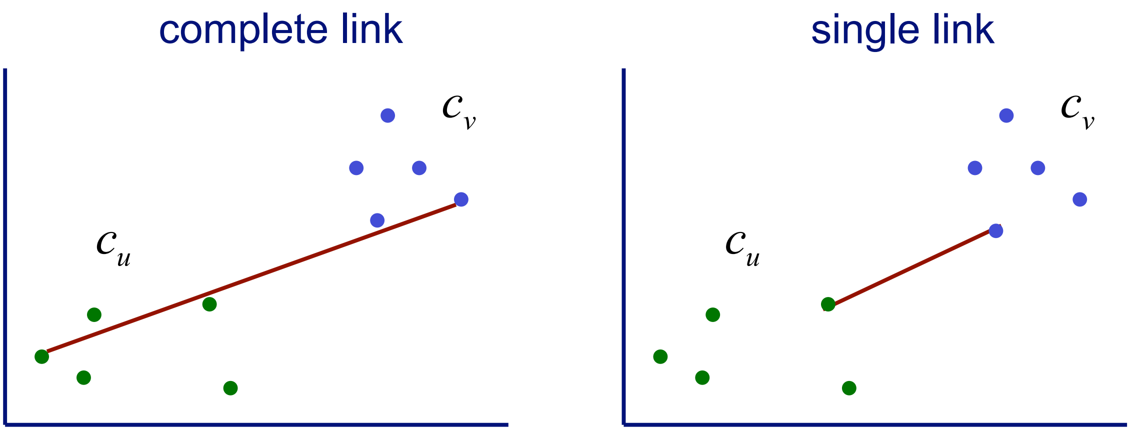
  - *complete link*: distance of two least similar profiles

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

  - *average link*: average distance between profiles

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

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:

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

Computational complexity

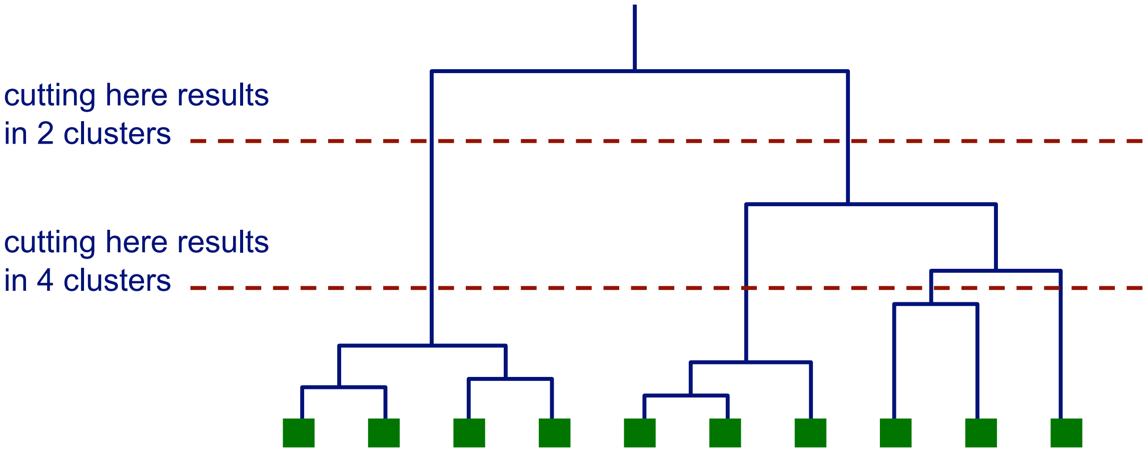
- the naïve implementation of hierarchical clustering has $O(n^3)$ time complexity, where n is the number of instances
  - computing the initial distance matrix takes $O(n^2)$ time
  - there are $O(n)$ merging steps
  - on each step, we have to update the distance matrix $O(n)$ and select the next pair of clusters to merge $O(n^2)$

Computational complexity

- using more sophisticated data structures to maintain the pairwise distance data we improve the time complexity
  - for single-link clustering, we can update and pick the next pair in $O(n)$ time, resulting in an $O(n^2)$ algorithm
  - for complete-link and average-link we can do these steps in $O(n \log n)$ time resulting in an $O(n^2 \log n)$ method

Flat clustering from a hierarchical clustering

- we can always generate a flat clustering from a hierarchical clustering by “cutting” the tree at some distance threshold


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

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

- 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.

Internal validation

- there are many different measures for assessing internal validation
- one such measure is the *Silhouette index*

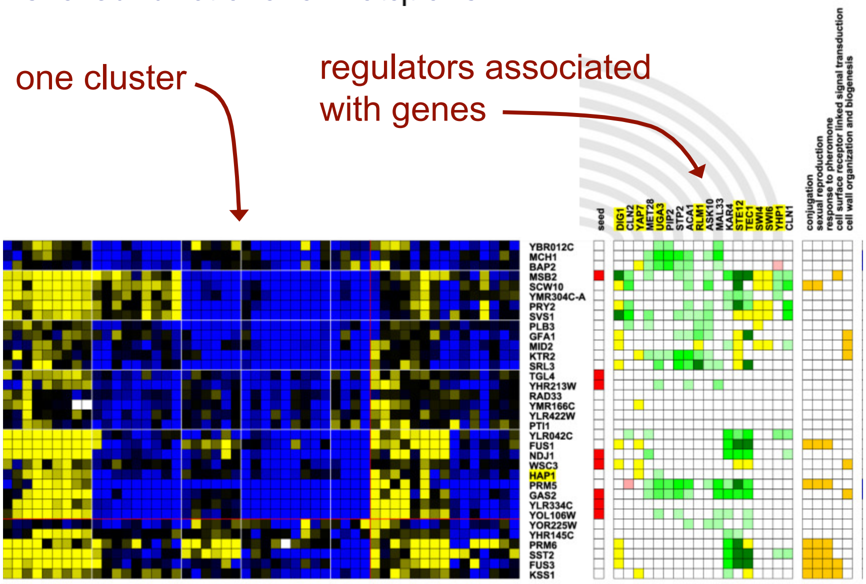
$$\frac{1}{k}\sum_{k}\left(\frac{1}{|c_k|}\sum_{\vec{x}_i \in c_k}\frac{b(\vec{x}_i)-a(\vec{x}_i)}{\max\left[b(\vec{x}_i),a(\vec{x}_i)\right]}\right)$$

$a(\vec{x}_i)$ average distance from $\vec{x}_i$ to other instances in same cluster

$b(\vec{x}_i)$ average distance from $\vec{x}_i$ to instances in next closest cluster

External validation

- can determine if a cluster seems to be correlated with other relevant information
- e.g. do the genes have
  - binding sites for common regulators
  - shared functional annotations



one cluster

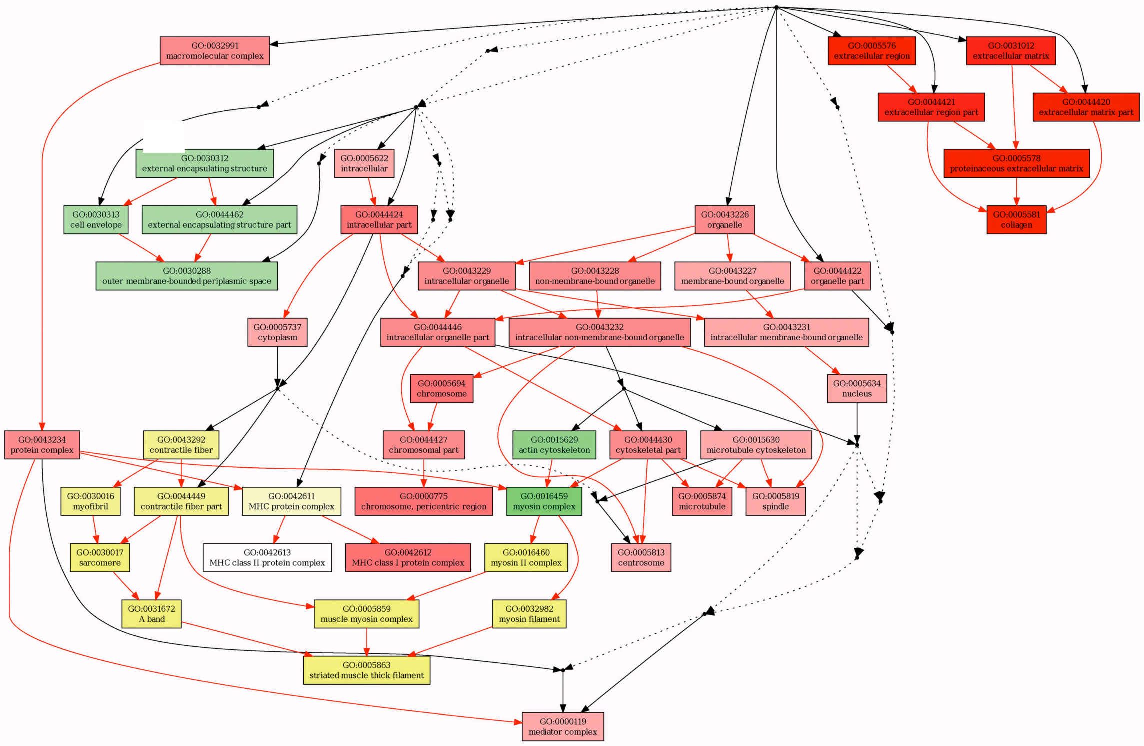
regulators associated with genes

Gene Ontology terms associated with the genes

figure from: Maere et al. BMC Systems Biology 2, 2008

The Gene Ontology

- a controlled vocabulary of more than 30K concepts describing molecular functions, biological processes, and cellular components


GO:0032991 macromolecular complex
GO:0005576 extracellular region
GO:0031012 extracellular matrix
GO:0044421 extracellular region part
GO:0044420 extracellular matrix part
GO:0030312 external encapsulating structure
GO:0005622 intracellular
GO:0005578 proteinaceous extracellular matrix
GO:0030313 cell envelope
GO:0044462 external encapsulating structure part
GO:0044424 intracellular part
GO:0043226 organelle
GO:0005581 collagen
GO:0030288 outer membrane-bounded periplasmic space
GO:0043229 intracellular organelle
GO:0043228 non-membrane-bound organelle
GO:0043227 membrane-bound organelle
GO:0044422 organelle part
GO:0005737 cytoplasm
GO:0044446 intracellular organelle part
GO:0043232 intracellular non-membrane-bound organelle
GO:0043231 intracellular membrane-bound organelle
GO:0005694 chromosome
GO:0005634 nucleus
GO:0043234 protein complex
GO:0043292 contractile fiber
GO:0044427 chromosomal part
GO:0015629 actin cytoskeleton
GO:0044430 cytoskeletal part
GO:0015630 microtubule cytoskeleton
GO:0030016 myofibril
GO:0044449 contractile fiber part
GO:0042611 MHC protein complex
GO:0000775 chromosome, pericentric region
GO:0016459 myosin complex
GO:0005874 microtubule
GO:0005819 spindle
GO:0030017 sarcomere
GO:0042613 MHC class II protein complex
GO:0042612 MHC class I protein complex
GO:0016460 myosin II complex
GO:0005813 centrosome
GO:0031672 A band
GO:0005859 muscle myosin complex
GO:0032982 myosin filament
GO:0005863 striated muscle thick filament
GO:0000119 mediator complex

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