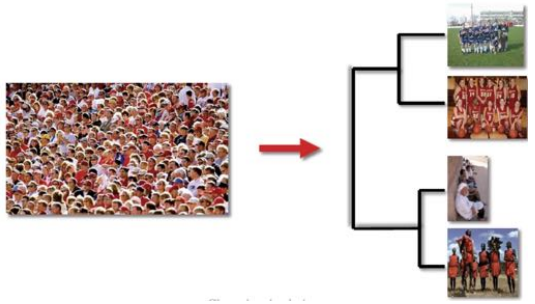
Clustering

Mikhail Dozmorov
Fall 2016

What is clustering

- Partitioning of a data set into subsets.
- A cluster is a group of relatively homogeneous cases or observations



What is clustering

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

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

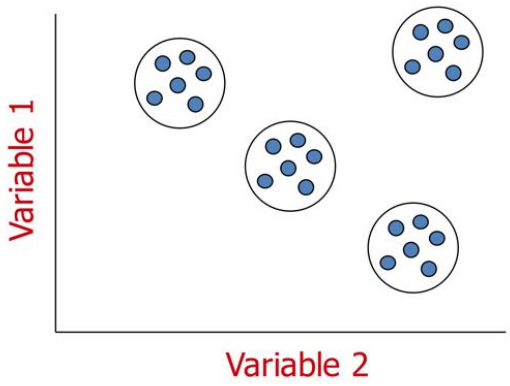
Clustering impossible

- **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

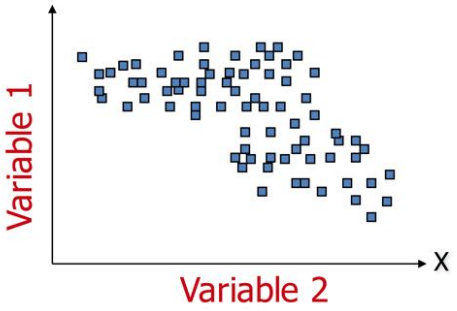
No function exists that satisfies all three.

J. Kleinberg. "**An Impossibility Theorem for Clustering. Advances in Neural Information Processing Systems**" (NIPS) 15, 2002.
https://www.cs.cornell.edu/home/kleinber/nips15.pdf

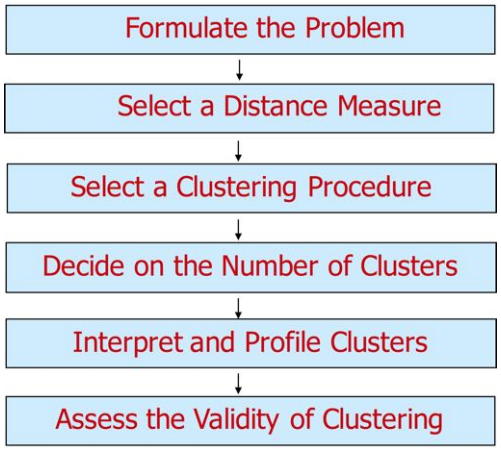
Clustering utopia



Clustering reality



Conducting Cluster Analysis



Clustering gene expression

Gene expression matrix

Samples

Genes

$$\begin{bmatrix} x_{11} & x_{12} & \mathsf{L} & x_{1n} \\ x_{21} & x_{22} & \mathsf{L} & x_{2n} \\ \mathsf{M} & \mathsf{M} & \mathsf{L} & \mathsf{M} \\ x_{g1} & x_{g2} & \mathsf{L} & x_{gn} \end{bmatrix}$$

Formulating the Problem

- Most important is **selecting the variables** on which the clustering is based.
- Inclusion of even one or two irrelevant variables may distort a clustering solution.
- Variables selected should describe the similarity between objects in terms that are relevant to the marketing research problem.
- Should be selected based on past research, theory, or a consideration of the hypotheses being tested.

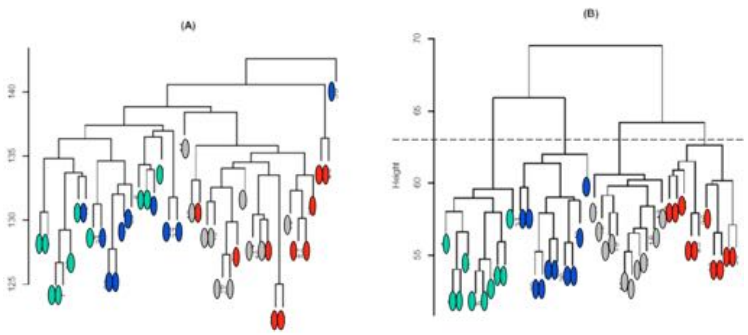
Filtering

- Non-informative genes contribute random terms in the calculation of distances
- The resulting effect is that they hide the useful information provided by other genes
- Therefore, assign non-informative genes zero weight, i.e., exclude them from the cluster analysis

Filtering examples

- **% Present >= X** - remove all genes that have missing values in greater than (100-X) percent of the columns
- **SD (Gene Vector) >= X** - remove all genes that have standard deviations of observed values less than X
- **At least X Observations with abs(Val) >= Y** - remove all genes that do not have at least X observations with absolute values greater than Y
- **MaxVal-MinVal >= X** - remove all genes whose maximum minus minimum values are less than X

Clustering noise



A: 450 relevant genes plus 450 "noise" genes.

B: 450 relevant genes.

Cluster the right data

Clustering works as expected when the data to be clustered is processed correctly

- **Log Transform Data** - replace all data values x by log2 (x). Why?
- **Center genes [mean or median]** - subtract the row-wise mean or median from the values in each row of data, so that the mean or median value of each row is 0.
- **Center arrays [mean or median]** - subtract the column-wise mean or median from the values in each column of data, so that the mean or median value of each column is 0.

Cluster the right data

Clustering works as expected when the data to be clustered is processed correctly

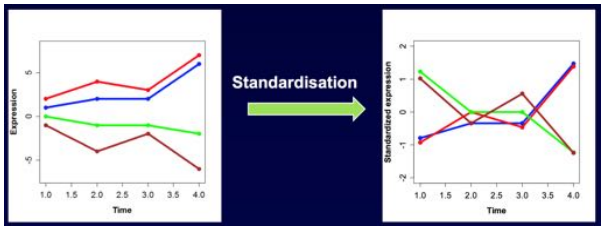
- **Normalize genes** - multiply all values in each row of data by a scale factor S so that the sum of the squares of the values in each row is 1.0 (a separate S is computed for each row).
- **Normalize arrays** - multiply all values in each column of data by a scale factor S so that the sum of the squares of the values in each column is 1.0 (a separate S is computed for each column).
- These operations are not associative, so the order in which these operations is applied is very important
- Log transforming centered genes are not the same as centering log transformed genes.

Standartization

In many cases, we are not interested in the absolute amplitudes of gene expression, but in the relative changes. Then, we standardize:

$$g_s = (g - \hat{g})/\sigma(g)$$

Standardized gene expression vectors have a mean value of zero and a standard deviation of one.



How to define (dis)similarity among objects

Distance

- Clustering organizes things that are close into groups
- What does it mean for two genes to be close?
- What does it mean for two samples to be close?
- Once we know this, how do we define groups?

Distance

- We need a mathematical definition of distance between two points
- What are points?
- If each gene is a point, what is the mathematical definition of a point?

Points

$Gene_1 = (E_{11}, E12, \ldots, E1N)$

$Gene_2 = (E_{21}, E22, \ldots, E2N)$

$Sample_1 = (E_{11}, E_{21}, \ldots, E_{G1})$

$Sample_2 = (E_{12}, E_{22}, \ldots, E_{G2})$

$E_{gi} = expression\ gene\ g,\ sample\ i$

Distance definition

For all objects i, j, and h

$$d(i,j) \geq 0$$
$$d(i,i) = 0$$
$$d(i,j) = d(j,i)$$
$$d(i,j) \leq d(i,h) + d(h,j)$$

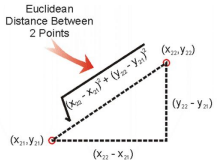
Most famous distance

Euclidean distance

Example distance between gene 1 and 2:

– Sqrt of Sum of $(E_{1i} - E_{2i})^2$, $i = 1, \ldots, N$

- When N is 2, this is distance as we know it:



- When N is 20,000 you have to think abstractly

Distance measures

- Euclidean distance

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{in} - x_{jn})^2}$$

- Manhattan distance

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{in} - x_{jn}|$$

- Minkowski distance (L_q metric)

$$d(i,j) = \left(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \cdots + |x_{in} - x_{jn}|^q \right)^{1/q}$$

- Disadvantages: not scale invariant, not for negative correlations

Distance measures

- When deciding on an appropriate value of q, the investigator must decide whether emphasis should be placed on large differences.
- Larger values of q give relatively more emphasis to larger differences.

Distance measures

- Canberra distance

$$d(\mathbf{p},\mathbf{q}) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i| + |q_i|}$$

- Binary (0/1 vectors), aka Jaccard distance

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

- Maximum distance between two components of x and y

Similarity definition

- For all objects i, j

$$0 \leq sim(i,j) \leq 1$$
$$sim(i,i) = 1$$
$$sim(i,j) = sim(j,i)$$

Similarity measures

- Gene expression profiles represent comparative expression measures
- Euclidean distance may not be meaningful
- Need distance measure that score based on similarity
- The more objects i and j are alike (or close) the larger $s(i,j)$ becomes

Similarity measures

Cosine similarity. From Euclidean dot product between two non-zero vectors:

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos\theta$$

The cosine similarity is

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{k=1}^{n} x_{ik} x_{jk}}{\left[\sum_{k=1}^{n} x_{ik}^2 \sum_{k=1}^{n} x_{jk}^2\right]^{1/2}}$$

Similarity measures

Pearson correlation coefficient [-1, 1]

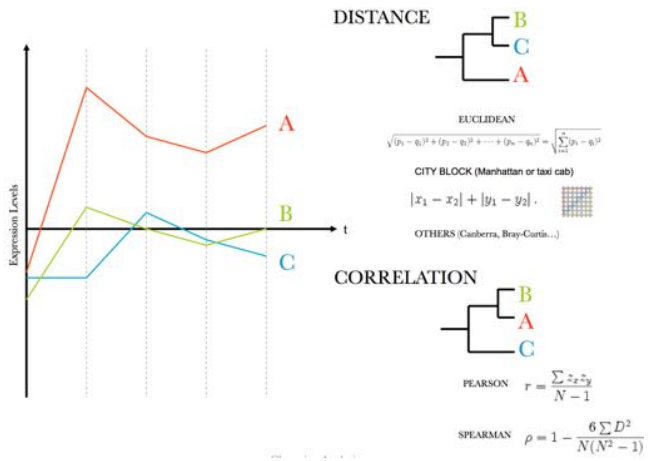
Vectors are normalized to the vector's means

$$s(i,j)=\frac{\sum_{k=1}^{n}(x_{ik}-\bar{x}_{i\cdot})(x_{jk}-\bar{x}_{j\cdot})}{\left[\sum_{k=1}^{n}(x_{ik}-x_{i\cdot})^2\sum_{k=1}^{n}(x_{jk}-\bar{x}_{j\cdot})^2\right]^{1/2}}$$

Convert to dissimilarity [0, 1]

$d(i,j) = (1 - s(i,j))/2$

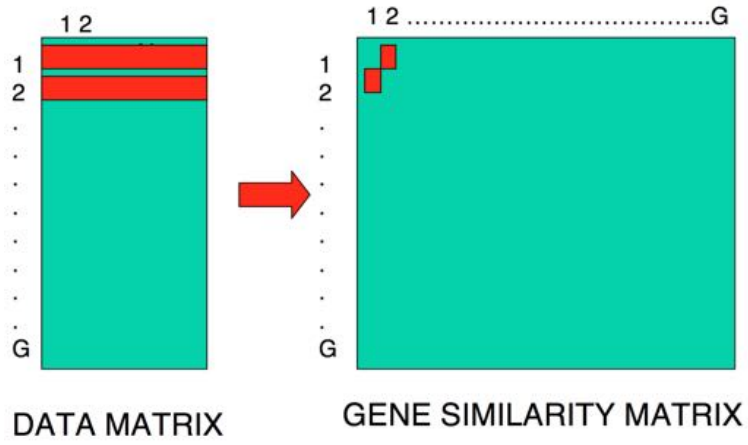
Distances between gene expression profiles



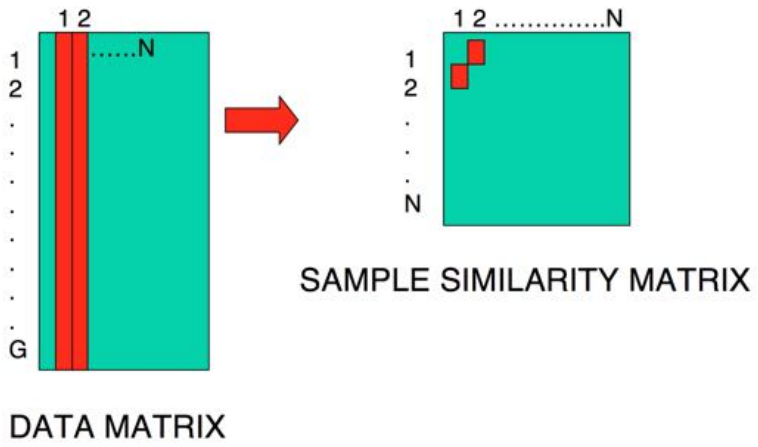
Convert correlation to dissmilarity

$$d(X_i, X_j) = \frac{1 - Cor(X_i, X_j)}{2}$$

The (dis-)similarity matrixes



The (dis-)similarity matrixes



Clustering binary data

- Two columns with binary data, encoded 0 and 1
- a - number of rows where both columns are 1
- b - number of rows where this and not the other column is 1
- c - number of rows where the other and not this column is 1
- d - number of rows where both columns are 0

Jaccard distance

$$\frac{a}{a+b+c}$$

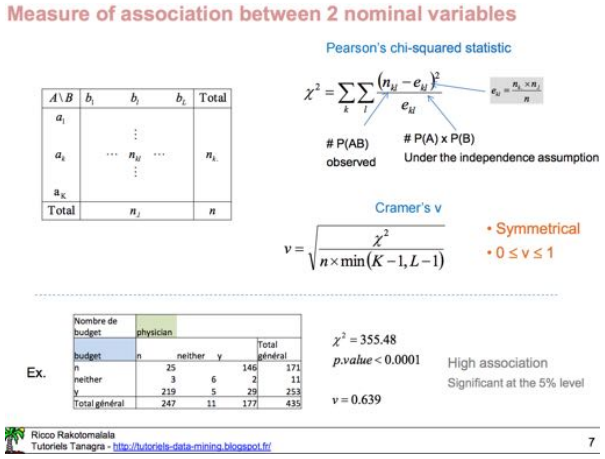
Clustering binary data

- Two columns with binary data, encoded 0 and 1
- a - number of rows where both columns are 1
- b - number of rows where this and not the other column is 1
- c - number of rows where the other and not this column is 1
- d - number of rows where both columns are 0

Tanimoto distance

$$\frac{a+d}{a+d+2(b+c)}$$

Clustering categorical data



Clustering mixed data

Gower distance

J. C. Gower "**A General Coefficient of Similarity and Some of Its Properties**" Biometrics 1971
http://venus.unive.it/romanaz/modstat_ba/gowdis.pdf

- Idea: Use distance measure between 0 and 1 for each pair of variables: $d_{ij}^{(f)}$
- Aggregate: $d(i,j) = \frac{1}{p}\sum_{i=1}^{p} d_{ij}^{(f)}$

Gower distance

How to calculate distance measure for each pair of variables

- **Quantitative**: interval-scaled distance $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{R_f}$, where x_{if} is the value for object i in variable f, and R_f is the range of variable f for all objects
- **Categorical**: use "1" when x_{if} and x_{jf} agree, and "0" otherwise
- **Ordinal**: Use normalized ranks, then like interval-scaled based on range

Choose (dis-)similarity metric

- Think hard about this step!
- Remember: garbage in - garbage out
- The metric that you pick should be a valid measure of the distance/similarity of genes.

Examples

- Applying correlation to highly skewed data will provide misleading results.
- Applying Euclidean distance to data measured on categorical scale will be invalid.

Distances in R

Function	Package	Distances
dist	stats	Euclidean, Manhattan, Canberra, max, binary
daisy	cluster, bioDist	Euclidean, Manhattan
distancematrix, distancevector	hopach	Euclidean, cor, cosine-angle (abs versions)
vegdist	vegan	Jaccard, Gower, many others

Other packages: `cclust`, `e1071`, `flexmix`, `fpc`, `mclust`, `Mfuzz`, `class`

Assembling objects into clusters

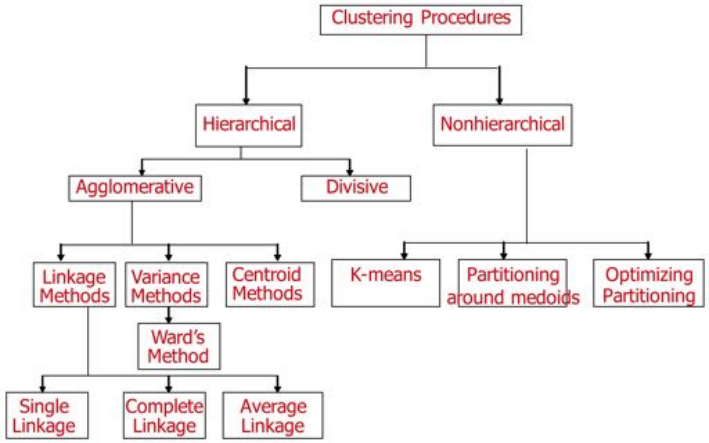
Assembling objects into clusters

- The number of ways to partition a set of n objects into k non-empty classes

$$S(n,k) = \frac{1}{k!}\sum_{i=0}^{k-1}(-1)^i\binom{k}{i}(k-i)^n$$

- $S(n,1) = 1$ - one way to partition n object in to 1 group, or n disjoint groups
- $S(n,2) = 2^{n-1} - 1$ ways to partition n objects into two non-empty groups

Classification of Clustering Procedures



Hierarchical Clustering

- Allows organization of the clustering data to be represented in a tree (dendrogram)
- **Agglomerative** (Bottom Up): each observation starts as own cluster. Clusters are merged based on similarities
- **Divisive** (Top Down): all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

Agglomerative clustering (bottom-up)

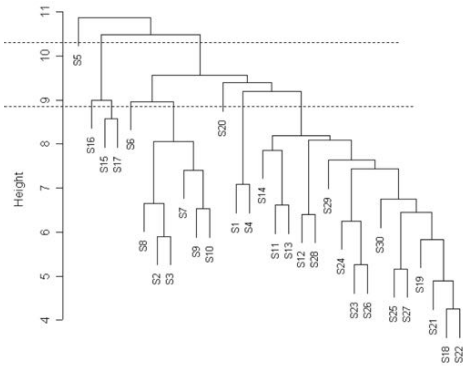
- Idea: ensure nearby points end up in the same cluster
- Starts with as each gene in its own cluster
- Joins the two most similar clusters
- Then, joins next two most similar clusters
- Continues until all genes are in one cluster

Divisive clustering (top-down)

- Starts with all genes in one cluster
- Choose split so that genes in the two clusters are most similar (maximize “distance” between clusters)
- Find next split in same manner
- Continue until all genes are in single gene clusters

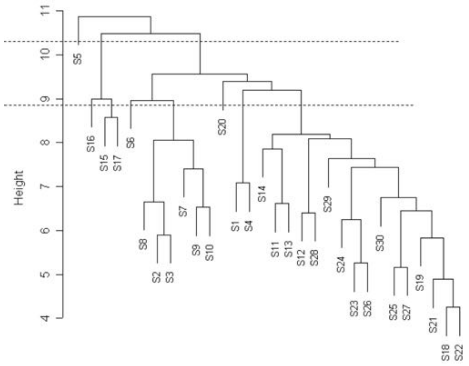
Dendrograms

- We can then make dendrograms showing divisions
- The y-axis represents the distance between the groups divided at that point



Dendrograms

- Note: Left and right is assigned arbitrarily. Vertical distance is what's matter
- Look at the height of division to find out distance. For example, S5 and S16 are very far.



Which to use?

- Both agglomerative and divisive are only ‘step-wise’ optimal: at each step the optimal split or merge is performed
- Outliers will irreversibly change clustering structure

Which to use?

Agglomerative/Bottom-Up

– Computationally simpler, and more available.

– More "precision" at bottom of tree

– When looking for small clusters and/or many clusters, use agglomerative

Which to use?

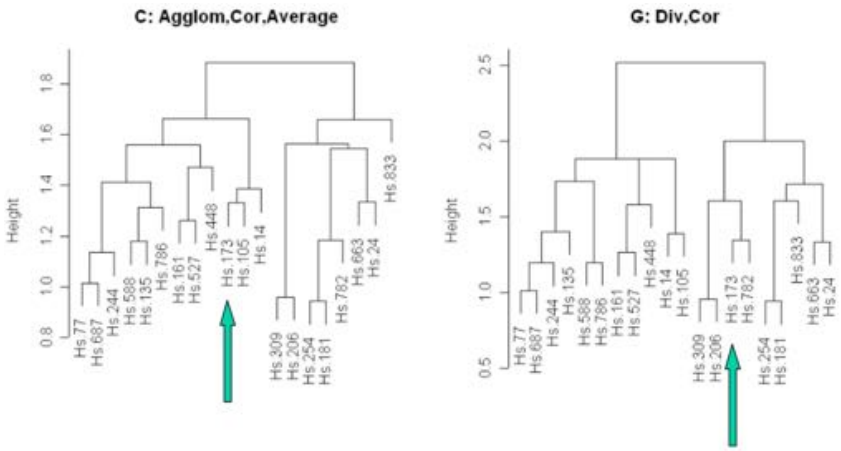
Divisive/Top-Down

– More "precision" at top of tree.

– When looking for large and/or few clusters, use divisive

Results ARE sensitive to choice!

Which to use?

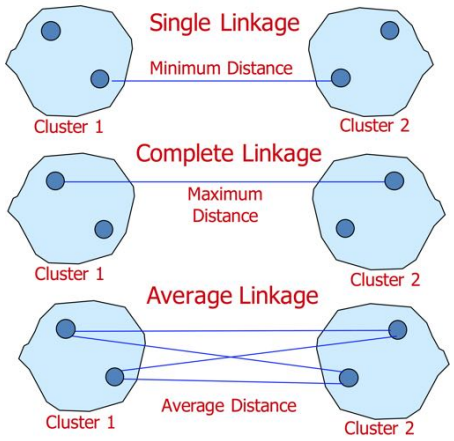


Linking objects based on the distance between them

Linkage between clusters

- **Single Linkage** - join clusters whose distance between closest genes is smallest (elliptical)
- **Complete Linkage** - join clusters whose distance between furthest genes is smallest (spherical)
- **Average Linkage** - join clusters whose average distance is the smallest.

Linkage between clusters

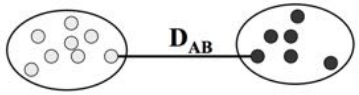


Single linkage

Cluster-to-cluster distance is defined as the *minimum distance* between members of one cluster and members of the another cluster. Single linkage tends to create 'elongated' clusters with individual genes chained onto clusters.

$$D_{AB} = \min (d(u_i, v_j))$$

where $u \in A$ and $v \in B$
for all $i = 1$ to N_A and $j = 1$ to N_B



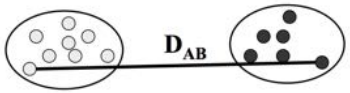
5

Complete linkage

Cluster-to-cluster distance is defined as the *maximum distance* between members of one cluster and members of the another cluster. Complete linkage tends to create clusters of similar size and variability.

$$D_{AB} = \max (d(u_i, v_j))$$

where $u \in A$ and $v \in B$
for all $i = 1$ to N_A and $j = 1$ to N_B



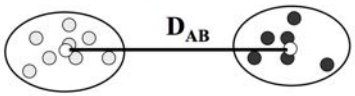
7

Average linkage

Cluster-to-cluster distance is defined as the *average distance* between all members of one cluster and all members of another cluster. Average linkage has a slight tendency to produce clusters of similar variance.

$$D_{AB} = 1/(N_A N_B) \Sigma \Sigma (d(u_i, v_j))$$

where $u \in A$ and $v \in B$
for all $i = 1$ to N_A and $j = 1$ to N_B



Ward's method

- **Ward's procedure** is commonly used. For each cluster, the sum of squares is calculated. The two clusters with the smallest increase in the overall sum of squares within cluster distances are combined.

$$\begin{aligned}\Delta(A,B) &= \sum_{i\in A\cup B}\|\vec{x}_i - \vec{m}_{A\cup B}\|^2 - \sum_{i\in A}\|\vec{x}_i - \vec{m}_A\|^2 - \sum_{i\in B}\|\vec{x}_i - \vec{m}_B\|^2 \\ &= \frac{n_A n_B}{n_A + n_B}\|\vec{m}_A - \vec{m}_B\|^2\end{aligned}$$

- Δ - Merging cost of combining the clusters A and B. m_j is the center of cluster j, and n_j is the number of points in it.
- The sum of squares starts at 0 (each point is in its own cluster), and grows as clusters are merged. Ward's method keep this growth to minimum.

Ward, J. H., Jr. (1963), "**Hierarchical Grouping to Optimize an Objective Function**", Journal of the American Statistical Association
http://iv.slis.indiana.edu/sw/data/ward.pdf

Ward's method

- The distance d between two clusters C_i and C_j is defined as the loss of information (or: the increase in error) in merging two clusters.
- The error of a cluster C is measured as the sum of distances between the objects in the cluster and the cluster centroid $cenC$.
- When merging two clusters, the error of the merged cluster is larger than the sum or errors of the two individual clusters, and therefore represents a loss of information.
- The merging is performed on those clusters which are most homogeneous, to unify clusters such that the variation inside the merged clusters increases as little as possible.
- Ward's method tends to create compact clusters of small size. It is a least squares method, so implicitly assumes a Gaussian model.

Ward's method

An important issue though is the form of input that is necessary to give Ward's method. For an input data matrix, x, in R's `hclust` function the following command is required: `hclust(dist(x)^2, method="ward")` although this is not mentioned in the function's documentation file.

Fionn Murtagh "**Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?**" Journal of Classification 2014 https://link.springer.com/article/10.1007/s00357-014-9161-z

K-means clustering

- k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.
- It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data.

K-means statistics

- The basic idea behind K-means clustering consists of defining clusters so that the total intra-cluster variation (known as total within-cluster variation) is minimized

$$minimize\left(\sum_{i=1}^{k} W(C_k)\right)$$

where C_k is the k^{th} cluster and $W(C_k)$ is the within-cluster variation of the cluster C_k.

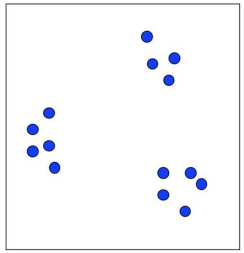
K-means - Algorithm

```
Begin
    Assign each item a class in 1 to K (randomly)
    For 1 to max-iteration {
        For each class 1 to K {
            Calculate centroid (one of the "K means")
            Calculate distance from centroid to each item
        }
        Assign each item the class of the nearest centroid
        Exit if no items are re-assigned (convergence)
    }
End
```

J. B. MacQueen "**Some Methods for classification and Analysis of Multivariate Observations**" 1967 https://projecteuclid.org/euclid.bsmsp/1200512992

K-means steps

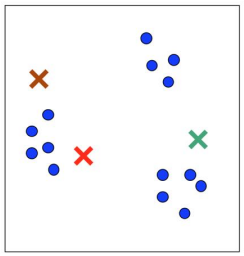
- Simplified example
  – Expression for two genes for 14 samples
- Some structure can be seen



Iteration = 0

K-means steps

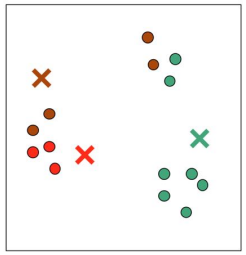
- Choose K centroids
- These are starting values that the user picks.
- There are some data driven ways to do it



Iteration = 0

K-means steps

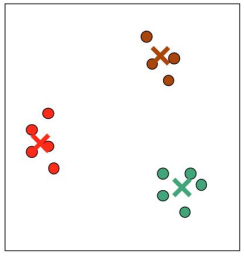
– Find the closest centroid for each point
- This is where distance is used
- This is "first partition" into K clusters



Iteration = 1

K-means steps

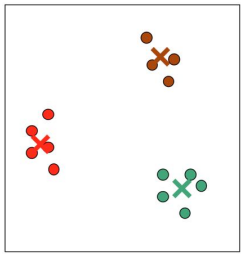
– Take the middle of each cluster
- Re-compute centroids in relation to the middle
- Use the new centroids to calculate distance



Iteration = 3

K-means steps

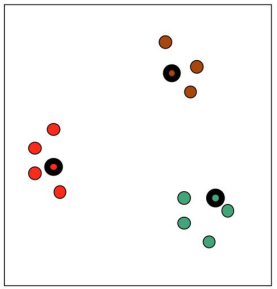
– Expression for two genes for 14 samples



Iteration = 3

PAM (K-medoids)

- **Centroid** - The average of the samples within a cluster
- **Medoid** - The "representative object" within a cluster
- Initializing requires choosing medoids at random.



K-means limitations

- Final results depend on starting values
- How do we chose K? There are methods but not much theory saying what is best.
- Where are the pretty pictures?

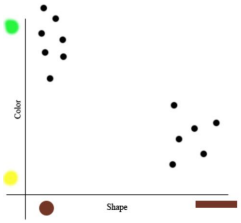
Self-organizing (Kohonen) maps

- Self organizing map (SOM) is a learning method which produces low dimension data (e.g. $2D$) from high dimension data (nD) through the use of self-organizing neural networks
- E.g. an apple is different from a banana in more then two ways but they can be differentiated based on their size and color only.

Self-organizing (Kohonen) maps

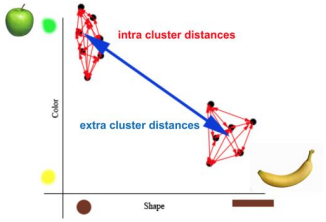
If we present apples and bananas with points and similarity with lines then

- Two points connected by a shorter line are of same kind
- Two points connected by a longer line are of different kind
- Threshold t is chosen to decide if the line is longer/shorter



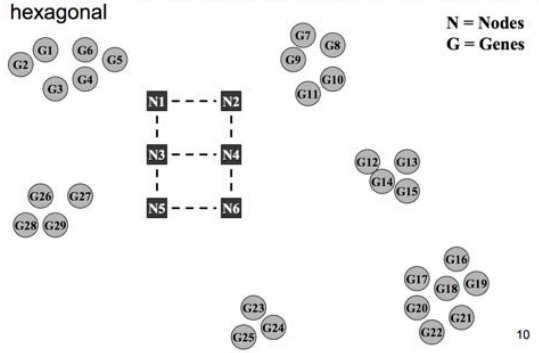
Self-organizing (Kohonen) maps

- We just created a map to differentiate an apple from banana based on two traits only.
- We have successfully “trained” the SOM, now anyone can use to “map” apples from banana and vice versa



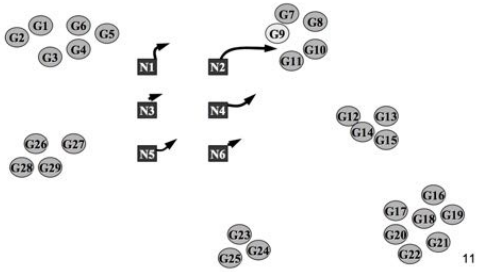
SOM in gene expression studies

1. Specify the number of nodes (clusters) desired, and also specify a 2-D geometry for the nodes, e.g., rectangular or hexagonal



SOM example

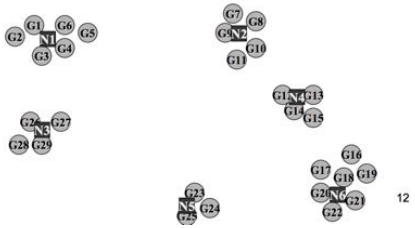
2. Choose a random gene, say, G9
3. Move the nodes in the direction of G9. The node closest to G9 (N2) is moved the most, and the other nodes are moved by smaller varying amounts. The farther away the node is from N2, the less it is moved.



SOM example

4. Repeat Steps 2 and 3 several thousand times; with each iteration, the amount that the nodes are allowed to move is decreased.

5. Finally, each node will "nestle" among a cluster of genes, and a gene will be considered to be in the cluster if its distance to the node in that cluster is less than its distance to any other node.



12

Application of SOM

Genome Clustering

- Goal: trying to understand the phylogenetic relationship between different genomes.
- Compute: bootstrap support of individual genomes for different phylogentic tree topologies, then cluster based on the topology support.

Clustering Proteins based on the architecture of their activation loops

- Align the proteins under investigation
- Extract the functional centers
- Turn 3D representation into 1D feature vectors
- Cluster based on the feature vectors

Other approaches

- **Bi-clustering** - cluster both the genes and the experiments simultaneously to find appropriate context for clustering
- R packages: `iBBiG`, `FABIA`, `biclust`
- stand-alone: `BicAT` (Biclustering Analysis Toolbox))

Dimensionality reduction techniques

Principal Components Analysis

- Principal component analysis (PCA) is a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components
- Also know as Independent component analysis or *dimension reduction technique*
- PCA decomposes complex data relationship into simple components
- New components are linear combinations of the original data

Principal Components Analysis

- Performs a rotation of the data that maximizes the variance in the new axes
- Projects high dimensional data into a low dimensional sub-space (visualized in 2-3 dims)
- Often captures much of the total data variation in a few dimensions (< 5)
- Exact solutions require a fully determined system (matrix with full rank), i.e. a “square” matrix with independent rows

Principal Components Analysis

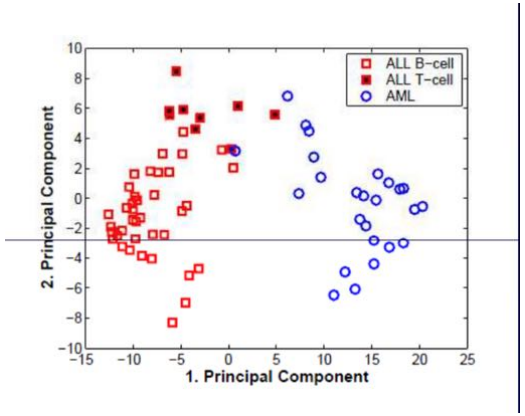
- **PCA** - linear projection of the data onto major principal components defined by the eigenvectors of the covariance matrix.
- Criterion to be minimised: square of the distance between the original and projected data.

$$x_P = Px$$

P is composed by eigenvectors of the covariance matrix

$$C = \frac{1}{n-1}\sum_i (x_i - \mu)(x_i - \mu)^t$$

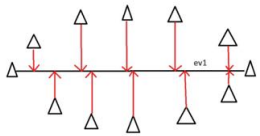
Principal Components Analysis



Example: Leukemia data sets by Golub et al.: Classification of ALL and AML

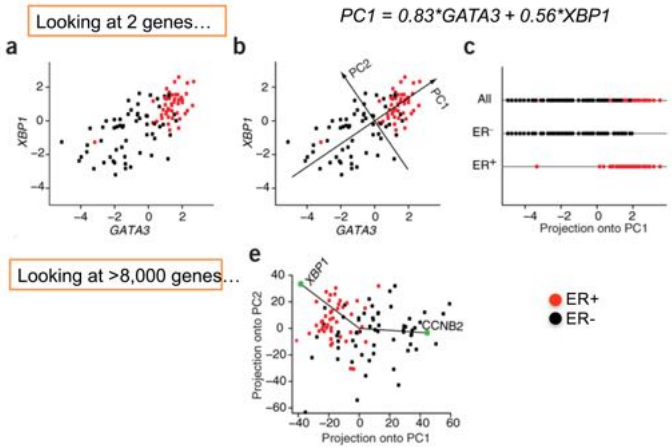
Principal Components Analysis

- Eigenvalue: describes the total variance in an eigenvector.
- The eigenvector with the largest eigenvalue is the first principal component. The second largest eigenvalue will be the direction of the second largest variance.



Eigenvector

Principal Components Analysis



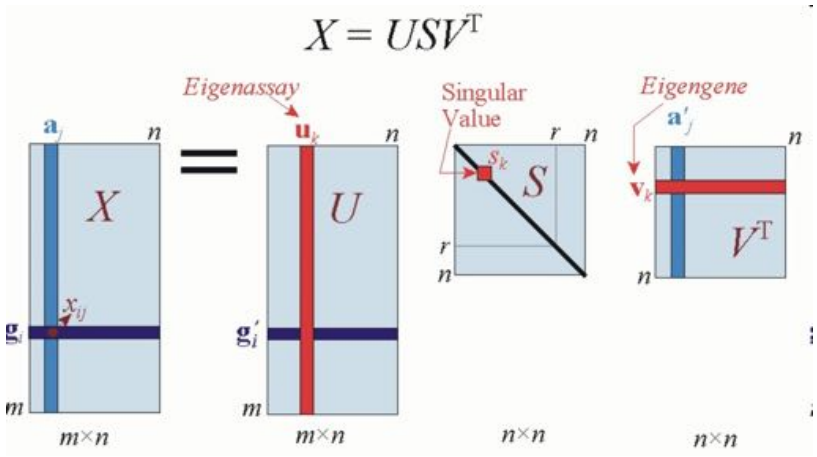
PCA for gene expression

- Given a gene-by-sample matrix X we decompose (centered and scaled) X as USV^T
- We don't usually care about total expression level and the dynamic range which may be dependent on technical factors
- U, V are orthonormal
- S diagonal-elements are eigenvalues = variance explained

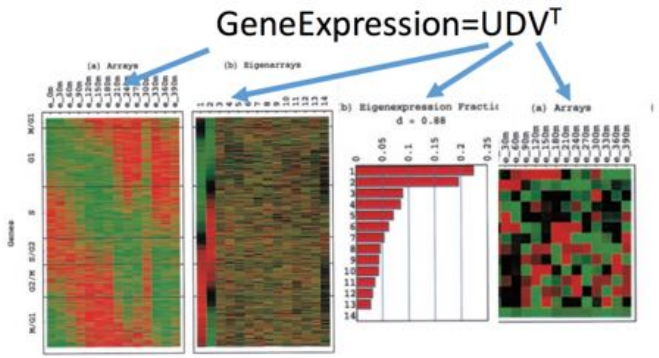
PCA for gene expression

- Columns of V are
  - Principle components
  - Eigengenes/metagenes that span the space of the gene transcriptional responses
- Columns of U are
  - The "loadings", or the correlation between the column and the component
  - Eigenarrays/metaarrays - span the space of the gene transcriptional responses
- Truncating U, V, D to the first k dimensions gives the best k-rank approximation of X

Singular Value Decomposition



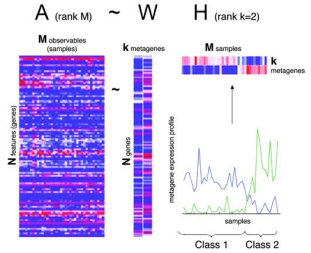
PCA applied to cell cycle data



Singular value decomposition for genome-wide expression data processing and modeling. *PNAS*

Other decomposition techniques

- Non-negative matrix factorization
- $A = WH$ (A, W, H are non-negative)
- H defined a meta-gene space: similar to eigengenes
- Classification can be done in the meta-gene space



Jean-Philippe Brunet et al. PNAS 2004;101:4164-4169

NMF

- Many computational methods
  - Cost function $|A - WH|$
  - Squared error - aka Frobenius norm
  - Kullback–Leibler divergence
- Optimization procedure
  - Most use stochastic initialization, and the results don't always converge to the same answer

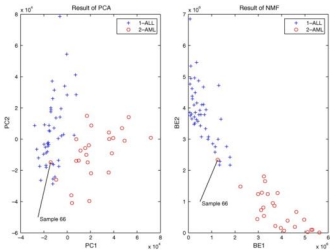
NMF

- $A = WH$: Toy Biological interpretation
- Assume $k = 2$
- We have 2 transcription factors that activate gene signatures $W1$ and $W2$
- H represents the activity of each factor in each sample
- TF effects are additive

NMF

- NMF operates in the original non-negative measurement space
- Highly expressed genes matter more
- Positivity constraint is advantageous: positive correlation among genes is more likely to be biologically meaningful
- NMF may more accurately capture the data generating process

NMF vs. PCA



- Results of PCA vs NMF for reducing the leukemia data with 72 samples in visualization. Sample 66 is mislabeled. However in 2-D display, the reduced data by NMF can clearly show this mistake while that by PCA cannot demonstrate the wrong. ‘PC’ stands for principal component and ‘BE’ means basis experiment.

Weixiang Liu, Kehong Yuan, Datian Ye “**Reducing microarray data via nonnegative matrix factorization for visualization and clustering analysis**” Journal of Biomedical Informatic 2008,

Multidimensional scaling

MDS attempts to

- Identify abstract variables which have generated the inter-object similarity measures
- Reduce the dimension of the data in a non-linear fashion
- Reproduce non-linear higher-dimensional structures on a lower-dimensional display

Kruskal's stress

$$stress = \sqrt{\frac{\sum (d_{ij} - \hat{d}_{ij})^2}{\sum d_{ij}^2}}$$

- Goodness-of-fit - Measures degree of correspondence between distances among points on the MDS map and the matrix input.
- Start with distances d_{ij}
- Fit decreasing numbers $\hat{d}_{ij}$
- Subtract, square, sum
- Take a square root
- Divide by a scaling factor

MDS Basic Algorithm

- Obtain and order the M pairs of similarities
- Try a configuration in q dimensions
  - Determine inter-item distances and reference numbers
  - Minimize Kruskal's stress
- Move the points around to obtain an improved configuration
- Repeat until minimum stress is obtained

Comparison Between PCA, MDS, and SOM

- **PCA** tries to preserve the covariance of the original data
- **MDS** tries to preserve the metric (ordering relations) of the original space
- **SOM** tries to preserve the topology (local neighborhood relations), items projected to nearby locations are similar

How good is your clustering?

Assess cluster fit and stability

- Most often ignored.
- Cluster structure is treated as reliable and precise
- BUT! Clustering is generally VERY sensitive to noise and to outliers
- Measure cluster quality based on how “tight” the clusters are.
- Do genes in a cluster appear more similar to each other than genes in other clusters?

Clustering evaluation methods

- Sum of squares
- Homogeneity and Separation
- Cluster Silhouettes and Silhouette coefficient: how similar genes within a cluster are to genes in other clusters
- Rand index
- Gap statistics
- Cross-validation

Sum of squares

- A good clustering yields clusters where genes have small within-cluster sum-of-squares (and high between-cluster sum-of-squares).

Homogeneity

- **Homogeneity** is calculated as the average distance between each gene expression profile and the center of the cluster it belongs to

$$H_k = \frac{1}{N_g} \sum_{i \in k} d(X_i, C(X_i))$$

N_g - total number of genes in the cluster

Separation

– **Separation** is calculated as the weighted average distance between cluster centers

$$S_{ave} = \frac{1}{\sum_{k \neq l} N_k N_l} \sum_{k \neq l} N_k N_l d(C_k, C_l)$$

Homogeneity and separation

– Homogeneity reflects the compactness of the clusters while S reflects the overall distance between clusters
– Decreasing Homogeneity or increasing Separation suggest an improvement in the clustering results

Variance Ratio Criterion (VCR)

$$VRC_k = (SS_B/(K-1))/(SS_W/(N-K))$$

- SS_B – between-cluster variation
- SS_W – within-cluster variation

The goal is to maximize VRC_k over the clusters

$$\kappa_k = (VRC_{k+1} - VRC_k) - (VRC_k - VRC_{k-1})$$

- Select K to minimize the value of kappaK
- Calinski & Harabasz (1974)

Silhouette

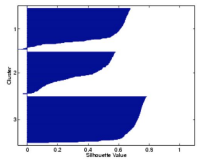
- Good clusters are those where the genes are close to each other compared to their next closest cluster.

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

- $b(i) = min(AVGD_{BETWEEN}(i,k))$
- $a(i) = AVGD_{WITHIN}(i)$
- How well observation i matches the cluster assignment. Ranges $-1 < s(i) < 1$
- Overall silhouette: $SC = \frac{1}{N_g} \sum_{i=1}^{N_g} s(i)$
- Rousseeuw, Peter J. "**Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis.**" Journal of Computational and Applied Mathematics 1987
http://www.sciencedirect.com/science/article/pii/0377042787901257

Silhouette plot

- The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters.
- Silhouette width near +1 indicates points that are very distant from neighboring clusters
- Silhouette width near 0 indicate points that are not distinctly in one cluster or another
- Negative width indicates points are probably assigned to the wrong cluster.



Rand index

Cluster multiple times

- Clustering A: 1, 2, 2, 1, 1
- Clustering B: 2, 1, 2, 1, 1

Compare pairs

- $a :=\ and\ =$, the number of pairs assigned to the same cluster in A and in B
- $b :\neq\ and\ \neq$, … different clusters in A and in B
- $c :\neq\ and\ =$, … same in A, different in B
- $d :=\ and\ \neq$, … same in B, different in A

Rand index

$$R = \frac{a+b}{a+b+c+d}$$

- Adjust the Rand index to make it vary between -1 and 1 (negative if less than expected)
- $AdjRand = (Rand - expect(Rand))/(max(Rand) - expect(Rand))$

Gap statistics

- Cluster the observed data, varying the total number of clusters $k = 1, 2, \ldots K$
- For each cluster, calculate the sum of the pairwise distances for all points

$$D_r = \sum_{i,i' \in C_r} d_{ii'}$$

- Calculate within-cluster dispersion measures

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r$$

Gap statistics

1. Cluster the observed data, varying the total number of clusters from $k = 1, 2, \ldots K$, giving within dispersion measures $W_k, k = 1, 2, \ldots K$.

2. Generate B reference datasets, using the uniform prescription (a) or (b) above, and cluster each one giving within dispersion measures W^*_{kb}, $b = 1, 2, \ldots B$, $k = 1, 2, \ldots K$. Compute the (estimated) Gap statistic:

$$\text{Gap}(k) = (1/B)\sum_b \log(W^*_{kb}) - \log(W_k)$$

3. Let $\bar{l} = (1/B)\sum_b \log(W^*_{kb})$, compute the standard deviation $\text{sd}_k = [(1/B)\sum_b(\log(W^*_{kb}) - \bar{l})^2]^{1/2}$, and define $s_k = \text{sd}_k\sqrt{1 + 1/B}$. Finally choose the number of clusters via

$$\hat{k} = \text{smallest } k \text{ such that } \text{Gap}(k) \geq \text{Gap}(k+1) - s_{k+1}$$

Cross-validation approaches

- Cluster while leave-out k experiments (or genes)
- Measure how well cluster groups are preserved in left out experiment(s)
- Or, measure agreement between test and training set

Clustering validity

- Hypothesis: if the clustering is valid, the linking of objects in the cluster tree should have a strong correlation with the distances between objects in the distance vector

Suppose that the original data $\{X_i\}$ have been modeled using a cluster method to produce a dendrogram $\{T_i\}$; that is, a simplified model in which data that are "close" have been grouped into a hierarchical tree. Define the following distance measures.

- $x(i, j) = |X_i - X_j|$, the ordinary Euclidean distance between the ith and jth observations.
- $t(i, j)$ = the dendrogrammatic distance between the model points T_i and T_j. This distance is the height of the node at which these two points are first joined together.

Then, letting $\bar{x}$ be the average of the $x(i, j)$, and letting $\bar{t}$ be the average of the $t(i, j)$, the cophenetic correlation coefficient c is given by[4]

$$c = \frac{\sum_{i<j}(x(i,j) - \bar{x})(t(i,j) - \bar{t})}{\sqrt{[\sum_{i<j}(x(i,j) - \bar{x})^2][\sum_{i<j}(t(i,j) - \bar{t})^2]}}.$$

WADP - robustness of clustering

- 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

Bittner M. et.al. "**Molecular classification of cutaneous malignant melanoma by gene expression profiling**" Nature 2000
http://www.nature.com/nature/journal/v406/n6795/full/406536A0.html

WADP - robustness of clustering

- Perturb each original gene expression profile by $N(0, 0.01)$
- Re-normalize the data, cluster
- Cluster-specific discrepancy rate: D/M. That is, for the M pairs of genes in an original cluster, count the number of gene pairs, D, that do not remain together in the clustering of the perturbed data, and take their ratio.
- The overall discrepancy ratio is the weighted average of the cluster-specific discrepancy rates.

WADP - robustness of clustering

- If there were originally m_j genes in the cluster j, then there are $M_j = m_j(m_j - 1)/2$ pairs of genes
- In the new clustering, identify how many of these paris (D_j) still remain in the cluster
- Calculate D_j/M_j

$$WADP = \frac{\sum_{j=1}^{k} m_j D_j/M_j}{\sum_{j=1}^{k} m_j}$$

Clustering pitfalls

- Any data – even noise – can be clustered
- It is quite possible for there to be several different classifications of the same set of objects.
- It should be clear that any clustering produced should be related to the features in which the investigator in interested.

Summary