

What is Clustering?

Given n objects, assign them to groups (clusters) based on their similarity

- Unsupervised Machine Learning
- Class Discovery
- Difficult, and maybe ill-posed problem!

Cluster These ...



**The Real Clusters Are in
the Eye of the Beholder**

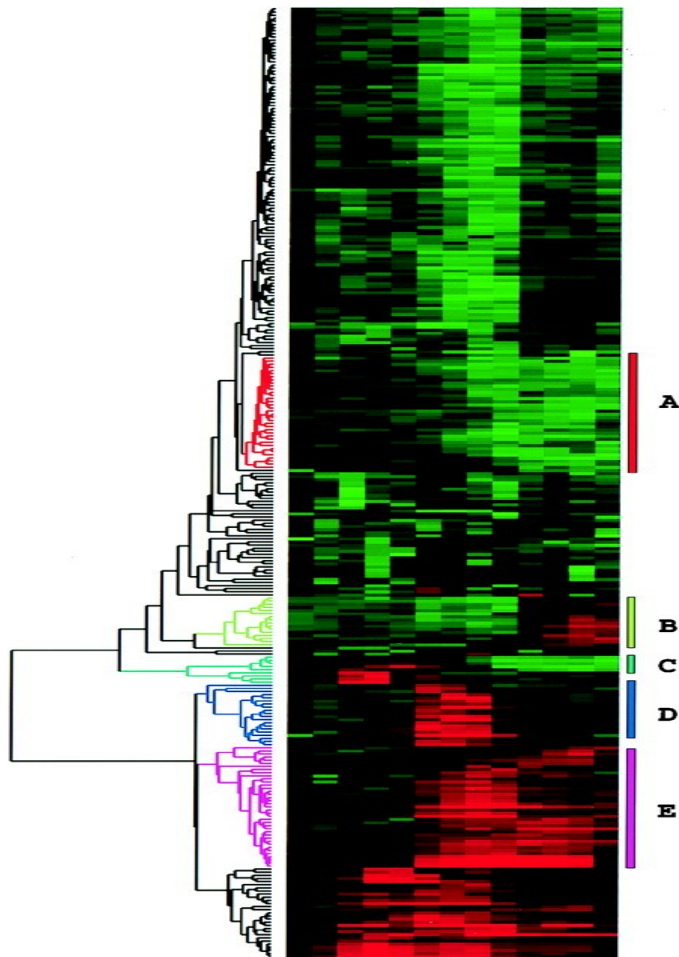
Impossibility of Clustering

- Scale-invariance: meters vs inches
- Richness: all partitions as possible solutions
- Consistency: increasing distances between clusters and decreasing distances within clusters should yield the same solution

No function exists that satisfies all three.

Kleinberg, NIPS 2002

Clustering Microarray Data

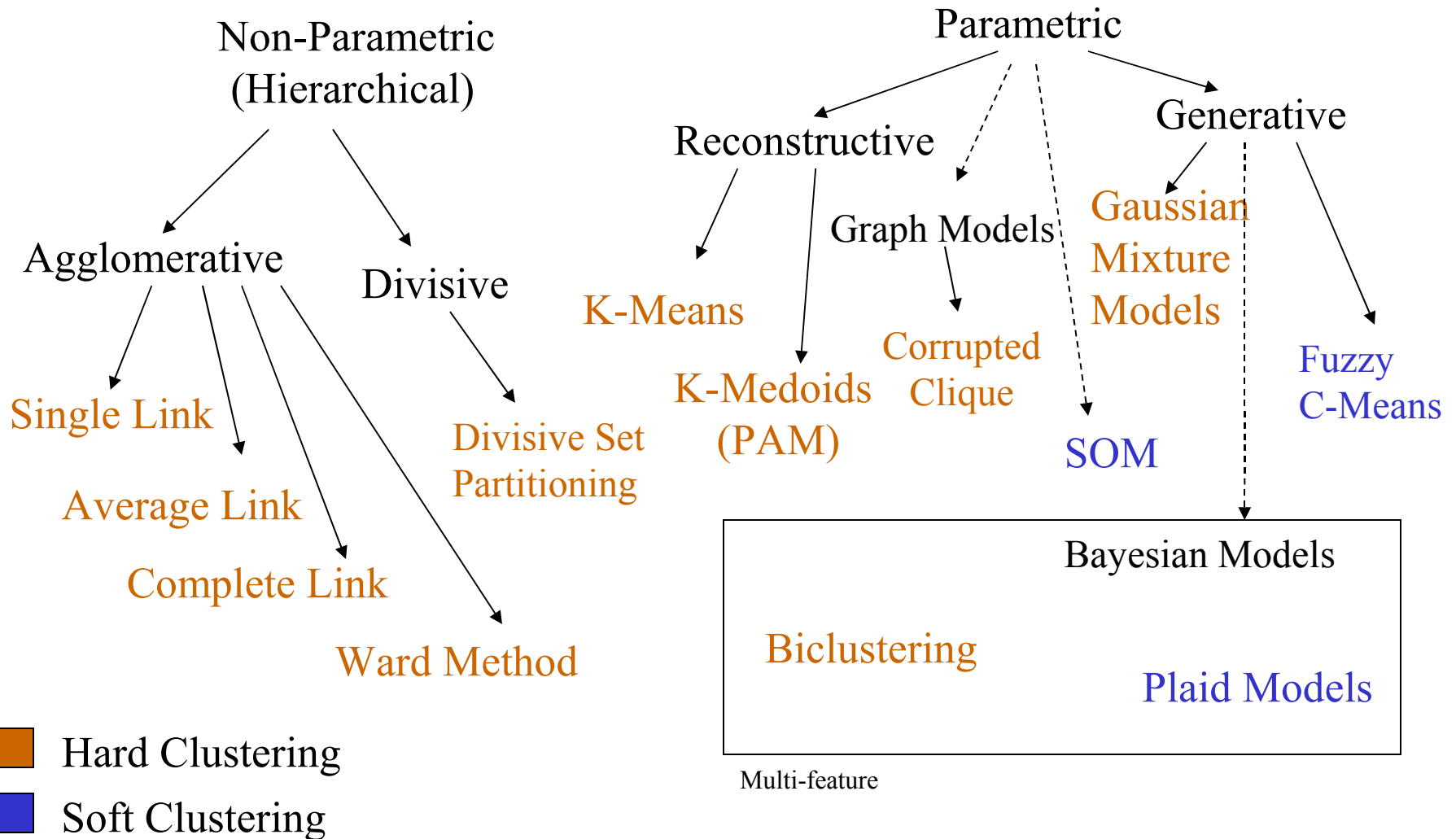


Clustering reveals similar expression patterns, in particular in time-series expression data

Guilt-by-association: a gene of unknown function has the same function as a similarly expressed gene of known function

Genes of similar expression might be similarly regulated

Clustering Approaches



How To Choose the Right Clustering?

- Data Type
 - Independent Experiments (e.g. knockouts)
 - Dependent experiments (e.g. time series)
- Parametric vs. non-parametric clustering
- Quality of Clustering
- Software Availability
- Features of the Methods
 - Computing averages (sometimes impossible or too slow)
 - Stability analysis
 - Properties of the clusters
 - Speed
 - Memory

Clustering Meta-Procedure

1. Compare the similarity of all pairs of objects
2. Group the most similar ones together into clusters
3. Reason about the resulting groups of clusters

Distance Measures, $d(x,y)$

Certain properties are expected from distance measures

1. $d(x,y)=0$
 - $d(x,y)>0, x \neq y$
1. $d(x,y)=d(y,x)$
 - $d(x,y) \leq d(x,z)+d(z,y)$ the triangle inequality

If properties 1-4 are satisfied, the distance measure is a metric

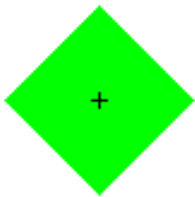
The L_p norm

$$d(x, y) = \sqrt[p]{|x_1 - y_1|^p + \dots + |x_n - y_n|^p}$$

$p = 2$, Euclidean Dist.

$p = \infty$, Manhattan Dist.(downtown Davis distance)

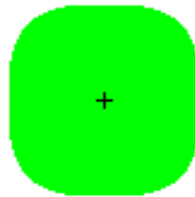
Equidistant points from a center, for different norms



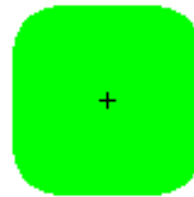
$p=1$



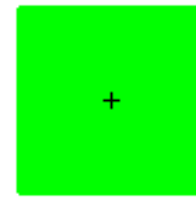
$p=2$



$p=3$



$p=4$



$p=20$

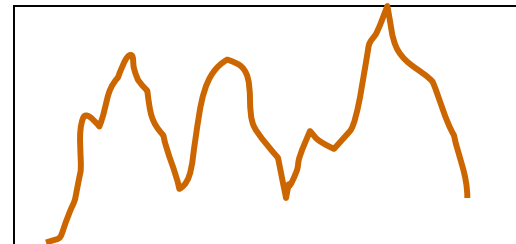
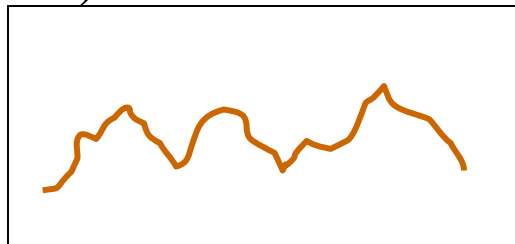
Pearson Correlation Coefficient

(Normalized vector dot product)

$$r(x, y) = \frac{\sum_k x_k y_k - \frac{\sum_k x_k \sum_k y_k}{n}}{\sqrt{\left(\sum_k x_k^2 - \frac{(\sum_k x_k)^2}{n}\right) \left(\sum_k y_k^2 - \frac{(\sum_k y_k)^2}{n}\right)}}$$

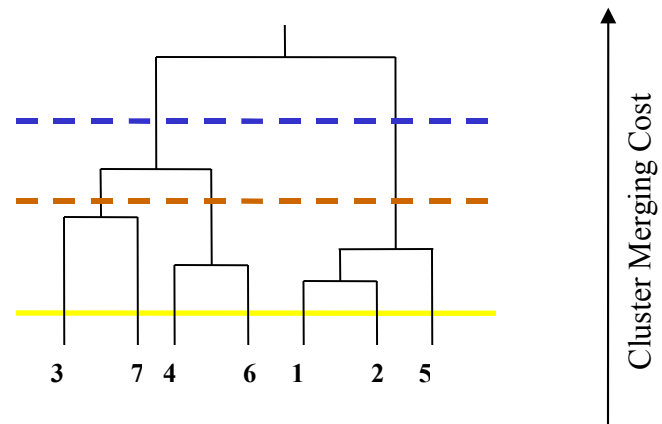
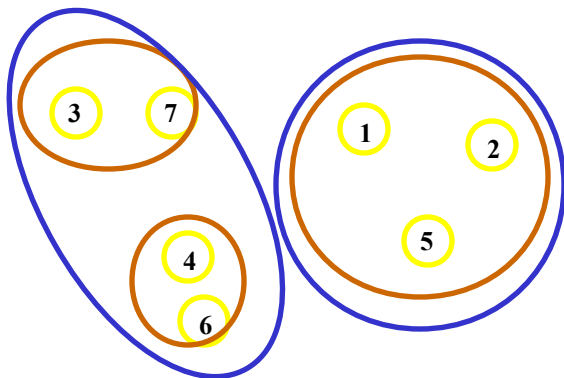
Not a metric!

Good for comparing expression profiles because it is insensitive to scaling (but data should be normally distributed, e.g. log expression)!



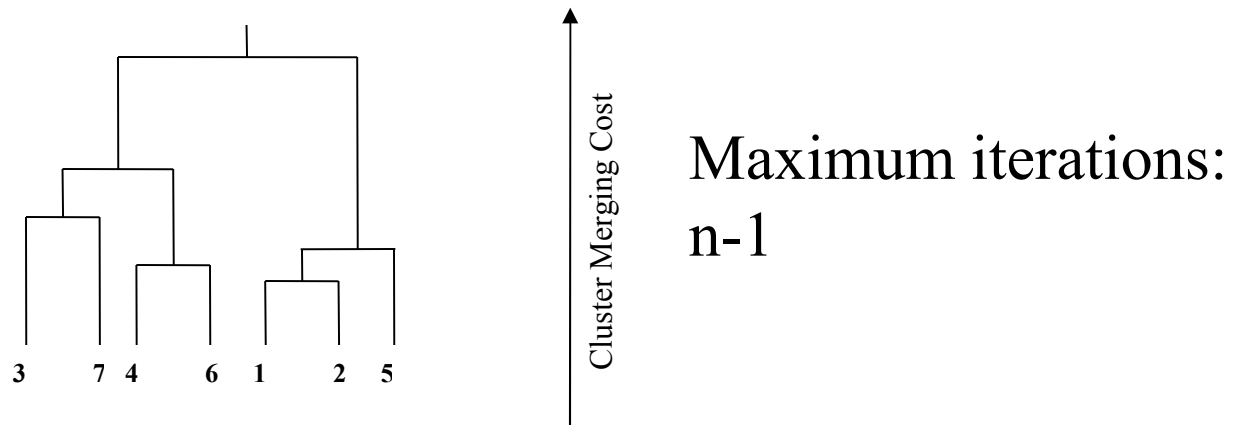
Hierarchical Clustering

- Input: Data Points, x_1, x_2, \dots, x_n
- Output: Tree
 - the data points are leaves
 - Branching points indicate similarity between sub-trees
 - Horizontal cut in the tree produces data clusters



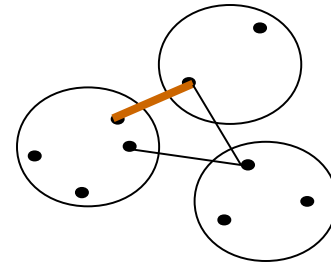
General Algorithm

- Place each element in its own cluster, $C_i = \{x_i\}$
 - Compute (update) the merging cost between every pair of elements in *the set of clusters* to find the two cheapest to merge clusters C_i, C_j ,
 - Merge C_i and C_j in a new cluster C_{ij} which will be the parent of C_i and C_j in the result tree.
1. Go to (2) until there is only one set remaining

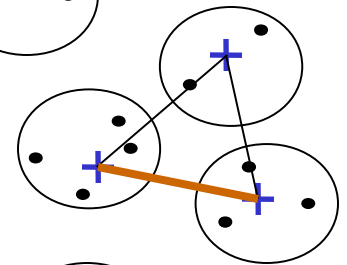


Different Types of Algorithms Based on The Merging Cost

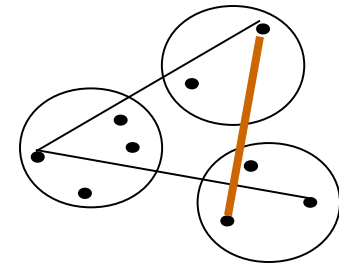
- Single Link, $\min_{x \in C_i, y \in C_j} d(x, y)$



- Average Link, $\frac{1}{|C_i| |C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$



- Complete Link, $\max_{x \in C_i, y \in C_j} d(x, y)$

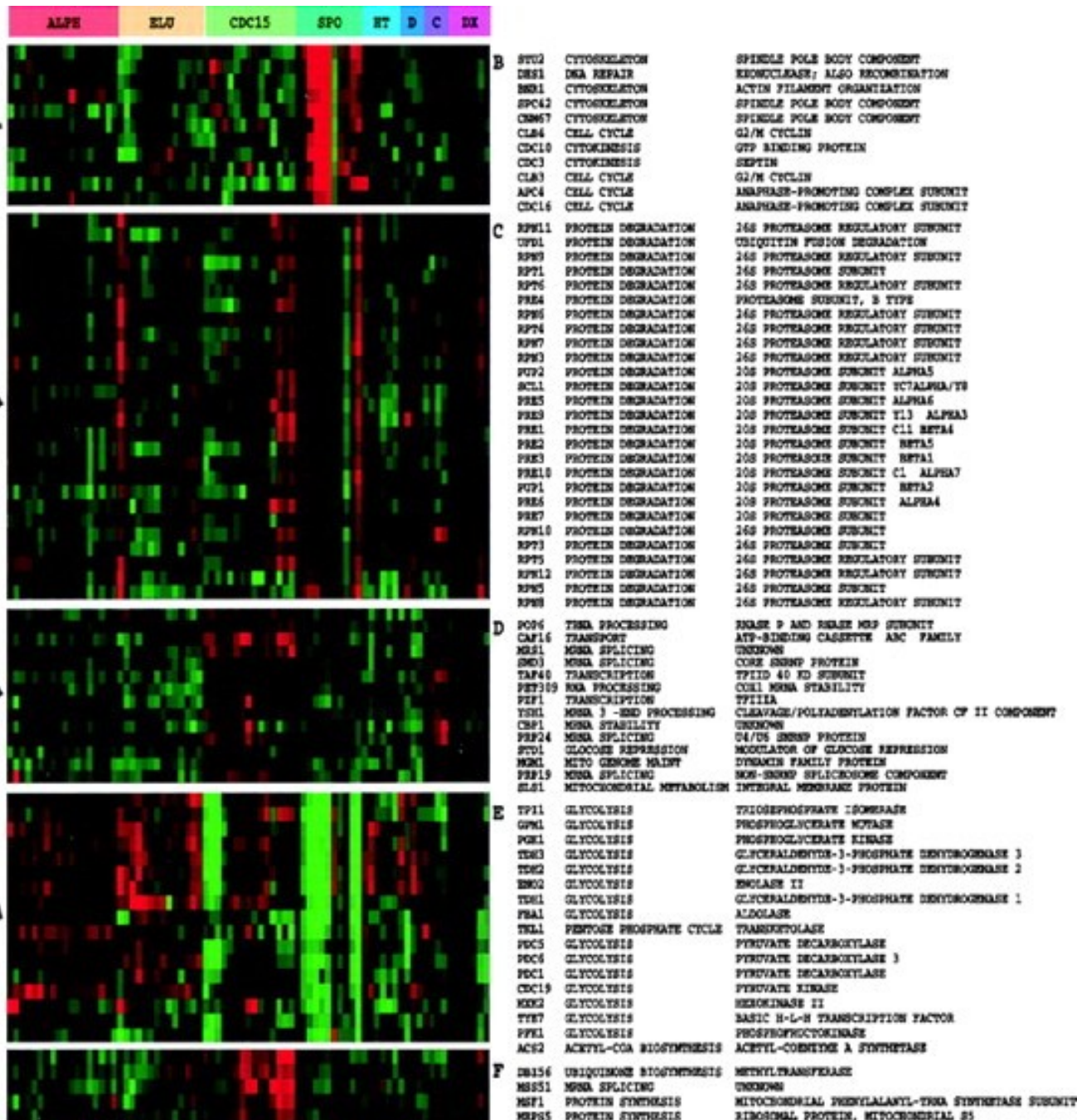
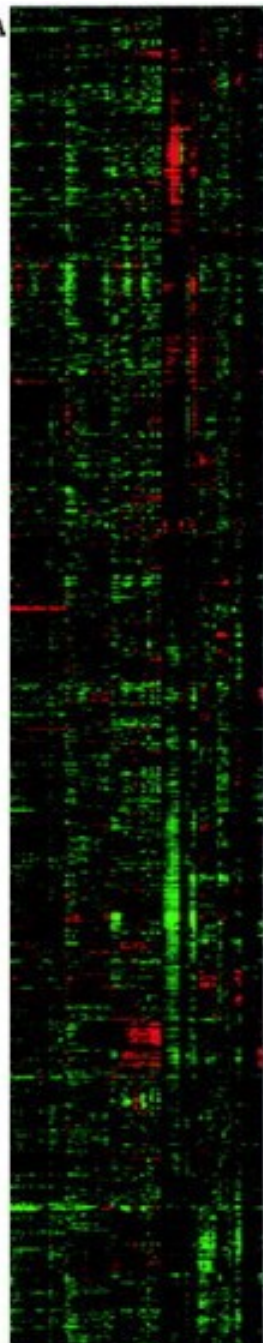


- Others (Ward method-least squares)

Characteristics of Hierarchical Clustering

- Greedy Algorithms – suffer from local optima, and build a few big clusters
- A lot of guesswork involved:
 - Number of clusters
 - Cutoff coefficient
 - Size of clusters
- Average Link is fast and not too bad: biologically meaningful clusters are retrieved

A



K-Means

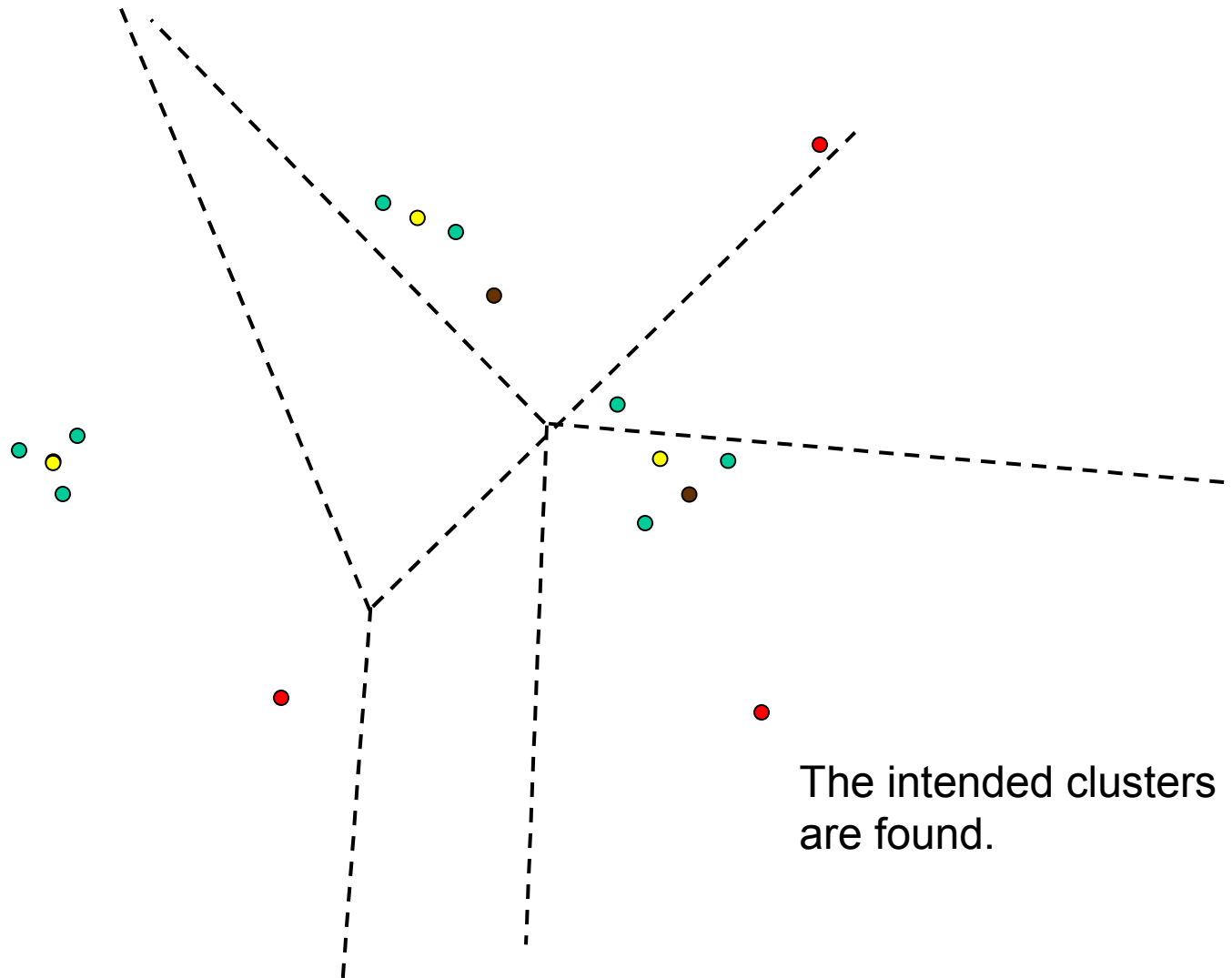
Input: Data Points, Number of Clusters (K)

Output: K clusters

Algorithm: Starting from k -centroids assign data points to them based on proximity, updating the centroids iteratively

- Select K initial cluster centroids, $c_1, c_2, c_3, \dots, c_k$
- Assign each element x to nearest centroid
 1. For each cluster, re-compute its centroid by averaging the data points in it
 2. Go to (2) until convergence is achieved

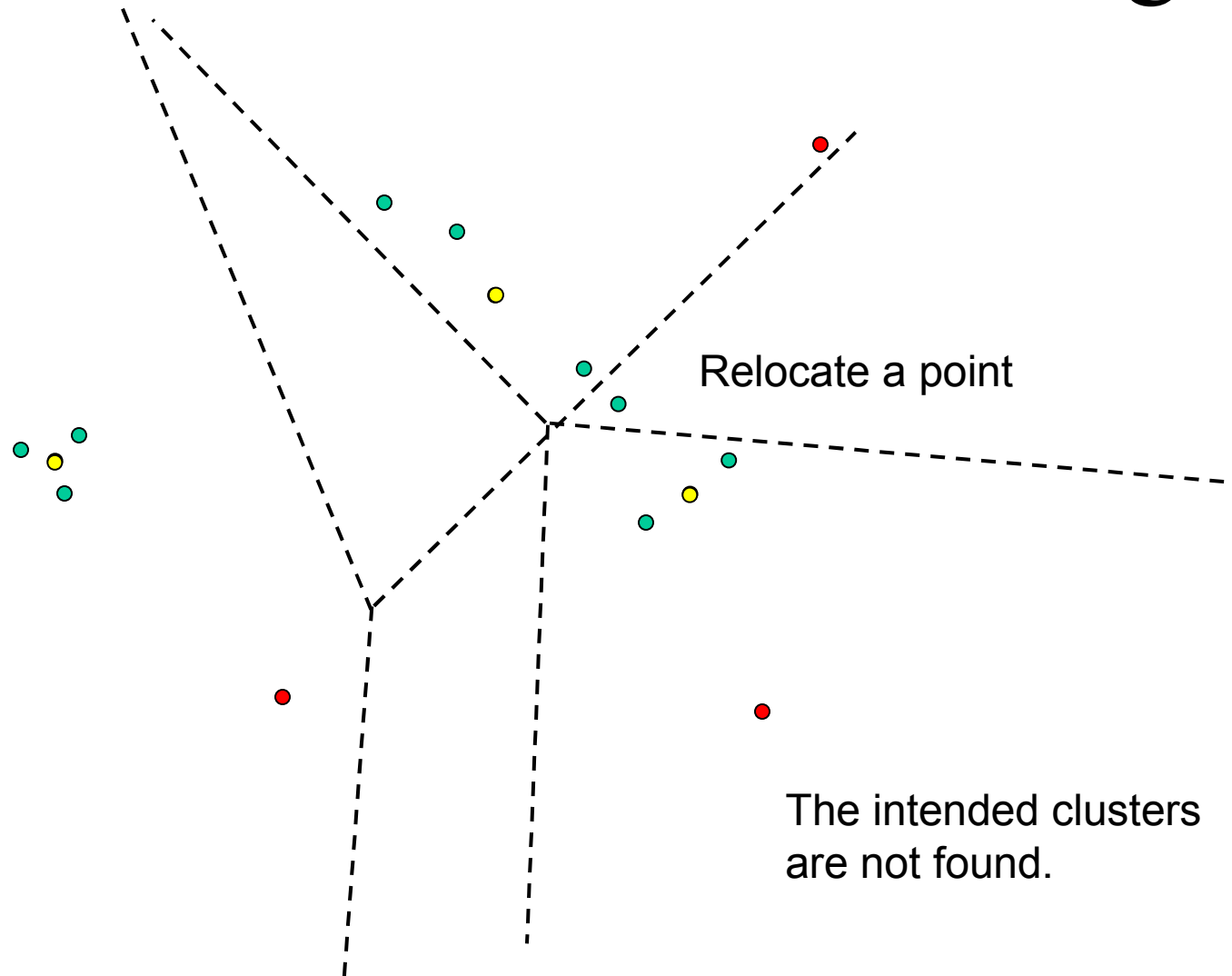
K-means Clustering



K-Means Properties

- Must know the number of clusters before hand
- Sensitive to perturbations
- Clusters formed ad hoc with no indication of relationships among them
- Results depend on initial choice for centers
- In general, better than average link clustering

Properties of K-means Clustering



Self Organizing Maps Clustering

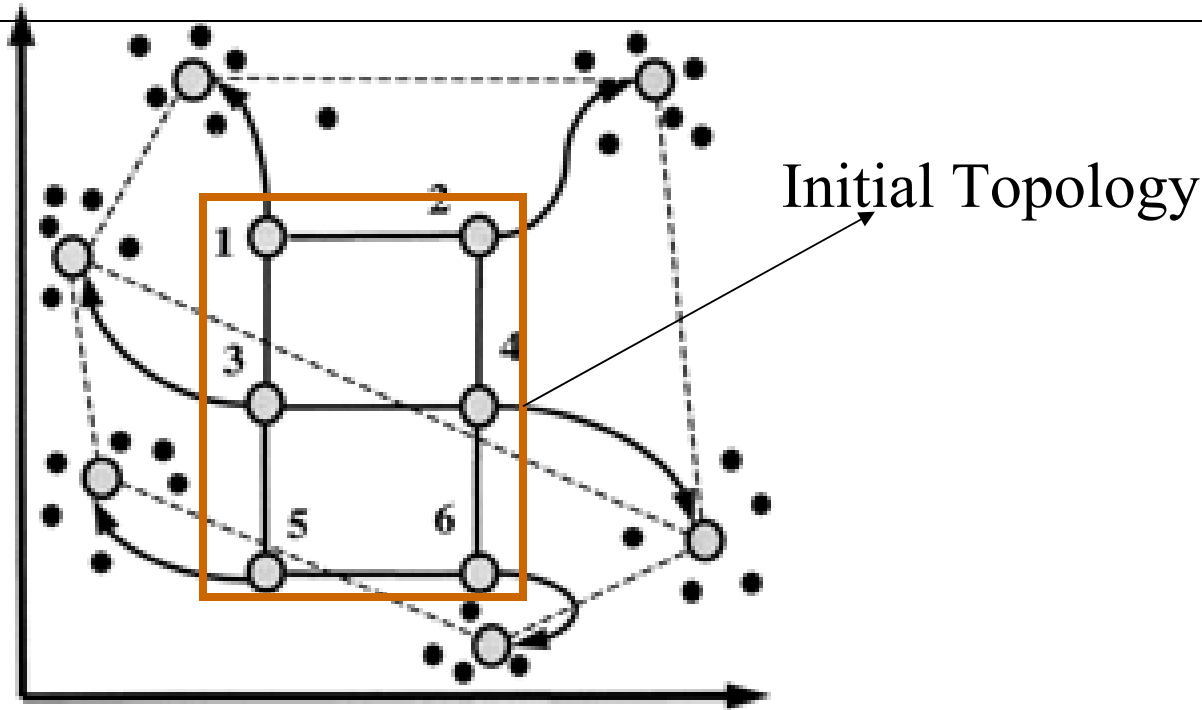
Input: Data Points, SOM Topology (K nodes and a distance function)

Output: K clusters, (near clusters are similar)

Algorithm: Starting with a simple topology (connected nodes) iteratively move the nodes “closer” to the data

- 1.
- 2.
- 3.
- 4.

$$f_{i+1}(N) = f_i(N) + \tau (d(N, N_p), i)(P - f_i(N))$$



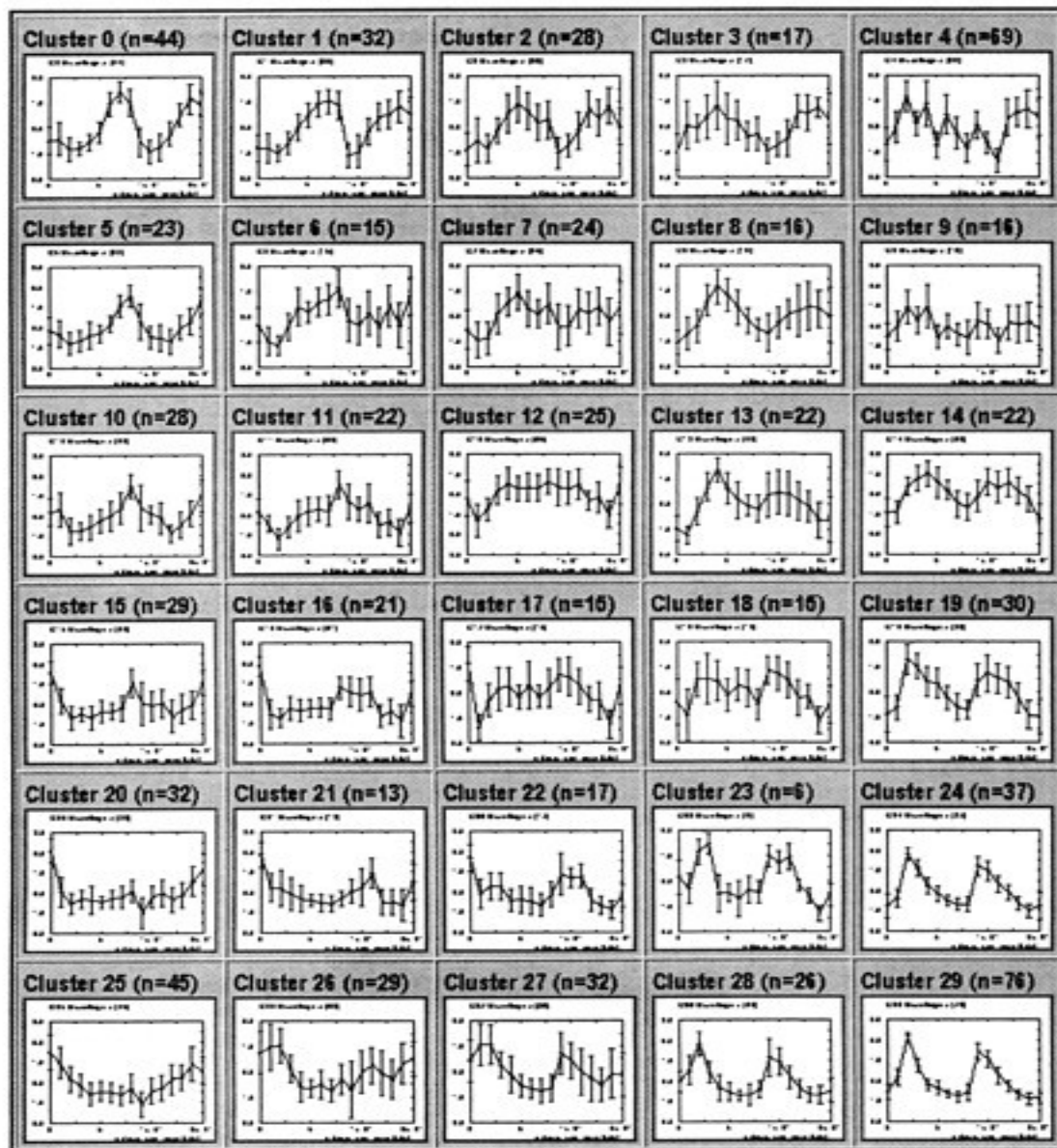
N = Node; P = Random point P ; N_p = Node closest to P

$d(N, N_p)$ = Distance between N and N_p

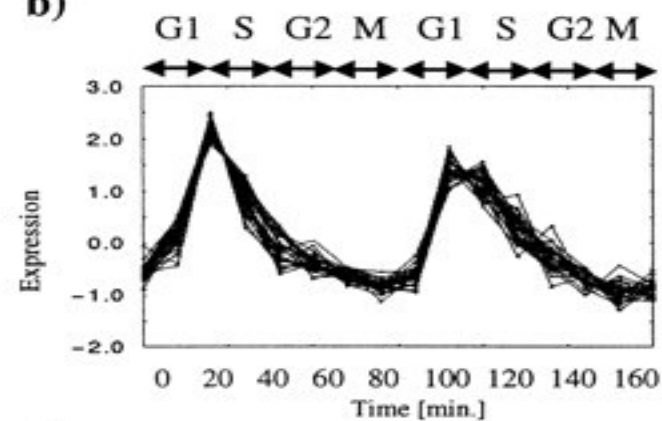
$f_i(N)$ = Position of node N at iteration i

τ is the learning rate (decreases with d and I)

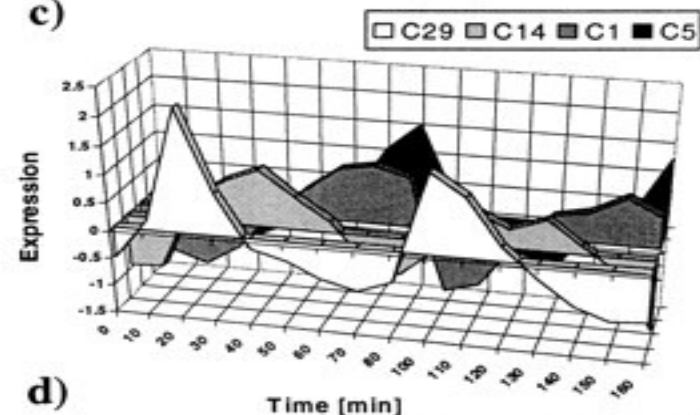
a)



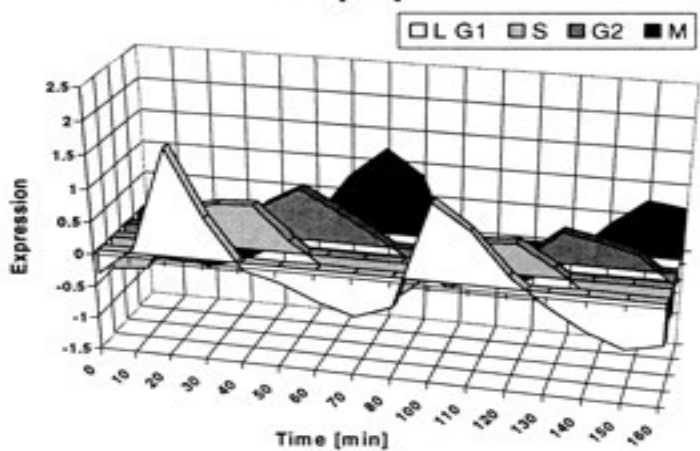
b)



c)



d)

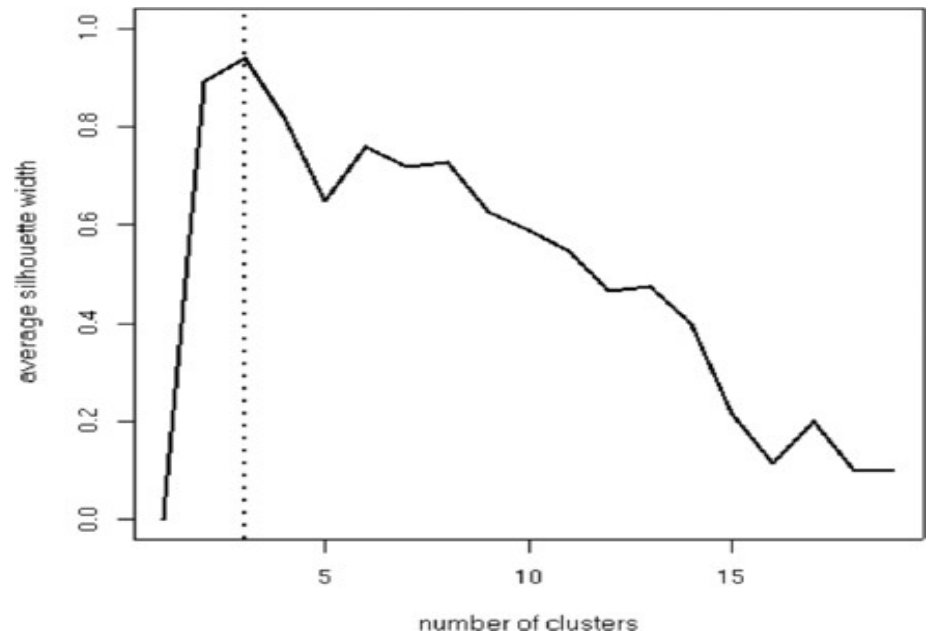
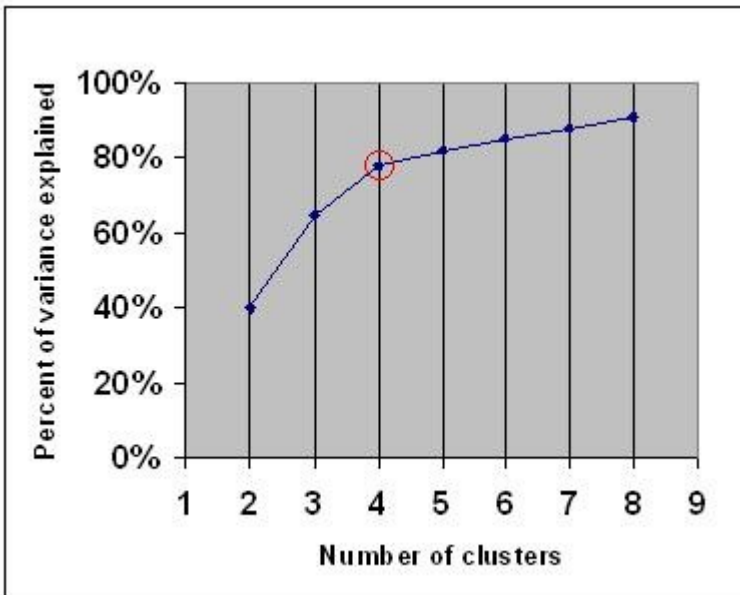


SOM Properties

- Neighbouring clusters are similar
- Element on the borders belong to both clusters
- Very robust
- Works for short profile data too

What if the number of clusters is not known?

- Elbow criterion: look for a clustering that explains most of the variance or stability in data with the fewest clusters
- Information theoretic: maximize (or minimize) some Information Criterion (like BIC or AIC or MDL)
- Within/between cluster distance/separation: silhouettes

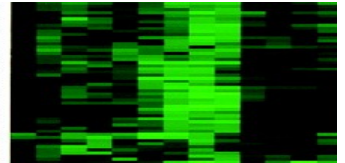


Note on Missing Values

- Microarray experiments often have missing values, as a result of experimental error, values out of bound, spot reading error, batch errors, etc.
- Many clustering algorithms (all of the ones presented here) are sensitive to missing data
- Filling in the holes:
 - All 0s
 - Average
 - Better: weighted K-nearest neighbor, or SVD based methods (SVDimpute, KNNimpute) Troyanskaya et al. 2000 (AVAILABLE FOR DOWNLOAD)
 - Robust
 - Do better than average

Cluster Visualization

- How to “see” the clusters effectively?
- Present gene expressions in different colors
- Plot similar genes close to each other
- R
- GeneXPress
- Expander
- CytoScape



Algorithm Comparison and Cluster Validation

- Paper: Chen et al. 2001
- Data: embryonic stem cells expression data
- Results: evaluated advantages and weaknesses of algorithms w/respect to both internal and external quality measures
- Used known and developed novel indices to measure clustering efficacy

Algorithms Compared

- Average Link Hierarchical Clustering,
- K-Means and PAM , and
- SOM, two different neighborhood radii
 - $R=0$ (theoretically approaches K-Means)
 - $R=1$
- Compared them for different numbers of clusters

Clustering Quality Indices

- Homogeneity and Separation
 - Homogeneity is calculated as the average distance between each gene expression profile and the center of the cluster it belongs to
 - Separation is calculated as the weighted average distance between cluster centers
 - H reflects the compactness of the clusters while S reflects the overall distance between clusters
 - Decreasing H or increasing S suggest an improvement in the clustering results

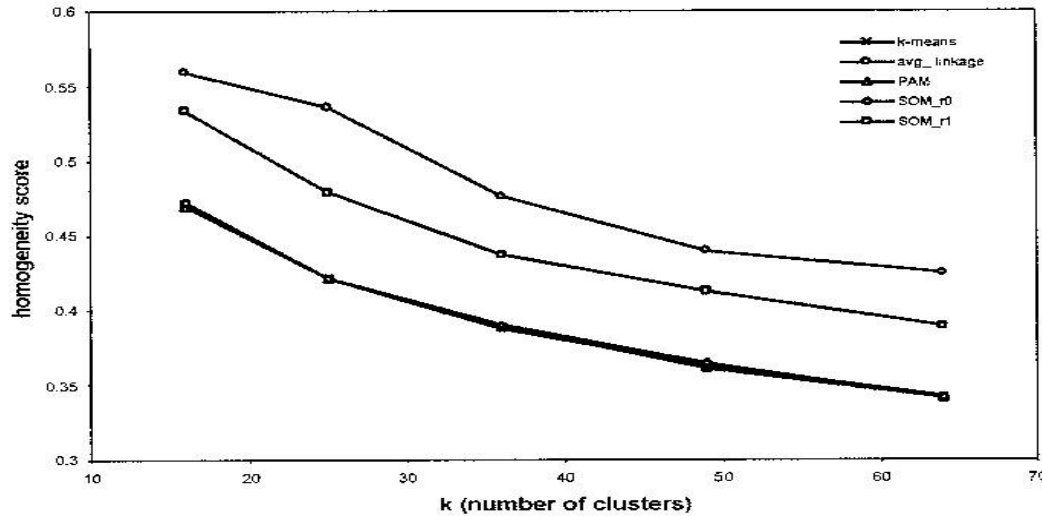


Figure 1a Comparing homogeneity scores among different algorithms

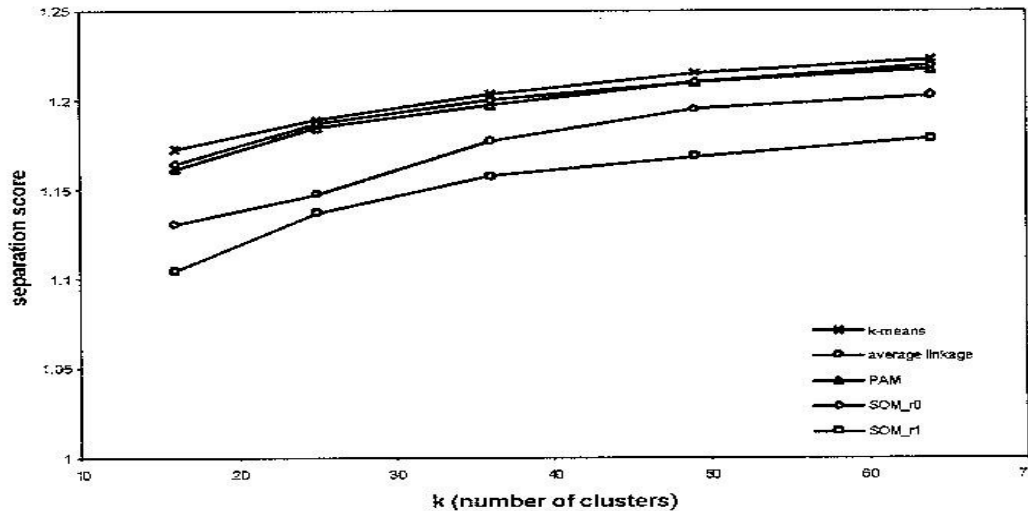


Figure 1b Comparing separation scores among different algorithms

Results:

- K-Means and PAM scored identically
- SOM_r0 very close to both above
- All three beat ALHC
- SOM_r1 worst

- Silhouette Width

—

—

•

•

•

•

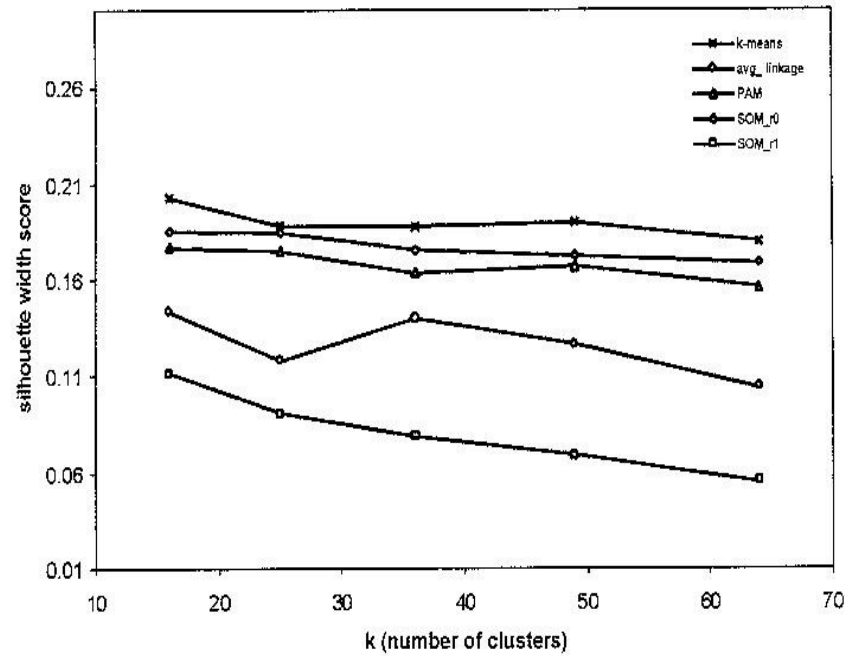


Figure 2 Comparison of average silhouette width among different algorithms

- Redundant Scores (external validation)

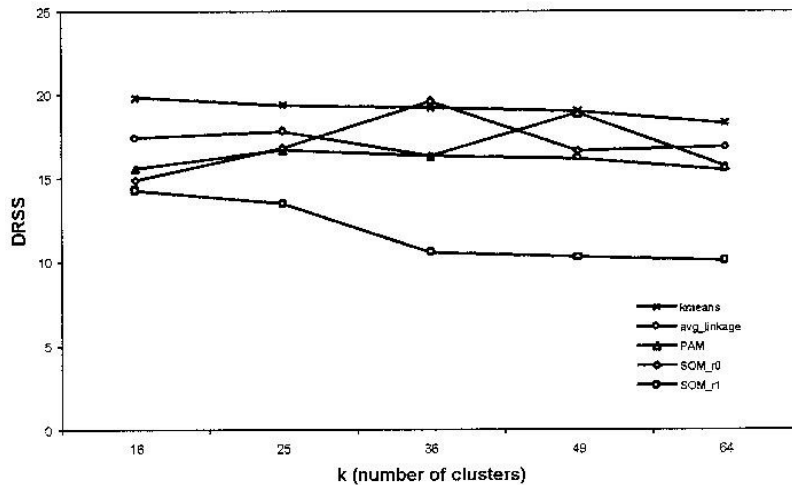


Figure 3 Comparison of DRSS among different algorithms

Results:

- K-means consistently better than ALHC
- PAM and SOM_r0 close to the above
- SOM_r1 was consistently the worst

- WADP – Measure of Robustness

-

-

-

-

-

-

-

Results:

- SOM_r1 clusters are the most robust of all
- K-means and ALHC were high through all cluster numbers
- PAM and SOM_r1 were better for small number of clusters

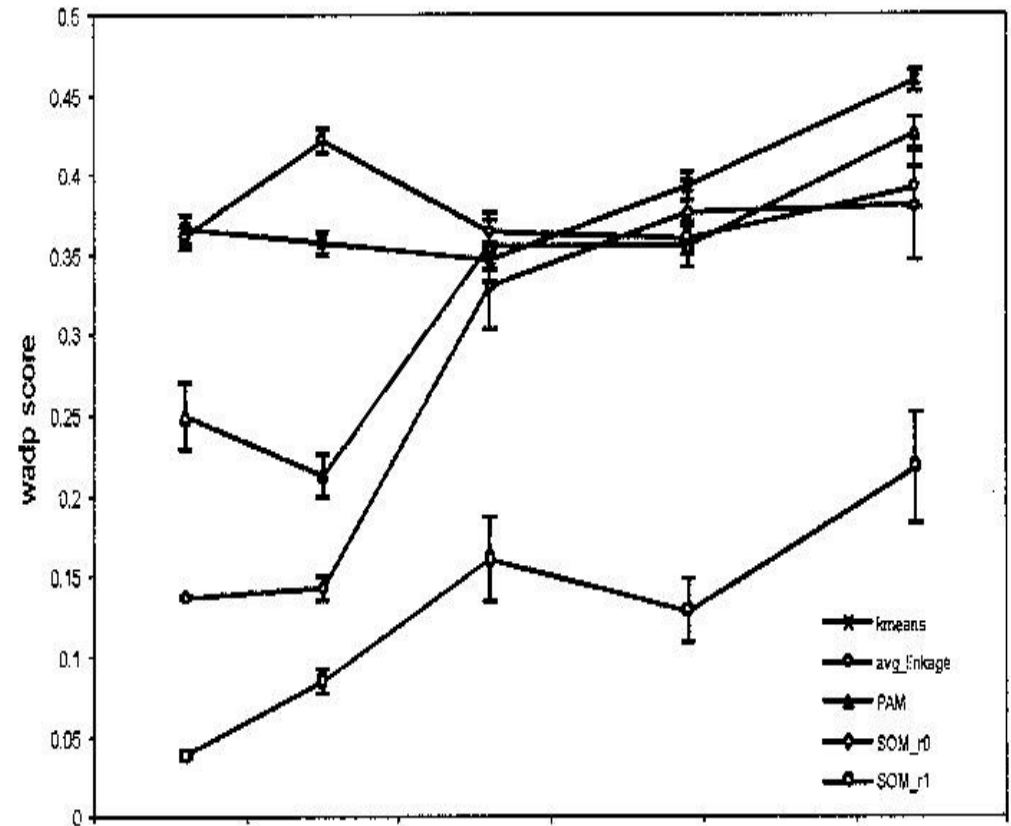
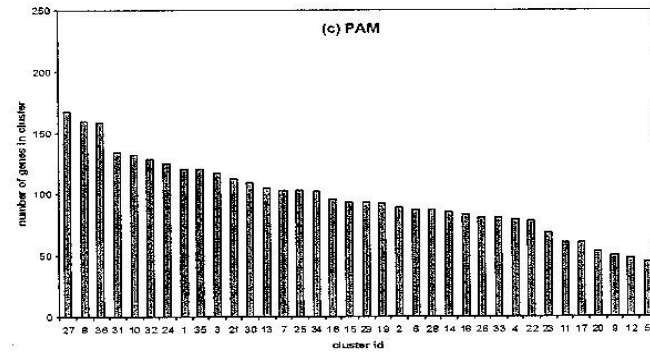
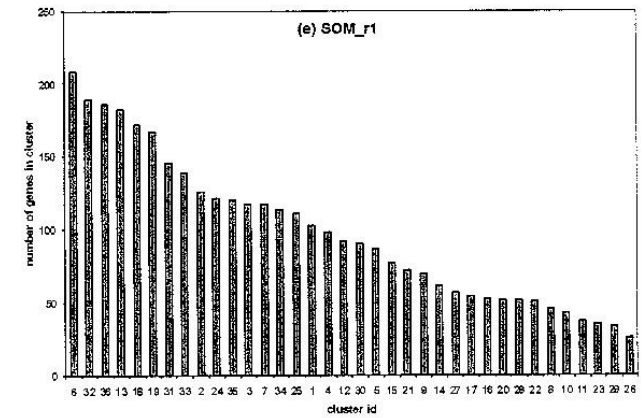
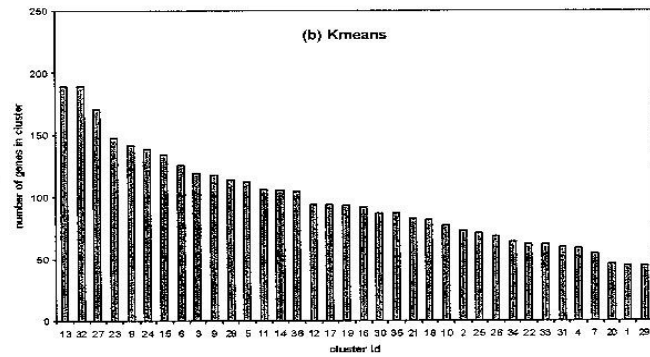
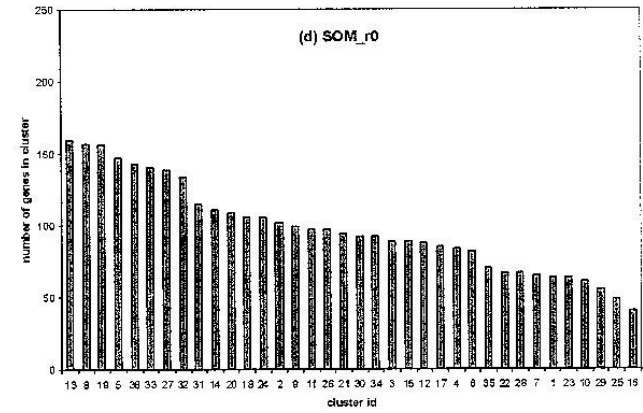
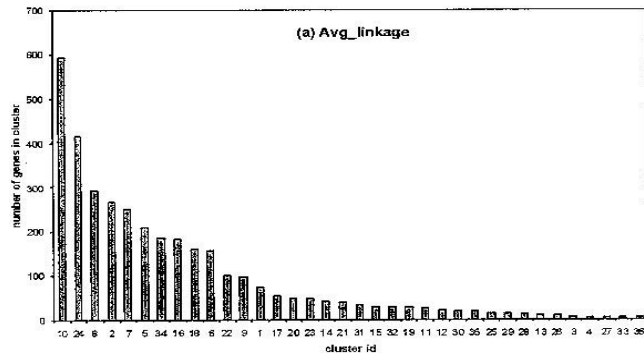


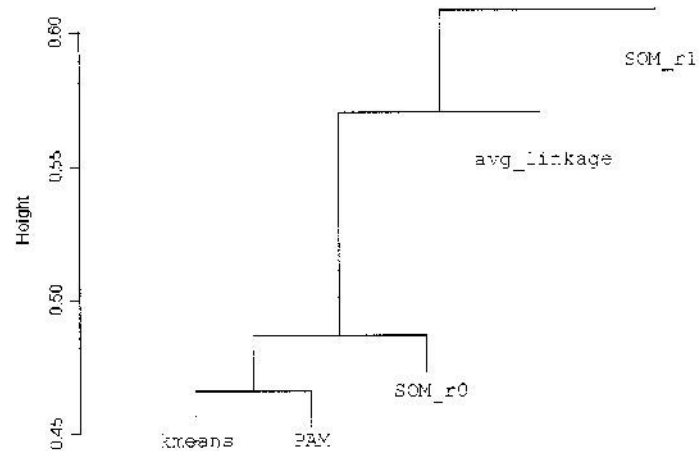
Figure 4 Comparison of WADP scores among different algorithms

Comparison of Cluster Size and Consistency



Comparison of Cluster Content

- How similar are two clusterings in all the methods?
 - WADP



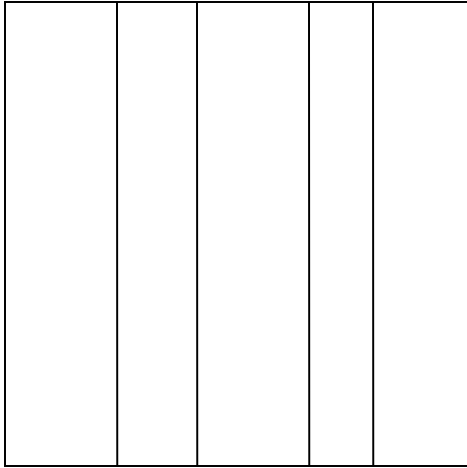
- Other measures of similarity based on co-clusteredness of elements
 - Rand index
 - Adjusted Rand
 - Jaccard

Clustering: Conclusions

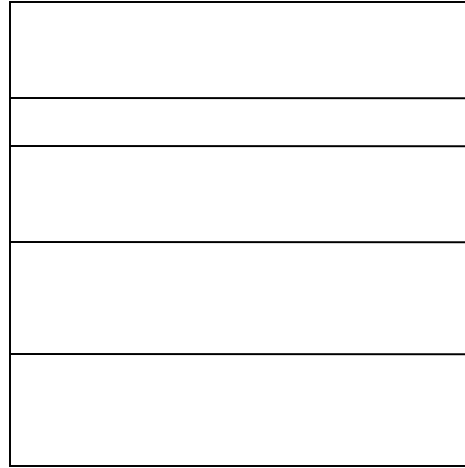
- K-means outperforms ALHC
- SOM_r0 is almost K-means and PAM
- Tradeoff between robustness and cluster quality: SOM_r1 vs SOM_r0, based on the topological neighborhood
- When should we use which? Depends on what we know about the data
 - Hierarchical data – ALHC
 - Cannot compute mean – PAM
 - General quantitative data - K-Means
 - Need for robustness – SOM_r1
 - Soft clustering: Fuzzy C-Means
 - Clustering genes and experiments - Biclustering

Biclustering

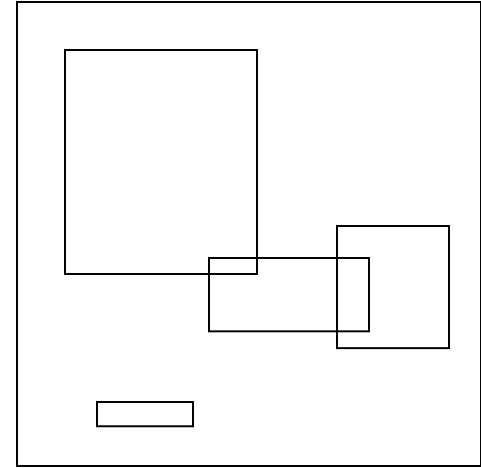
- Problem with clustering:
 - Clustering the same genes under different subsets of conditions can result in very different clusterings
- Additional Motivation
 - sometimes only subset of genes are interesting and one wants to cluster those
 - Genes expressed differentially in different conditions and pathways
- Proposed solutions: cluster simultaneously the genes and the conditions



Clustering
conditions



Clustering
Genes



Biclustering

The biclustering methods look for submatrices in the expression matrix which show coordinated differential expression of subsets of genes in subsets of conditions. The biclusters are also statistically significant.

Clustering is a **global** similarity method, while biclustering is a **local** one.

Biclustering Methods

- Biclustering
- Coupled Two-way Clustering
- Iterative Signature Algorithm
- SAMBA
- Spectral Biclustering
- Plaid Models

Other Dimension Reduction Techniques

- All (including clustering) are based on the premise that not all genes (or experiments) show different behavior, so groups of similar genes (experiments) are sought
- Principal Component Analysis
 - Identifies the underlying classes or “base” genes of the data representing most variability (best separating the genes)
 - All other genes expressions are linear combination of those
 - Classes are built around a few top “base” genes
 - Typically used for 2D or 3D data visualization and seeding k-means
- Independent Component Analysis
 - Similar as PCA but here the “base” components are required to be statistically independent
- Non-zero Matrix Factorization

Expander example...