Clustering

Lecture 6, 1/24/03

What is Clustering?

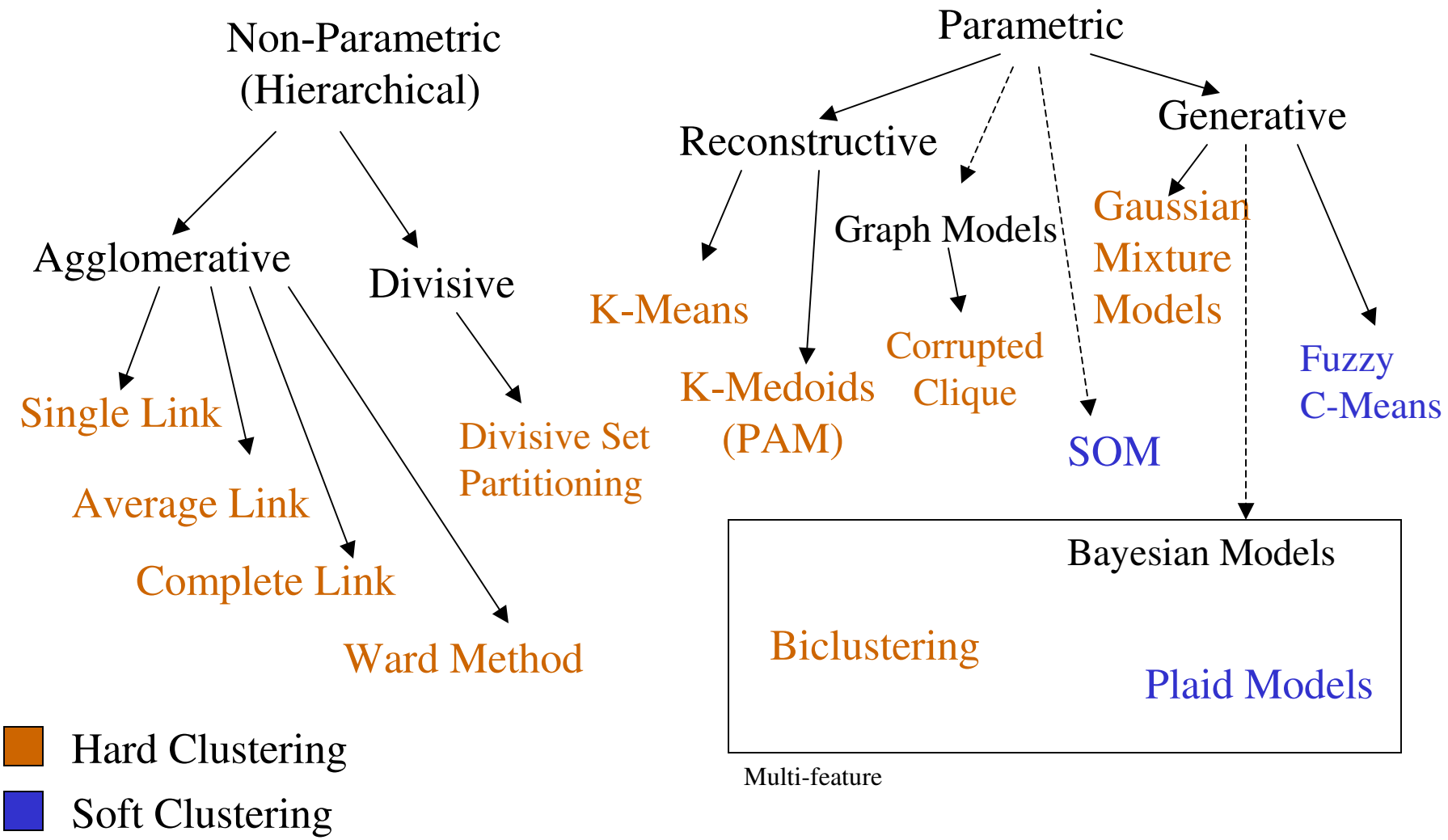
Given n objects, assign them to groups (clusters) based on their similarity

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

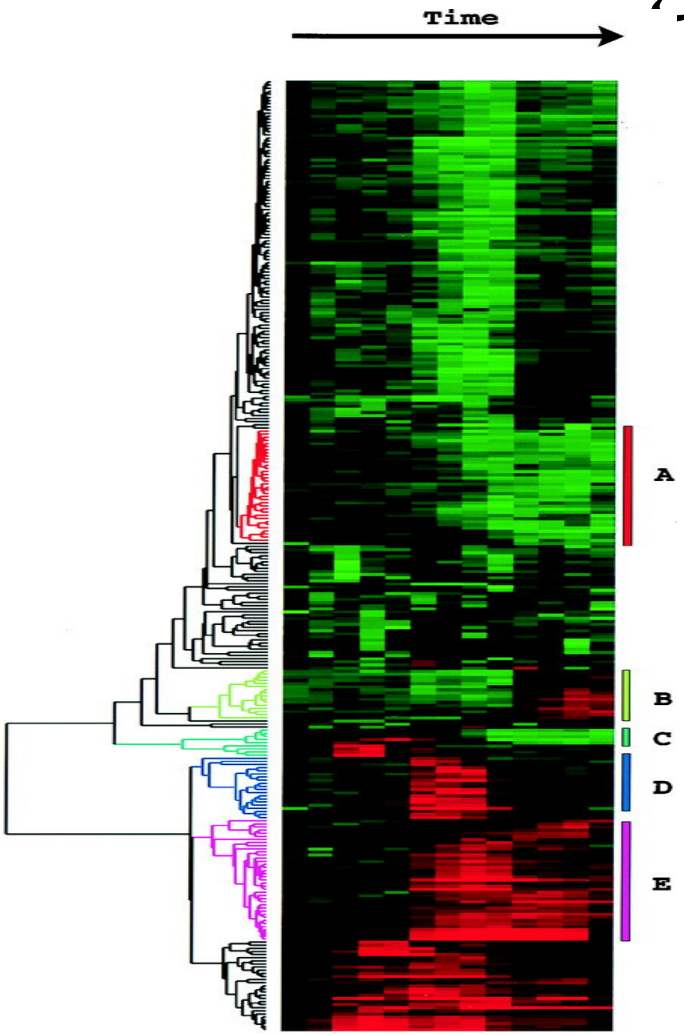
Cluster These …

The Real Clusters Are in the Eye of the Beholder

Clustering Approaches



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

How To Choose the Right Clustering?

- Data Type:
  - Single array measurement?
  - Series of experiments
- Quality of Clustering
- Code Availability
- Features of the Methods
  - Computing averages (sometimes impossible or too slow)
  - Sensitivity to Perturbation and other indices
  - Properties of the clusters
  - Speed
  - Memory

Distance Measures, *d(x,y)*

Certain properties are expected from distance measures

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

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

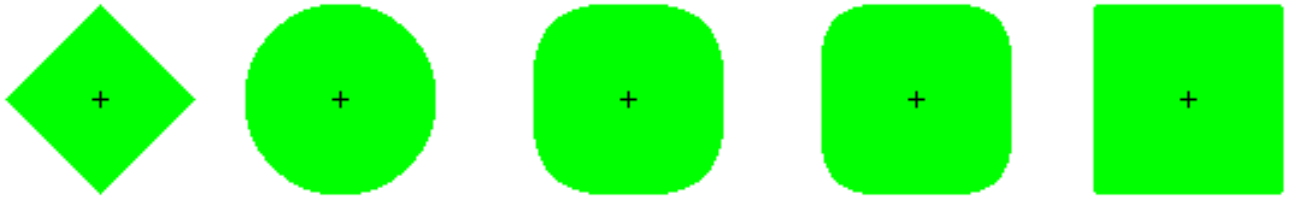
The L_p norm

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

p = 2, Euclidean Dist.

p = ∞, Manhattan Dist.(downtown Davis distance)

Equidistant points from a center, for different norms



p=1 p=2 p=3 p=4 p=20

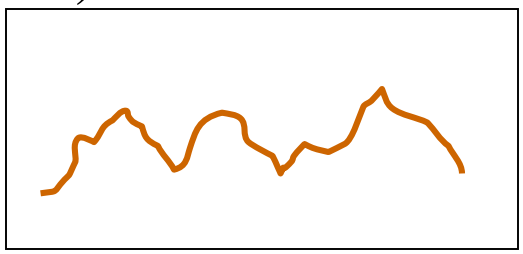
Pearson Correlation Coefficient

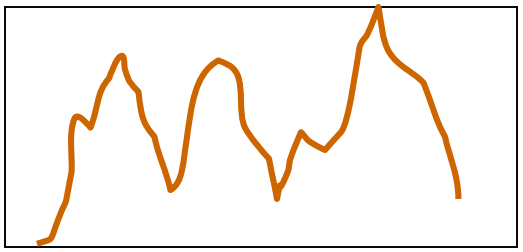
(Normalized vector dot product)

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

Not a metric!

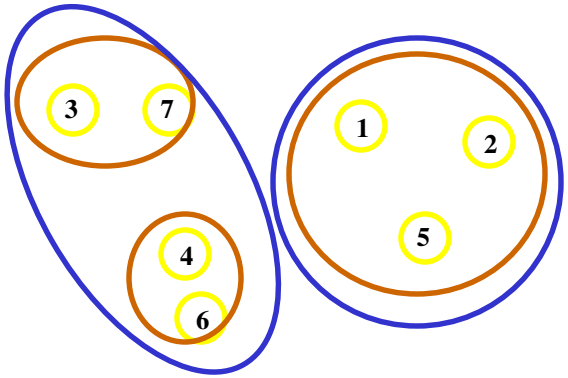
Good for comparing expression profiles because it is insensitive to scaling (but data should be normally distributed, e.g. log expression)!

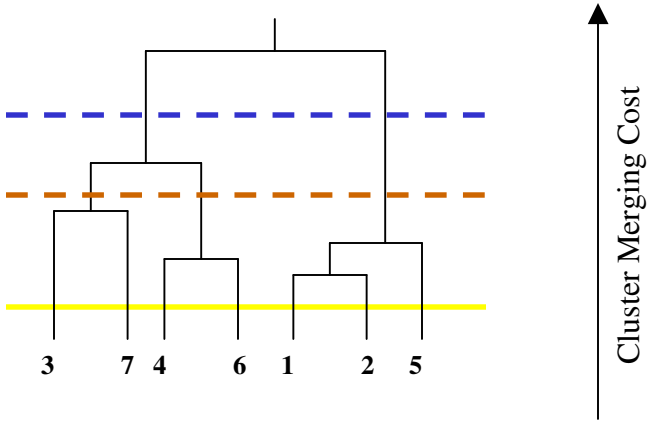




Hierarchical Clustering

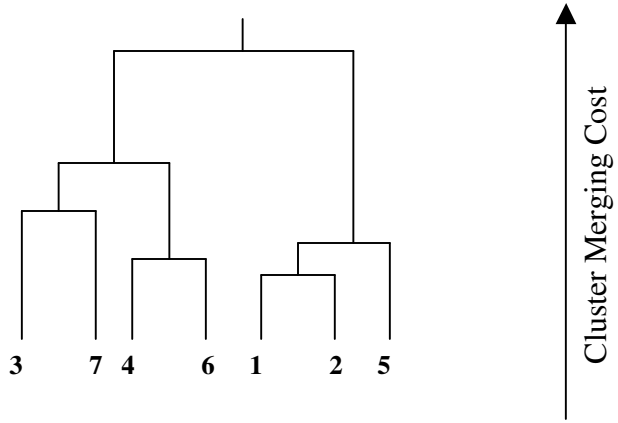
- Input: Data Points, $x_1, x_2, \ldots, x_n$
- Output:Tree
  - the data points are leaves
  - Branching points indicate similarity between sub-trees
  - Horizontal cut in the tree produces data clusters





General Algorithm

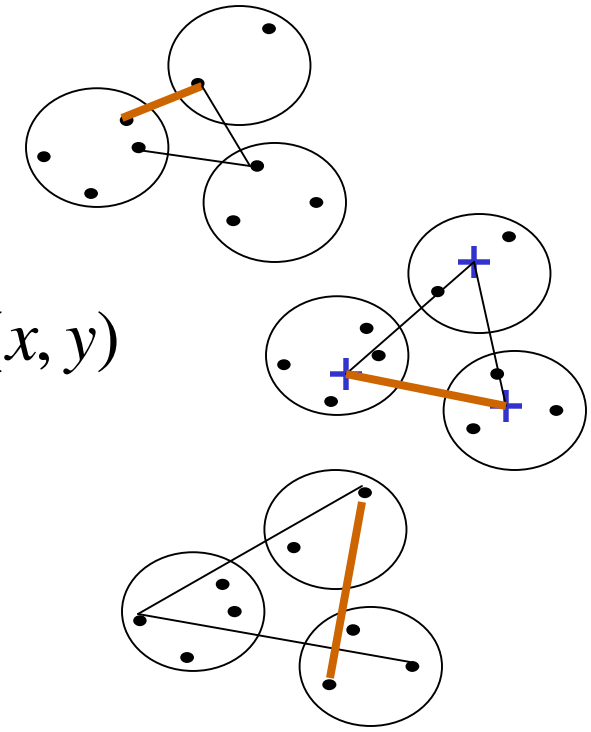
1. Place each element in its own cluster, $C_i=\{x_i\}$
2. Compute (update) the merging cost between every pair of elements in *the set of clusters* to find the two cheapest to merge clusters C_i, C_j,
3. Merge C_i and C_j in a new cluster C_{ij} which will be the parent of C_i and C_j in the result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



Maximum iterations: n-1

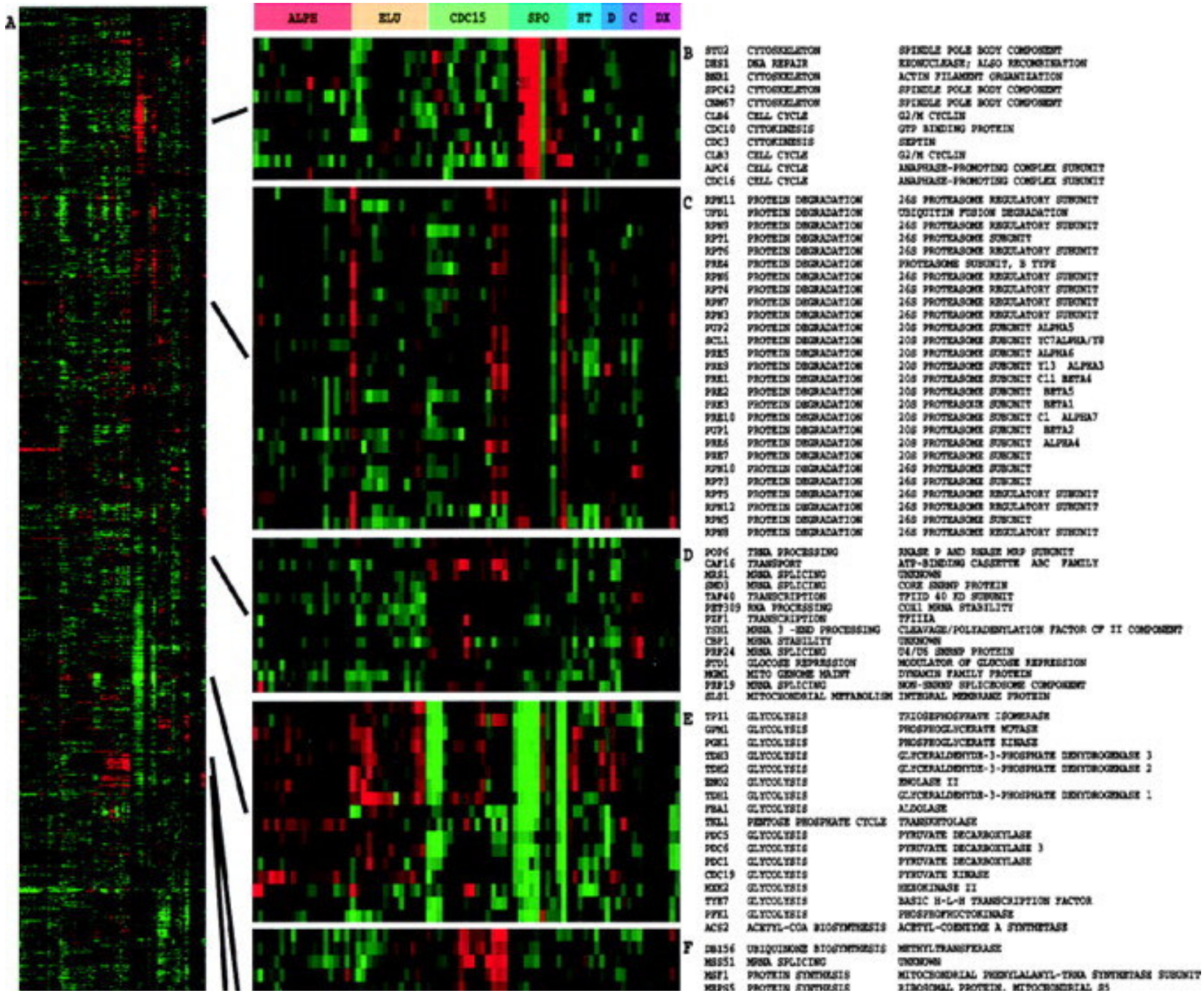
Different Types of Algorithms Based on The Merging Cost

- Single Link, $\min_{x \in C_i, y \in C_j} d(x, y)$
- Average Link, $\frac{1}{|Ci||Cj|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$
- 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
ALPH
ELU
CDC15
SPO
HT
D
C
DX
B
STU2 CYTOSKELETON SPINDLE POLE BODY COMPONENT
DBS1 DNA REPAIR RECNUCLEASE; ALSO RECOMBINATION
BBS1 CYTOSKELETON ACTIN FILAMENT ORGANIZATION
SPC42 CYTOSKELETON SPINDLE POLE BODY COMPONENT
CNM67 CYTOSKELETON SPINDLE POLE BODY COMPONENT
CLB6 CELL CYCLE G2/M CYCLIN
CDC10 CYTOKINESIS GTP BINDING PROTEIN
CDC3 CYTOKINESIS SEPTIN
CLB3 CELL CYCLE G2/M CYCLIN
APC4 CELL CYCLE ANAPHASE-PROMOTING COMPLEX SUBUNIT
CDC16 CELL CYCLE ANAPHASE-PROMOTING COMPLEX SUBUNIT
C
RPN11 PROTEIN DEGRADATION 26S PROTEASOME REGULATORY SUBUNIT
UFD1 PROTEIN DEGRADATION UBIQUITIN FUSION DEGRADATION
RPN9 PROTEIN DEGRADATION 26S PROTEASOME REGULATORY SUBUNIT
RPT1 PROTEIN DEGRADATION 26S PROTEASOME SUBUNIT
RPT6 PROTEIN DEGRADATION 26S PROTEASOME REGULATORY SUBUNIT
PRE4 PROTEIN DEGRADATION PROTEASOME SUBUNIT, B TYPE
RPN6 PROTEIN DEGRADATION 26S PROTEASOME REGULATORY SUBUNIT
RPT4 PROTEIN DEGRADATION 26S PROTEASOME REGULATORY SUBUNIT
RPN7 PROTEIN DEGRADATION 26S PROTEASOME REGULATORY SUBUNIT
RPN3 PROTEIN DEGRADATION 26S PROTEASOME REGULATORY SUBUNIT
PUP2 PROTEIN DEGRADATION 20S PROTEASOME SUBUNIT ALPHA5
SCL1 PROTEIN DEGRADATION 20S PROTEASOME SUBUNIT Y7ALPHA/Y8
PRE5 PROTEIN DEGRADATION 20S PROTEASOME SUBUNIT ALPHA6
PRE9 PROTEIN DEGRADATION 20S PROTEASOME SUBUNIT Y13 ALPHA3
PRE1 PROTEIN DEGRADATION 20S PROTEASOME SUBUNIT C11 BETA4
PRE2 PROTEIN DEGRADATION 20S PROTEASOME SUBUNIT BETA5
PRE3 PROTEIN DEGRADATION 20S PROTEASOME SUBUNIT BETA1
PRE10 PROTEIN DEGRADATION 20S PROTEASOME SUBUNIT C1 ALPHA7
PUP1 PROTEIN DEGRADATION 20S PROTEASOME SUBUNIT BETA2
PRE6 PROTEIN DEGRADATION 20S PROTEASOME SUBUNIT ALPHA4
PRE7 PROTEIN DEGRADATION 20S PROTEASOME SUBUNIT
RPN10 PROTEIN DEGRADATION 26S PROTEASOME SUBUNIT
RPT3 PROTEIN DEGRADATION 26S PROTEASOME SUBUNIT
RPT5 PROTEIN DEGRADATION 26S PROTEASOME REGULATORY SUBUNIT
RPN12 PROTEIN DEGRADATION 26S PROTEASOME REGULATORY SUBUNIT
RPN5 PROTEIN DEGRADATION 26S PROTEASOME SUBUNIT
RPN8 PROTEIN DEGRADATION 26S PROTEASOME REGULATORY SUBUNIT
D
POP6 TRNA PROCESSING RNASE P AND RNASE MRP SUBUNIT
CAF16 TRANSPORT ATP-BINDING CASSETTE ABC FAMILY
MRS1 MRNA SPLICING UNKNOWN
SMD3 MRNA SPLICING CORE SNRNP PROTEIN
TAF40 TRANSCRIPTION TFIID 40 KD SUBUNIT
PET309 RNA PROCESSING COB1 MRNA STABILITY
PEF1 TRANSCRIPTION TFIIEA
YTH1 MRNA 3 -END PROCESSING CLEAVAGE/POLYADENYLATION FACTOR CF II COMPONENT
CBP1 MRNA STABILITY UNKNOWN
PRP24 MRNA SPLICING U4/U6 SNRNP PROTEIN
STD1 GLUCOSE REPRESSION MODULATOR OF GLUCOSE REPRESSION
MGM1 MITO GENOME MAINT DYNAMIN FAMILY PROTEIN
PRP19 MRNA SPLICING NOS-SNRNP SPLICEOSOME COMPONENT
SLS1 MITOCHONDRIAL METABOLISM INTEGRAL MEMBRANE PROTEIN
E
TPI1 GLYCOLYSIS TRIOSEPHOSPHATE ISOMERASE
GPM1 GLYCOLYSIS PHOSPHOGLYCERATE MUTASE
PGK1 GLYCOLYSIS PHOSPHOGLYCERATE KINASE
TDH3 GLYCOLYSIS GLYCERALDEHYDE-3-PHOSPHATE DEHYDROGENASE 3
TDH2 GLYCOLYSIS GLYCERALDEHYDE-3-PHOSPHATE DEHYDROGENASE 2
ENO2 GLYCOLYSIS ENOLASE II
TDH1 GLYCOLYSIS GLYCERALDEHYDE-3-PHOSPHATE DEHYDROGENASE 1
FBA1 GLYCOLYSIS ALDOLASE
TKL1 PENTOSE PHOSPHATE CYCLE TRANSKETOLASE
PDC5 GLYCOLYSIS PYRUVATE DECARBOXYLASE
PDC6 GLYCOLYSIS PYRUVATE DECARBOXYLASE 3
PDC1 GLYCOLYSIS PYRUVATE DECARBOXYLASE
CDC19 GLYCOLYSIS PYRUVATE KINASE
HXK2 GLYCOLYSIS HEXOKINASE II
TYE7 GLYCOLYSIS BASIC H-L-H TRANSCRIPTION FACTOR
PFK1 GLYCOLYSIS PHOSPHOFRUCTOKINASE
ACS2 ACETYL-COA BIOSYNTHESIS ACETYL-COENZYME A SYNTHETASE
F
COQ4 UBIQUINONE BIOSYNTHESIS METHYLTRANSFERASE
MES1 MRNA SPLICING UNKNOWN
MSF1 PROTEIN SYNTHESIS MITOCHONDRIAL PHENYLALANYL-TRNA SYNTHETASE SUBUNIT
MRP5 PROTEIN SYNTHESIS RIBOSOMAL PROTEIN, MITOCHONDRIAL S5

ECS289A

- Optimize a given function
- Combinatorial Optimization Problems
  - Enumerable space
  - Given a finite number of objects
  - Find an object which maximizes/minimizes a function

$$\min \sum_i d(x_i, x)$$

K-Means

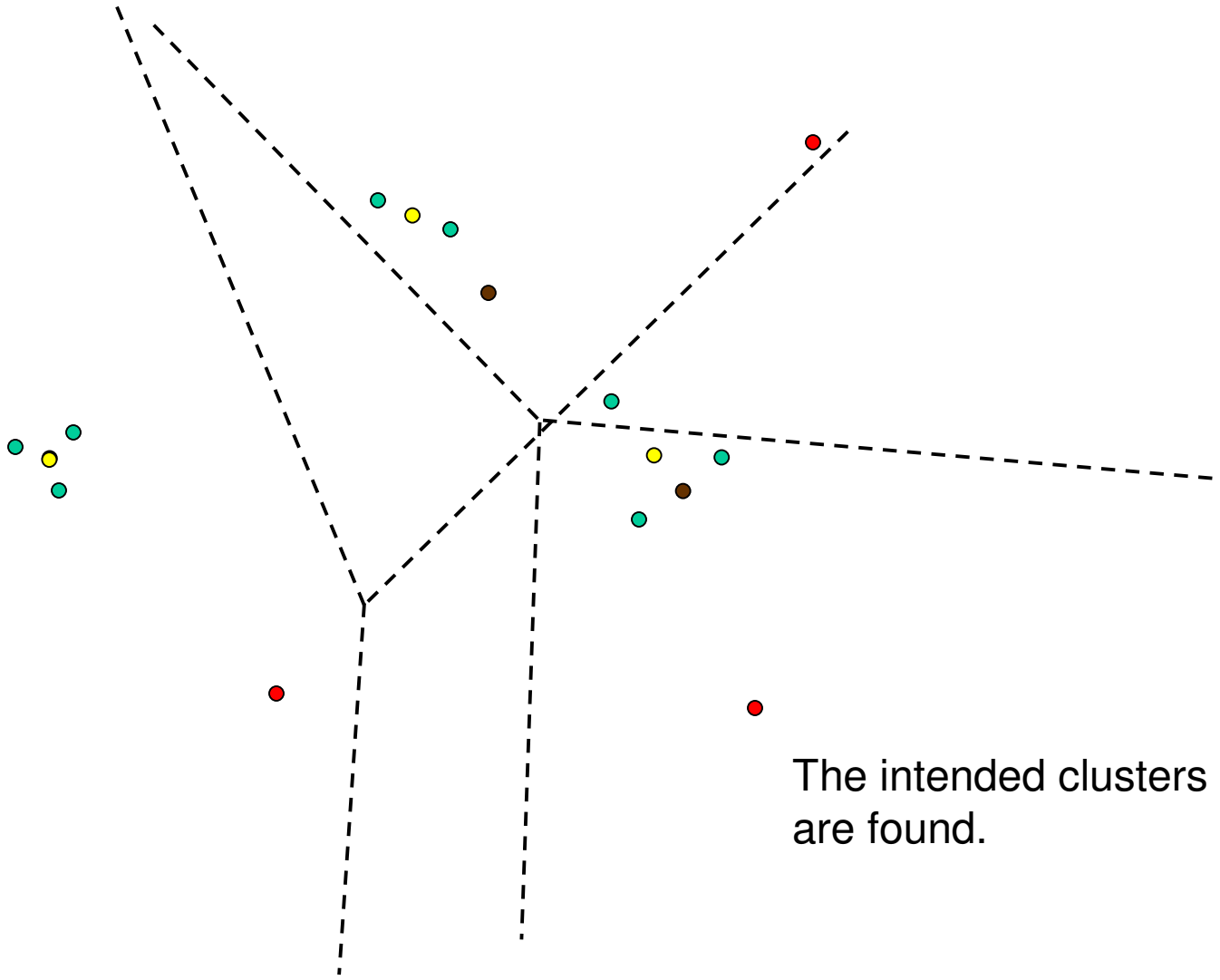
Input: Data Points, Number of Clusters (K)

Output: K clusters

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

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

K-means Clustering



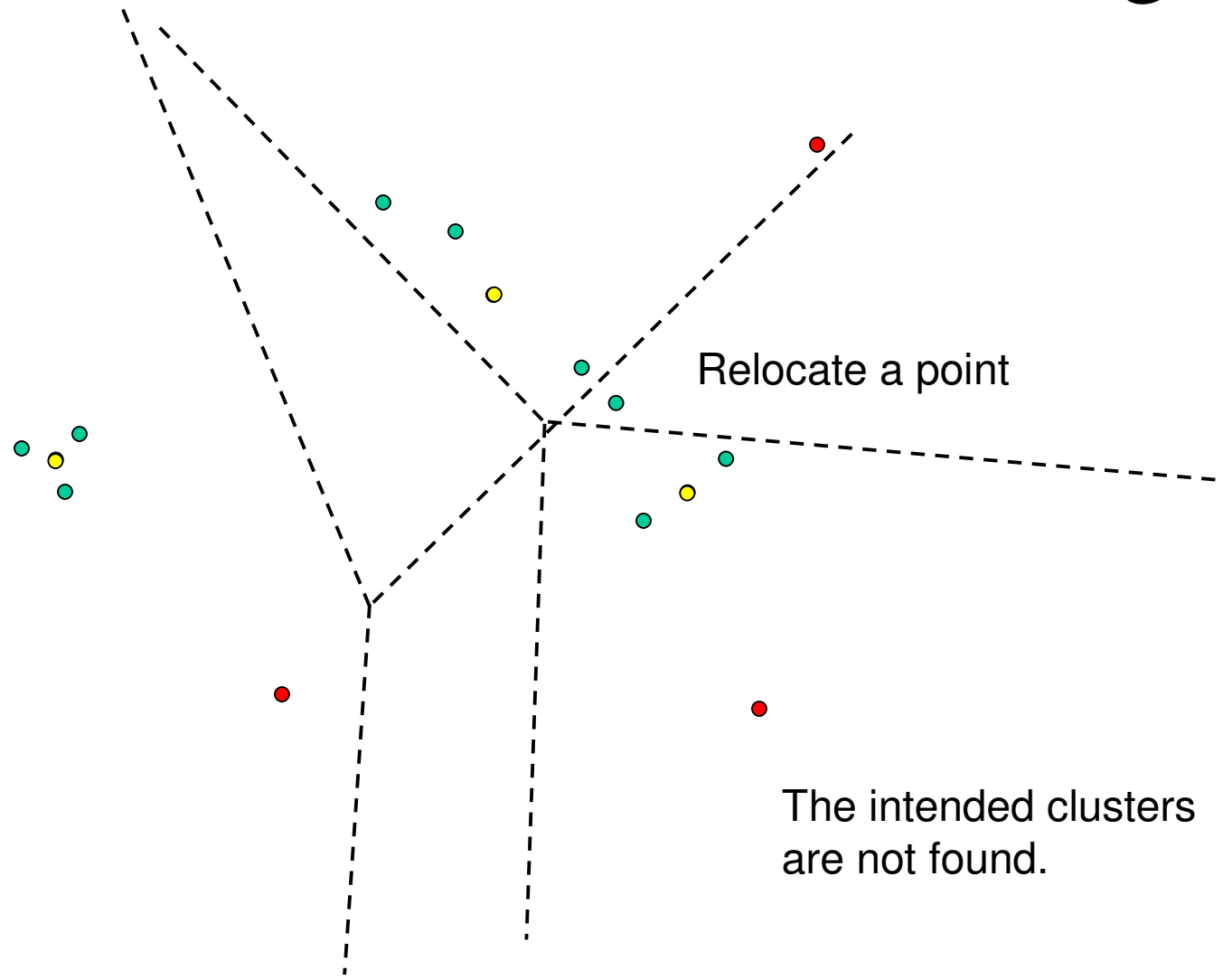
The intended clusters are found.

Ouyang et al.

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, betters average link clustering

Properties of K-means Clustering



Self Organizing Maps Clustering

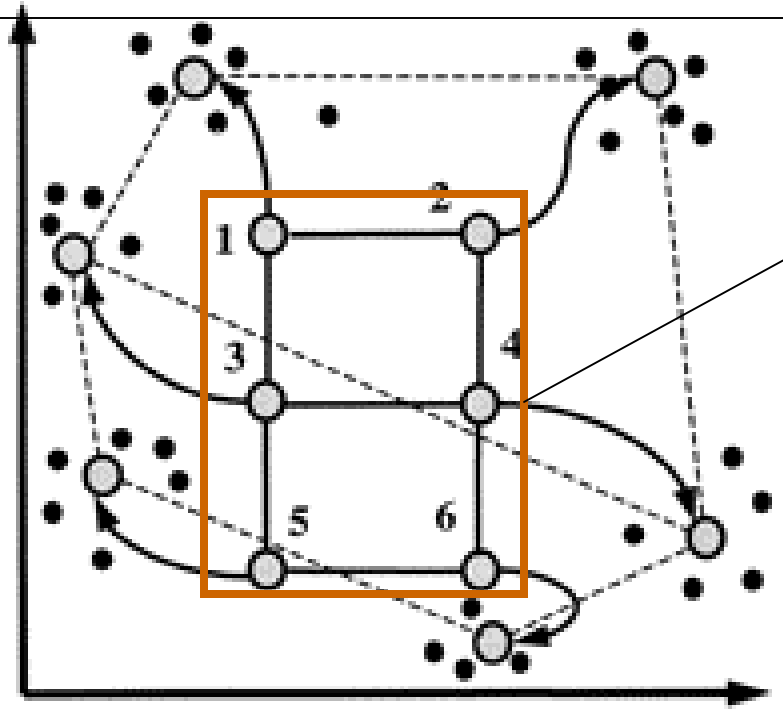
Input: Data Points, SOM Topology (K nodes and a distance function)

Output: K clusters, (near clusters are similar)

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

1. Select initial topology
2. Select a random data point P
3. Move all the nodes towards P by varying amounts
4. Go to (2) until convergence is achieved.

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



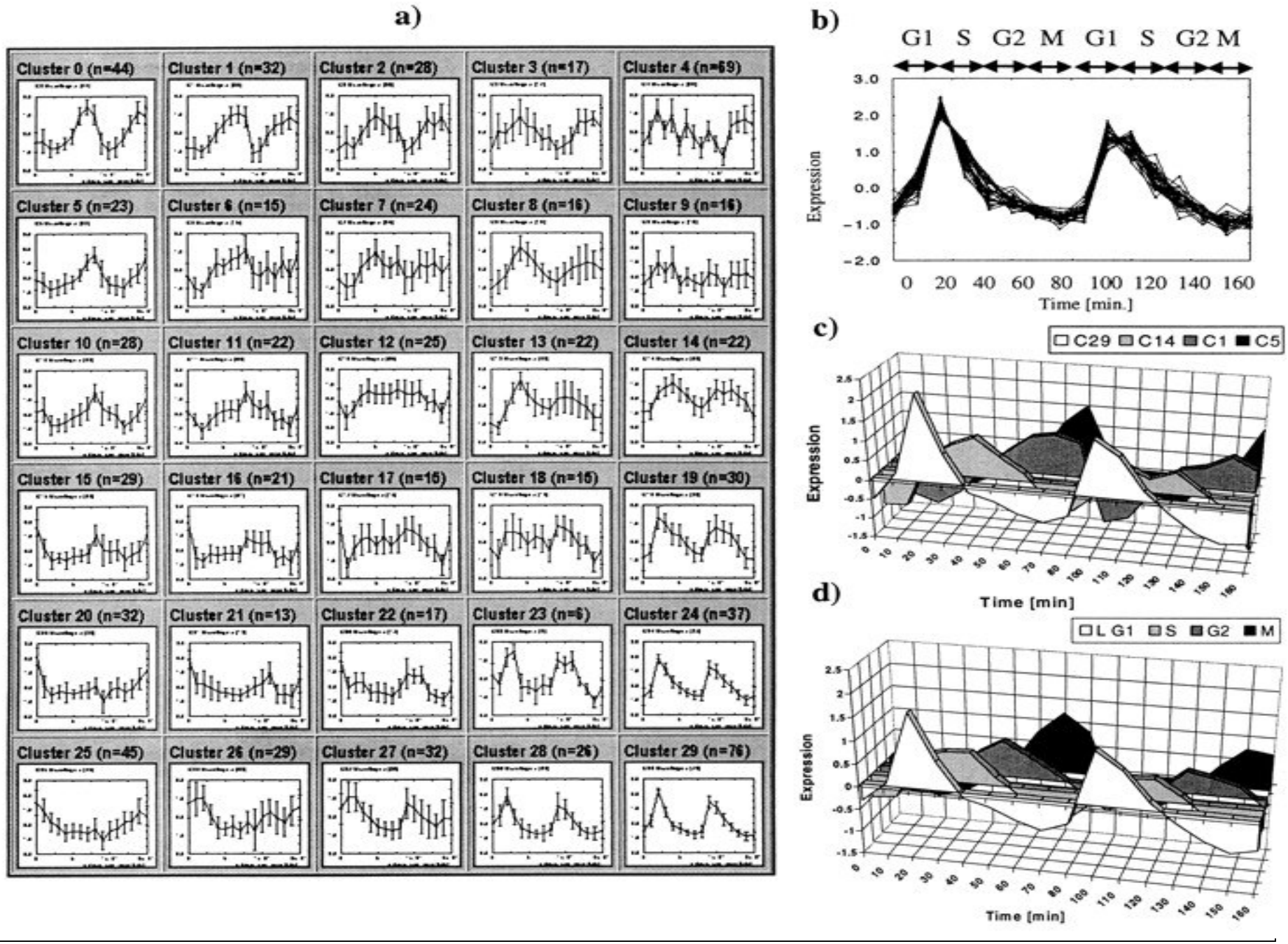
Initial Topology

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)
Cluster 0 (n=44)
Cluster 1 (n=32)
Cluster 2 (n=28)
Cluster 3 (n=17)
Cluster 4 (n=69)
Cluster 5 (n=23)
Cluster 6 (n=15)
Cluster 7 (n=24)
Cluster 8 (n=16)
Cluster 9 (n=16)
Cluster 10 (n=28)
Cluster 11 (n=22)
Cluster 12 (n=25)
Cluster 13 (n=22)
Cluster 14 (n=22)
Cluster 15 (n=29)
Cluster 16 (n=21)
Cluster 17 (n=15)
Cluster 18 (n=15)
Cluster 19 (n=30)
Cluster 20 (n=32)
Cluster 21 (n=13)
Cluster 22 (n=17)
Cluster 23 (n=6)
Cluster 24 (n=37)
Cluster 25 (n=45)
Cluster 26 (n=29)
Cluster 27 (n=32)
Cluster 28 (n=26)
Cluster 29 (n=76)
b)
G1 S G2 M G1 S G2 M
Expression
Time [min.]
c)
C29 C14 C1 C5
Expression
Time [min]
d)
L G1 S G2 M
Expression
Time [min]

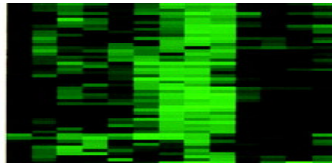
ECS289A

Properties

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

Cluster Presentation

- How to "see" the clusters effectively?
- Present gene expressions in different colors
- Plot similar genes close to each other



- Eisen's TreeView: minimize the sum of distances between clustered neighboring genes (2^{n-1} possible sub-tree flips, but can be done in polynomial time by dynamic programming)

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
    - Robust
    - Do better than average

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

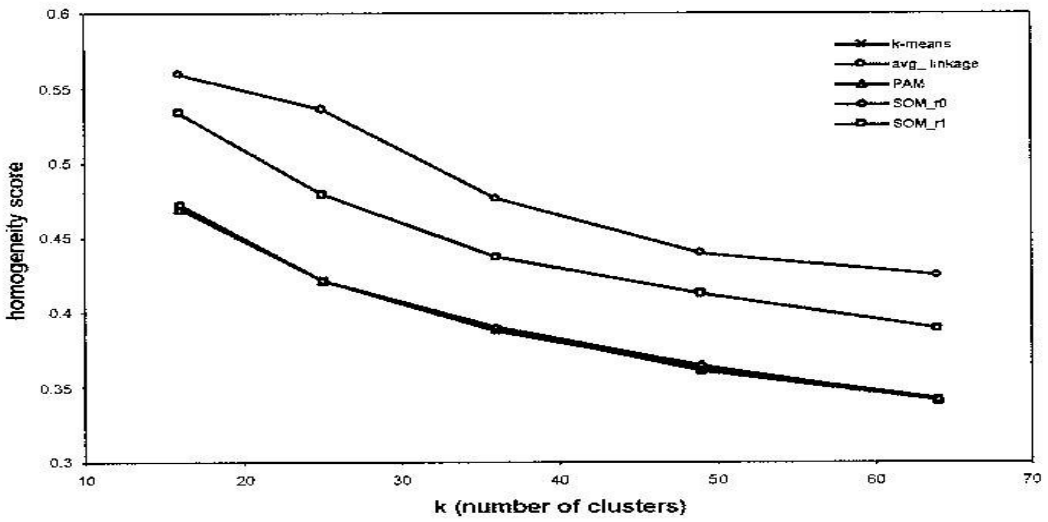


Figure 1a Comparing homogeneity scores among different algorithms

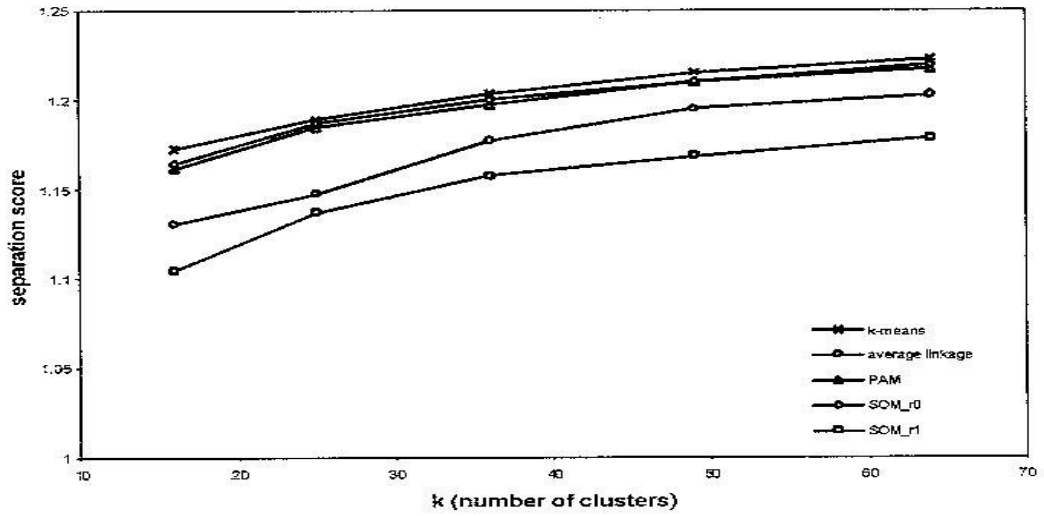


Figure 1b Comparing separation scores among different algorithms

Results:

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

- Silhouette Width
  - A composite index reflecting the compactness and separation of the clusters, and can be applied to different distance metrics
  - A larger value indicates a better overall quality of the clusters

Results:

•All had low scores indicating underlying "blurriness" of the data

•K-Means, PAM, SOM_r0 very close

•All three slightly better than ALHC

•SOM_r1 had the lowest score

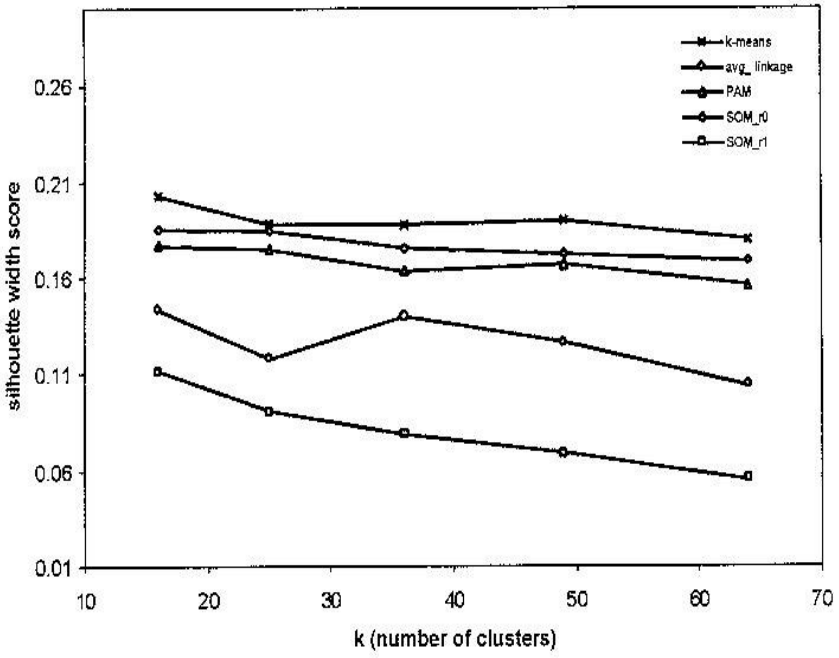


Figure 2 Comparison of average silhouette width among different algorithms

- Redundant Scores (external validation)
  - Almost every microarray data set has a small portion of duplicates, i.e. redundant genes (check genes)
  - A good clustering algorithm should cluster the redundant genes' expressions in the same clusters with high probability
  - DRRS (difference of redundant separation scores) between control and redundant genes was used as a measure of cluster quality
  - High DRRS suggests the redundant genes are more likely to be clustered together than randomly chosen genes

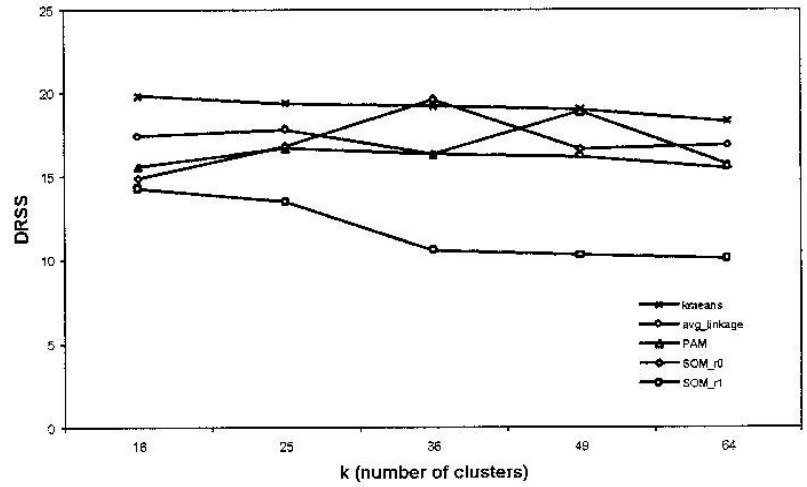


Figure 3 Comparison of DRSS among different algorithms

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
  - If the input data deviate slightly from their current value, will we get the same clustering?
  - Important in Microarray expression data analysis because of constant noise
  - Experiment:
    - each gene expression profile was perturbed by adding to it a random vector of the same dimension
    - values for the random vector generated from a Gaussian distr. (mean zero, and stand. dev.=0.01)
    - data was renormalized and clustered
    - WADP Cluster discrepancy: measure of inconsistent clusterings after noise. WADP=0 is perfect.

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

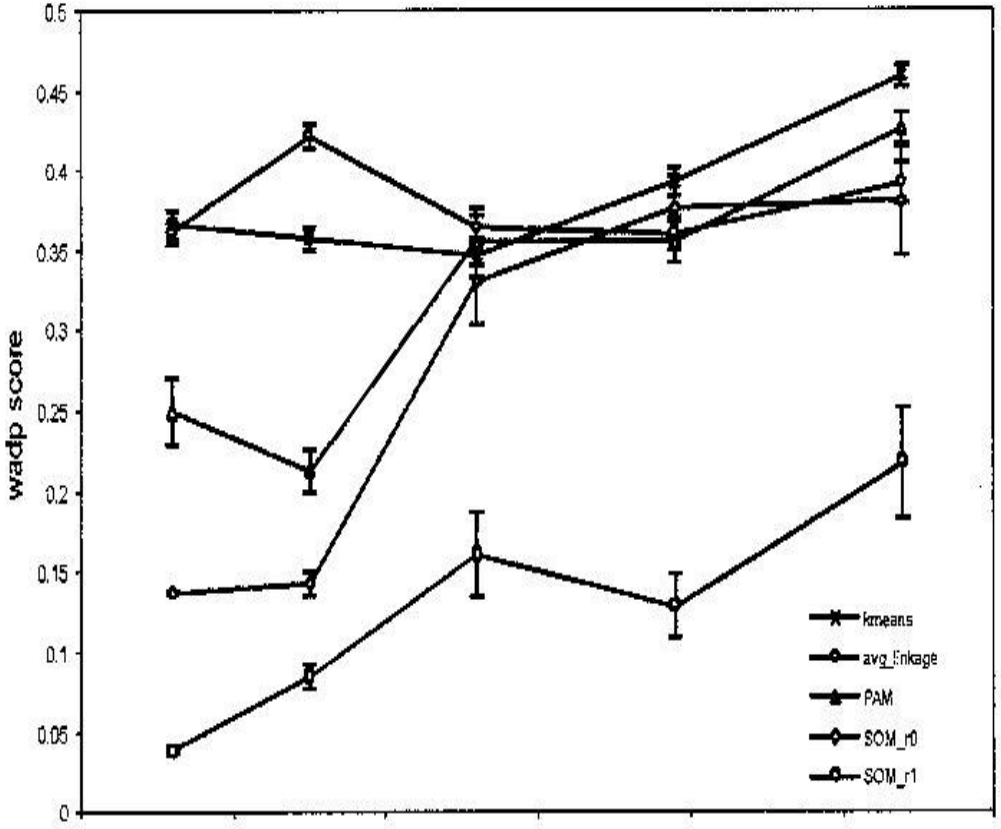
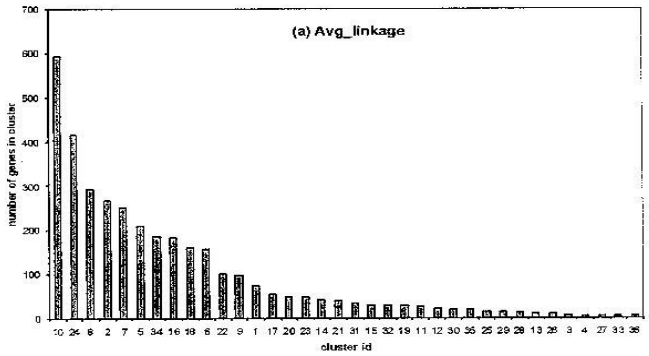
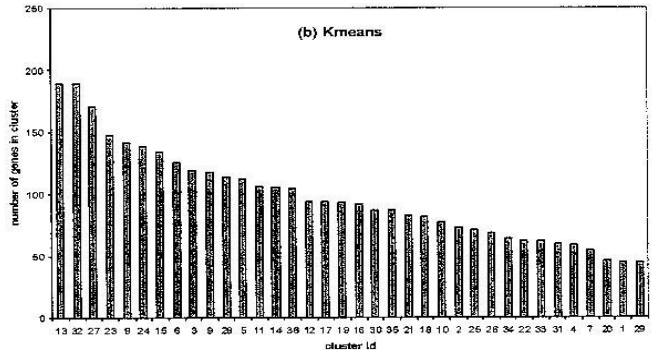
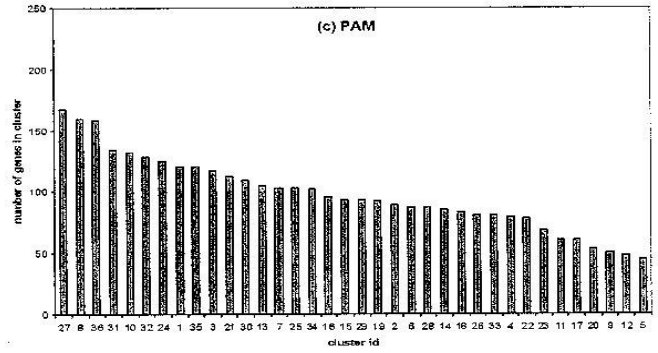


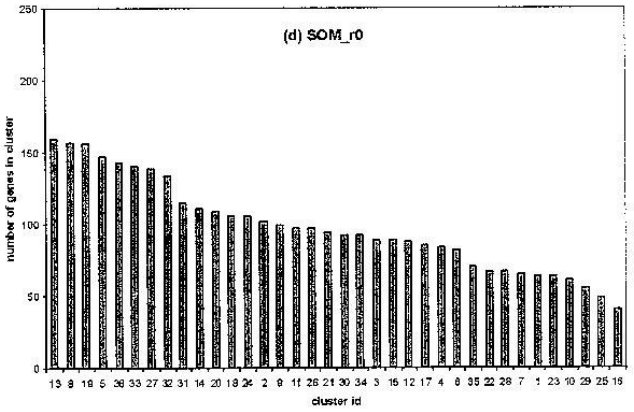
Figure 4 Comparison of WADP scores among different algorithms

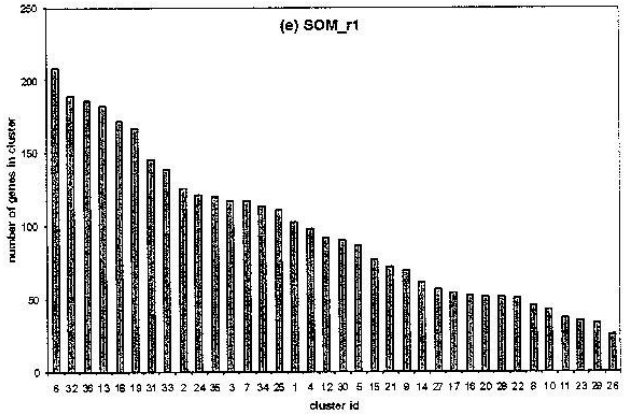
Comparison of Cluster Size and Consistency





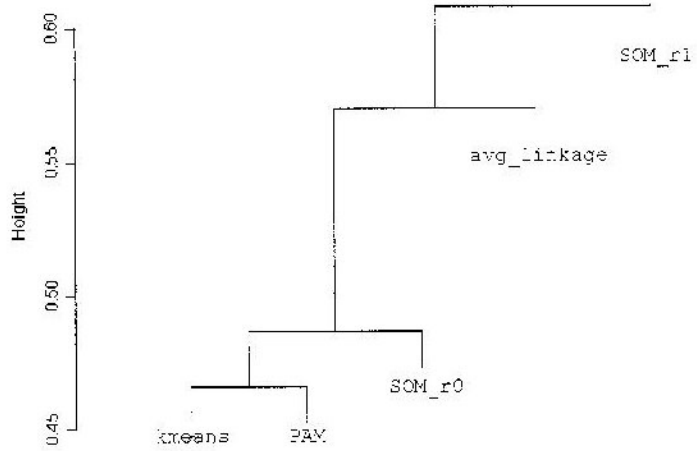






Comparison of Cluster Content

- 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

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
- Whan 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

References

- Eisen et al., Cluster analysis and display of genome-wide expression patterns, 1998. PNAS, v. 95, 14863-14868
- Tamayo et al., Interpreting patterns of gene expression with self organizing maps, 1999. PNAS, v. 96, 2907-2912
- Chen G, et al., Cluster analysis of microarray gene expression data, 2001. Statistica Sinica, 12:241-262
- Troyanskaya et al., Missing value estimation methods for DNA microarrays, Bioinformatics 2001 Jun;17(6):520-5