10-810 /02-710 Computational Genomics

Clustering expression data

Goal

- Data organization (for further study)
- Functional assignment
- Determine different patterns

- Classification
- Relations between experimental conditions

 Subsets of genes related to subset of experiments Genes

Experiments

Both

Clustering metric

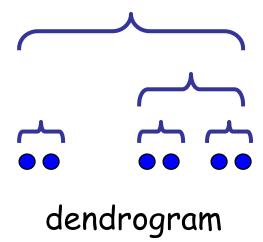
- A key issue in clustering is to determine the similarity / distance metric.
- Often, such metric has a bigger impact on the results than the actual clustering algorithm used
- When determining the metric we should take into account our assumptions about the data and the goals of the clustering algorithm.

Clustering algorithms

- We can divide clustering methods into roughly three types:
- 1. hierarchical agglomerative clustering
 - For example, hierarchical clustering
- 2. Model based
 - For example, k-means, Gaussian mixtures
- 3. Iterative partitioning (top down)
 - For example, graph based algorithms

Hierarchical clustering

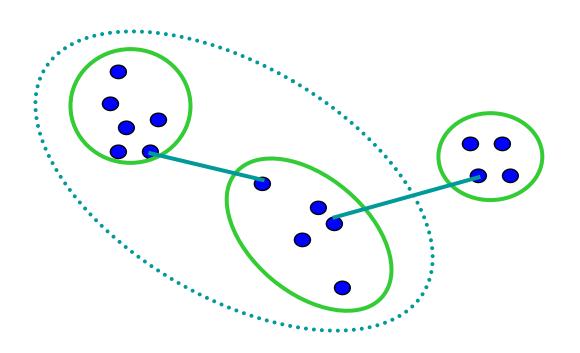
- Probably the most popular clustering algorithm in this area
- First presented in this context by Eisen in 1998



- Agglomerative (bottom-up)
- Algorithm:
 - 1. Initialize: each item a cluster
 - 2. Iterate:
 - select two most similar clusters
 - merge them
 - 3. Halt: when there is only one cluster left

Similarity criteria: Single Link

cluster similarity = similarity of two most similar members



Potentially
 long and skinny
 clusters

Example: single link

$$\begin{aligned} &d_{(1,2),3} = \min\{d_{1,3}, d_{2,3}\} = \min\{6,3\} = 3\\ &d_{(1,2),4} = \min\{d_{1,4}, d_{2,4}\} = \min\{10,9\} = 9\\ &d_{(1,2),5} = \min\{d_{1,5}, d_{2,5}\} = \min\{9,8\} = 8 \end{aligned}$$

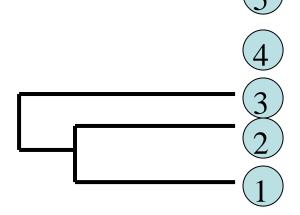






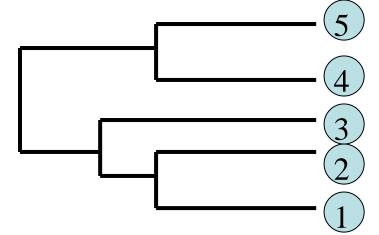
Example: single link

$$\begin{aligned} d_{(1,2,3),4} &= \min\{d_{(1,2),4}, d_{3,4}\} = \min\{9,7\} = 7 \\ d_{(1,2,3),5} &= \min\{d_{(1,2),5}, d_{3,5}\} = \min\{8,5\} = 5 \end{aligned}$$



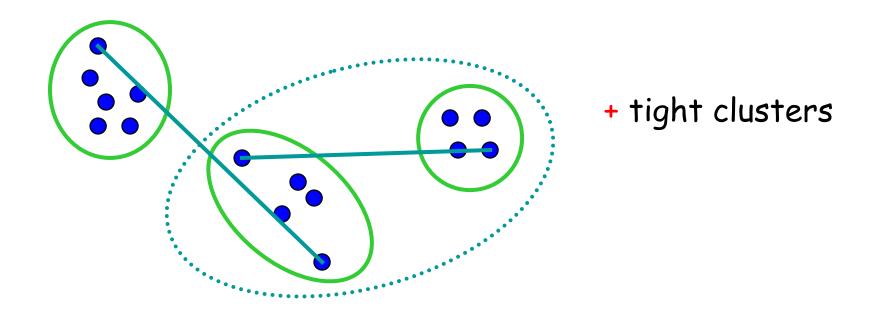
Example: single link

$$d_{(1,2,3),(4,5)} = \min\{d_{(1,2,3),4},d_{(1,2,3),5}\} = 5$$



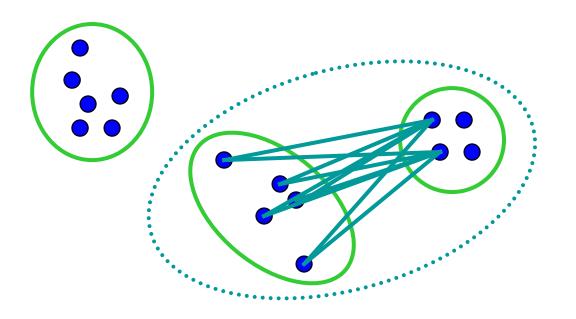
Hierarchical: Complete Link

cluster similarity = similarity of two least similar members



Hierarchical: Average Link

cluster similarity = average similarity of all pairs



the most widely used similarity measure

Robust against noise

Similarity measure

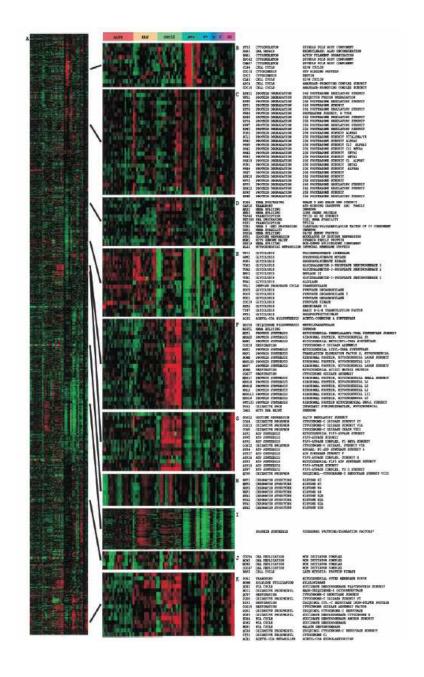
- In most cases the correlation coefficient ((normalized dot product) is used
- The correlation coefficient is defined as:

$$\rho_{x,y} = \frac{\text{cov}(x,y)}{\text{std}(x)\text{std}(y)} = \frac{\sum_{i} (x_i - \mu_x)(y_i - \mu_y)}{\sigma_x \sigma_y}$$

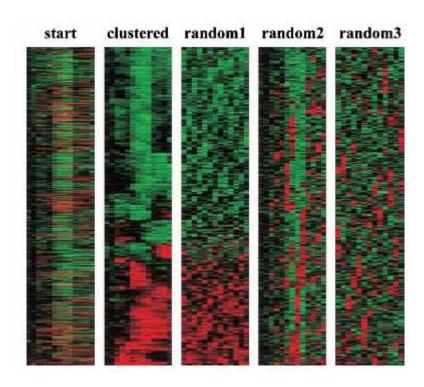
- Advantages:
 - Identifies relationships regardless of absolute unit changes
 - A simple way around missing values
- Disadvantages
 - Not a metric

Cluster results

Combining several time series yeast datasets



Validation



Model based clustering

- In model based clustering methods we assume a generative model by which the data was generated
- Our goal is to recover the parameters of such model, and use these to cluster the genes

Model based clustering

For simplicity we'll start with the following assumptions:

- clusters are exclusive (single gene, single cluster)
- we are searching for a fixed number of clusters (k)
- variation of profiles within a cluster can be modeled as a multivariate Gaussian

Clustering algorithm

- 1. initialize cluster models
- 2. iterate until convergence:
 - assign genes to clusters
 - estimate cluster models on the basis of the genes assigned to them

Our model: Gaussian mixtures

- We assume a generative model that works in the following way
- In order to generate a new point, we first chose a cluster 1 i k
 according to p(i)
- Next, we select the point using i's probability distribution model
- We assume that the profiles (vectors $\mathbf{x} = [x_1, ..., x_n]$) within each cluster are normally distributed such that $x \sim N(\mu,)$.

$$p(x \mid \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{n/2} |\Sigma_i|^{1/2}} e^{-(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)/2}$$

Likelihood

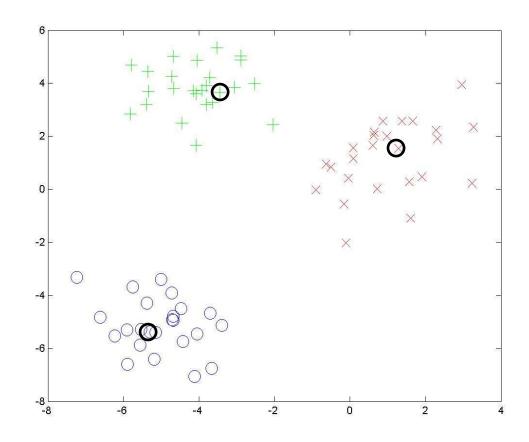
 Given our model, and a set of parameters for each of the clusters, we can compute the joint likelihood of our data.

$$L(D \mid M) = \prod_{i} \prod_{j} p(j) p(x_i \mid j)$$

 Our goal is to find a set of parameters that will maximize the above likelihood

Initialize

- The easiest way is to chose a random gene as a center for each of the clusters
- Initialization is a key aspect of this algorithm (and of other EM type algorithms we have discussed). It is wise to re-run the algorithm several times and chose the highest likelihood result as our clusters.
- We will need to chose the variance / covariance for each cluster



E step: Assigning profiles to clusters

 Simple way: assign each gene (profile x_j) to the cluster that gives the highest probability to it. In other words, gene j is assigned to cluster i when

$$p(x | \mu_i, \sigma_i) > p(x | \mu_j \sigma_j) \forall j \neq i$$

 Better way: assign each gene partially to different clusters based on the relative probabilities that the cluster models give to the profile

$$p(i \mid x) = \frac{p(x \mid \mu_i, \sigma_i) p(i)}{\sum_{i} p(x \mid \mu_i, \sigma_j)}$$

Each gene profile will consequently be associated with k assignment probabilities

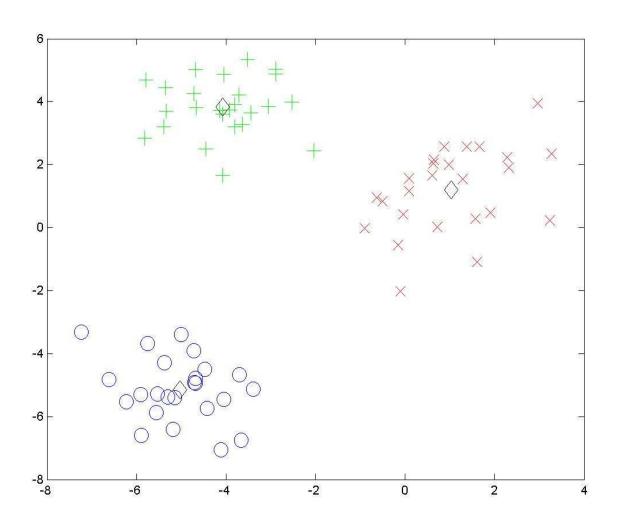
Re-computing the parameters

- We can re-estimate the Gaussian models on the basis of the partial (or simple) assignments
- Each cluster *i* sees a vector of *m* (the number of genes) assignment probabilities representing the degree to which profiles are assigned to the cluster

$$w_{i1} = P(i|x_1)$$
...
$$w_{im} = P(i|x_m)$$

 To re-estimate the cluster models we simply find the weighted mean and the covariance of the profiles, where the weighting is given by the above assignment probabilities

Re-computing the parameters



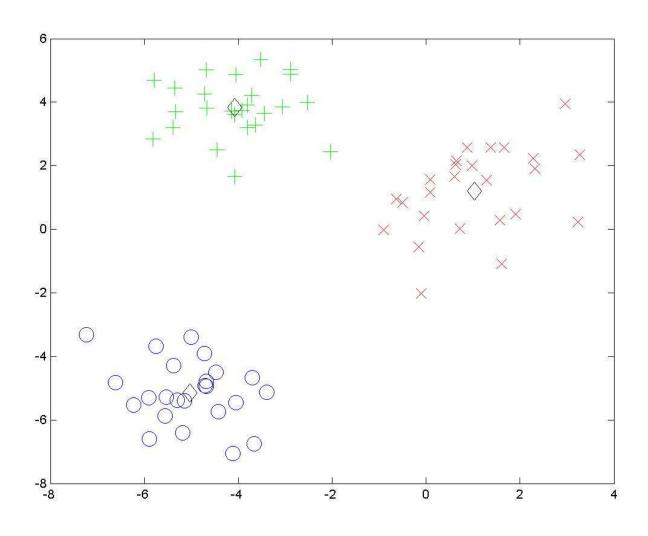
M step: Re-computing the parameters

- To re-estimate the cluster models we simply find the weighted mean and the covariance of the profiles, where the weighting is given by the above assignment probabilities
- We also determine the cluster distribution by setting

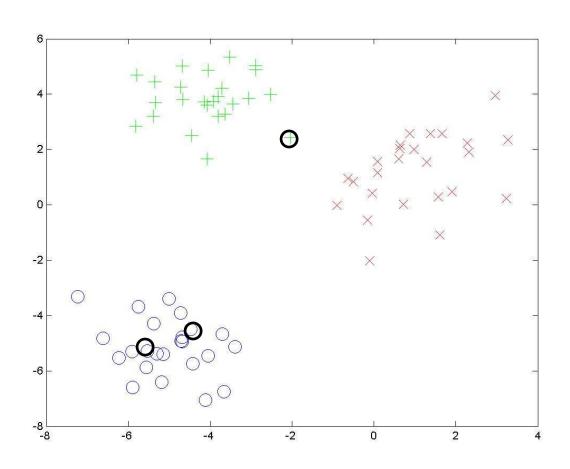
$$p(i) = \frac{\sum_{j} p(x_j \mid \mu_i, \sigma_i)}{\sum_{k} \sum_{j} p(x_j \mid \mu_k, \sigma_k)}$$

- It can be shown that such a computation is the MLE for the class parameters
- The two steps (E and M) are repeated until the parameters no longer change

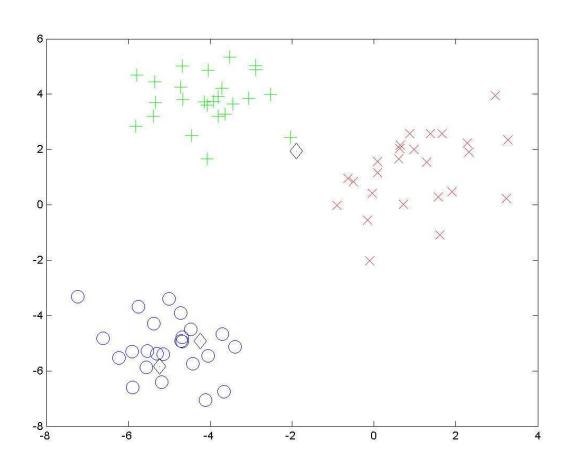
Second (and final) iteration



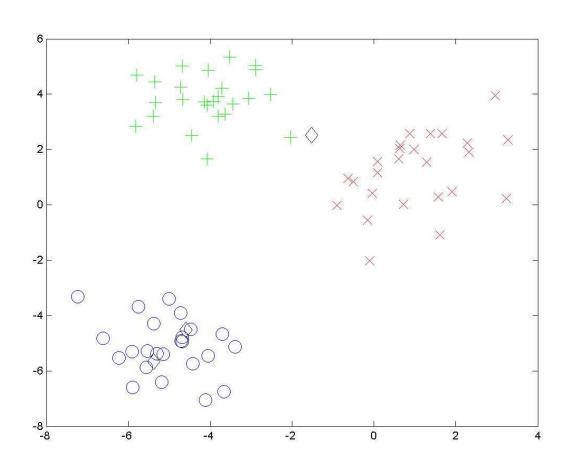
The importance of initializations



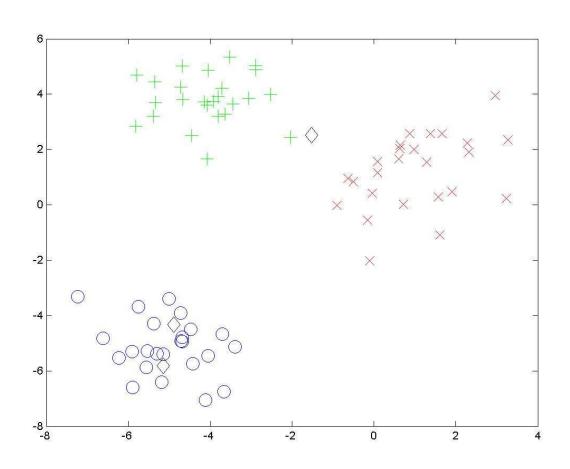
The importance of initializations: Step 1



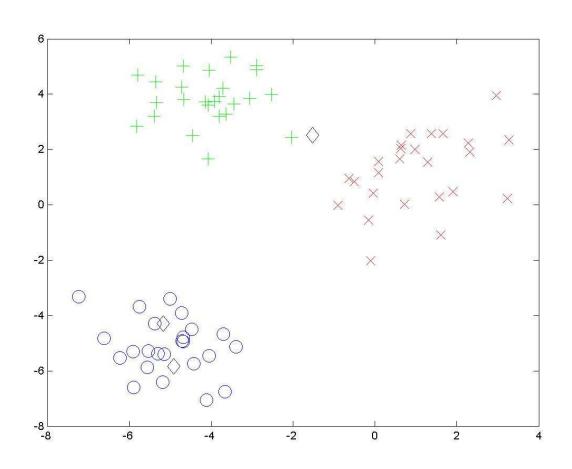
The importance of initializations: Step 2



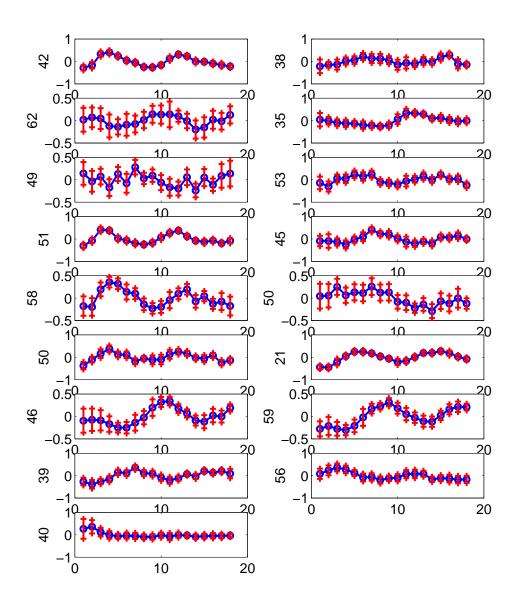
The importance of initializations: Step 5



The importance of initializations; Convergence

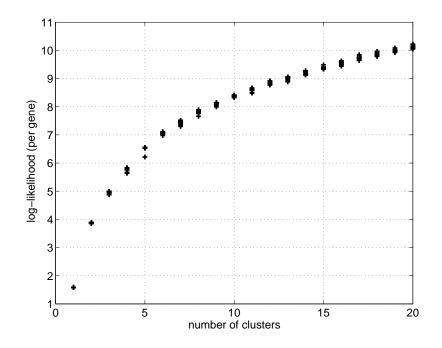


Example of clusters for the cell cycle expression dataset

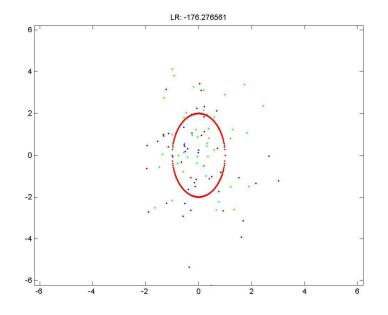


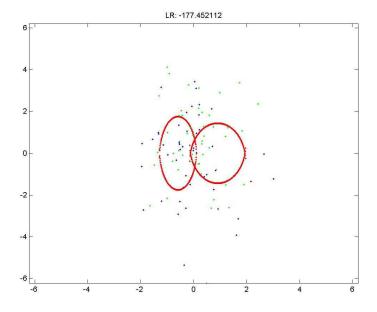
Number of clusters

- How do we find the right number of clusters?
- The overall log-likelihood of the profiles implied by the cluster models goes up as we add clusters
- One way is to use cross validation

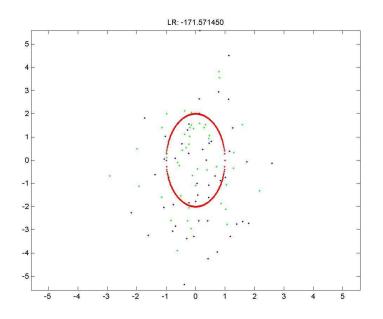


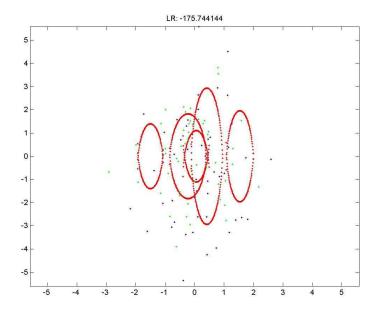
Cross validation





Cross validation



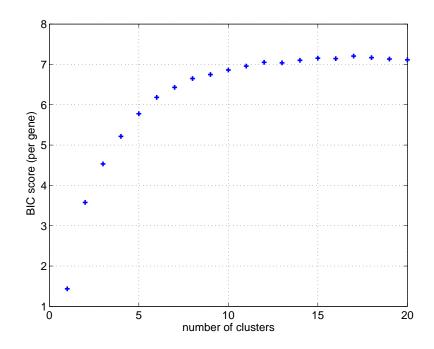


Number of clusters (cont.)

 Another possible solution: Bayesian information criterion (BIC):

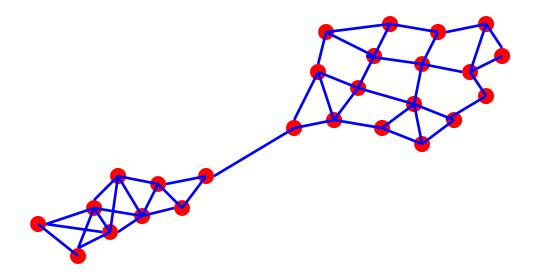
$$model - score = L(x \mid \Theta) - \frac{d}{2}\log(m)$$

 The log-likelihood is evaluated on the basis of the estimated cluster models (means, covariances, and frequencies), d is the number of independent parameters in the model, and m is the number of gene profiles



Top down: Graph based clustering

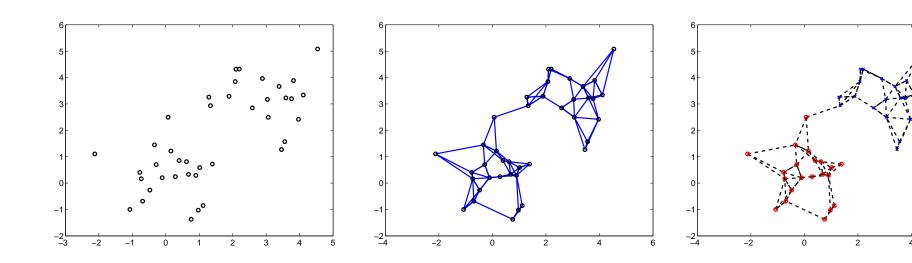
 Many top down clustering algorithms work by first constructing a neighborhood graph and then trying to infer some sort of connected components in that graph



Graph based clustering

- We need to clarify how to perform the following three steps:
 - 1. construct the neighborhood graph
 - 2. assign weights to the edges (similarity)
 - 3. partition the nodes using the graph structure

Example



Clustering methods: Comparison

	Bottom up	Model based	Top down
Running time	naively, O(n³)	fast (each iteration is linear)	could be slow (matrix transformation)
Assumptions	requires a similarity / distance measure	strong assumptions	general (except for graph structure)
Input parameters	none	k (number of clusters)	either <i>k</i> or distance threshold
Clusters	subjective (only a tree is returned)	exactly k clusters	depends on the input format

Cluster validation

- We wish to determine whether the clusters are real
 - internal validation (stability, coherence)
 - external validation (match to known categories)

Internal validation: Coherence

- A simple method is to compare clustering algorithm based on the coherence of their results
- We compute the average inter-cluster similarity and the average intra-cluster similarity
- Requires the definition of the similarity / distance metric

Internal validation: Stability

- If the clusters capture real structure in the data they should be stable to minor perturbation (e.g., subsampling) of the data.
- To characterize stability we need a measure of similarity between any two k-clusterings.
- For any set of clusters C we define L(C) as the matrix of 0/1 labels such that L(C)_{ij} =1 if genes i and j belong to the same cluster and zero otherwise.
- We can compare any two k clusterings C and C' by comparing the corresponding label matrices L(C) and L(C').

Internal validation

 We can compare any two k clusterings C and C' by comparing the corresponding label matrices L(C) and L(C'). For example, we can define their similarity as

$$Sim(L(C), L(C')) = \frac{N(1,1)}{N(1,1) + N(1,0) + N(0,1)}$$

where N(s,r) is the number of matrix elements (pairs of genes) such that the label in one clustering is s $(L(C)_{ij}=s)$ and r in the other $(L(C')_{ii}r)$.

Note that this method is independent of the similarity metric used

Validation by subsampling

- C is the set of k clusters based on all the gene profiles
- C' denotes the set of k clusters resulting from a randomly chosen subset (80-90\%) of genes
- We have high confidence in the original clustering if Sim(L(C),L(C'))
 approaches 1 with high probability, where the comparison is done
 over the genes common to both
- Another way to do this?

External validation

- More common (why?).
- Suppose we have generated k clusters (sets of gene profiles)
 C₁,...,C_k. How do we assess the significance of their relation to m known (potentially overlapping) categories G₁,...,G_m?
- Let's start by comparing a single cluster C with a single category
 G_j. The p-value for such a match is based on the hyper-geometric
 distribution.
- Board.
- This is the probability that a randomly chosen |C_i| elements out of n would have I elements in common with G_j.

P-value (cont.)

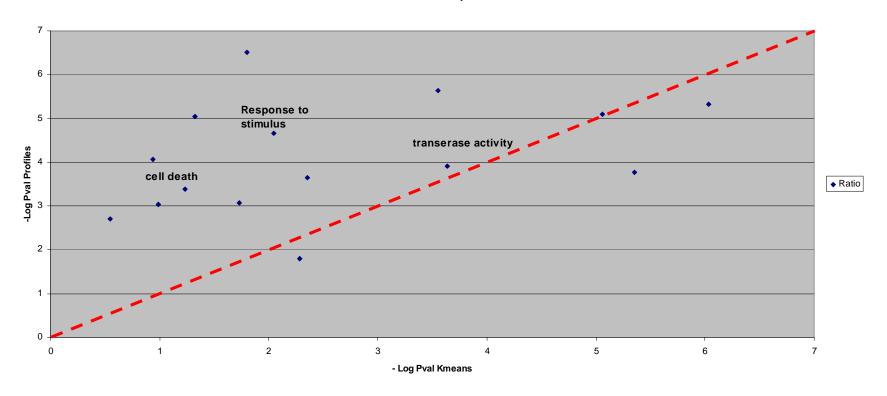
If the observed overlap between the sets (cluster and category) is I elements (genes), then the p-value is

$$p = prob(l \ge \hat{l}) = \sum_{j=l}^{\min(c,m)} prob(exactly - j - matches)$$

- Since the categories G₁,...,G_m typically overlap we cannot assume that each cluster-category pair represents an independent comparison
- In addition, we have to account for the multiple hypothesis we are testing.
- Solution ?

External validation: Example

P-value comparison



What you should know

- Why is clustering useful
- What are the different types of clustering algorithms
- What are the assumptions we are making for each, and what can we get from them
- Cluster validation: Internal and external