C02

群集分析
Cluster Analysis

吳漢銘

國立政治大學 統計學系

http://www.hmwu.idv.tw

Outlines

- Clustering Analysis
- Visualizing Clustering Results (DR, Heat Map)
- Distance and Similarity Measure
- **Clustering Algorithms**
  - K-Means **(K均值法)**, On-Line K-Means **(線上K均值法)**
  - kmodes, k-medoid, PAM, CLARA
  - Fuzzy C-Means **(模糊C均值法)**
  - Quality Threshold Clustering **(QT群集法)**
  - Hierarchical Clustering **(階層式分群法)**
  - Self-Organizing Maps (SOM) **(自我組織相關圖)**
  - Generated Association Plots **(廣義相關圖)**
- **How Many Clusters? Cluster Validation**

Cluster Analysis

Group a given collection of unlabeled patterns into meaningful clusters.

Data, X

Step1. Feature Extraction

Transformation/Normalization

Dimension Reduction

Similarity/Distance Measures

Step2. Clustering Algorithms

Clusters, y + **Dimension Reduction** + **Visualization** **Graphics Methods**

Step3. Cluster Validation

A Good Review Paper: Daxin Jiang, Chun Tang and Aidong Zhang, (2004), Cluster analysis for gene expression data: a survey, *IEEE Transactions on Knowledge and Data Engineering* 16(11), 1370- 1386.

Hierarchical Clustering

Hierarchical clustering can be perform using agglomerative and divisive approaches. The result is a tree that depicts the relationships between the objects.

- Divisive clustering:
  - begin at step 1 with all the data in one cluster, in each subsequent step a cluster is split off, until there are n clusters.
- Agglomerative clustering:
  - all the objects start apart.
  - There are n clusters at step 0, each object forms a separate cluster.
  - In each subsequent step two clusters are merged, until only cluster is left.

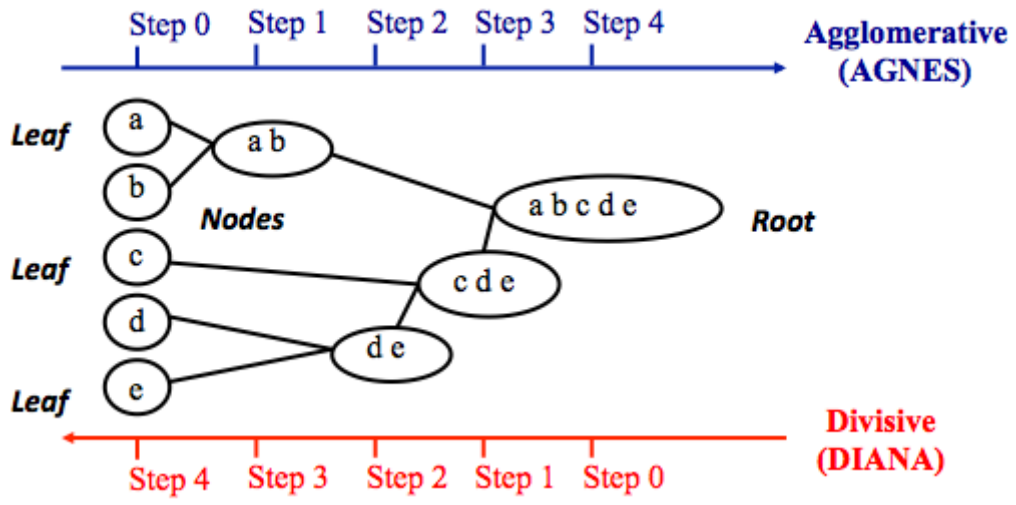


Image Source: http://www.sthda.com/english/wiki/print.php?id=237

Non-Hierarchical Clustering

- 分割演算法 (Partitioning Algorithm)
  - k-means, The EM algorithm, ...
- 密度型演算法 (Density-Based Algorithm)
  - Nearest Neighbor, ...

- NOTES:
  - Only use similarities of instances, without any other requirement of the data.
  - The aim is to find groups such that instances in a group are more similar to each other than instances in different groups.
  - Make sure that all attribute have the same scale.

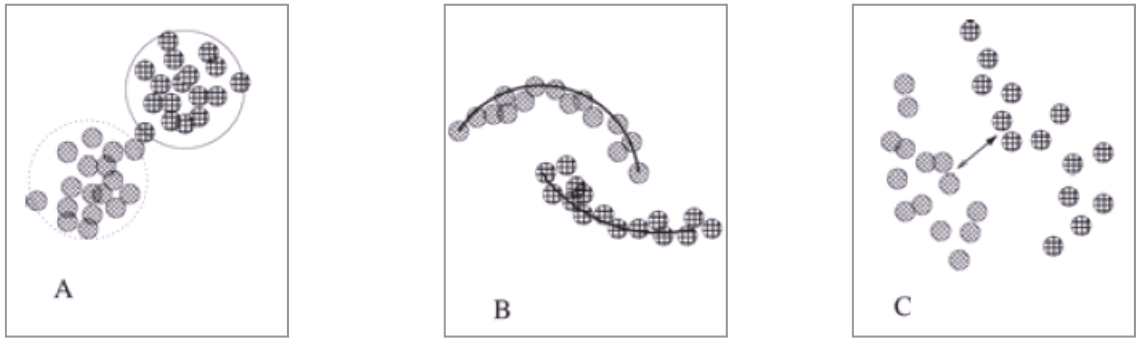
After Clustering

- Supervised learning after clustering
- Dimensionality Reduction vs. Clustering
- The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data.
  - **Marketing**: finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records;
  - **Biology**: classification of plants and animals given their features;
  - **Libraries**: book ordering;
  - **Insurance**: identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds;
  - **City-planning**: identifying groups of houses according to their house type, value and geographical location;
  - **Earthquake studies**: clustering observed earthquake epicenters to identify dangerous zones;
  - **WWW**: document classification; clustering weblog data to discover groups of similar access patterns.

http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/

Two Important Properties

- Most of data has been organized into non-overlapping clusters.
- A good cluster should have a small within variance and large between variance.



Visualizing Clustering Results: Heat Map (1/2)

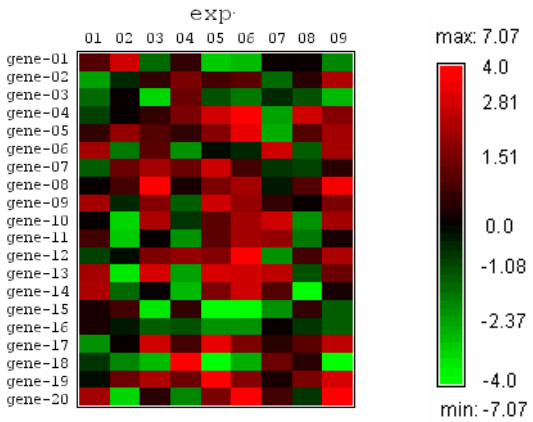
Samples/ conditions

genes

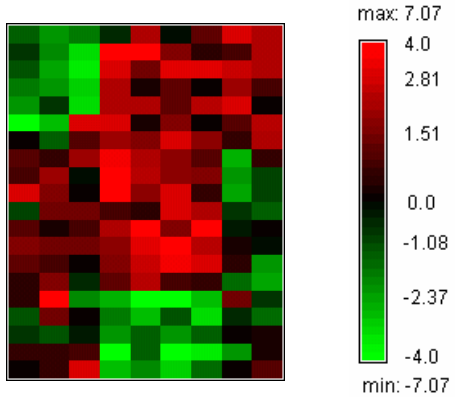
	A							H	I
1	-1.37							3.03	2.48
3	-1.19							2.92	2.48
4	-1.93	-2.28	-3.16			1.49	0.21	2.20	1.03
5	-2.21	-0.79	-3.29	2.55	2.44	1.45	2.68	3.03	0.19
6	-4.14	-2.91	-1.64	3.21	0.37	1.93	0.14	1.27	2.67
7	-1.91	-1.36	-0.44	2.22	1.85	3.11	2.03	0.67	2.40
		0.79	2.25	3.65	2.52	2.09	1.13	-2.59	0.67
		2.33	-0.07	3.89	2.72	2.13	1.75	-2.17	-0.90
		1.85	0.21	7.07	2.01	3.05	0.76	-2.58	-1.04
11	-2	1.65	1.53	0.95	0.60	3.12	2.52	-0.77	-1.40
12	1.21	0.24	1.04	2.50	3.69	1.81	3.98	-0.33	0.11
13	1.74	1.60	1.70	2.02	3.45	4.46	2.69	0.41	-0.09
14	1.34	1.06	0.06	1.81	2.90	3.64	3.04	0.49	-2.33
15	0.57	1.81	-0.47	1.40	2.70	0.99	0.82	-1.61	-2.56
16	0.61	4.22	-2.03	-2.61	-4.00	-4.64	-2.92	1.55	-0.71
17	-1.13	1.64	0.01	-1.77	-2.85	-1.24	-3.41	-0.59	-1.64
18	-0.86	-1.17	-0.41	-2.20	-1.30	-2.37	-1.41	0.08	0.25
19	0.75	0.66	1.04	-4.26	-1.41	-3.99	-3.53	-2.17	0.34
20	0.15	0.68	3.18	-2.86	-2.01	-3.18	-1.58	0.10	1.28

Color mapping

Without ordering

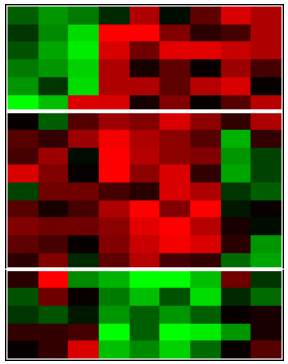


Ordering/Seriation/ Clustering

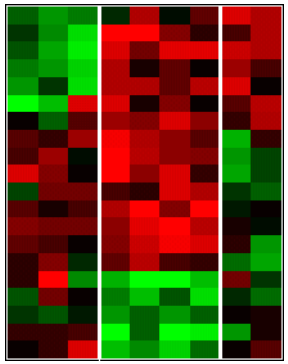


Visualizing Clustering Results: Heat Map (2/2)

Gene-based clustering

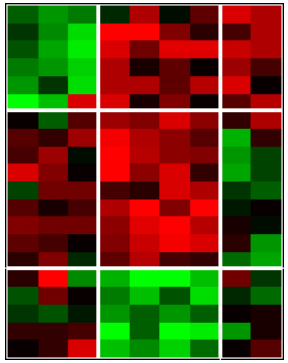


Sample-based clustering

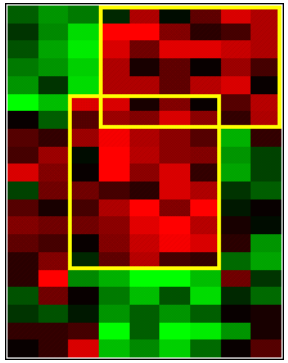


e.g., K-means, SOM, Hierarchical Clustering, Model-based clustering,…

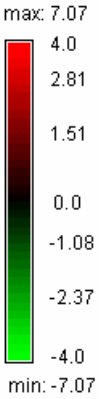
Twoway-based clustering

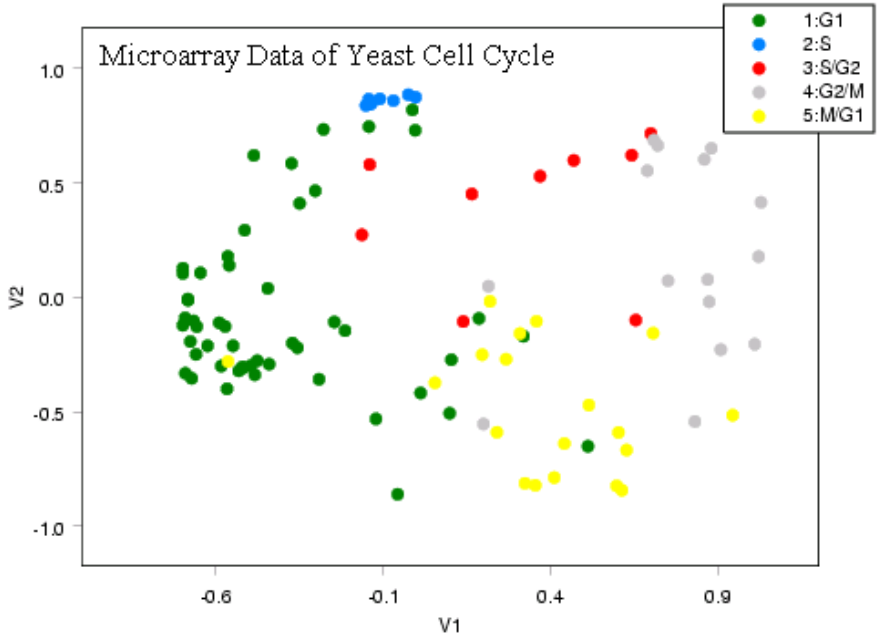


Subspace clustering



e.g., Bi-clustering

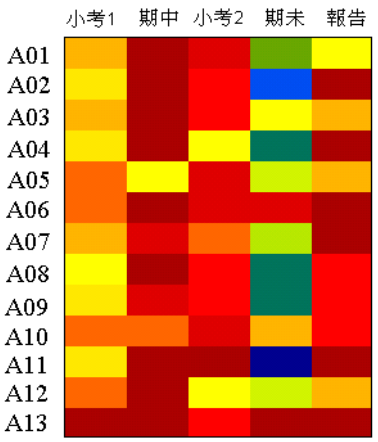




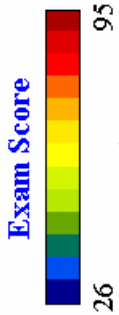
Dimension Reduction

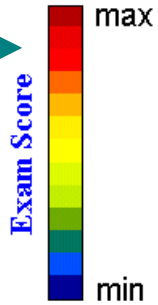
Heat Map: Data Image, Matrix Visualization

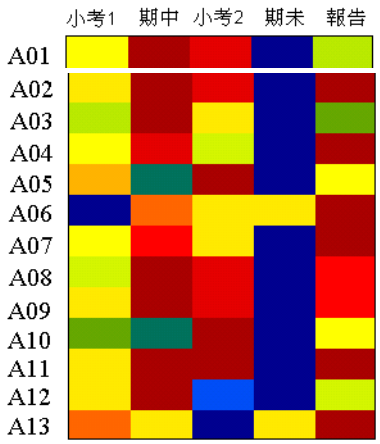
	A	B	C	D	E	F
1	學號	小考1	期中考	小考2	期末考	報告
2	A01	69	92	85	45	62
3	A02	66	90	83	36	90
4	A03	72	92	80	62	70
5	A04	68	90	60	37	95
6	A05	74	60	86	54	70
7	A06	77	90	88	88	95
8	A07	73	88	77	51	95
9	A08	61	90	84	40	82
10	A09	66	88	82	39	80
11	A10	76	75	87	72	80
12	A11	64	90	90	26	95
13	A12	75	90	60	55	70
14	A13	92	90	83	90	95



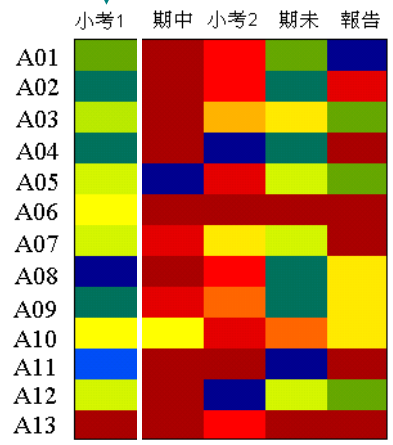
Range Matrix Condition







Range Row Condition

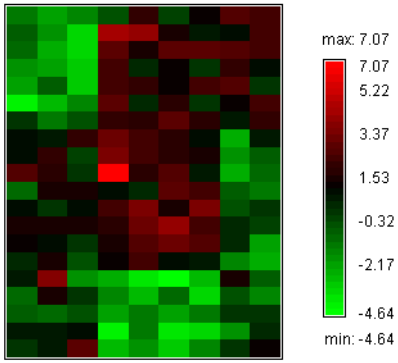


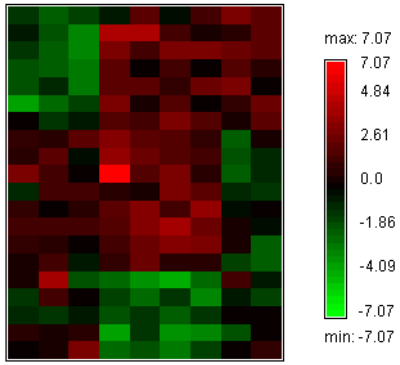
Range Column Condition

What about this one?

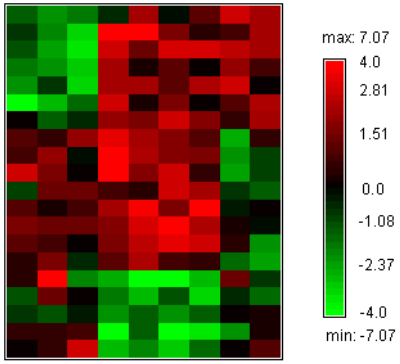
Heat Map: Display Conditions

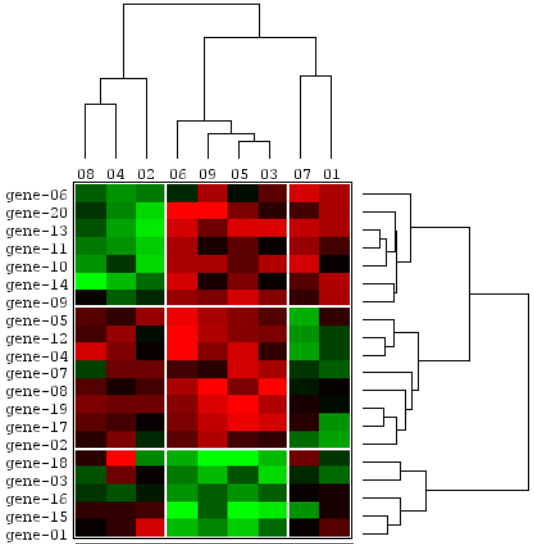
	A	B	C	D	E	F	G	H	I
1	-1.37	-2.30	-1.80	-0.55	2.45	-0.13	1.49	3.03	2.48
2	-0.68	-2.11	-3.42	4.67	4.57	1.75	0.61	0.92	2.52
3	-1.19	-2.49	-3.66	3.14	1.70	3.29	3.33	2.92	2.48
4	-1.93	-2.28	-3.16	2.51	0.32	1.49	0.21	2.20	1.03
5	-2.21	-0.79	-3.29	2.55	2.44	1.45	2.68	3.03	0.19
6	-4.14	-2.91	-1.64	3.21	0.37	1.93	0.14	1.27	2.67
7	0.21	-1.36	-0.44	2.22	1.85	3.11	2.03	0.67	2.40
8	1.13	0.79	2.25	3.65	2.52	2.09	1.13	-2.59	0.67
9	0.95	2.33	-0.07	3.89	2.72	2.13	1.75	-2.17	-0.90
10	3.04	1.85	0.21	7.07	2.01	3.05	0.76	-2.58	-1.04
11	-1.02	1.65	1.53	0.95	0.60	3.12	2.52	-0.77	-1.40
12	1.21	0.24	1.04	2.50	3.69	1.81	3.98	-0.33	0.11
13	1.74	1.60	1.70	2.02	3.45	4.46	2.69	0.41	-0.09
14	1.34	1.06	0.06	1.81	2.90	3.64	3.04	0.49	-2.33
15	0.57	1.81	-0.47	1.40	2.70	0.99	0.82	-1.61	-2.56
16	0.61	4.22	-2.03	-2.61	-4.00	-4.64	-2.92	1.55	-0.71
17	-1.13	1.64	0.01	-1.77	-2.85	-1.24	-3.41	-0.59	-1.64
18	-0.86	-1.17	-0.41	-2.20	-1.30	-2.37	-1.41	0.08	0.25
19	0.75	0.66	1.04	-4.26	-1.41	-3.99	-3.53	-2.17	0.34
20	0.15	0.68	3.18	-2.86	-2.01	-3.18	-1.58	0.10	1.28





Center Matrix Condition





fviz_cluster {factoextra}

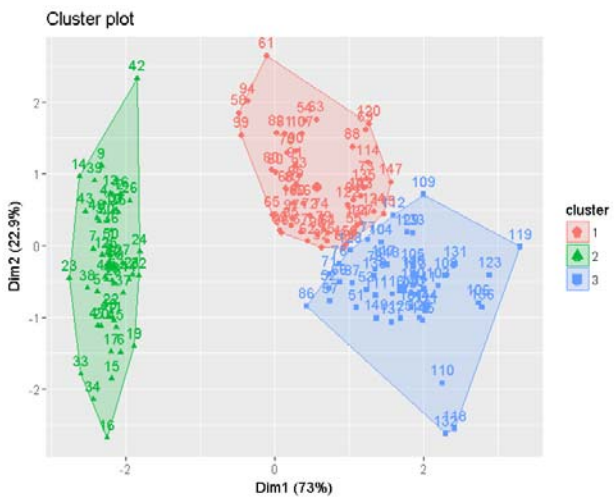
Visualize Clustering Results

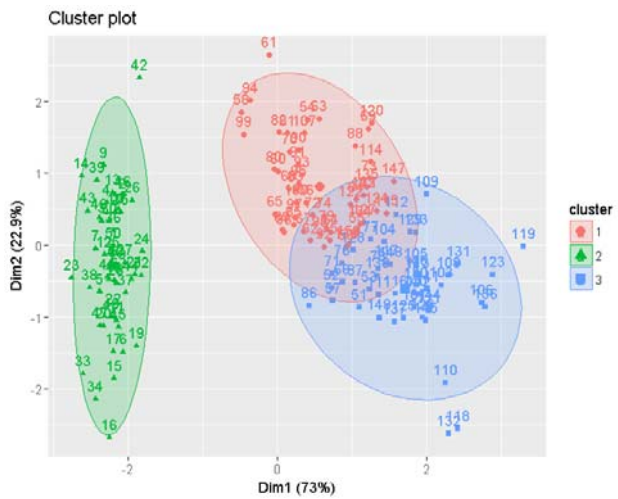
Usage

```
fviz_cluster(object, data = NULL, choose.vars = NULL, stand = TRUE,
  axes = c(1, 2), geom = c("point", "text"), repel = FALSE,
  show.clust.cent = TRUE, ellipse = TRUE, ellipse.type = "convex",
  ellipse.level = 0.95, ellipse.alpha = 0.2, shape = NULL,
  pointsize = 1.5, labelsize = 12, main = "Cluster plot", xlab = NULL,
  ylab = NULL, outlier.color = "black", outlier.shape = 19,
  ggtheme = theme_grey(), ...)
```

fviz_cluster provides ggplot2-based elegant visualization of partitioning methods including: **kmeans {stats}; pam, clara and fanny {cluster}; dbscan {fpc}; Mclust {mclust}; HCPC {FactoMineR}; hkmeans {factoextra}.** Observations are represented by points in the plot, using **principal components** if ncol(data) > 2. An ellipse is drawn around each cluster.

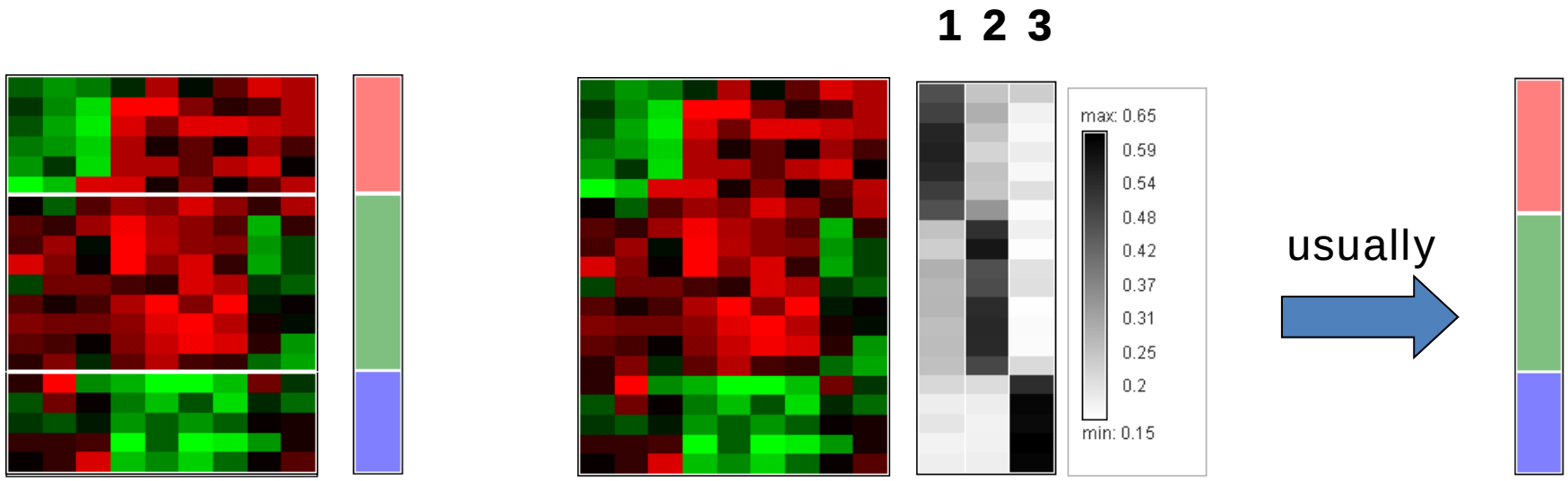
```
> iris.scaled <- scale(iris[, -5])
> iris.km <- kmeans(iris.scaled,  centers=3)
> fviz_cluster(iris.km, iris[, -5])
> fviz_cluster(iris.km, iris[, -5], ellipse.type = "norm")
```





Hard & Soft Clustering

- Hard Clustering: K-means, SOM, Hierarchical Clustering
- Soft Clustering: Fuzzy c-means, model-based clustering, ...



Class Labels **Class Membership**

R functions and packages for cluster analysis

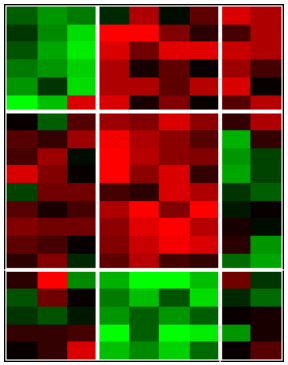
Package	Functions	Description
cclust		Convex clustering methods
class	`SOM`	Self-organizing maps
cluster	`agnes`	AGglomerative NESting
	`clara`	Clustering LARge Applications
	`diana`	DIvisive ANAlysis
	`fanny`	Fuzzy Analysis
	`mona`	MONothetic Analysis
	`pam`	Partitioning Around Medoids
e1071	`bclust`	Bagged clustering
	`cmeans`	Fuzzy C-means clustering
flexmix		Flexible mixture modeling
fpc		Fixed point clusters, clusterwise regression and discriminant plots
hopach	`hopach`, `boothopach`	Hierarchical Ordered Partitioning and Collapsing Hybrid
mclust		Model-based cluster analysis
stats	`hclust`, `cophenetic`	Hierarchical clustering
	`heatmap`	Heatmaps with row and column dendrograms
	`kmeans`	k-means

K.S. Pollard, M.J. van der Laan (2005). **Cluster Analysis of Genomic Data with Applications in R**. in *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, Springer.

Clustering Analysis in Microarray Experiments

Goals

- Find natural classes in the data
- Identify new classes/gene correlations
- Refine existing taxonomies
- Support biological analysis/discovery



- cluster genes based on samples profiles
- cluster samples based on genes profiles

Hypothesis:

- genes with similar biological function have similar expression profiles.

Characteristic of Microarray Data:

- High-throughput (large-scale)
- Noise
- Outliers
- Biological Knowledge

Distance and Similarity Measure

Cov	x1	x2	x3	x4		x p
x1	1.00	0.48	0.10	-0.10		-0.28
x2	0.48	1.00	0.41	0.22		-0.23
x3	0.10	0.41	1.00	0.36		-0.05
x4	-0.10	0.22	0.36	1.00		0.10
x p	-0.28	-0.23	-0.05	0.10		1.00

Proximity Matrix

Data Matrix

$x \quad y$

Data	x1	x2	x3	x4	...	x p
subject01	-0.48	-0.42	0.87	0.92		-0.18
subject02	-0.39	-0.58	1.03	1.21		-0.33
subject03	0.87	0.25	-0.17	0.18		-0.44
subject04	1.57	1.03	1.22	0.31		-0.49
subject05	-1.15	-0.86	1.21	1.62		0.16
subject06	0.04	-0.12	0.31	0.16		-0.06
subject07	2.95	0.45	-0.40	-0.66		-0.38
subject08	-1.22	-0.74	1.34	1.50		0.29
subject09	-0.73	-1.06	-0.79	-0.02		0.44
subject10	-0.58	-0.40	0.13	0.58		0.02
subject11	-0.50	-0.42	0.66	1.05		0.06
subject12	-0.86	-0.29	0.42	0.46		0.10
subject13	-0.16	0.29	0.17	-0.28		-0.55
subject14	-0.36	-0.03	-0.03	-0.08		-0.25
subject15	-0.72	-0.85	0.54	1.04		0.24
subject16	-0.78	-0.52	0.26	0.20		0.48
subject17	0.60	-0.55	0.41	0.45		-0.66
⋮						
subject n	-2.29	-0.64	0.77	1.60		0.55
mean	0.07	-0.04	0.44	0.31	...	-0.21

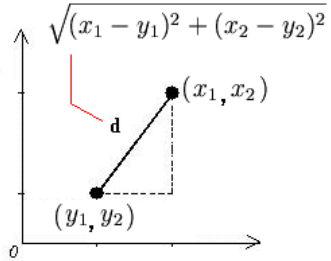
Pearson Correlation Coefficient

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

$$x = (x_1, x_2, \cdots, x_n)$$

$$y = (y_1, y_2, \cdots, y_n)$$

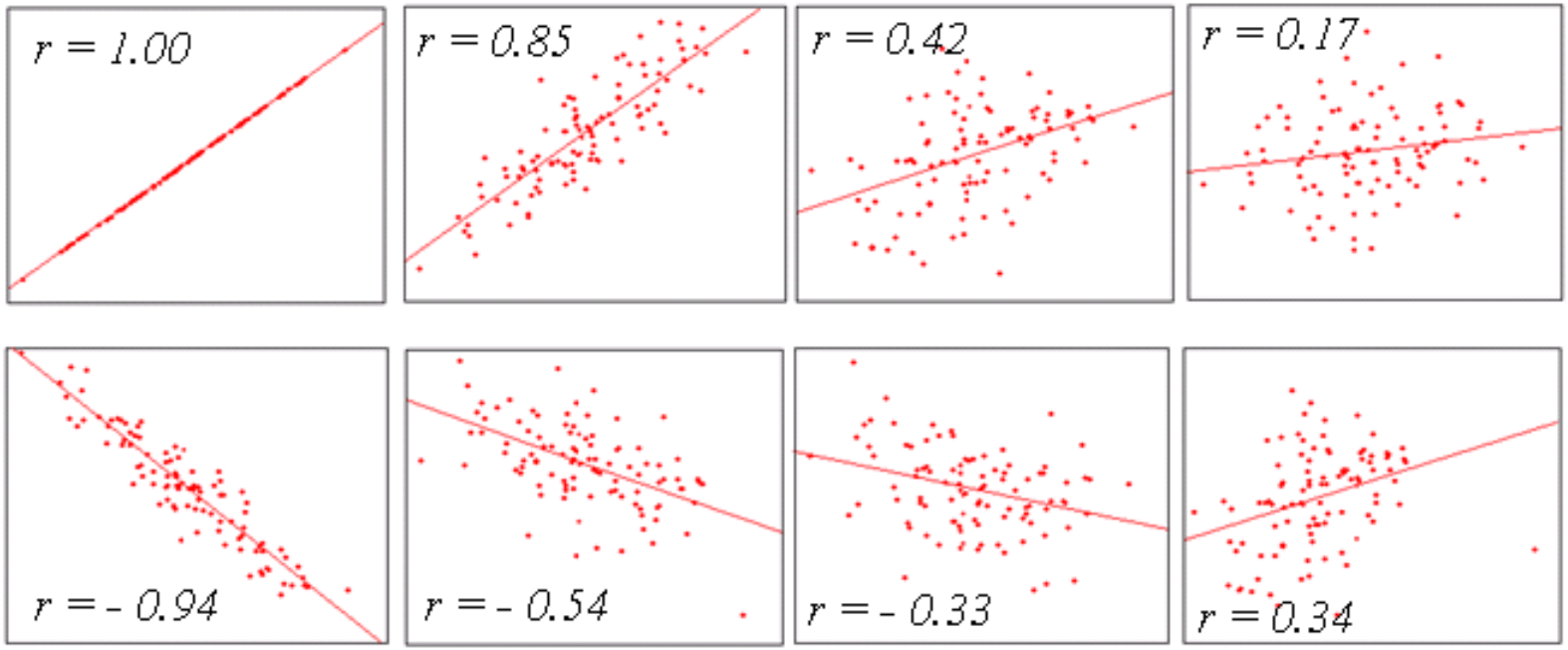
Euclidean Distance



$$d_{xy} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

- The standard transformation from a similarity matrix C to a distance matrix D is given by $d_{rs} = (c_{rr} - 2c_{rs} + c_{ss})^{1/2}$.
- (Eisen *et al.* 1998) $d_{rs} = 1 - c_{rs}$
- Other transformations (Chatfield and Collins 1980, Section 10.2)

Pearson Correlation Coefficient




```
dist(x, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
     method: one of "euclidean", "maximum", "manhattan", "canberra", "binary"
or "minkowski" distance measure.
cor(x, y = NULL, use = "everything",
    method = c("pearson", "kendall", "spearman"))
```

More Similarity Measures (1/4)

Dissimilarity/Similarity Measure for Quantitative Data

Similarity	Formula
Pearson correlation	$s(i,j) = \dfrac{\mathrm{cov}(x_i, x_j)}{\sqrt{\mathrm{var}(x_i)\,\mathrm{var}(x_j)}}$
Spearman correlation (r_i is ranked x_i)	$s(i,j) = \dfrac{\mathrm{cov}(r_i, r_j)}{\sqrt{\mathrm{var}(r_i)\,\mathrm{var}(r_j)}}$
Kendall's Tau	$s(i,j) = \dfrac{1}{\binom{p}{2}} \sum_{k \neq k'} sign\ [(x_{ik} - x_{ik'})(x_{jk} - x_{jk'})]$

All indices range from -1 to +1

Kendall's tau

Two pairs of observation (x_i, y_i) and (x_j, y_j)

- C: concordant pair: $(x_j - x_i)(y_j - y_i) > 0$
- D: discordant pair: $(x_j - x_i)(y_j - y_i) < 0$
- tie:

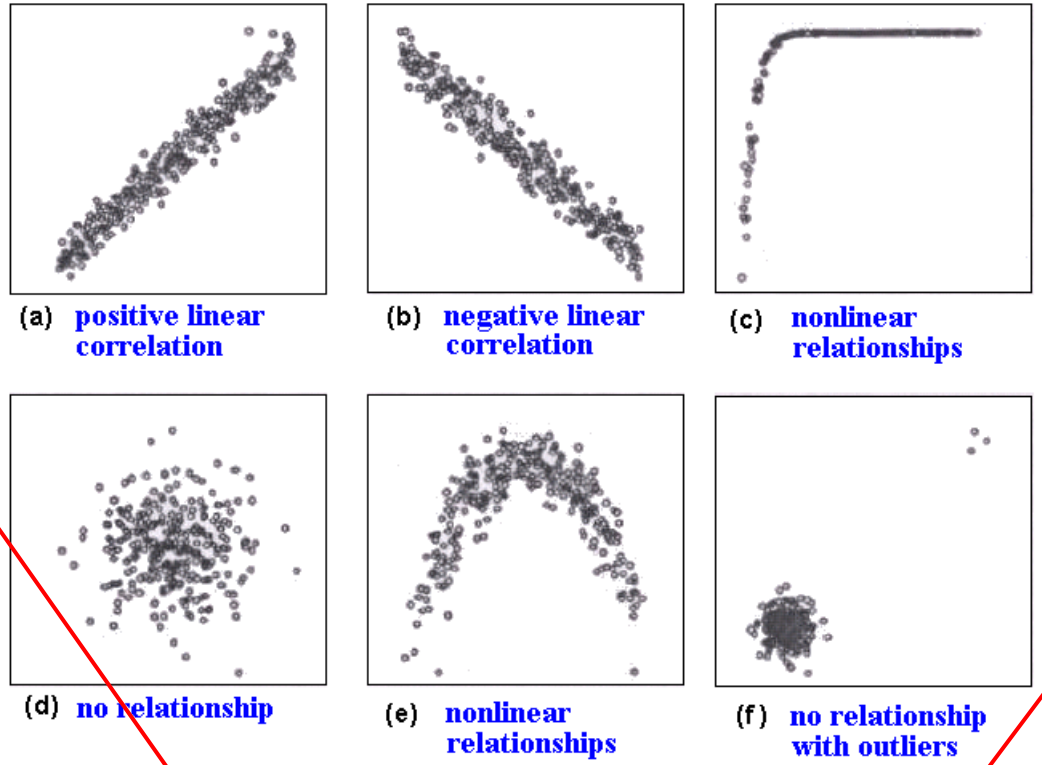
E_y: extra y pair in x's: $(x_j - x_i) = 0$

E_x : extra x pair in y's: $(y_j - y_i) = 0$

$$\tau = \frac{C - D}{\sqrt{C + D - E_y}\sqrt{C + D - E_x}}$$

More Similarity Measures (2/4)

measures the strength of a linear relationship



measure any monotonic relationship between two variables

non-monotonic, fail to detect the existence of a relationship

Data	*Pearson's rho*	*Spearman's rho*	*Kendall's tau*
(a)	0.98	0.98	0.87
(b)	-0.98	-0.98	-0.87
(c)	0.50	0.99	0.98
(d)	-0.02	-0.03	-0.02
(e)	-0.06	-0.02	-0.02
(f)	0.68	0.00	0.00

more robust

More Similarity Measures (3/4)

- Pearson's rho measures the strength of a linear relationship [(a), (b)].
- Spearman's rho and Kendall's tau measure any monotonic relationship between two variables [(a), (b) ,(c)].
- If the relationship between the two variables is non-monotonic, all three correlation coefficients fail to detect the existence of a relationship [(e)].
- Both Spearman's rho and Kendall's tau are rank-based non-parametric measures of association between variable X and Y.
- The rank-based correlation coefficients are more robust against outliers.
- Algorithm they use different logic for computing the correlation coefficient, they seldom lead to markedly different conclusions (Siegel and Castellan, 1988).

More Similarity Measures (4/4)

Pearson Correlation	Calculate the mean of all elements in vector $\mathbf{a}$. Then subtract that value from each element in $\mathbf{a}$. Call the resulting vector $\mathbf{A}$. Do the same for $\mathbf{b}$ to make a vector $\mathbf{B}$. *Result* = $\mathbf{A}.\mathbf{B}/(\lvert\mathbf{A}\rvert\lvert\mathbf{B}\rvert)$
Distance	Distance is not a correlation at all, but a measurement of dissimilarity. Distance is the measurement of Euclidian distance between the expression profile for gene $\mathbf{A}$ (defined by its expression values for each point in N-dimensional space, where N is the number of conditions with data in your experiment) and the expression profile for gene $\mathbf{B}$. *Result* = $\lvert\mathbf{a}-\mathbf{b}\rvert$ divided by the square root of the number of conditions with data
Spearman Correlation	Order all the elements of vector $\mathbf{a}$. Use this order to assign a rank to each element of $\mathbf{a}$. Make a new vector $\mathbf{a}'$ where the ith element in $\mathbf{a}'$ is the rank of a_i in $\mathbf{a}$. Now make a vector $\mathbf{A}$ from $\mathbf{a}'$ in the same way as $\mathbf{A}$ was made from $\mathbf{a}$ in the Pearson Correlation. Similarly, make a vector $\mathbf{B}$ from $\mathbf{b}$. *Result* = $\mathbf{A}.\mathbf{B}/(\lvert\mathbf{A}\rvert\lvert\mathbf{B}\rvert)$

Source: Chapter 14, GeneSpring Manual 7.2

K-Means Clustering (1/3)

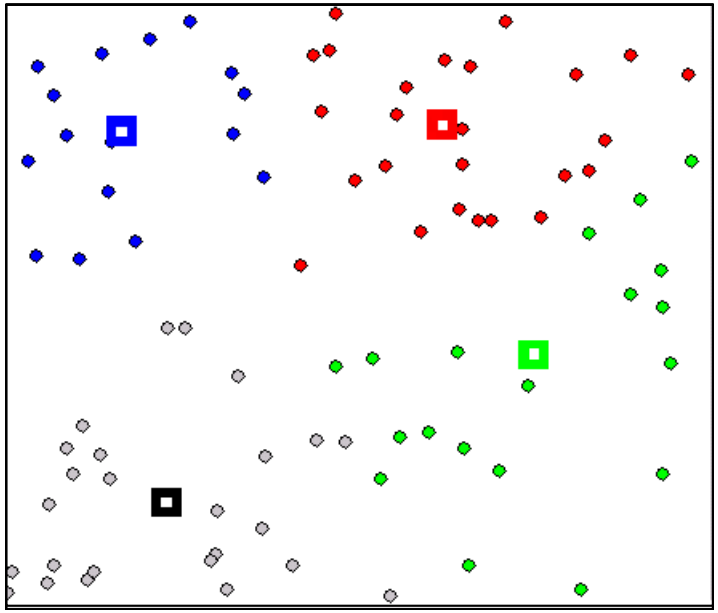
- K-means is a partition methods for clustering.
- Data are classified into k groups as specified by the user.
- Two different clusters cannot have any objects in common, and the k groups together constitute the full data set.

The K-Means Algorithm

1. The data points are randomly assigned to one of the K clusters.
2. The position of the K centroids are determined (initial group centroids).
3. For each data point:
   - Calculate the distance from the data point to each cluster.
   - Assign data point to the cluster that has the closest centroid.
4. Repeat the above step until the centroids no longer move.

The choice of initial partition can greatly affect the final clusters that result.

centroid (mean)



Converged

K-Means Clustering (2/3)

- The best reference vectors are those that minimize the total reconstruction error.
- Distanvatge:
  - a local search.
  - Final $\boldsymbol{m}_i$ highly depend on the initial $\boldsymbol{m}_i$.

(1) initialize

Initialize $\boldsymbol{m}_i, i = 1, \ldots, k$, for example, to k random $\boldsymbol{x}^t$

Repeat

(2) minimize

For all $\boldsymbol{x}^t \in \mathcal{X}$

$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

(3) estimate

For all $\boldsymbol{m}_i, i = 1, \ldots, k$

$$\boldsymbol{m}_i \leftarrow \sum_t b_i^t \boldsymbol{x}^t / \sum_t b_i^t$$

(4) stabilize

Until $\boldsymbol{m}_i$ converge

K-Means Clustering (3/3)

- When X_t is represented by m_i, there is an error that is proportional to the distance.
- Find k reference vectors m_j (prototypes/codebook vectors/codewords) which best represent data.
- K-means clustering a special case of the EM algorithm.

- Use nearest (most similar) reference:

$$\|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\|$$

- Reconstruction error

$$E(\{\boldsymbol{m}_i\}_{i=1}^k | \mathcal{X}) = \sum_t \sum_i b_i^t \|\boldsymbol{x}^t - \boldsymbol{m}_i\|^2$$

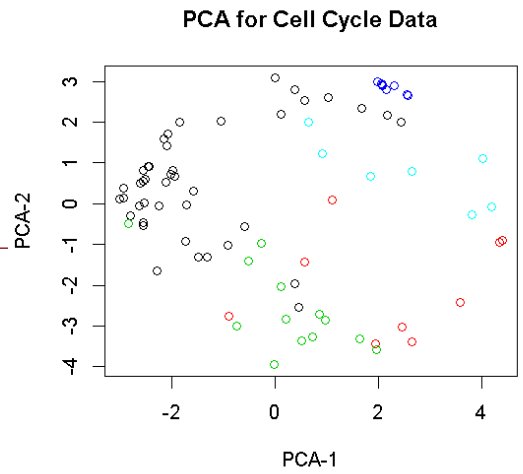
$$b_i^t = \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

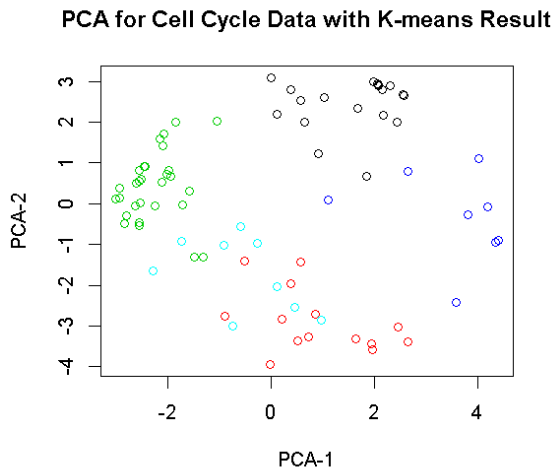
K-means for clustering
=> find groups in the data
=> groups are represented by their centers.

K-Means' Results Shown on the DR Space

```
kmeans(x, centers, iter.max = 10, nstart = 1,
       algorithm = c("Hartigan-Wong", "Lloyd", "Forgy",
                     "MacQueen"), trace=FALSE)
## S3 method for class 'kmeans'
fitted(object, method = c("centers", "classes"), ...)
```

```
> no.group <- 5
> no.iter <- 20
> cell.matrix  <- read.table("YeastCellCycle_alpha.txt",
+ header=TRUE, row.names=1)
> n <- dim(cell.matrix)[1]
> p <- dim(cell.matrix)[2]-1
> cell.data <- cell.matrix[,2:p+1]
> gene.phase <- cell.matrix[,1]
> phase <- unique(gene.phase)
> phase.name <- c("G1", "S", "S/G2", "G2/M", "M/G1")
> cell.sdata <- t(scale(t(cell.data)))
>
> cell.kmeans <- kmeans(cell.sdata, no.group, no.iter)
> plot(cell.sdata[,1:4], col = cell.kmeans$cluster)
> ## PCA
> pca.dim <- princomp(cell.sdata)$scores
> plot(pca.dim[,1], pca.dim[,2],  main="PCA for Cell Cycle Data",
+ xlab="PCA-1", ylab="PCA-2", col=gene.phase)
> plot(pca.dim[,1], pca.dim[,2],  main="PCA for Cell Cycle Data with K-means Result",
xlab="PCA-1", ylab="PCA-2", col=cell.kmeans$cluster)
```



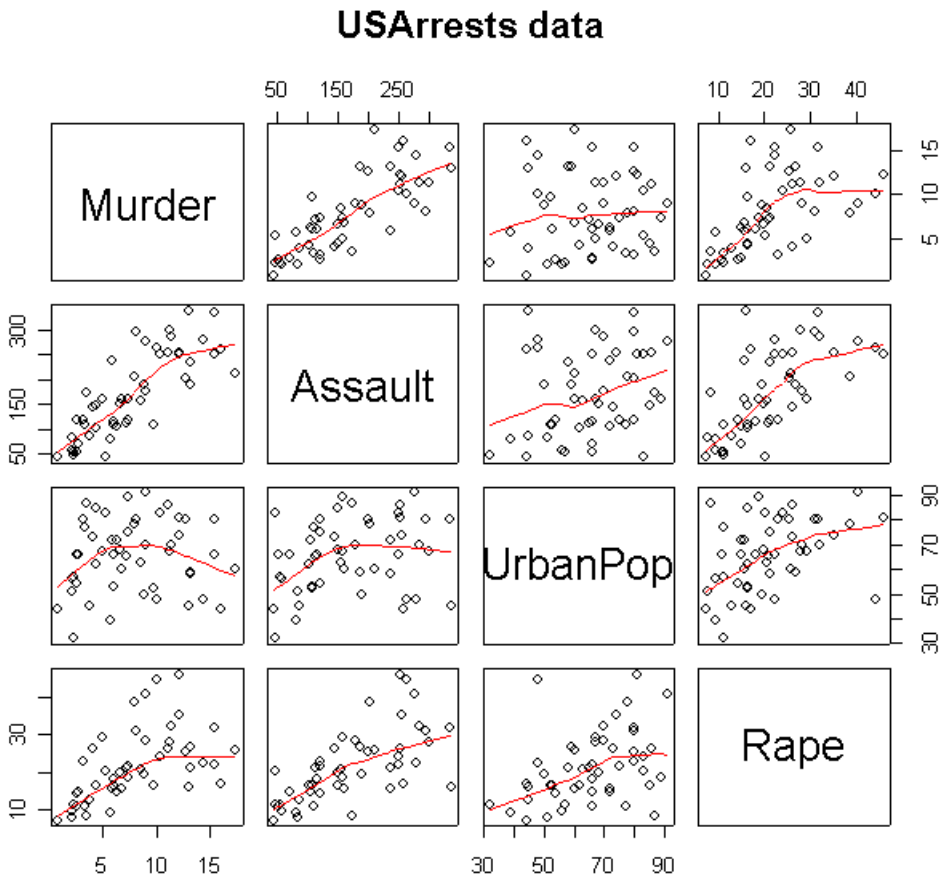


Example: Violent Crime Rates by US State

- This data set contains statistics, in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. Also given is the percent of the population living in urban areas.

- A data frame with 50 observations on 4 variables.

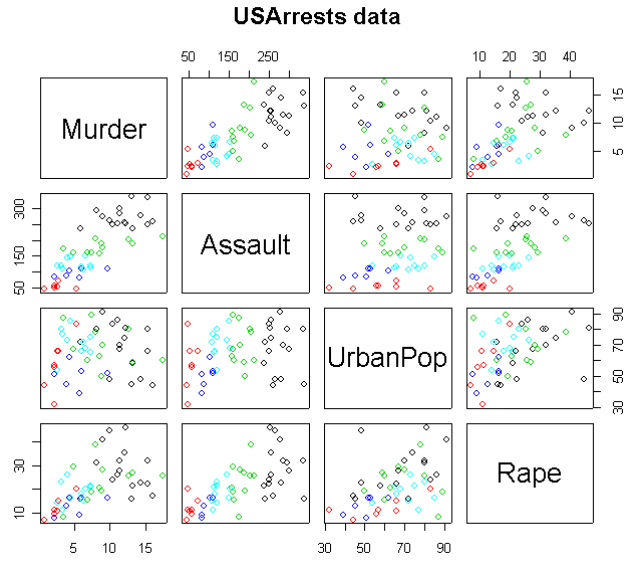
- [,1] Murder arrests (per 100,000)
- [,2] Assault arrests (per 100,000)
- [,3] Percent urban population
- [,4] Rape arrests (per 100,000)

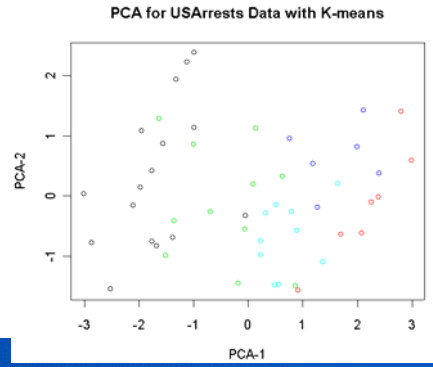


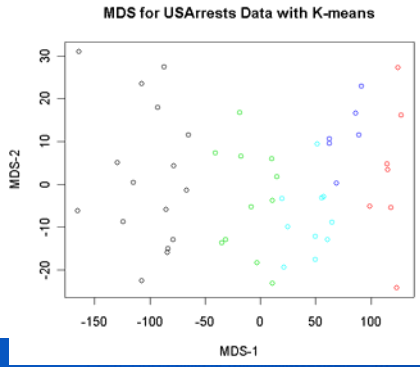

```
data(USArrests)
pairs(USArrests, panel = panel.smooth, main = "USArrests data")
```

R: kmeans

```
no.group <- 5
no.iter <- 20
USArrests.kmeans <- kmeans(USArrests, no.group, no.iter)
plot(USArrests, col = USArrests.kmeans$cluster,
       main = "K-means: USArrests data")

## PCA
USArrests.pca <- princomp(USArrests, cor=TRUE, scores=TRUE)
pca.dim1 <- USArrests.pca$scores[,1]
pca.dim2 <- USArrests.pca$scores[,2]
plot(pca.dim1, pca.dim2,  main="PCA for USArrests Data
      with K-means", xlab="PCA-1", ylab="PCA-2",
      col=USArrests.kmeans$cluster)

## MDS
USArrests.mds<- cmdscale(dist(USArrests))
mds.dim1 <- USArrests.mds[,1]
mds.dim2 <- USArrests.mds[,2]
plot(mds.dim1, mds.dim2, xlab="MDS-1", ylab="MDS-2", main="MDS for USArrests Data with
K-means", col = USArrests.kmeans$cluster)
```





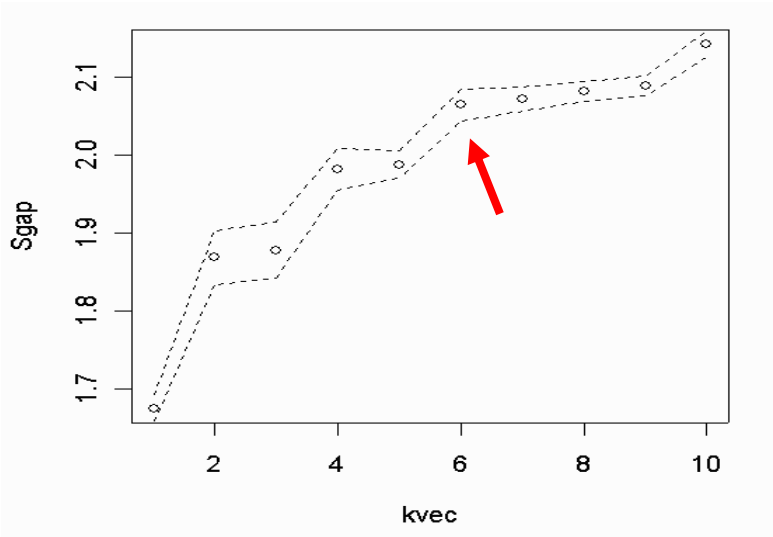


Example

- Data

Baseline: Culture Medium (CM-00h)
OH-04h, OH-12h, OH-24h
CA-04h, CA-24h
SO-04h, SO-24h

- A set of 359 genes was selected for clustering.



J. R. Statist. Soc. B (2001)
63, *Part* 2, *pp.* 411–423

Estimating the number of clusters in a data set via the gap statistic

Robert Tibshirani, Guenther Walther and Trevor Hastie

Stanford University, USA

BMC Genomics 2008, **9**:479 http://www.biomedcentral.com/1471-2164/9/479

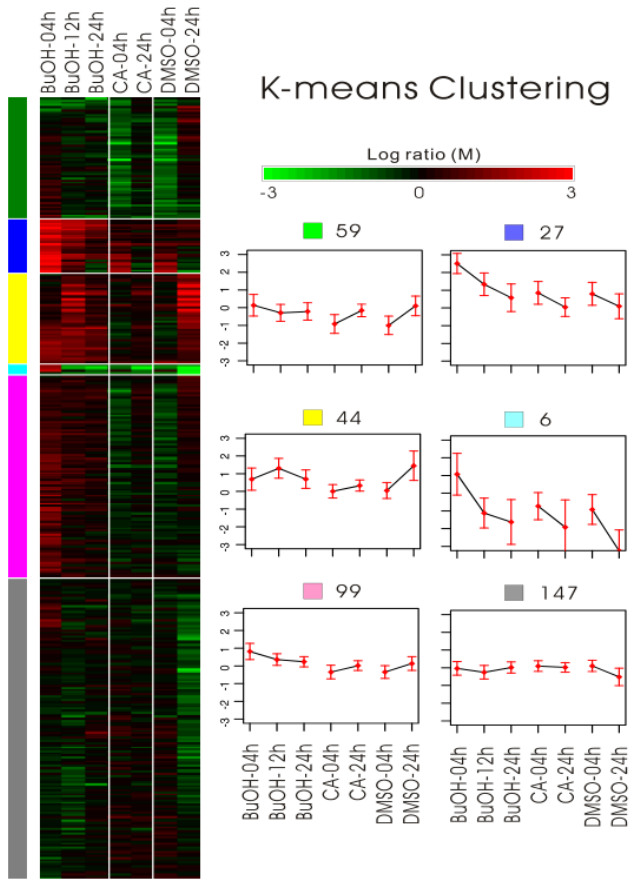
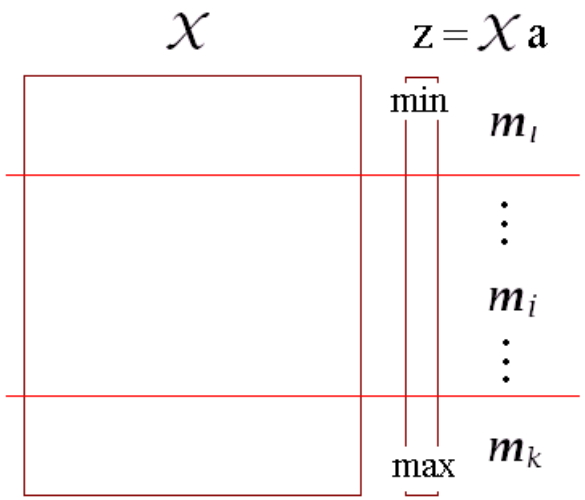


Figure 3
Comparative analysis of differential gene expression patterns in immature dendritic cells treated with [BF/S+L/Ep] and specific phytocompounds by K-means clustering analyses. Analysis of test iDC samples (including vehicle control [+DMSO only]) involved comparing the treatment values with untreated values (i.e., zero hour-treated). The left panel shows the heat map of the resulting clustered gene expression. The number of genes involved and the mean profile for each cluster is in the right panel. The mean profile was superimposed with error bars showing ± 1 standard error of the mean.

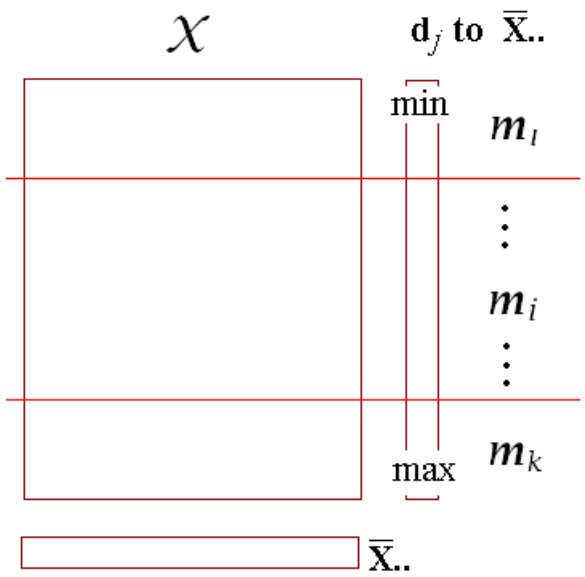
Initialization of K-means

1. Randomly select k instances from X.
2. mean{X} + (small random vector)$_i$
3. z=Xa (PCA).
4. (1) ordering the distances of data points by sample means,
   (2) {1+n(G_i-1)/K}-th point is selected to be the initial cluster centroid.



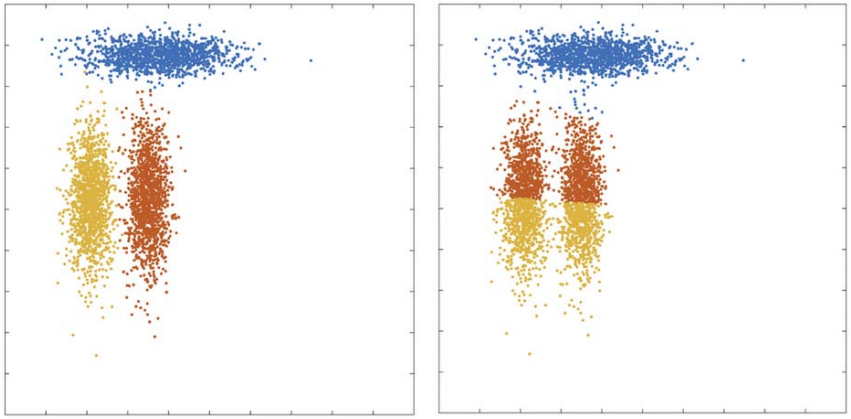
Leader Cluster Algorithm

- Incremental: an instance far away from existing m_i
  => new center at that point
- A center covers a large number of instance
  => split into two centers
- A center cover too few instance
  => removed

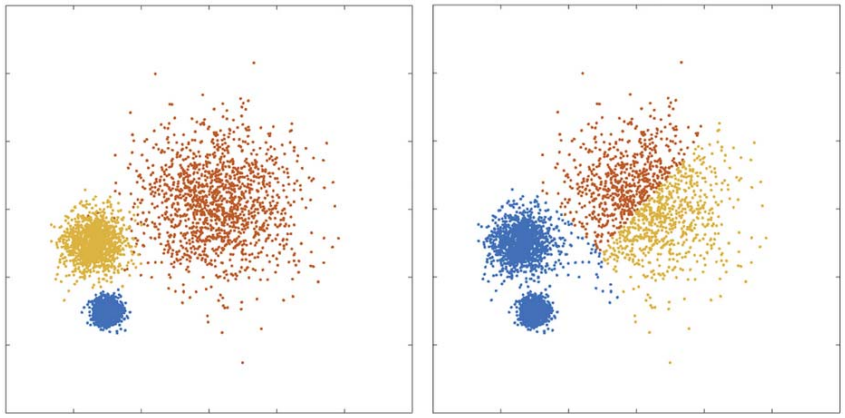


Properties of K-Means

- K-means is by far the most popular algorithm due to its interpretability, easy construction, and appealing computing speed.
- It aims at minimizing the within-cluster variability, given a pre-specified number of data groups
- The use of **Euclidean** distances at Step 1 implies that the performance of the algorithm is optimal when clusters have approximately spherical shapes and similar sizes.



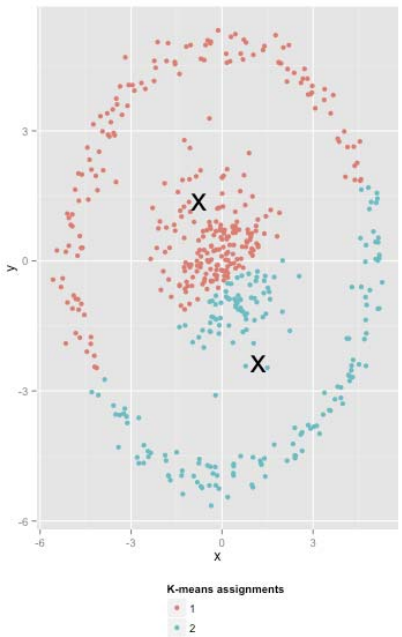
(a) Generated synthetic data (b) K-means



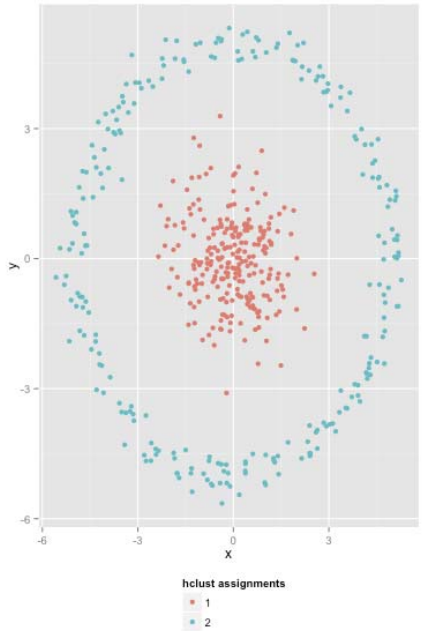
(a) Generated synthetic data (b) K-means

http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0162259

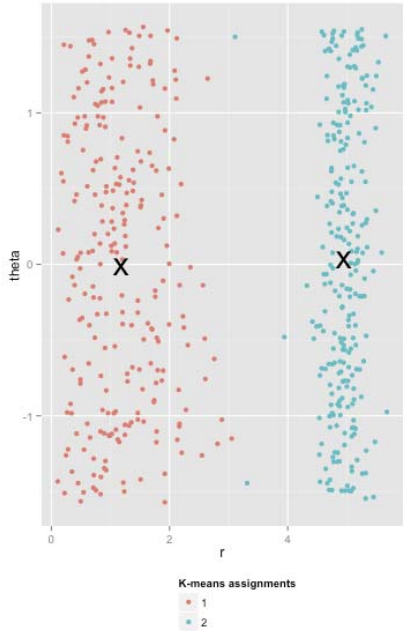
More examples



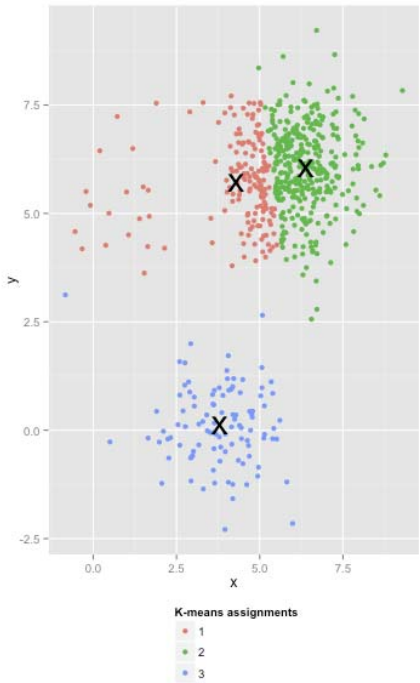
K-means



single-linkage
hierarchical clustering



K-means
(polar coordinates)



sizes: 20, 100, 500

http://varianceexplained.org/r/kmeans-free-lunch/

Assumptions are where your power comes from!

Competitive Learning

- **Online learning:**
  - Don’ t have the whole sample at hand during training.
  - Receive instances one by one and update model parameters as we get them.

- **Competitive learning:**
  - Groups compete among themselves.
  - Winner-take-all: one group wins and gets updated, and the others are not updated at all.

- **Advantage:**
  - Do not need extra memory to store the whole training set.
  - Updates at each step are simple to implement.
  - Model adapts itself to changes of the input distribution automatically.

Online K-Means

- Batch algorithm: all $\boldsymbol{m}_j$, j=1,..,k, compete and $\boldsymbol{m}_i$ wins the competition.
- The calculation of $\boldsymbol{b}$ and update of $\boldsymbol{m}$ are iterated until convergence.

- Reconstruction error

$$E(\{\boldsymbol{m}_i\}_{i=1}^k|\mathcal{X}) = \frac{1}{2}\sum_t\sum_i b_i^t\|\boldsymbol{x}^t - \boldsymbol{m}_i\|^2$$

$$b_i^t = \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

$$\boldsymbol{m}_i = \frac{\sum_t b_i^t \boldsymbol{x}^t}{\sum_t b_i^t}$$

- Online algorithm: stochastic gradient descent, considering the instances one by one, and doing a small update at each step, not forgetting the effect of the previous updates.

- Reconstruction error **for a single instance**

$$E^t(\{\boldsymbol{m}_i\}_{i=1}^k|\boldsymbol{x}^t) = \frac{1}{2}\sum_i b_i^t\|\boldsymbol{x}^t - \boldsymbol{m}_i\|^2$$

$$= \frac{1}{2}\sum_i\sum_j^d b_i^t(x_j^t - m_{ij})^2$$

Update Rule for each instance $\boldsymbol{x}^t$

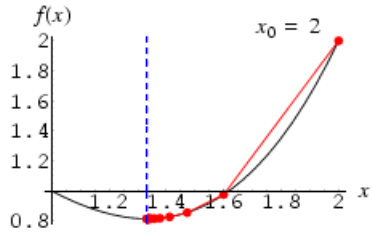
$$\Delta m_{ij} = -\eta\frac{\partial E^t}{\partial m_{ij}} = \eta b_i^t(x_j^t - m_{ij})$$

Online K-Means

- **Gradient Descent Method**: an algorithm for finding the nearest local minimum of a function which presupposes that the gradient of the function can be computed. Starts at a point P_0 and, as many times as needed, moves from P_i to P_{i+1} by minimizing along the line extending from P_i in the direction of $-\nabla f(P_i)$, the local downhill gradient.
- When applied to a 1-dimensional function $f(x)$, the method takes the form of iterating

$$x_i = x_{i+1} - \eta f'(x_{i-1})$$

from a starting point x0 for some small $\eta > 0$ until a fixed point is reached.



- Reconstruction error **for a single instance**

$$E^t(\{\boldsymbol{m}_i\}_{i=1}^k | \boldsymbol{x}^t) = \frac{1}{2} \sum_i b_i^t \|\boldsymbol{x}^t - \boldsymbol{m}_i\|^2$$

$$= \frac{1}{2} \sum_i \sum_j^d b_i^t (x_j^t - m_{ij})^2$$

Update Rule for each instance $\boldsymbol{x}^t$

$$\Delta m_{ij} = -\eta \frac{\partial E^t}{\partial m_{ij}} = \eta b_i^t (x_j^t - m_{ij})$$

Initialize $\boldsymbol{m}_i, i = 1, \ldots, k$, for example, to k random $\boldsymbol{x}^t$
Repeat
  For all $\boldsymbol{x}^t \in \mathcal{X}$ in random order
    $i \leftarrow \arg\min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\|$
    $\boldsymbol{m}_i \leftarrow \boldsymbol{m}_i + \eta(\boldsymbol{x}^t - \boldsymbol{m}_j)$
Until $\boldsymbol{m}_i$ converge

- This moves the closest center (b_i=1) toward the input by a factor given η.
- The other centers have their (b_l=0) equal to 0 and are not updated.

Handling categorical data: k-modes (Huang 1997)

k-modes is a variation of the K-Means Method

- Replacing means of clusters with **modes.**
- Using **new dissimilarity measures** to deal with categorical objects.
  The simple-matching distance is used to determine the dissimilarity of two objects. It is computed by counting the number of mismatches in all variables.
- Using a **frequency-based method** to update modes of clusters.
- A mixture of categorical and numerical data: k-prototype method.

```
> # install.packages("vcd")
> library(vcd) # Visualizing Categorical Data
> ?Arthritis # 關節炎Treatment Data
> data(Arthritis)
> str(Arthritis)
'data.frame':   84 obs. of  5 variables:
 $ ID       : int  57 46 77 17 36 23 75 39 33 55 ...
 $ Treatment: Factor w/ 2 levels "Placebo","Treated": 2 2 2 2 2 2 2 2 2 2 ...
 $ Sex      : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
 $ Age      : int  27 29 30 32 46 58 59 59 63 63 ...
 $ Improved : Ord.factor w/ 3 levels "None"<"Some"<..: 2 1 1 3 3 3 1 3 1 1 ...
> head(Arthritis)
  ID Treatment  Sex Age Improved
1 57   Treated Male  27     Some
2 46   Treated Male  29     None
3 77   Treated Male  30     None
4 17   Treated Male  32   Marked
5 36   Treated Male  46   Marked
6 23   Treated Male  58   Marked
```

Huang, Z. (1997) A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining. in KDD: Techniques and Applications (H. Lu, H. Motoda and H. Luu, Eds.), pp. 21-34, World Scientific, Singapore.

kmodes {klaR}

```
> # install.packages("klaR")
> library(klaR) # Classification and visualization
> res <- kmodes(Arthritis[,c(2,3,5)], modes=3)
> res
K-modes clustering with 3 clusters of sizes 31, 27, 26

Cluster modes:
  Treatment    Sex Improved
1   Treated   Male     None
2   Treated Female   Marked
3   Placebo Female     None

Clustering vector:
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  2  1  2
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
 2  2  2  2  1  2  2  2  2  2  2  2  2  2  2  1  2  2  1  1  1
47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67
 1  1  1  1  1  1  3  3  3  3  3  2  3  3  3  3  3  3  3  3  3
70 71 72 73 74 75 76 77 78 79 80 81 82 83 84
 2  3  3  3  3  2  3  3  3  2  3  3  3  3  2

Within cluster simple-matching distance by cluster:
[1] 25 11  7

Available components:
[1] "cluster"    "size"       "modes"      "withindiff" "iterations"
[6] "weighted"
```

```
> table(Arthritis[,c(2,3,5)])
, , Improved = None

         Sex
Treatment Female Male
  Placebo     19   10
  Treated      6    7

, , Improved = Some

         Sex
Treatment Female Male
  Placebo      7    0
  Treated      5    2

, , Improved = Marked

         Sex
Treatment Female Male
  Placebo      6    1
  Treated     16    5
```

K-medoid

- A medoid (中心點) can be defined as **the object of a cluster** whose **average dissimilarity to all the objects in the cluster is minimal**. i.e. it is a most centrally located point in the cluster.

- K-medoid is more robust to noise and outliers as compared to k-means because it minimizes a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances.

- PAM uses a greedy search which may not find the optimum solution, but it is faster than exhaustive search.
  - Initialize: select *k* of the *n* data points as the medoids
  - Associate each data point to the closest medoid.
  - While the cost (sum of distances of points to their medoid) of the configuration decreases:
    - For each medoid *m*, for each non-medoid data point *o*:
    - Swap *m* and *o*, recompute the cost.
    - If the **total cost** of the configuration increased in the previous step, undo the swap

pam {cluster}

Partitioning Around Medoids

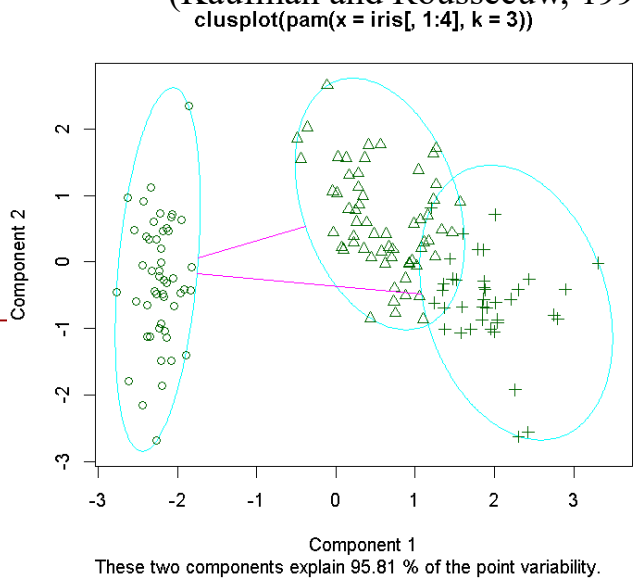
(Kaufman and Rousseeuw, 1990)

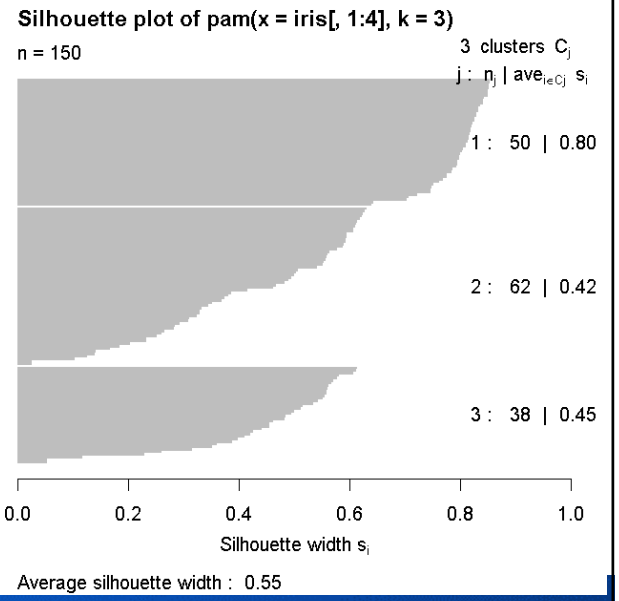
Usage

```
pam(x, k, diss = inherits(x, "dist"), metric = "euclidean",
    medoids = NULL, stand = FALSE, cluster.only = FALSE,
    do.swap = TRUE,
    keep.diss = !diss && !cluster.only && n < 100,
    keep.data = !diss && !cluster.only,
    pamonce = FALSE, trace.lev = 0)
```

```
> iris.pam <- pam(iris[,1:4], k=3)
> iris.pam
Medoids:
      ID Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]   8          5.0         3.4          1.5         0.2
[2,]  79          6.0         2.9          4.5         1.5
[3,] 113          6.8         3.0          5.5         2.1
Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
...
[149] 3 2
Objective function:
    build      swap
0.6709391 0.6542077

Available components:
 [1] "medoids"    "id.med"     "clustering" "objective"  "isola
 [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
> plot(iris.pam)
```

```
> clusplot(iris[, 1:4], iris.pam)
```





clara {cluster}

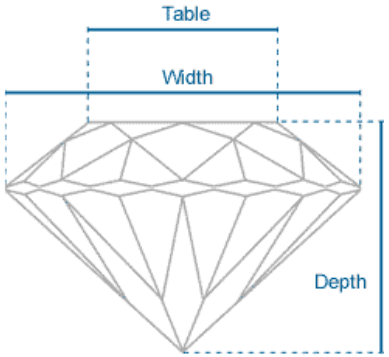
Clustering Large Applications

- Split randomly the data sets in multiple **subsets** with fixed size (sampsize)
- Compute **PAM** algorithm on each subset and choose the corresponding k medoids.
- Assign each observation of the entire data set to the closest medoid.
- Calculate the mean of the dissimilarities of the observations to their closest medoid.
  (This is used as a measure of the goodness of the clustering.)
- Retain the sub-dataset for which the mean is minimal.

Usage

```
clara(x, k, metric = "euclidean", stand = FALSE, samples = 5,
      sampsize = min(n, 40 + 2 * k), trace = 0, medoids.x = TRUE,
      keep.data = medoids.x, rngR = FALSE, pamLike = FALSE, correct.d = TRUE)
```

```
> require(ggplot2)
> data(diamonds)
> dim(diamonds)
[1] 53940    10
> head(diamonds)
# A tibble: 6 x 10
  carat       cut color clarity depth table price     x     y     z
  <dbl>     <ord> <ord>   <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1  0.23     Ideal     E     SI2  61.5    55   326  3.95  3.98  2.43
2  0.21   Premium     E     SI1  59.8    61   326  3.89  3.84  2.31
3  0.23      Good     E     VS1  56.9    65   327  4.05  4.07  2.31
4  0.29   Premium     I     VS2  62.4    58   334  4.20  4.23  2.63
5  0.31      Good     J     SI2  63.3    58   335  4.34  4.35  2.75
6  0.24 Very Good     J    VVS2  62.8    57   336  3.94  3.96  2.48
```



clara {cluster}

Clustering Large Applications

```
> system.time(
+   clara.res <- clara(diamonds[, -(2:4)], k=10, stand=T, samples=500, pamLike=TRUE))
   user  system elapsed
   3.17    0.01    3.20
> clara.res
Call:    clara(x = diamonds[, -(2:4)], k = 10, stand = T, samples = 500,      pamLike =
TRUE)
Medoids:
      carat depth table price    x    y    z
 [1,]  0.38  61.7    56  1087 4.63 4.66 2.86
 [2,]  0.33  61.9    59   579 4.40 4.46 2.74
 [3,]  0.53  59.8    60  2542 5.28 5.35 3.18
...
 [7,]  1.01  60.1    60  5902 6.46 6.51 3.90
 [8,]  1.09  61.9    56  4784 6.61 6.64 4.10
 [9,]  1.51  62.6    56  9234 7.32 7.27 4.57
[10,]  2.12  62.3    58 12693 8.25 8.16 5.12
Objective function:      1.410032
Clustering vector:       int [1:53940] 1 2 3 2 2 2 2 1 2 3 1 1 2 1 2 2 1 1 ...
Cluster sizes:           10935 5368 4108 6187 6099 5164 4237 5800 3158 2884
Best sample:
 [1]  1052  1556  1866  3298  3359  3811  3917  4379  4531  6145  6412  7057
...
[49] 43265 43851 44593 45306 45472 46776 47279 48835 50549 52598 53314 53607

Available components:
 [1] "sample"     "medoids"    "i.med"      "clustering" "objective"
 [6] "clusinfo"   "diss"       "call"       "silinfo"    "data"
```

Fuzzy C-Means Clustering (Bezdek, 1981)

Minimization of the objective function

m: fuzziness index (often m=2)

$$J_m = \sum_{i=1}^{n} \sum_{k=1}^{K} u_{ik}^m \; \| \mathbf{x}_i - \mathbf{c}_k \|^2$$

fuzzy centroids (center vectors)

$U = [u_{ik}]$ fuzzy partition matrix

membership degree of $\mathbf{x}_i$ to c_j

$$\mathbf{c}_k = \frac{\sum_{i=1}^{n} u_{ik} \cdot \mathbf{x}_i}{\sum_{i=1}^{n} u_{ik}}$$

Algorithm of FCM

1. Initialize $U = [u_{ik}]$ matrix, $U^{(t)}, t = 0$.
2. At step t, Compute fuzzy centroids (center vectors) $C^{(t)} = [\mathbf{c}_k]$ with $U^{(t)}$

$$\mathbf{c}_k = \frac{\sum_{i=1}^{n} u_{ik} \cdot \mathbf{x}_i}{\sum_{i=1}^{n} u_{ik}}, \quad k = 1, \cdots, K$$

3. Compute the degree of membership of all data points for all clusters to update the partition matrix, i.e. obtain $U^{(t+1)}$, as follows:

$$u_{ik} = \frac{1}{\sum_{j=1}^{K} \left(\frac{\|\mathbf{x}_i - \mathbf{c}_k\|}{\|\mathbf{x}_i - \mathbf{c}_j\|} \right)^{\frac{2}{m-1}}}$$

4. Repeat step 2 $\sim$ 3 unless $\| U^{(t+1)} - U^{(t)} \| < \epsilon$.

Several Soft Clustering Algorithms in R

- Fuzzy c-means: **`cmeans {e1071}`**
- Fuzzy Analysis Clustering (Kaufman and Rousseeuw, 1990) : **`fanny {cluster}`**

$$\sum_{k=1}^{K}\left(\sum_{i,j}^{n} u_{ik}^{m} u_{jk}^{m} \parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2\right)/\left(2\sum_{j=1}^{n} u_{jk}^{m}\right).$$

- Fuzzy c-shell clustering (Dave, 1996): **`cshell {e1071}`**

$$J_K = \sum_{i=1}^{n}\sum_{k=1}^{K} u_{ik}^{m}(\parallel \mathbf{x}_i - \mathbf{v}_k \parallel - r_k)^2$$

the data point $\mathbf{x}_i$ to a circle $(\mathbf{v}, r)$.

- Model-based clustering (Fraley and Raftery, 2002) : **`mclust {Mclust}`**
- Consensus clustering: **`cl_bag {clue}`**

cmeans {e1071}

Fuzzy C-Means Clustering

Usage

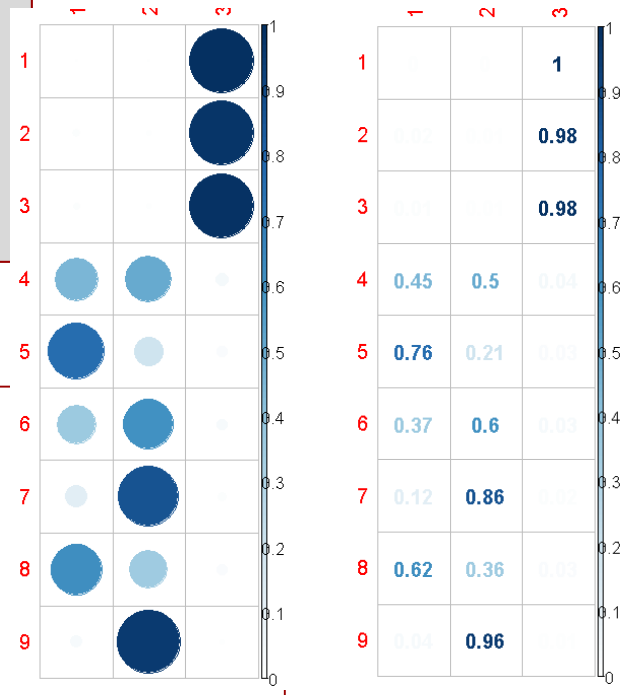
```
cmeans (x, centers, iter.max=100,verbose=FALSE,
        dist="euclidean",
        method="cmeans", m=2, rate.par = NULL)
```

Arguments

`centers:` #clusters or initial values for cluster centers

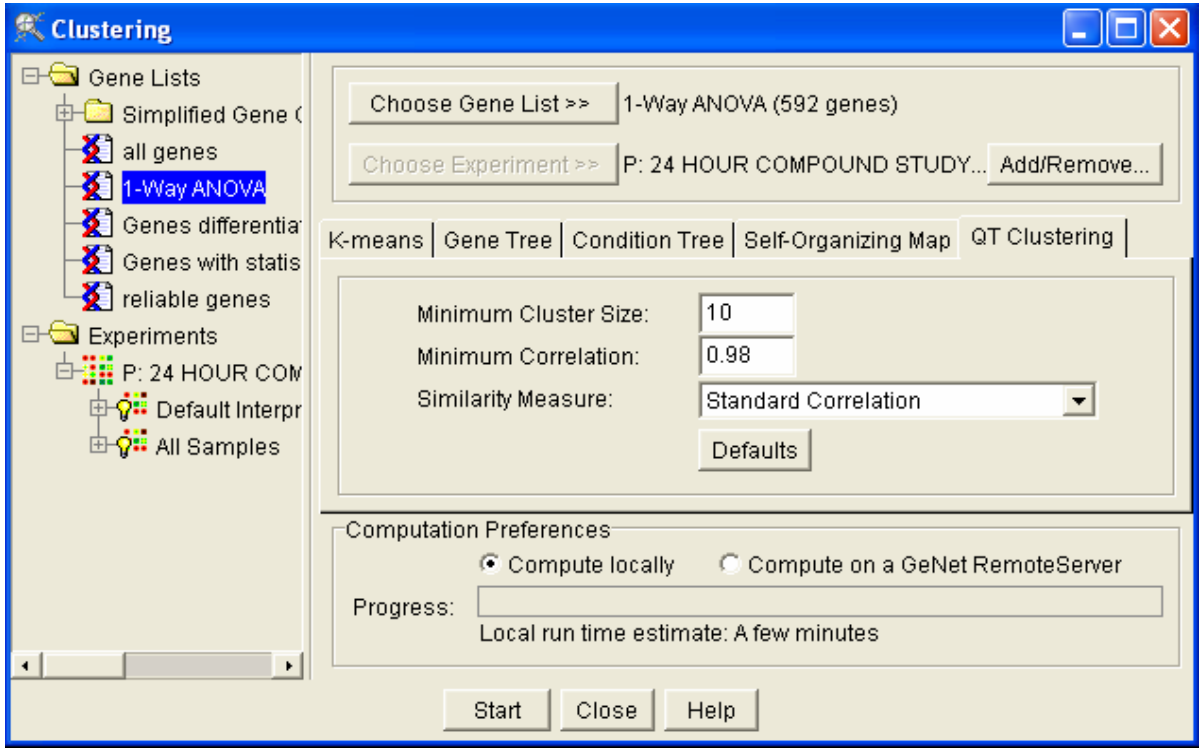
`dist: "euclidean"/"manhattan"`.

`method: "cmeans"/"ufcl"` **(on-line update)**



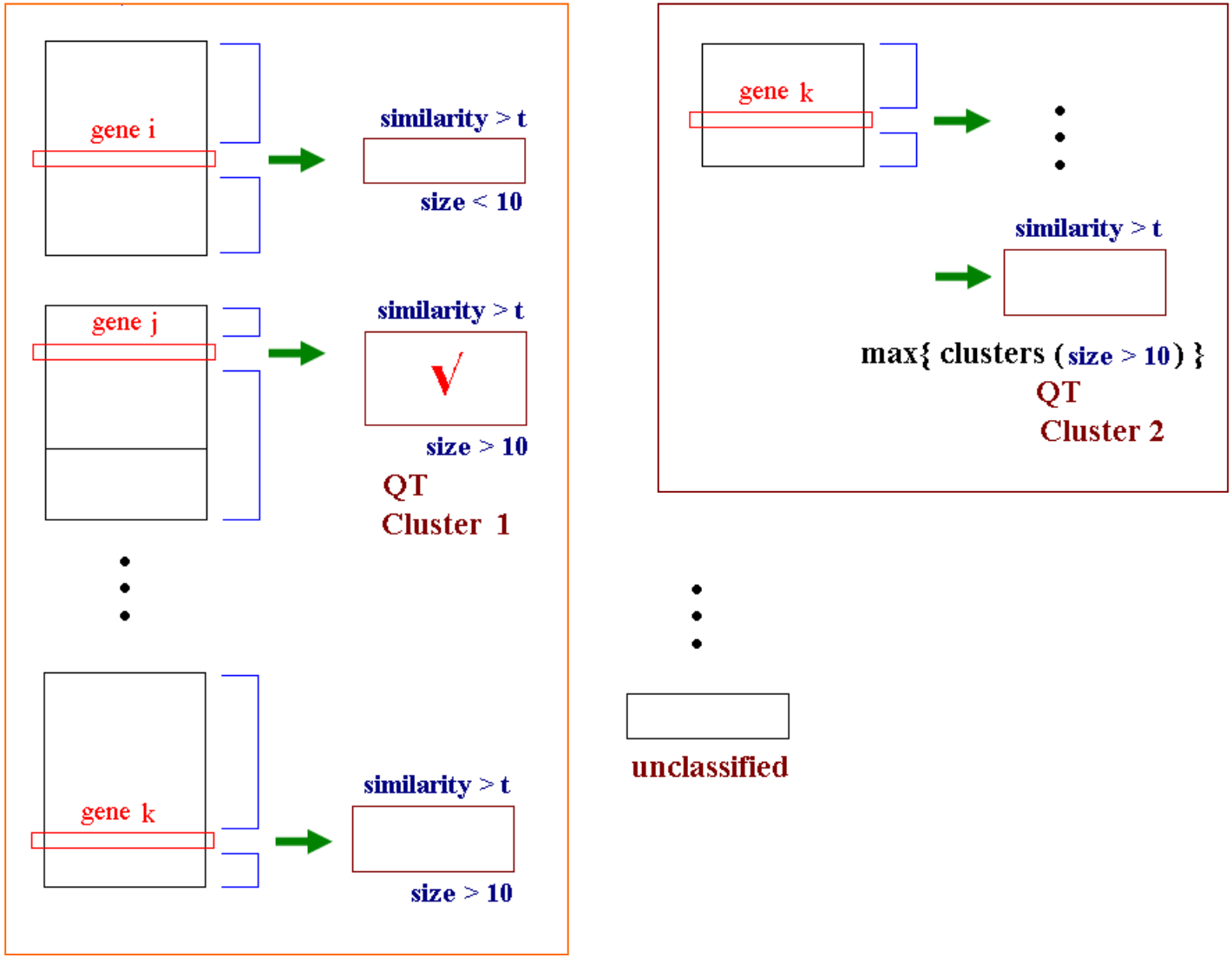

```
> library(e1071)
> iris.fc <- cmeans(iris[,1:4], centers=3)
> names(iris.fc)
[1] "centers"  "size"  "cluster"  "membership"  "iter"
[6] "withinerror" "call"
> library(corrplot) # A visualization of a correlation matrix
> id <- c(1:3, 51:53, 101:103)
> corrplot(iris.fc$membership[id,], is.corr = FALSE)
> corrplot(iris.fc$membership[id,], method = "number", is.corr = FALSE)
> iris.fc$cluster
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
...
[129] 2 2 2 2 2 1 2 2 2 2 1 2 2 2 1 2 2 2 1 2 2 1
```

QT (Quality Threshold) Clustering



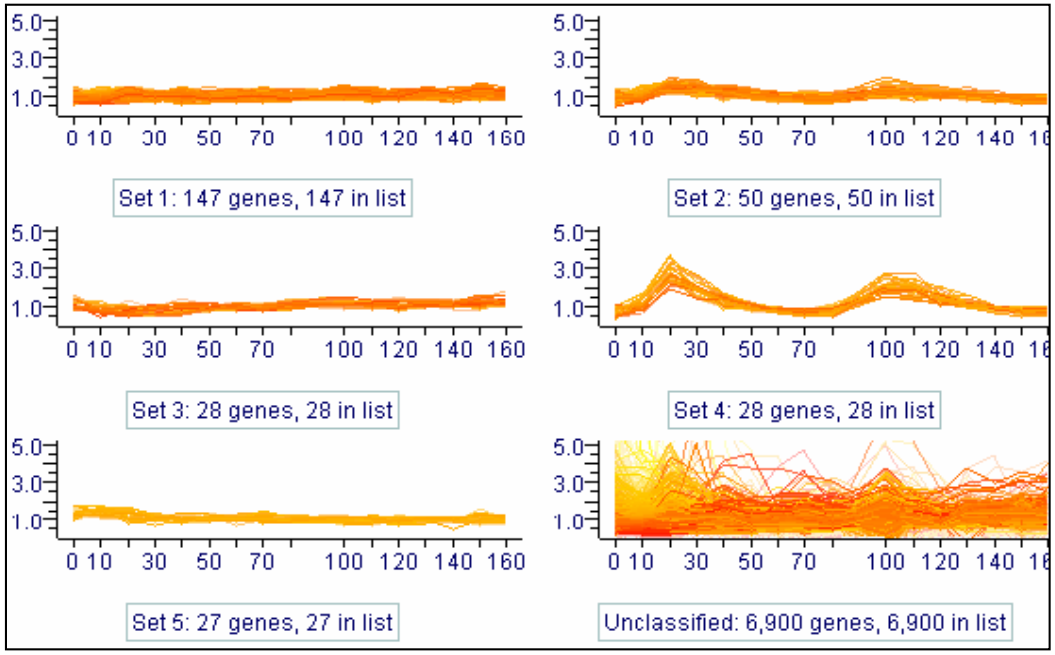
- **Minimum Cluster Size:** Minimum number of genes that you would like to have in each cluster.
- **Minimum Correlation:** Minimum correlation that genes within each cluster must have to one another.
- The diameter is the equivalent of 1 minus the minimum correlation.

Algorithm of QT Clustering



Interpreting the Results

- QT Clusters are displayed according to the cluster size, from the largest to the smallest.
- All sets will have **at least** the user-defined minimum cluster size and the minimum correlation (diameter).
- For example, all 147 genes in Set 1 below are at least 0.98 correlated to each other.
- Genes that did not meet the minimum quality are grouped under the "unclassified" category.



QT Clustering: Advantages

Advantages

- Quality Guarantee
- Number of clusters is not specified a priori
- All possible clusters are considered

Disadvantages

- Computationally Intensive/Time Consuming

Main differences between QT clustering and K-means clustering?

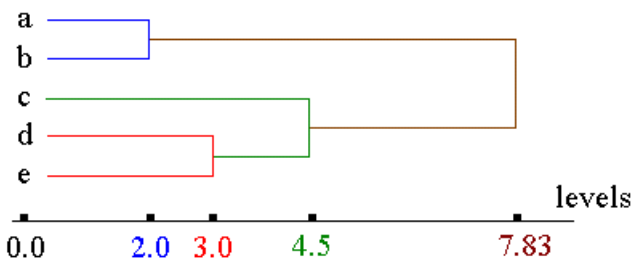
	K-means	QT clustering	Consequence
Need to specify cluster number?	Yes	No	K-means: if users specify too few clusters, genes that are not similar will be forced to group together.
Very computationally intensive?	No	Yes	QT clustering: may be too computationally intensive, depending on available RAM and number of genes in starting gene list, for some desktop computer.
Every gene must be clustered?	Yes	No	K-means: every gene on the selected gene list must belong to a cluster. This could potentially group genes that are not very similar into the same cluster. QT clustering: only cluster with user-specified quality will be formed.

Hierarchical Clustering and Dendrogram (Agglomerative Nesting, AGNES)

Example: Average-Linkage

distance matrix

	a	b	c	d	e
a	0	2	6	10	9
b		0	5	9	8
c			0	4	5
d				0	3
e					0



	{a, b}	c	d	e
{a, b}	0	5.5	9.5	8.5
c		0	4	5
d			0	3
e				0

$$D(\{a,b\},\{c\}) = \frac{1}{2}[D(a,c)+D(b,c)]$$

$$= \frac{1}{2}(6+5) = 5.5$$

	{a, b}	c	{d, e}
{a, b}	0	5.5	9.0
c		0	4.5
{d, e}			0

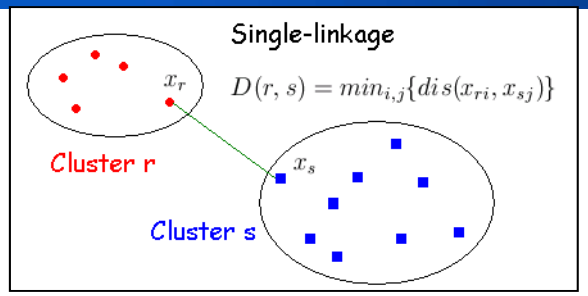
$$D(\{a,b\},\{d,e\})$$

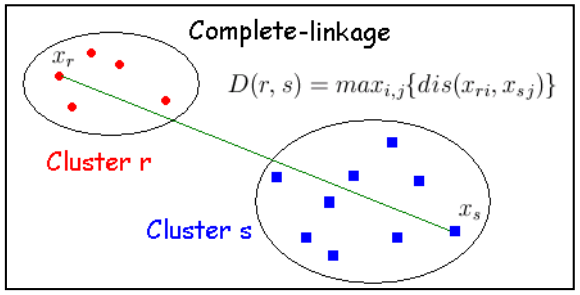
$$= \frac{1}{4}[D(a,d)+D(a,e)+D(b,d)+D(b,e)]$$

$$= \frac{1}{4}(10+9+9+8) = 9$$

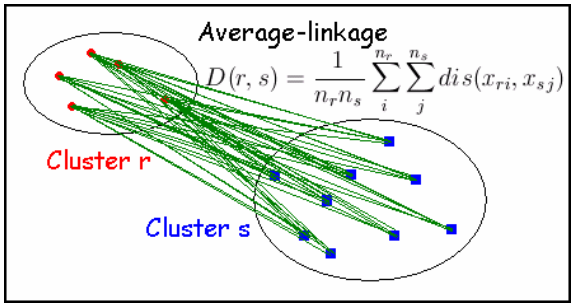
	{a, b}	{c, d, e}
{a, b}	0	7.83
{c, d, e}		0

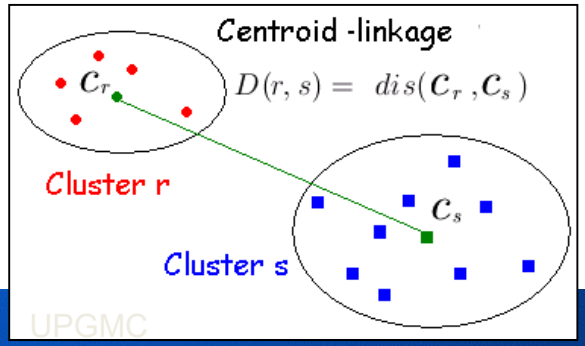
(Kaufman and Rousseeuw, 1990)





UPGMA (Unweighted Pair-Groups Method Average)





Ward's Method

- The Ward's method forms clusters by maximizing within-clusters homogeneity.
- The within-group (i.e., within-cluster) sum of squares is used as the measure of homogeneity.
- The within-cluster sums of squares that is minimized is also known as the error sums of squares (ESS).

Example:
Charles H. Romesburg (1984)

Toy Data

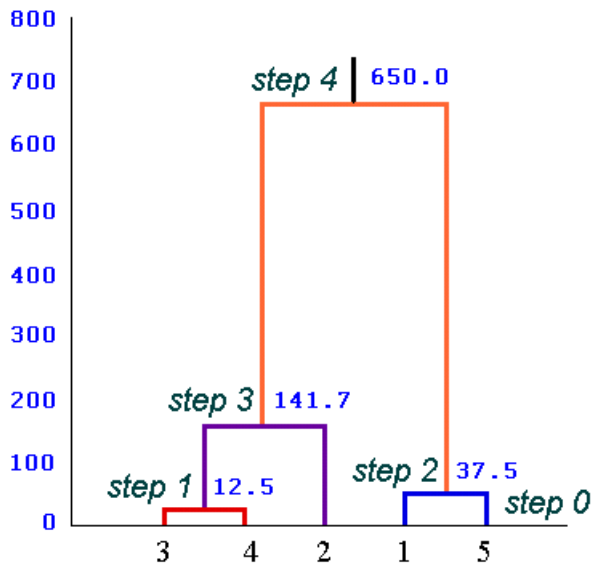
data	x1	x2
1	10	5
2	20	20
3	30	10
4	30	15
5	5	10

step	Possible Partitions				ESS
1	(12)	3	4	5	?

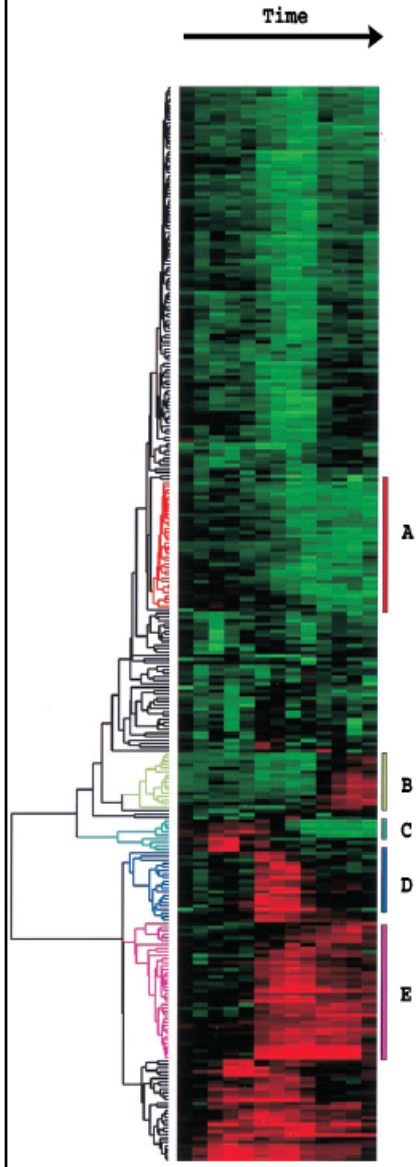
$$\{\overline{12}\} = [\ (10+20)/2, (5+20)/2\ ]$$
$$= [\ 15, 12.5]$$

$$\begin{aligned} ESS &= wss\{12\} + wss\{3\} + wss\{4\} + wss\{5\} \\ &= ss(1, \{\overline{12}\}) + ss(2, \{\overline{12}\}) \\ &= (10-15)^2 + (5-12.5)^2 + (20-15)^2 + (2-12.5)^2 \\ &= 162.5 \end{aligned}$$

step	Possible Partitions				ESS
1	(12)	3	4	5	162.5
	(13)	2	4	5	212.5
	(14)	2	3	5	250.0
	(15)	2	3	4	25.0
	(23)	1	4	5	100.0
	(24)	1	3	5	62.5
	(25)	1	3	4	162.5
	(34)	1	2	5	12.5*
	(35)	1	2	4	312.5
	(45)	1	2	3	325.0
2	(34)	(12)	5		175.0
	(34)	(15)	2		37.5*
	(34)	(25)	1		175.0
	(134)	2	5		316.7
	(234)	1	5		116.7
	(345)	1	2		433.3
3	(234)	(15)			141.7*
	(125)	(34)			245.9
	(1345)	2			568.8
4	(12345)				650.0



Display of Genome-Wide Expression Patterns



Proc. Natl. Acad. Sci. USA
Vol. 95, pp. 14863–14868, December 1998
Genetics

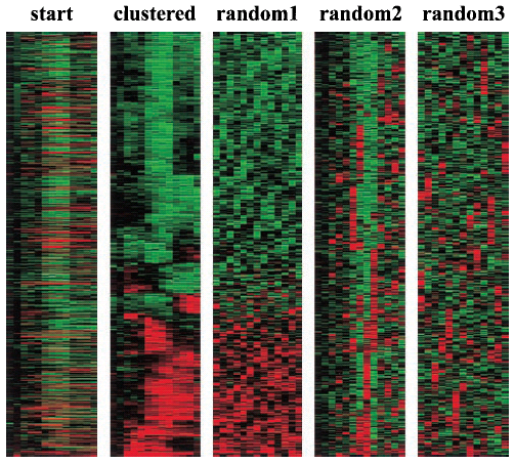
Cluster analysis and display of genome-wide expression patterns

MICHAEL B. EISEN*, PAUL T. SPELLMAN*, PATRICK O. BROWN†, AND DAVID BOTSTEIN*‡

FIG. 1. Clustered display of data from time course of serum stimulation of primary human fibroblasts. Experimental details are described elsewhere (11). Briefly, foreskin fibroblasts were grown in culture and were deprived of serum for 48 hr. Serum was added back and samples taken at time 0, 15 min, 30 min, 1 hr, 2 hr, 3 hr, 4 hr, 8 hr, 12 hr, 16 hr, 20 hr, 24 hr. The final datapoint was from a separate unsynchronized sample. Data were measured by using a cDNA microarray with elements representing approximately 8,600 distinct human genes. All measurements are relative to time 0. Genes were selected for this analysis if their expression level deviated from time 0 by at least a factor of 3.0 in at least 2 time points. The dendrogram and colored image were produced as described in the text; the color scale ranges from saturated green for log ratios −3.0 and below to saturated red for log ratios 3.0 and above. Each gene is represented by a single row of colored boxes; each time point is represented by a single column. Five separate clusters are indicated by colored bars and by identical coloring of the corresponding region of the dendrogram. As described in detail in ref. 11, the sequence-verified named genes in these clusters contain multiple genes involved in (*A*) cholesterol biosynthesis, (*B*) the cell cycle, (*C*) the immediate–early response, (*D*) signaling and angiogenesis, and (*E*) wound healing and tissue remodeling. These clusters also contain named genes not involved in these processes and numerous uncharacterized genes. A larger version of this image, with gene names, is available at http://rana.stanford.edu/clustering/serum.html.

Software:
Cluster and TreeView

FIG. 3. To demonstrate the biological origins of patterns seen in Figs. 1 and 2, data from Fig. 1 were clustered by using methods described here before and after random permutation within rows (random 1), within columns (random 2), and both (random 3).



Minimal Spanning Tree

Single-linkage correspond to minimal spanning tree:

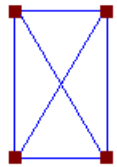
- Consider a weighted, completely connected ***graph*** with nodes corresponding to instance and **edge** between **nodes** with weights equal to the distances between the instance.

Spanning trees:

- A spanning tree of a graph is just a ***subgraph*** that contains all the vertices and is a tree. A graph may have many spanning trees; for instance the complete graph on four vertices

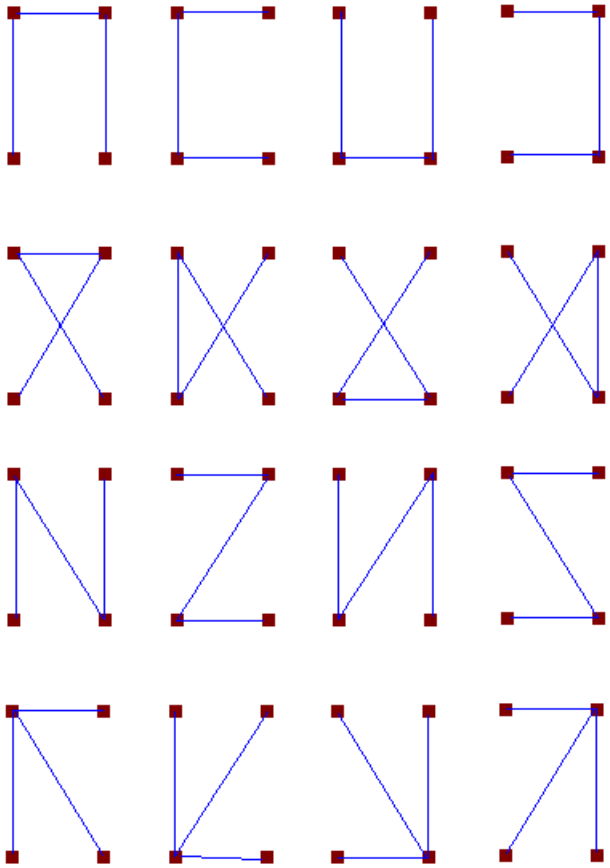
Minimum spanning trees

- Now suppose the edges of the graph have weights or lengths.
- The weight of a tree is just the sum of weights of its edges.
- Different trees have different lengths. The problem: how to find the minimum length spanning tree?
- Application: the minimum spanning tree can be used to approximately solve the traveling salesman problem.
- A convenient formal way of defining this problem is to find the shortest path that visits each point at least once.



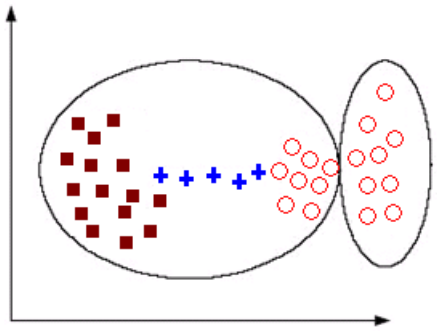
The complete graph on four vertices

Sixteen Spanning Trees



Comparisons

- Singe-linkage: clusters may be elongated due to the chaining effect.
- Complete-linkage: clusters tend to be more compact.



Method	can be used for cluster-ing cases	useful for clustering variables	different dissimi-larity and similarity measures	agglomera-tion level has a clear inter-pretation	avoids chaining or dilatation	avoids reversals (d)	invariant to mono-tonic transfor-mation
complete linkage	yes	yes	yes	yes	yes (a)	yes	yes (e)
single linkage	yes	yes	yes	yes	yes (b)	yes	yes (e)
average linkage	yes	yes	yes	no	no	yes	(yes (f))
weighted average linkage	yes	yes	yes	yes	no	yes	no
within average linkage	yes	yes	yes	yes	no	no	no
median Linkage	yes	no	no	no	(no) (c)	no	no
centroid Linkage	yes	no	no	yes	no	no	no
Ward's Linkage	yes	no	no	yes	no	yes	no

(a) chaining occurs (see text), (b) dilatation occurs (see text), (c) tendency to chaining (Gordon 1999: 67), (d) see text, (e) invariant to monotonic transformation of dissimilarities resp. similiarities (Bacher 1996: 146), (f) invariant to linear transformation of dissimilarities resp. similarities (Bacher 1996: 271)

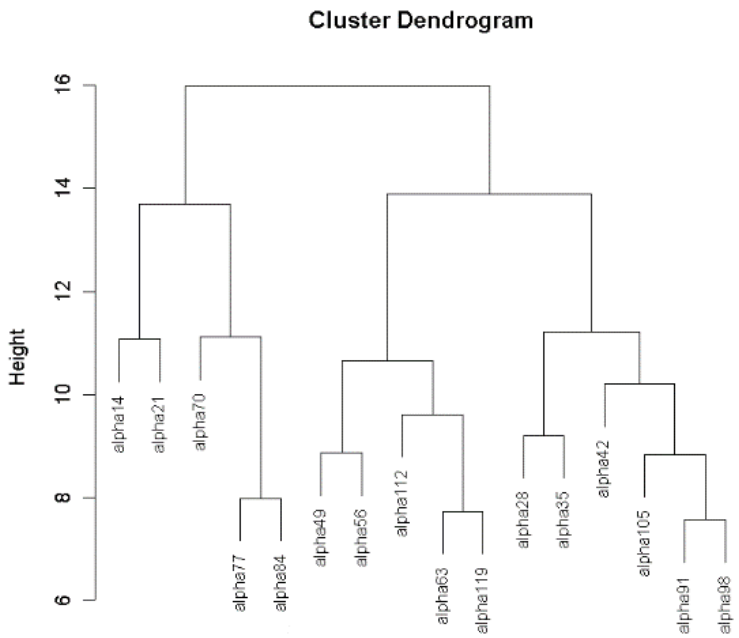
http://www.soziologie.wiso.uni-erlangen.de/koeln/script/chap4.pdf

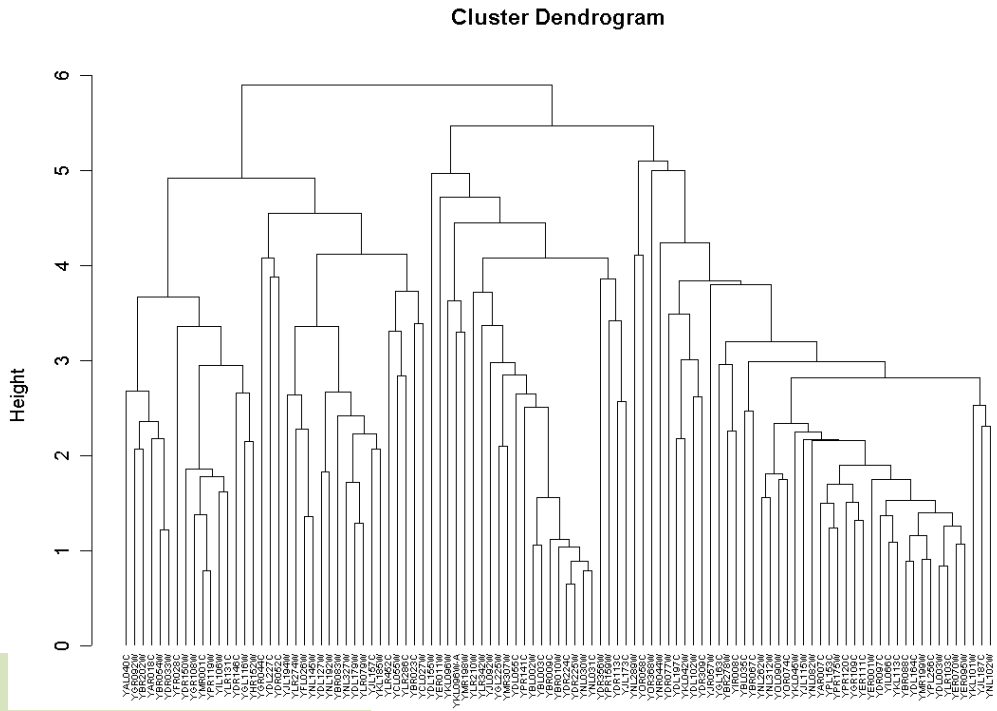
hclust {stats}

Hierarchical Clustering

See A05: page 134/172 or C01: page 31/95

```
> gene.name <-  rownames(cell.matrix)
## Hierarchical Clustering on genes
> cell.gene.hc.ave <- hclust(dist(cell.sdata), method="ave")
> plot(cell.gene.hc.ave, hang=-1, cex=0.5, labels=gene.name)
## Hierarchical Clustering on experiments
> cell.exp.hc.ave <- hclust(dist(t(cell.sdata)), method="ave")
> plot(cell.exp.hc.ave, cex=0.8)
```





See also:

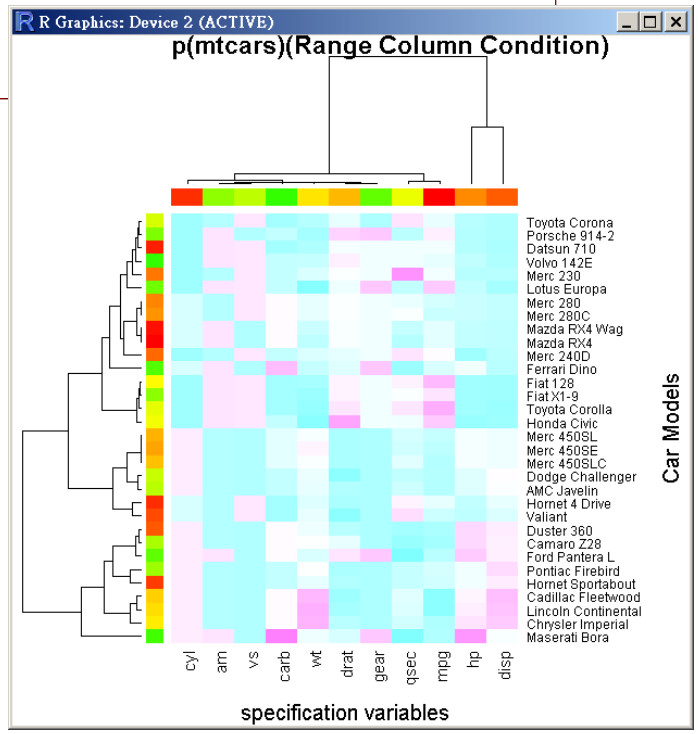
agnes {cluster}: Agglomerative Nesting (Hierarchical Clustering)

diana {cluster}: DIvisive ANAlysis Clustering

Heatmap (Hierarchical Clustering + Data Image)

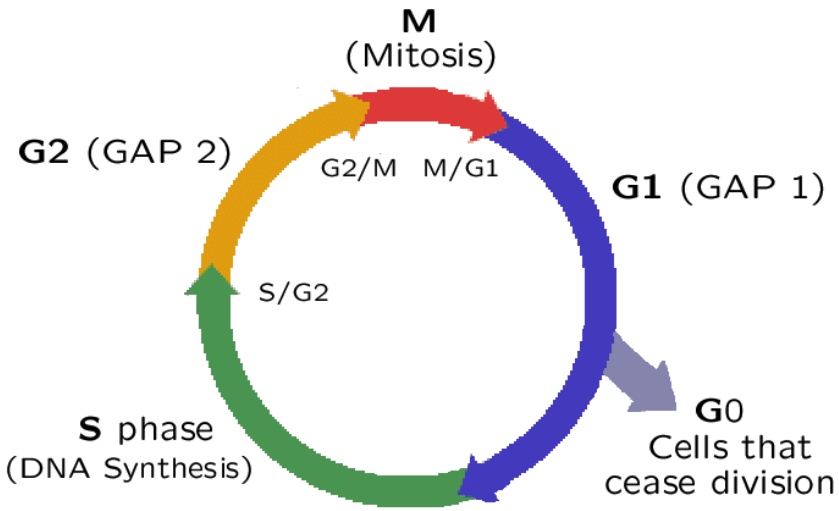
```
x  <- as.matrix(mtcars)
?heatmap
row.color <- rainbow(nrow(x), start=0, end=.3)
col.color <- rainbow(ncol(x), start=0, end=.3)
hv <- heatmap(x, col = cm.colors(256), scale="column",
              RowSideColors = row.color, ColSideColors = col.color,
              margins=c(5,10),
              xlab = "specification variables", ylab= "Car Models",
              main = "Heatmap(mtcars)(Range Column Condition)")
names(hv)
[1] "rowInd" "colInd" "Rowv"   "Colv"
```

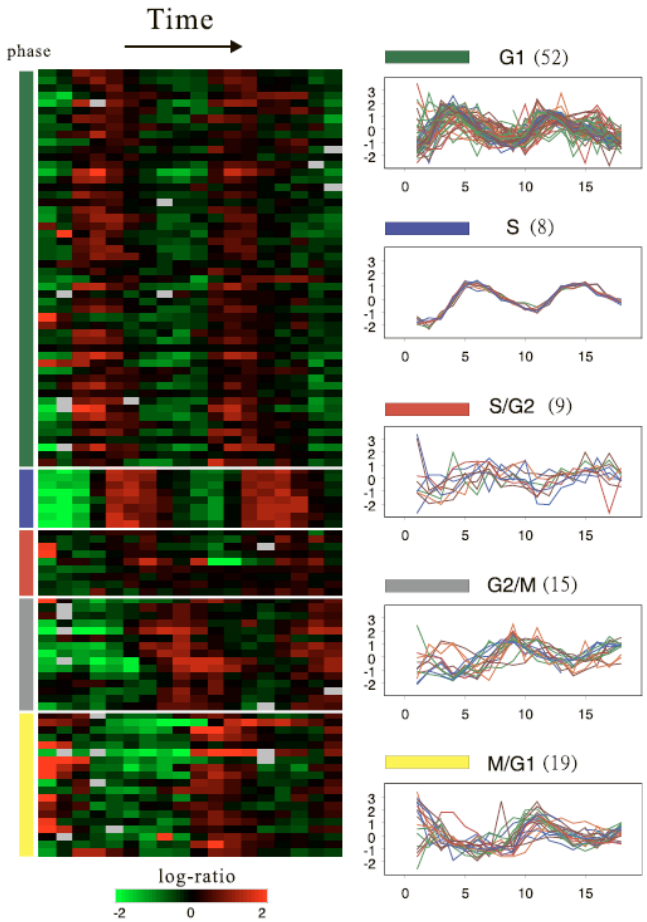
```
> mtcars
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
```



課堂練習: Microarray Data

- Lu and Wu (2010)
  - Time course data: every 7 minutes and totally 18 time points.
  - Known genes: there are 103 cell cycle-regulated genes by traditional method in G1, S, S/G2, G2/M, or M/G1. (Remove NA’ s: 79.)



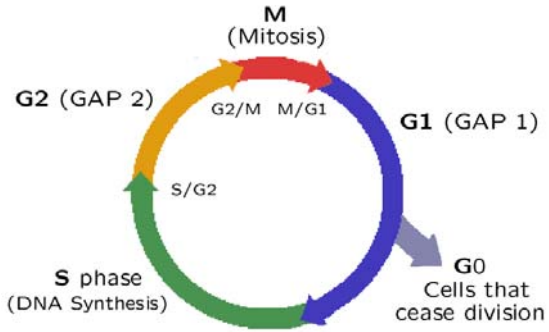


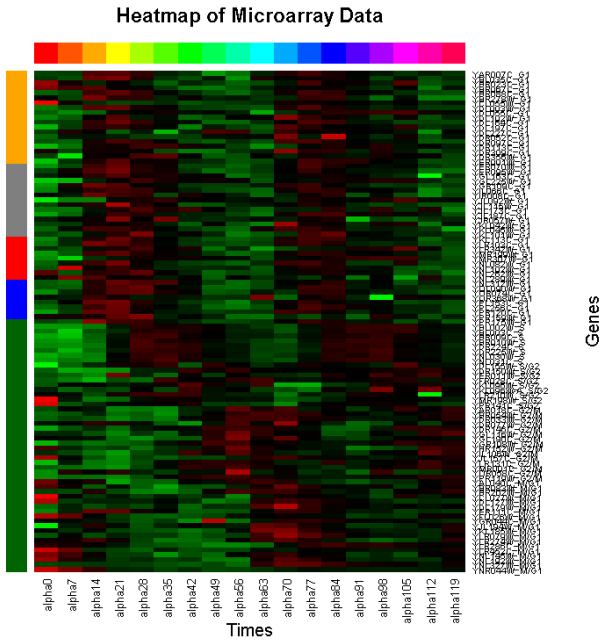
See also: Using R to draw a Heatmap from Microarray Data
http://www2.warwick.ac.uk/fac/sci/moac/people/students/peter_cock/r/heatmap/

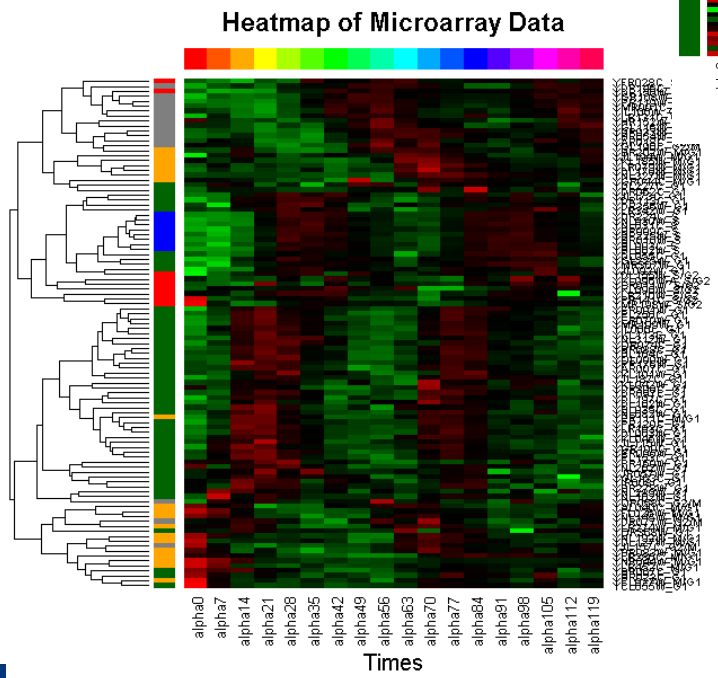
課堂練習: Microarray Data

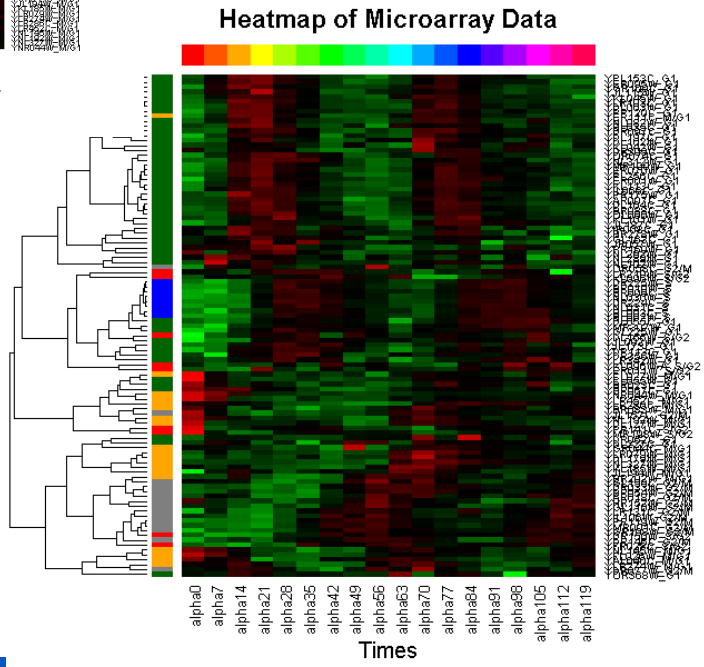
```
# install.packages("fields")
library(fields)
gbr <- two.colors(start="green", middle="black", end="red")
cell.raw <- read.table("trad_alpha103.txt", row.names=1, header=T)
cell.data <- t(scale(t(cell.raw[,2:19]), center=T, scale=T))
n <- nrow(cell.data)
p <- ncol(cell.data)
gene.phase <- cell.raw[,1]
range(cell.data)
cell.data[cell.data > 2.802712] <- 2.802712
cellcycle.color <- c("darkgreen", "blue", "red", "gray50", "orange")
rc <- cellcycle.color[gene.phase+1]
cc <- rainbow(ncol(cell.data))

hv1 <- heatmap(cell.data[n:1,], col = gbr, Colv=NA, Rowv=NA,
               RowSideColors = rc,
               ColSideColors = cc, margins = c(5,10),
               xlab = "Times", ylab =  "Genes",main = "Heatmap of Microarray Data")

hv2 <- heatmap(cell.data, col = gbr, Colv=NA, Rowv=NULL,
               RowSideColors = rc,
               ColSideColors = cc, margins = c(5,10),
               xlab = "Times", ylab =  "Genes",main = "Heatmap of Microarray Data")

dd <- as.dendrogram(hclust(as.dist(1-cor(t(cell.data)))))
hv3 <- heatmap(cell.data, col = gbr, Colv=NA, Rowv=dd,
              RowSideColors = rc,
              ColSideColors = cc, margins = c(5,10),
              scale = "row",
              xlab = "Times", ylab =  "Genes",main = "Heatmap of Microarray Data")
```

課堂練習: Microarray Data









gplots: Heatmap.2

- gplots: Various R programming tools for plotting data

```
> install.packages("gplots")
> library(gplots)
> ?heatmap.2
> library(affy)
> data(SpikeIn)
> pms <- SpikeIn@pm

# just the data, scaled across rows
heatmap.2(pms, col=rev(heat.colors(16)), main="SpikeIn@pm",
              xlab="Relative Concentration", ylab="Probeset",
              scale="row")

# fold change vs "12.50" sample
data <- pms / pms[, "12.50"]
data <- ifelse(data>1, data, -1/data)
heatmap.2(data, breaks=16, col=redgreen, tracecol="blue",
          main="SpikeIn@pm Fold Changes\nrelative to 12.50 sample",
          xlab="Relative Concentration", ylab="Probeset")
```

More Examples

http://www2.warwick.ac.uk/fac/sci/moac/currentstudents/peter_cock/r/heatmap/

Self-Organizing Maps (SOM)

- SOMs were developed by Kohonen in the early 1980's, original area was in the area of speech recognition.

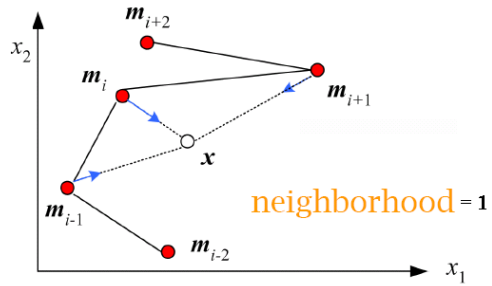
- *Idea:*
  Organise data on the basis of **similarity** by putting entities **geometrically** close to each other.

- SOM is unique in the sense that it combines both aspects:
  - It can be used at the same time both to reduce the amount of data by clustering, and
  - to construct a nonlinear projection of the data onto a low-dimensional display.

Self-Organizing Maps (SOM)

■ SOMs were developed by **Kohonen** in the early **1980's**, original area was in the area of speech recognition.

■ ***Idea:*** Organise data on the basis of **similarity** by putting entities **geometrically** close to each other.



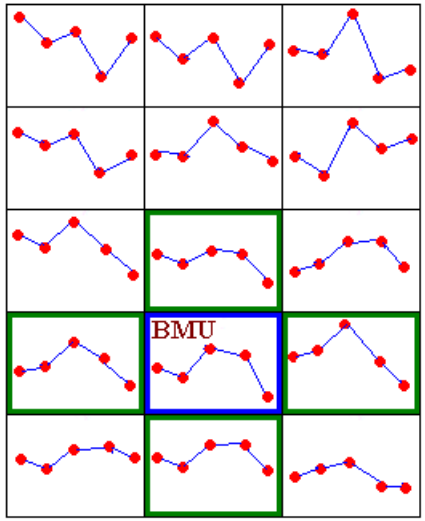
■ SOM is unique in the sense that it combines both aspects. It can be used at the same time both to reduce the amount of data by clustering, and to construct a nonlinear projection of the data onto a low-dimensional display.

Step 0:
Initialize weights $\mathbf{w}_i(t)$.
Set $\alpha(t)$ and $h_{ci}(t)$.

Learning process:

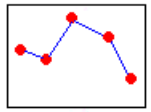
$$\mathbf{w}_i(t+1) = \begin{cases} \mathbf{w}_i(t) + h_{ci}(t)[\,\mathbf{x}(t) - \mathbf{w}_i(t)\,] & i \in N_c(t) \\ \mathbf{w}_i(t), \quad \text{o.w.} & \end{cases}$$

5 x 3 output node



Data Matrix

Table	X01	X02	X03	...	Xp
obs 001	-0.48	-0.42	0.87		-0.35
obs 002	-0.39	-0.58	1.08		-0.58
obs 003	0.87	0.25	-0.17		-0.13
obs 004	1.57	1.03	1.22		-1.02
obs 005	-1.15	-0.86	1.21		-0.44
obs 006	0.04	-0.12	0.31		0.08
obs 007	2.95	0.45	-0.40		-0.76
obs 008	-1.22	-0.74	1.34		-0.55
obs 009	-0.73	-1.06	-0.79		0.03
obs 010	-0.58	-0.40	0.13		-0.45
obs 011	-0.50	-0.42	0.66		0.01
obs 012	-0.86	-0.29	0.42		-0.63
obs 013	-0.16	0.29	0.17		-0.04
obs ...					
obs n	-1.79	0.94	2.13		-0.66



X01 ... Xp

input node

Incrementally decrease the learning rate and the neighborhood size, and repeat

Self-Organizing Maps (SOM)

(1)

Data Matrix

Table	X01	X02	X03	•••	Xp
obs 001	-0.48	-0.42	0.87		-0.35
obs 002	-0.39	-0.58	1.08		-0.58
obs 003	0.87	0.25	-0.17		-0.13
obs 004	1.57	1.03	1.22		-1.02
obs 005	-1.15	-0.86	1.21		-0.44
obs 006	0.04	-0.12	0.31		0.08
obs 007	2.95	0.45	-0.40		-0.76
obs 008	-1.22	-0.74	1.34		-0.55
obs 009	-0.73	-1.06	-0.79		0.03
obs 010	-0.58	-0.40	0.13		-0.45
obs 011	-0.50	-0.42	0.66		0.01
obs 012	-0.86	-0.29	0.42		-0.63
obs 013	-0.16	0.29	0.17		-0.04
obs •••					
obs n	-1.79	0.94	2.13		-0.66

(3)

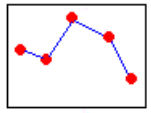
Step 0:
Initialize weights $\mathbf{w}_i(t)$.
Set $\alpha(t)$ and $h_{ci}(t)$.

(5)

Learning process:

$$\mathbf{w}_i(t+1) = \begin{cases} \mathbf{w}_i(t) + h_{ci}(t)[\,\mathbf{x}(t) - \mathbf{w}_i(t)\,] & i \in N_c(t) \\ \mathbf{w}_i(t), \quad \text{o.w.} & \end{cases}$$

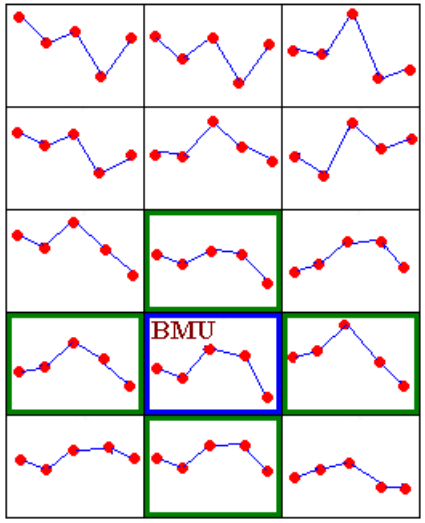
(4)



X01 ••• Xp

input node

5 x 3 output node



(2)

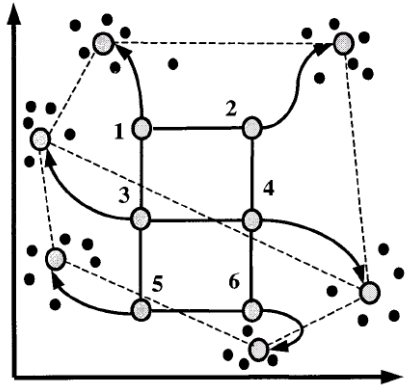
Best Match Unit

(6)

Incrementally decrease the learning rate and the neighborhood size, and repeat

(7)

Algorithm of SOM



Step 0: Initialize weights $\mathbf{w}_i(t)$.
Set topological neighborhood parameters $N_c(t)$.
Set learning rate parameters $\alpha(t)$ and $h_{ci}(t)$.

Step 1: For each input vector $\mathbf{x}(t)$, do

a. Finding a BMU: $\|\mathbf{x}(t) - \mathbf{w}_c(t)\| = \min_i \|\mathbf{x}(t) - \mathbf{w}_i(t)\|$

b. Learning process:

$$\mathbf{w}_i(t+1) = \begin{cases} \mathbf{w}_i(t) + h_{ci}(t)[\,\mathbf{x}(t) - \mathbf{w}_i(t)\,], & i \in N_c(t) \\ \mathbf{w}_i(t), & \text{o.w.} \end{cases}$$

c. Go to the next unvisited input vector. If there are no unvisited input vector left then go back to the very first one and go to Step 2.

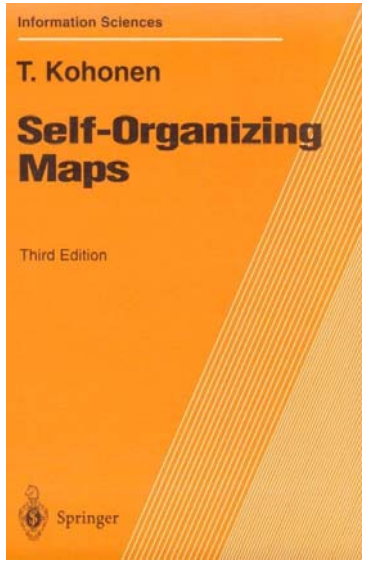
Step 2: Incrementally decrease the learning rate and the neighborhood size, and repeat Step 1.

Step 3: Keep doing Steps 1 and 2 for a sufficient number of iterations.

Apply SOM to Microarray Data

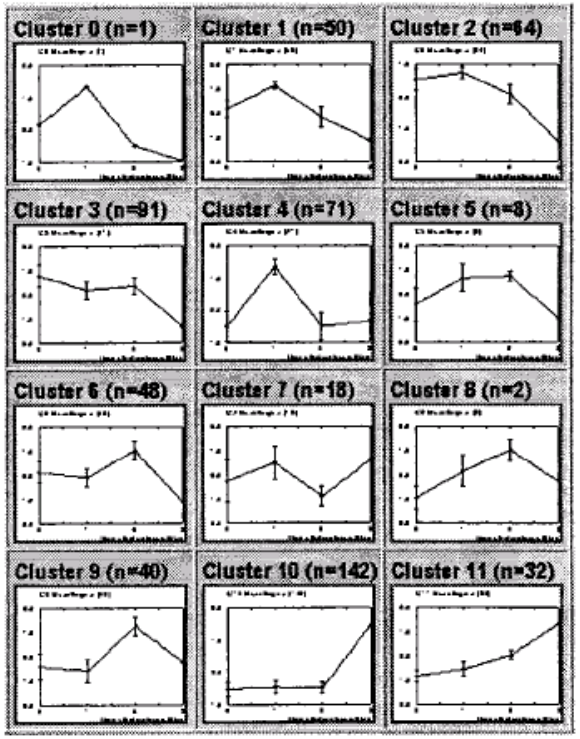
Macrophage Differentiation in HL-60 cells

Tamayo, P. et al. (1999). Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation.
Proc Natl Acad Sci 96:2907-2912.



1995, 1997, 2001

HL-60 4 × 3 SOM 567 genes



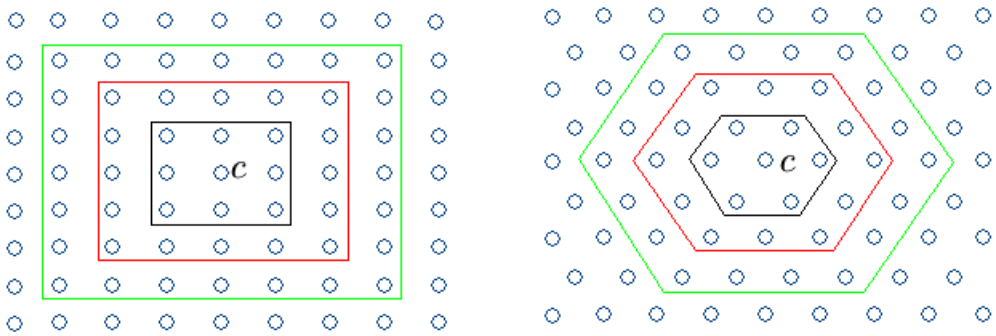
SOM - Initialization

Step 0: Initialize weights $\mathbf{w}_i(t)$.

Set topological neighborhood parameters $N_c(t)$.

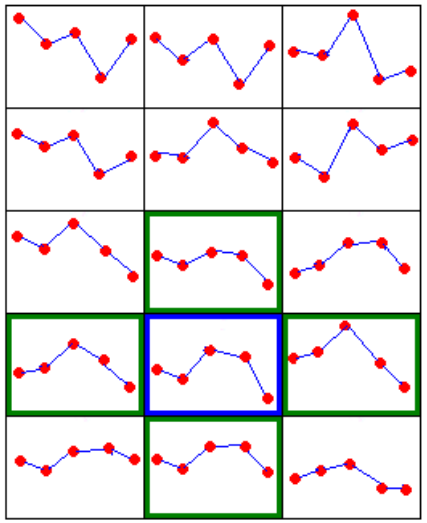
Set learning rate parameters $\alpha(t)$ and $h_{ci}(t)$.

Two examples of topological neighborhood.

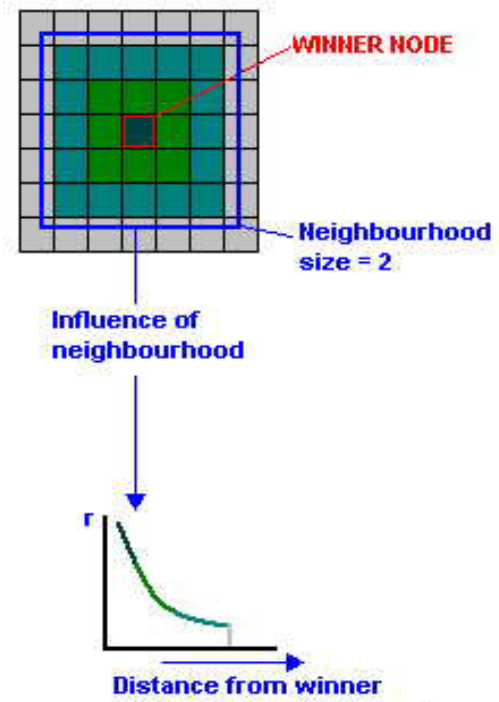


■ $N_c(t_1) = 1$, ■ $N_c(t_2) = 2$, ■ $N_c(t_3) = 3$, $t_1 < t_2 < t_3$

5 x 3 output node

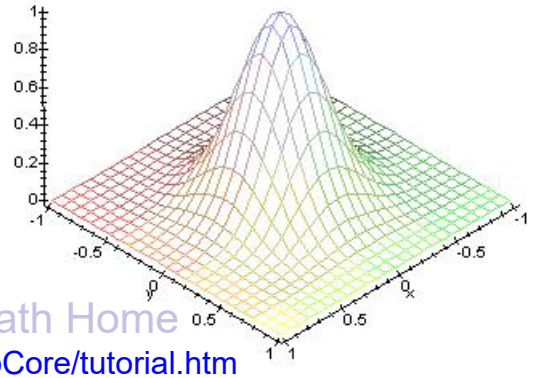


Neighborhood Functions



The winner node's weight is *modified* such that it becomes even more *similar* to the original input node's vector.

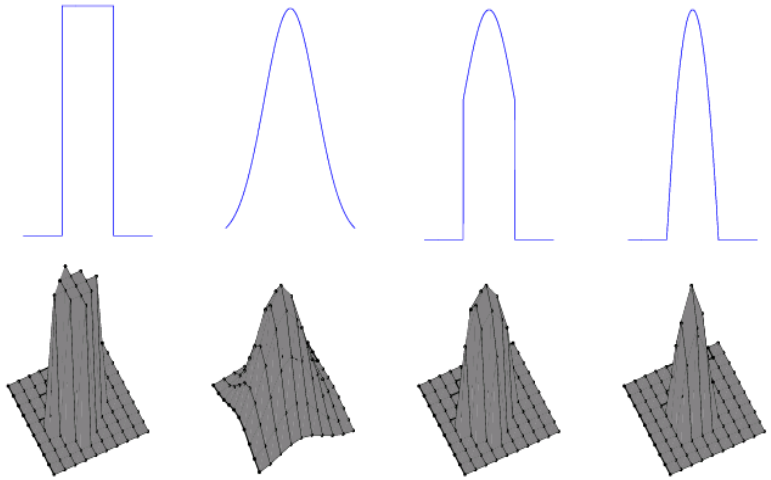
The neighborhood value has a two-fold character - a *size* and a *function of distance to influence*. One could even define a further third character - the *shape* of the neighborhood (in this case, a square - highlighted in blue).



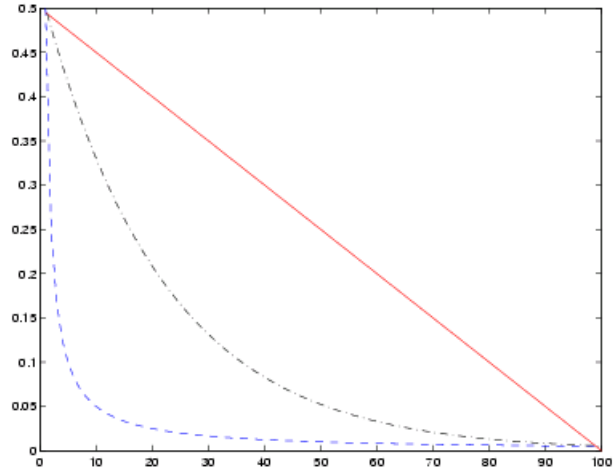
The peak of the Gaussian function would be the location of the winner node. As one moves out from that location, the *r* value decreases.

Figures source from: *SCI*path Home
http://www.ucl.ac.uk/oncology/MicroCore/tutorial.htm

Neighborhood Functions and Learning Rate



Different neighborhood functions. From the left
'bubble' $h_{ci}(t) = \mathbf{1}(\sigma_t - d_{ci})$,
'gaussian' $h_{ci}(t) = e^{-d_{ci}^2/2\sigma_t^2}$,
'cutgauss' $h_{ci}(t) = e^{-d_{ci}^2/2\sigma_t^2}\mathbf{1}(\sigma_t - d_{ci})$, and
'ep' $h_{ci}(t) = \max\{0, 1 - (\sigma_t - d_{ci})^2\}$, where
σ_t is the neighborhood radius at time t,
$d_{ci} = ||\mathbf{r}_c - \mathbf{r}_i||$ is the distance between map units c and i on the map grid
$\mathbf{1}(x)$ is the step function: $\mathbf{1}(x) = 0$ if $x < 0$ and $\mathbf{1}(x) = 1$ if $x \geq 0$.
The neighborhood radius used is $\sigma_t = 2$.
Source from Technical report on SOM Toolbox 2.0 for Matlab.

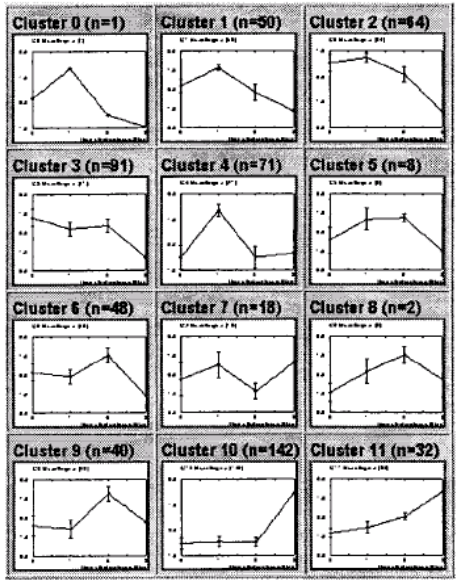


Different learning rate functions:
'linear' (solid line) $\alpha(t) = \alpha_0\,(1 - t/T)$,
'power' (dot-dashed) $\alpha(t) = \alpha_0(0.005/\alpha_0)^{t/T}$ and
'inv' (dashed) $\alpha(t) = \alpha_0/(1 + 100\,t/T)$, where T is the training length and α_0 is the initial learning rate.

Possible Parameters used in SOM

1. Grid dimension: 2D, 3D
2. Grid shape: in 2D → Rectangle, Hexagon, ...
3. Number of node: in 2D Rectangle → 4×6, 5×5, 3×8,...
4. Neighborhood function: Bubble kernel, Gaussian kernel, ...
5. Neighborhood size: radius of $N_c(t)$
6. Learning rate function: $\alpha(t)$
7. Initial weights: random, use input vector
8. Order of input vectors: random, ...
9. Ways of learning: number of iteration,...

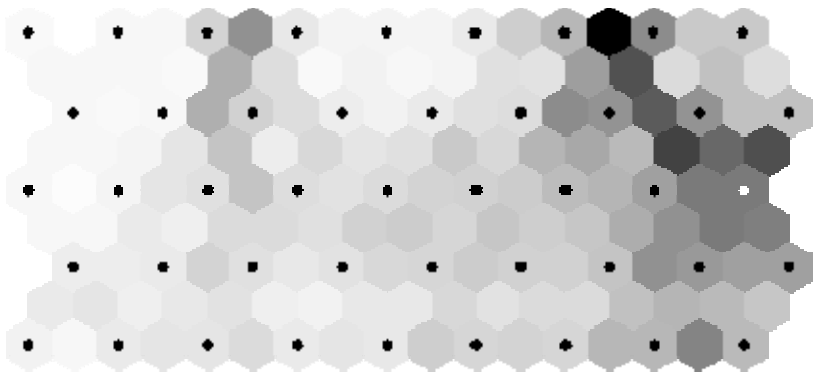
HL-60 4×3 SOM 567 genes



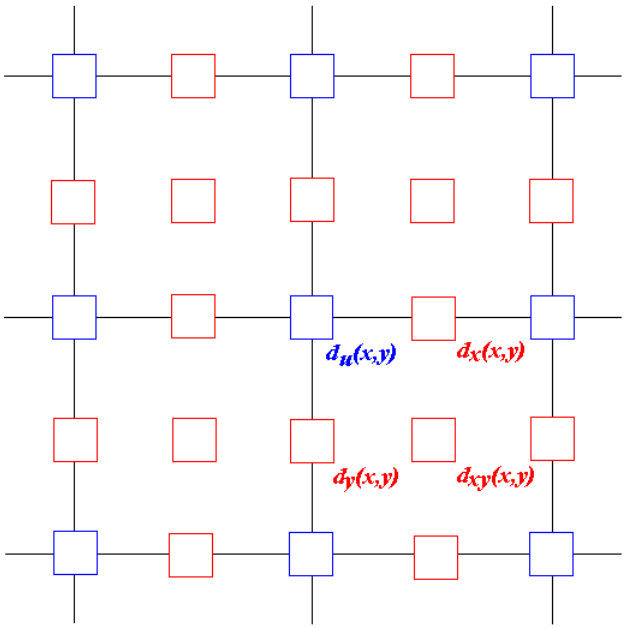
U-matrix: Unified Matrix Method

(Ultsch and Siemon 1989, Ultsch 1993)

U-matrix representation of SOM visualizes the distance between the neurons. The distance between the adjacent neurons is calculated and presented with different colorings between the adjacent nodes.



U-matrix representation of the SOM



$b(x,y)$: matrix of neurons, of size $n_x \times n_y$.

$w_i(x,y)$: matrix of weights.

$u(x,y)$: U-matrix of size $(2n_x - 1) \times (2n_y - 1)$.

$d_x(x,y)$: $\|b(x,y) - b(x+1,y)\| = \sqrt{\sum_i [w_i(x,y) - w_i(x+1,y)]^2}$

$d_y(x,y)$: $\|b(x,y) - b(x,y+1)\| = \sqrt{\sum_i [w_i(x,y) - w_i(x,y+1)]^2}$

$d_{xy}(x,y)$: $\frac{1}{2}[\frac{\|b(x,y)-b(x+1,y+1)\|}{\sqrt{2}} + \frac{\|b(x,y+1)-b(x+1,y)\|}{\sqrt{2}}]$

$d_u(x,y)$: the median of the surrounding elements.

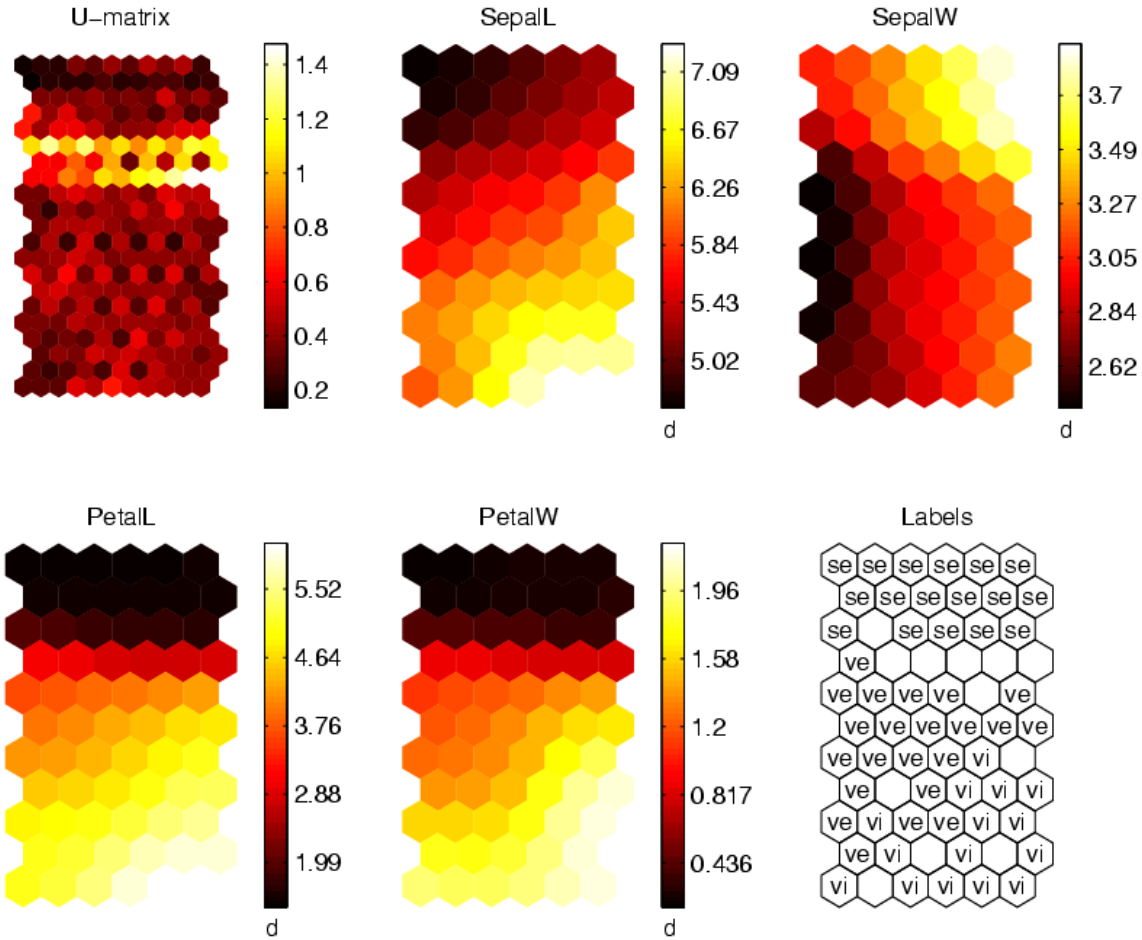
SOM: iris example

no.	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
...					
76	6.6	3.0	4.4	1.4	versicolor
...					
150	5.9	3.0	5.1	1.8	virginica

Iris Flowers

○ *Iris Setosa* ○ *Iris Versicolor* ○ *Iris Virginica*

The sepal length, sepal width, petal length, and petal width are measured in centimeters on fifty iris specimens from each of three species, *Iris setosa*, *I. versicolor*, and *I. virginica*. Fisher (1936)

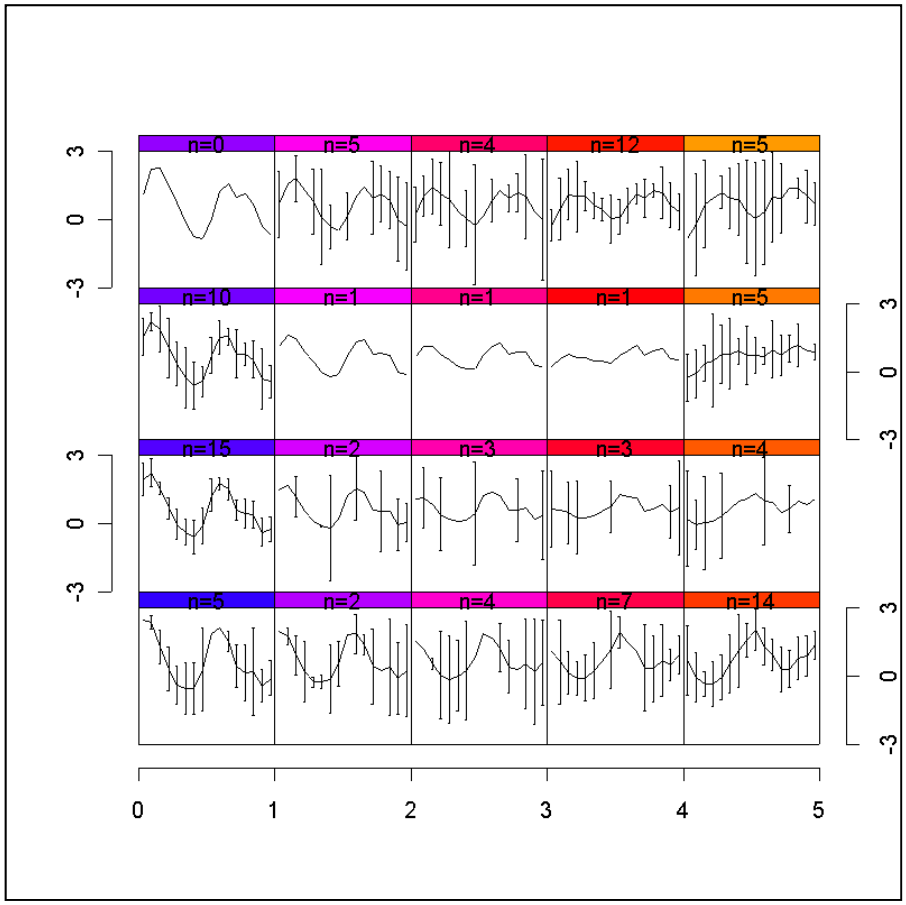


Source from technical Report on SOM Toolbox 2.0 for Matlab

Software: SOM Toolbox 2.0 for Matlab

R: SOM

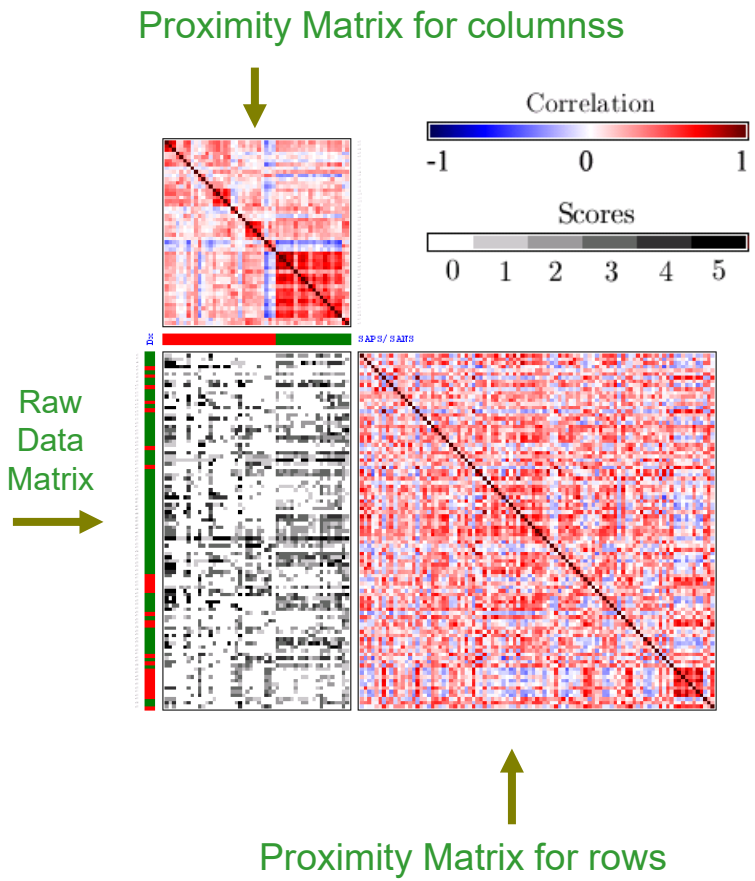
```
library(som)
cell.som <- som(cell.sdata, xdim=5, ydim=4, topol="rect", neigh="gaussian")
plot(cell.som)
```

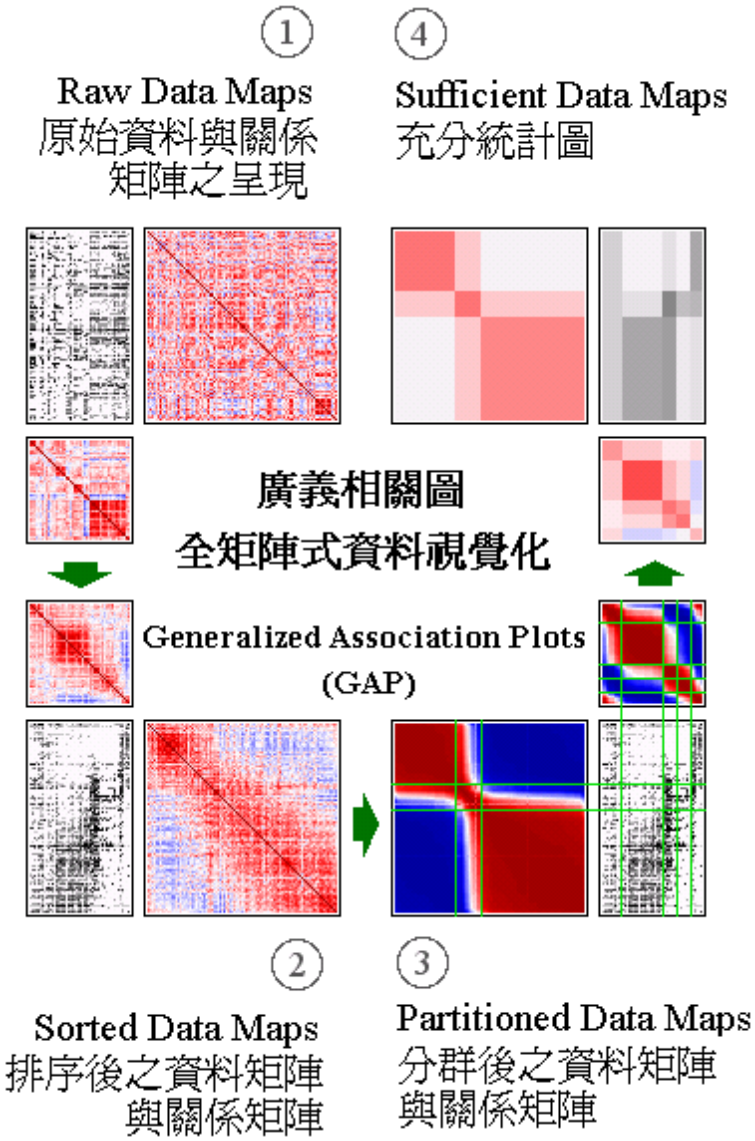


Generalized Association Plots (GAP)

(Chen, 2002)

- 95 patients: 69 schizophrenic and 26 bipolar disorders
- SAPS: 30 items, SANS: 20 items
- Six point scale (0-5).

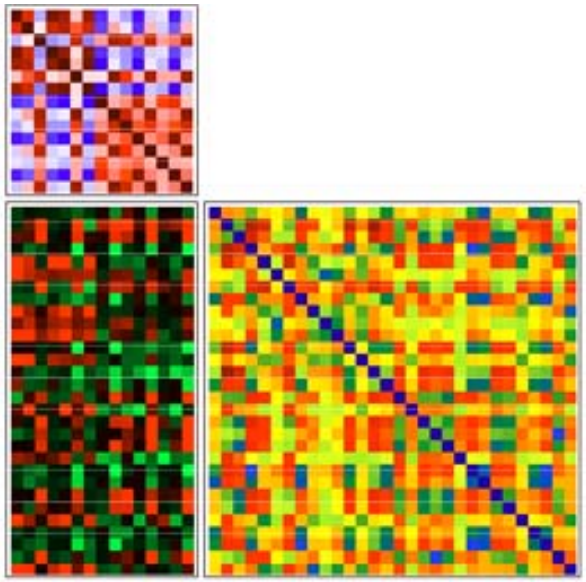


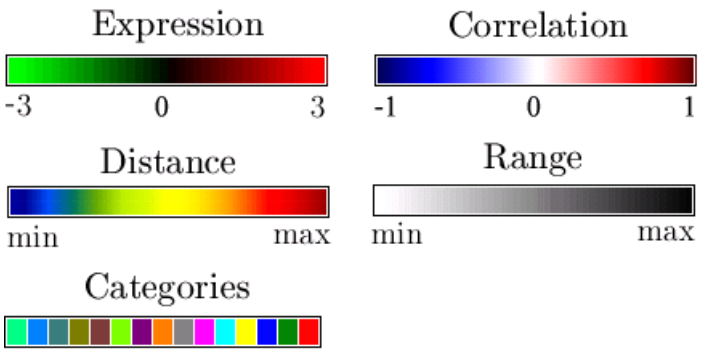


Presentation of Raw Data Matrix

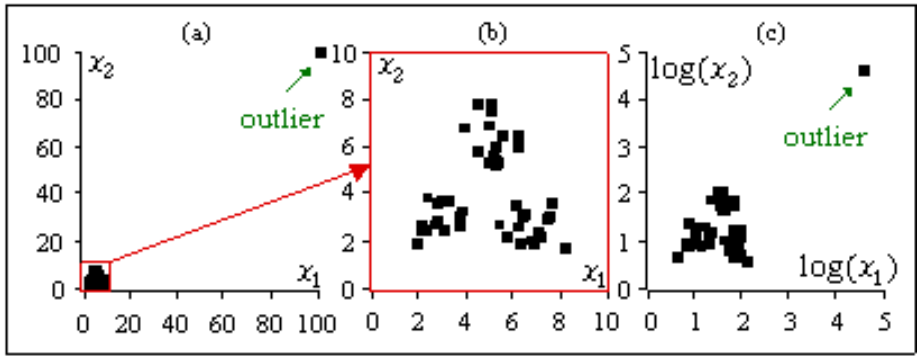
Image source: Dr. Chen Chun-houh's Silde

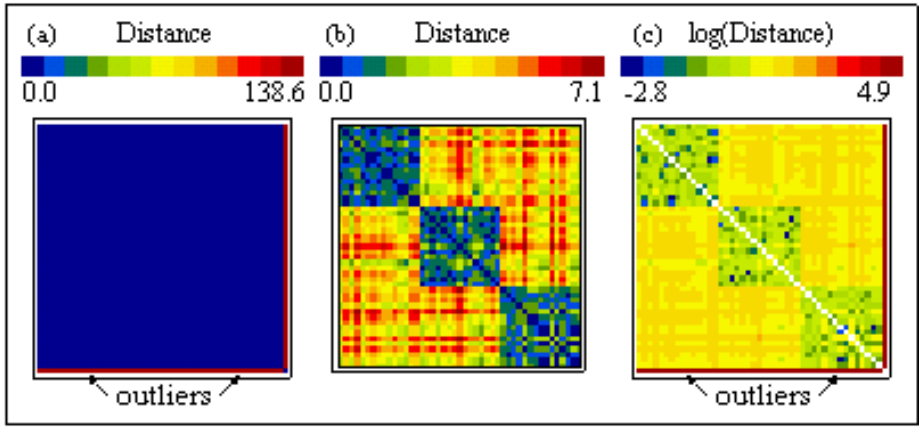
1. Color spectrum
2. Variable transformation
3. Selection of proximity

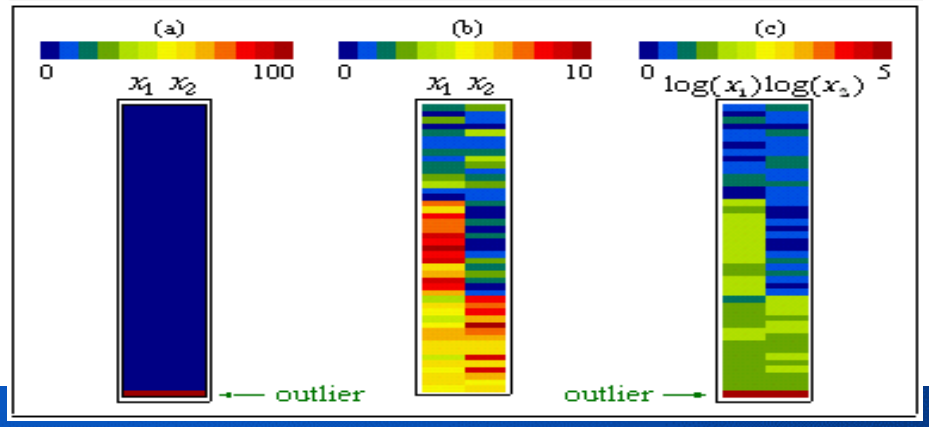




"Resolution" of a Statistical Graph

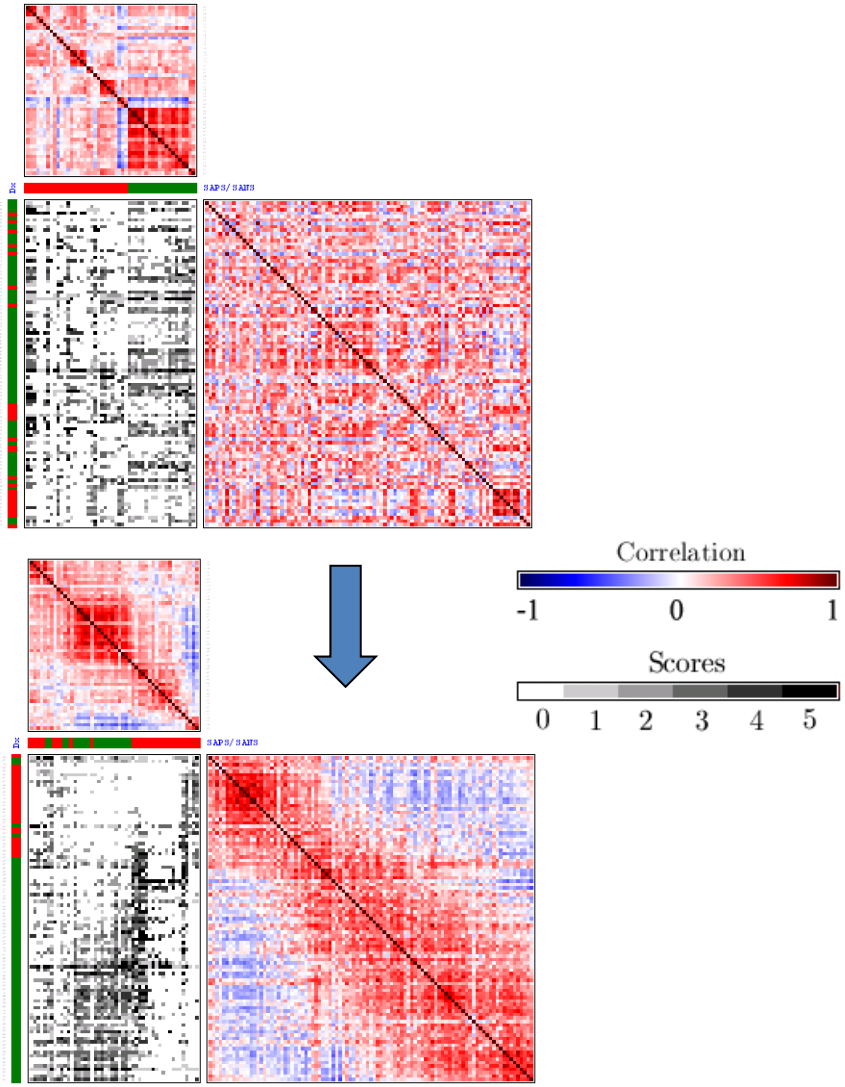


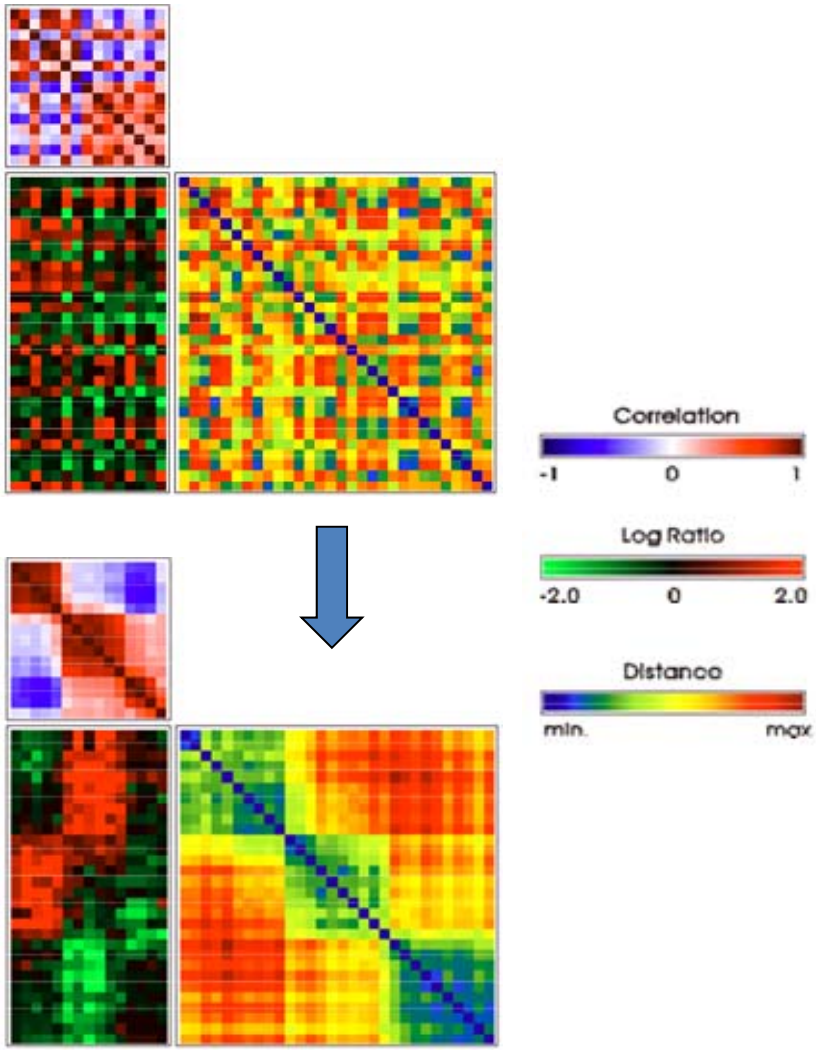




Concept of Relativity of a Statistical Graph

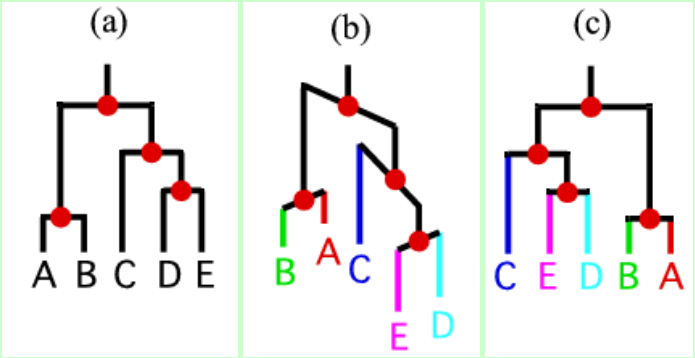
Placing similar (different) objects at closer (distant) positions



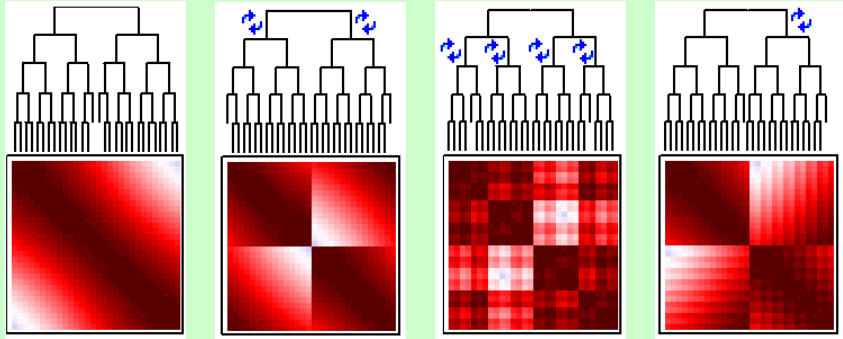


Seriation Problem for Hierarchical Clustering

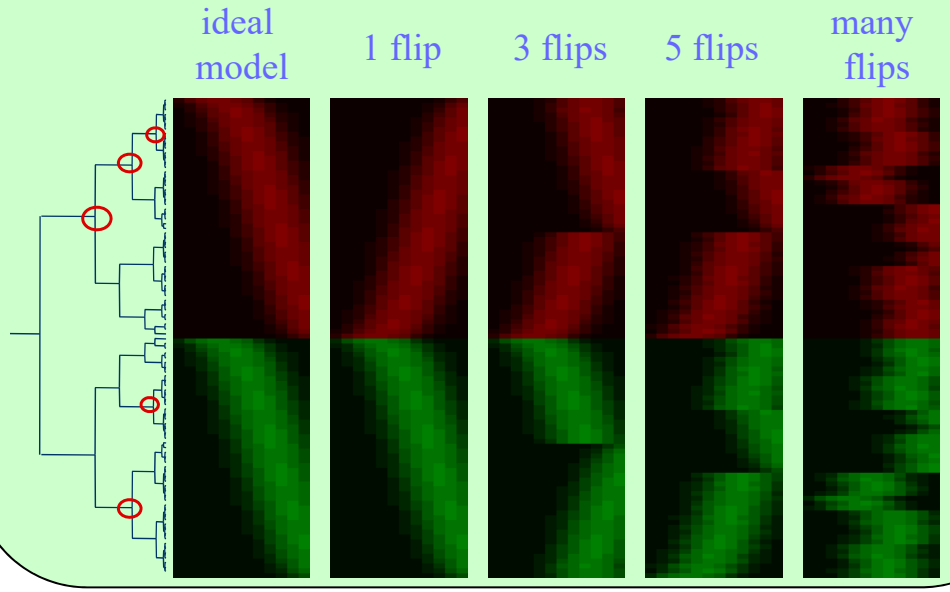
Tree seriation



Tree seriation for proximity matrices

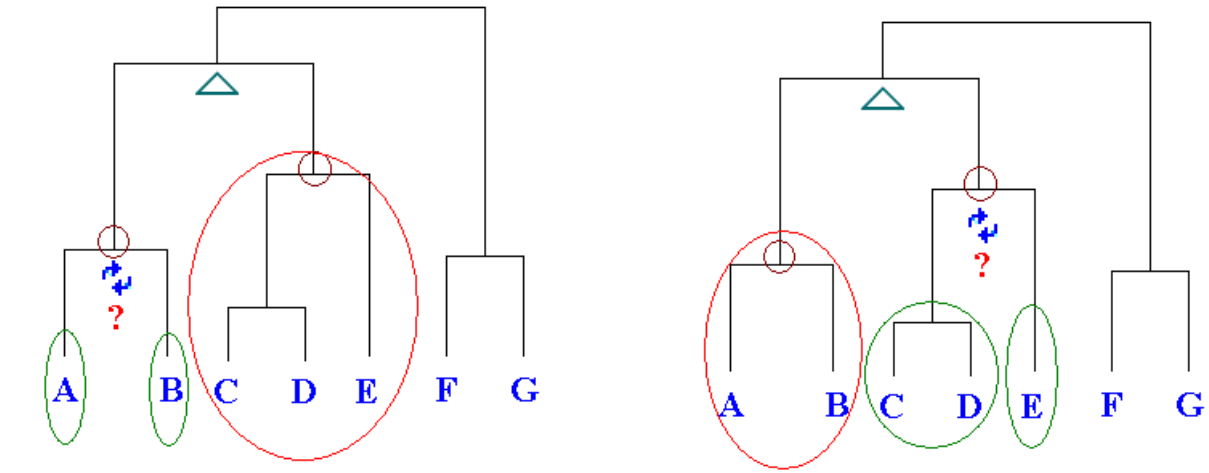


Tree seriation for raw data matrices



Different Seriations Generated from Identical Tree Structure

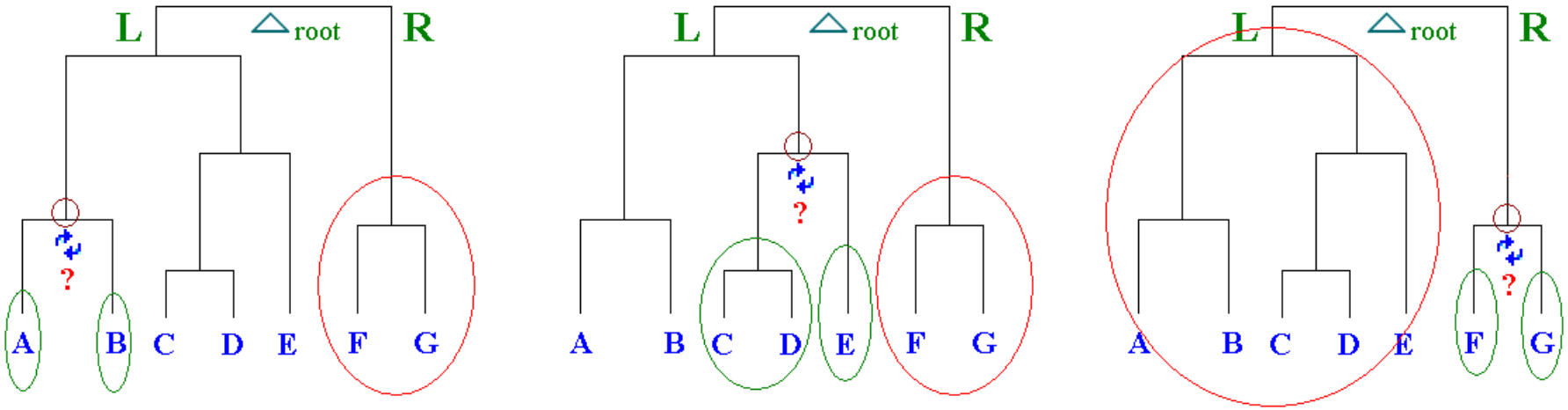
Internal Tree Flips: Uncle Approach



if d(A, {C,D, E}) < d(B, {C,D, E}) then flip

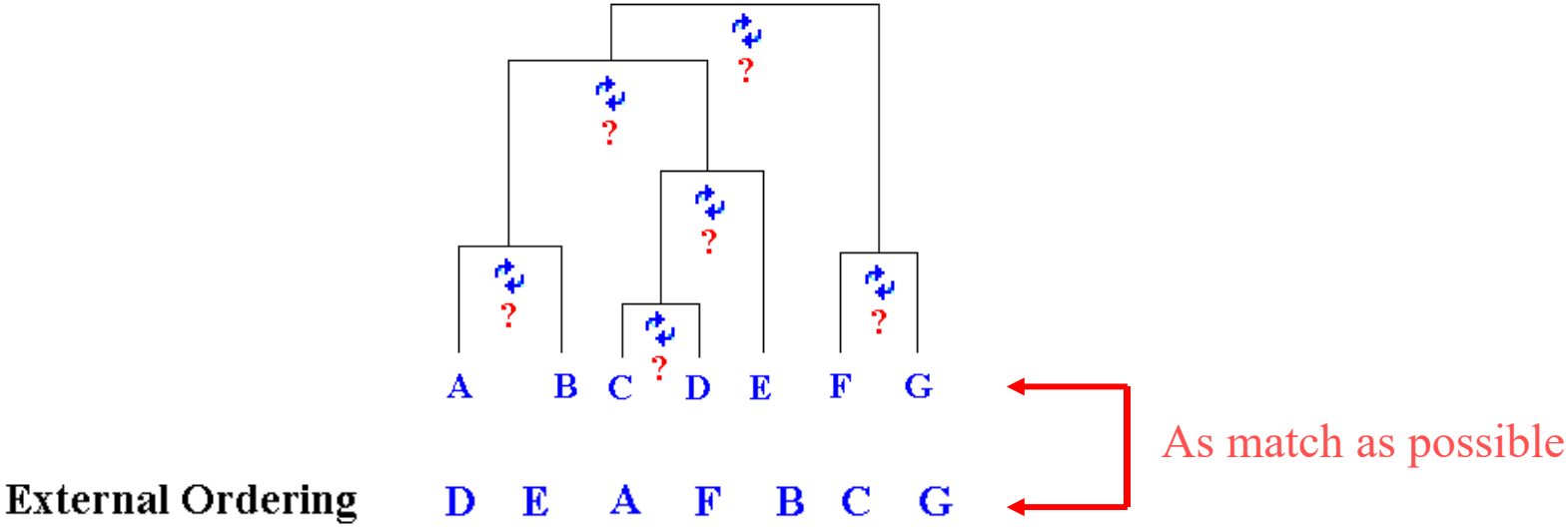
Further reading: Ziv Bar-Joseph, David K. Gifford, and Tommi S. Jaakkola, (2001), **Fast Optimal Leaf Ordering** for Hierarchical Clustering. Bioinformatics 17(Suppl. 1):S22–S29.

Internal Tree Flips: GrandPa Approach



Further reading: Ziv Bar-Joseph, David K. Gifford, and Tommi S. Jaakkola, (2001), **Fast Optimal Leaf Ordering** for Hierarchical Clustering. Bioinformatics 17(Suppl. 1):S22–S29.

External Tree Flips

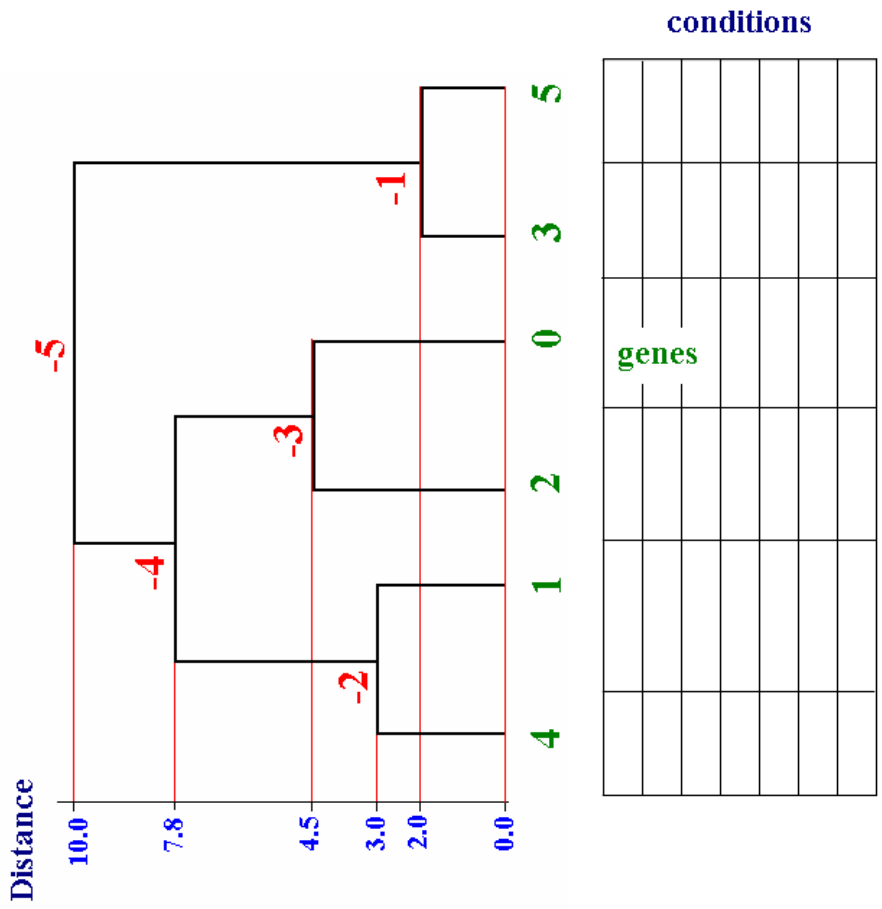


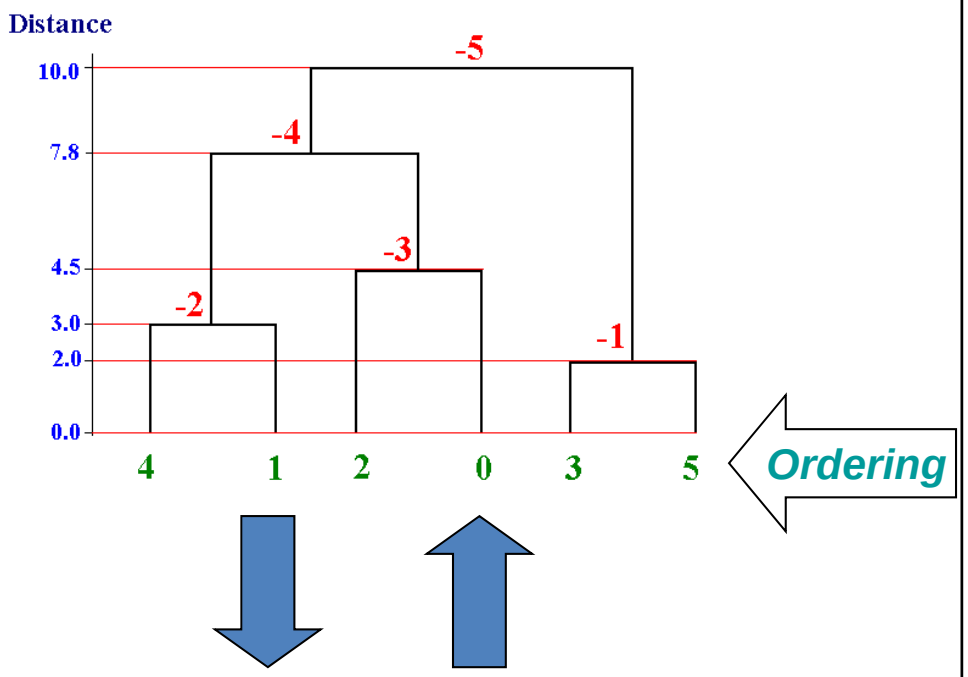
How to build an external ordering?

(1) Based on average expression level (Cluster Software, Eisen et al 1998)
(2) Using the results of a one-dimensional SOM
(3) …

Further reading: Tien, Y. J., Lee, Y. S, Wu, H. M. and Chen, C. H. (2006) Integration of clustering and visualization methods for simultaneously identifying coherent local clusters with smooth global patterns in gene expression profiles. BMC Bioinformatics.

Dendrogram and Tree Storage



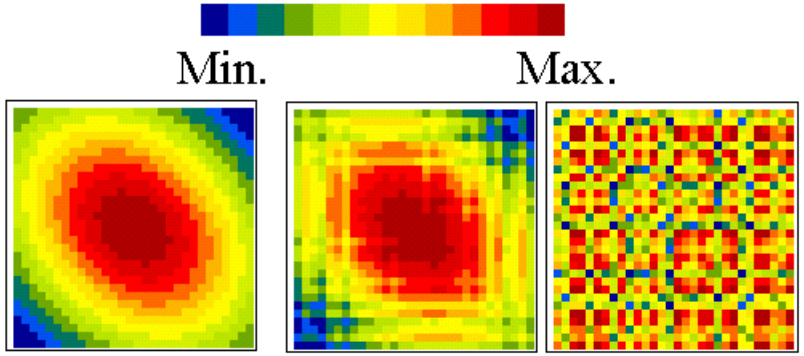


no	NodeID	Left	Right	Distance
0	-1	3	5	2
1	-2	4	1	3
2	-3	2	0	4.5
3	-4	-2	-3	7.8
4	-5	-4	-1	10

For example:

Cluster and TreeView, R

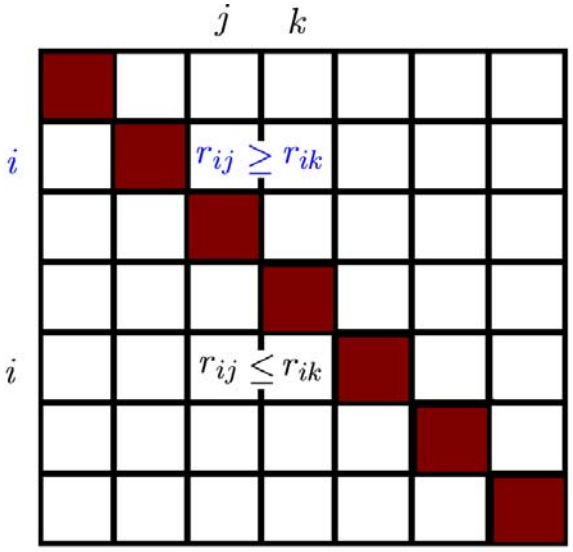
Criteria for a "good" Permutation



Robinson **pre-Robinson**

When T is symmetric, we usually want T' to approximate a Robinson form (Robinson (1951)).

Robinson Form



$r_{ij} \leq r_{ik}$ if $j < k < i$, $r_{ij} \geq r_{ik}$ if $i < j < k$

Global/local Criterion:

Anti-Robinson Measurements

permuted matrix, $D = [d_{ij}]$

$$AR(i) = \sum_{i=1}^{p} \Big[\sum_{j<k<i} I(d_{ij} < d_{ik}) + \sum_{i<j<k} I(d_{ij} > d_{ik}) \Big],$$

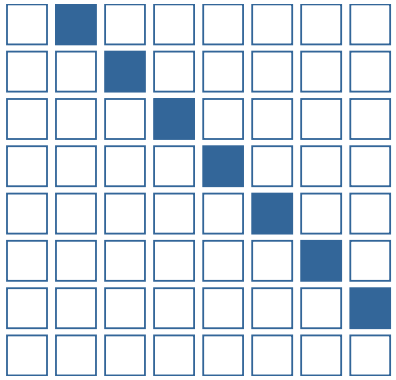
$$AR(s) = \sum_{i=1}^{p} \Big[\sum_{j<k<i} I(d_{ij} < d_{ik}) \cdot |d_{ij} - d_{ik}| + \sum_{i<j<k} I(d_{ij} > d_{ik}) \cdot |d_{ij} - d_{ik}| \Big],$$

$$AR(w) = \sum_{i=1}^{p} \Big[\sum_{j<k<i} I(d_{ij} < d_{ik}) |j-k| |d_{ij} - d_{ik}| + \sum_{i<j<k} I(d_{ij} > d_{ik}) |j-k| |d_{ij} - d_{ik}| \Big].$$

Criteria for a "good" Permutation

$$\text{permuted matrix, } D = [d_{ij}]$$

Local criterion: **Minimal Span Loss Function**

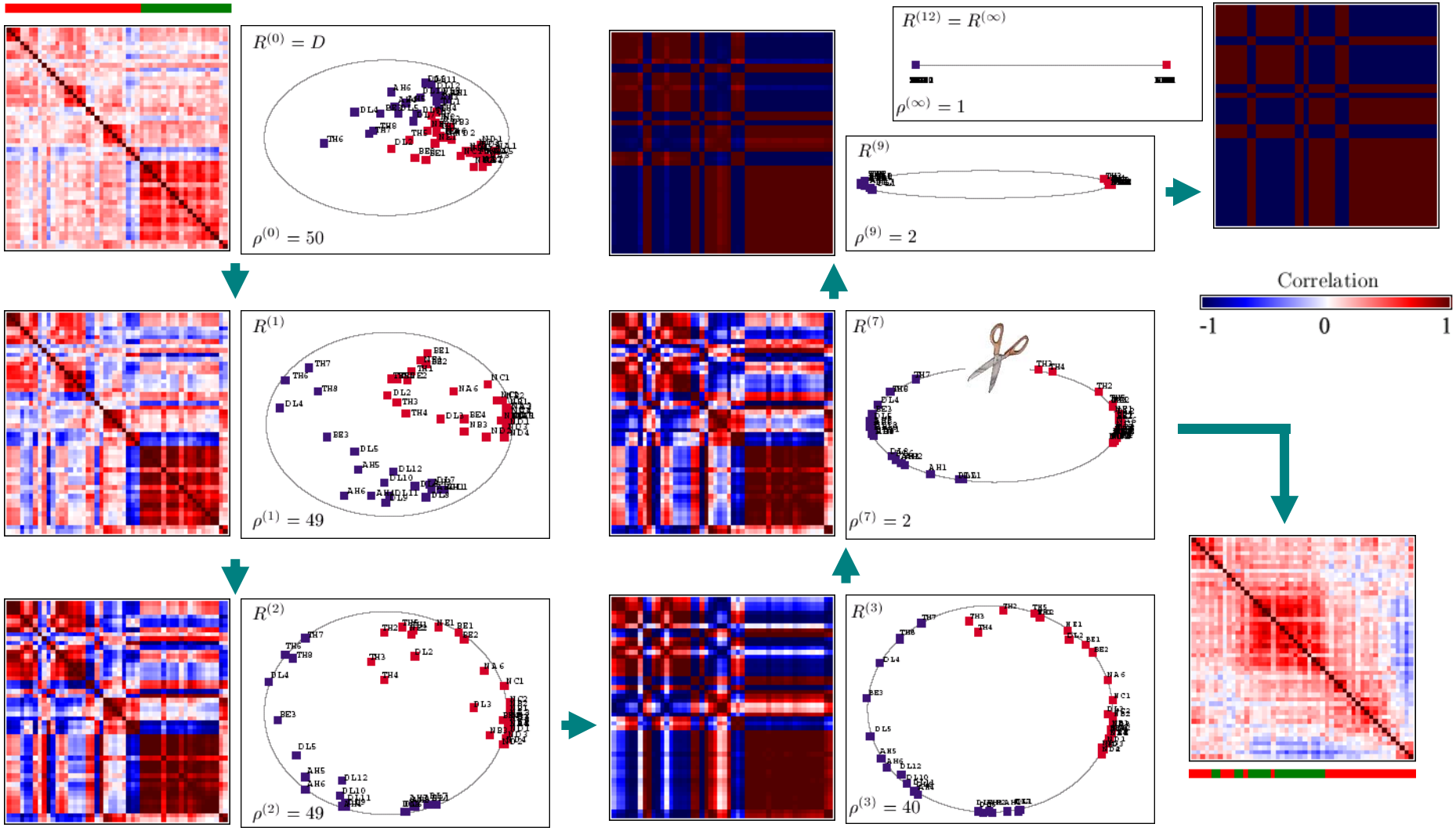


$$MS = \sum_{i=1}^{n-1} d_{i,i+1}$$

Further Reading: Michael Friendly , Ernest Kwan, (2003) Effect ordering for data displays, Computational Statistics & Data Analysis, v.43 n.4, p.509-539.

GAP Rank-Two Elliptical Seriation

- Seriation Algorithms with Converging Correlation Matrices.
- When the sequence reaches an iteration with rank two, the p objects fall on an ellipse and have unique relative position on the ellipse.



Global vs Local Seriation

GAP Elliptical Seriation

An algorithm for identifying global clustering patterns and smoothing temporal expression profiles

GAP Elliptical Seriation

Michael Eisen Tree Seriation

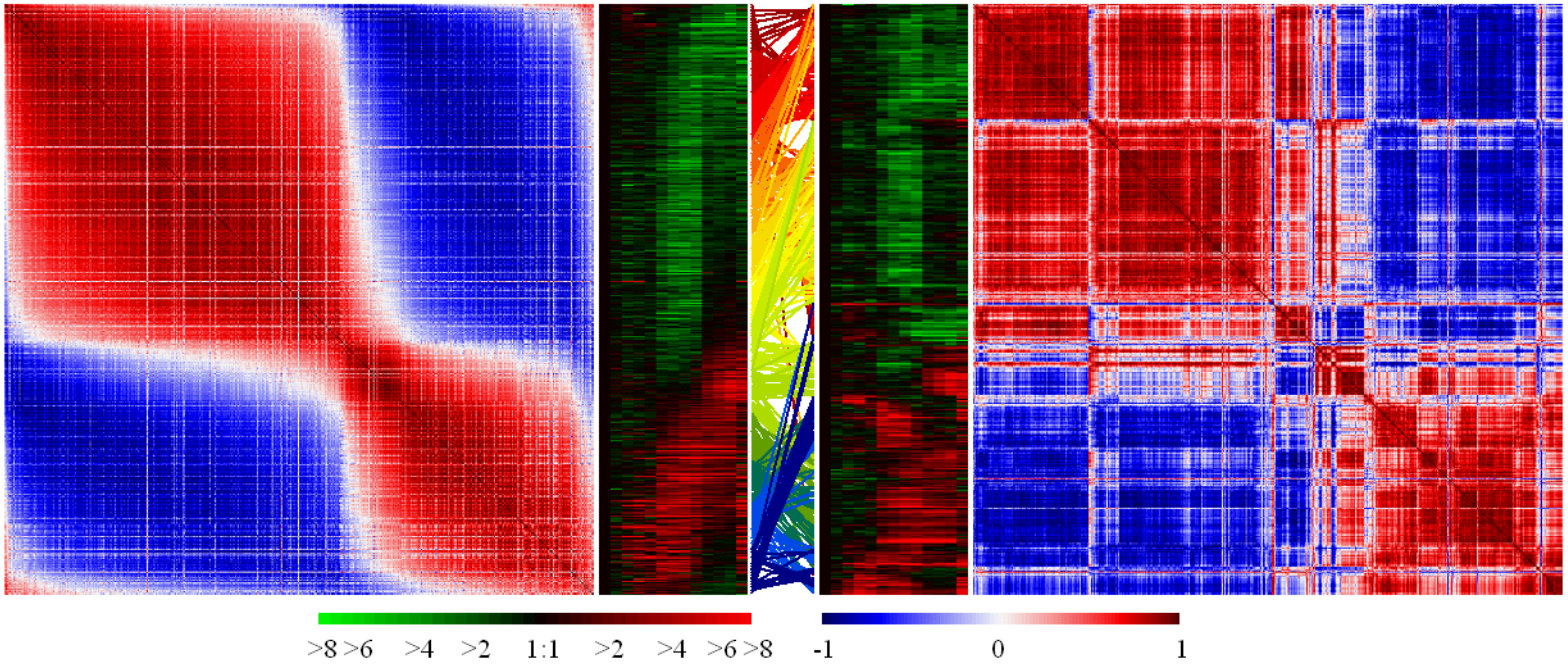


Image source: Dr. Chen Chun-houh's slide

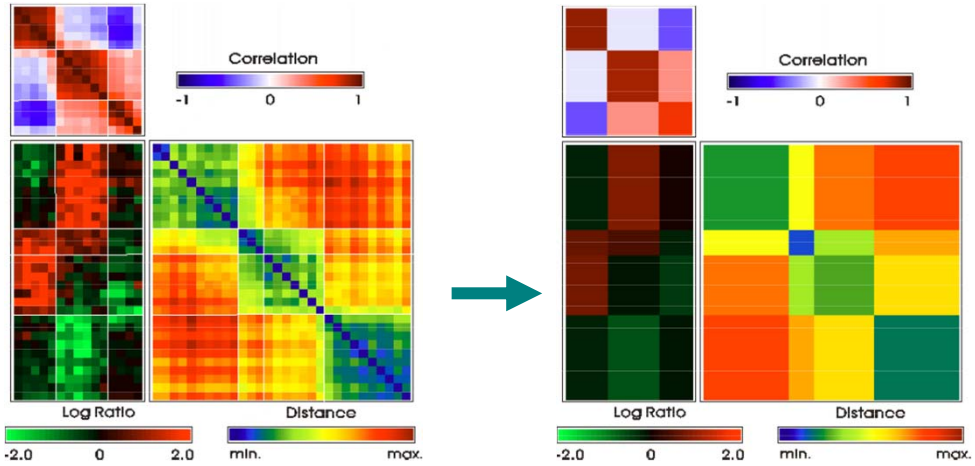
Sufficient Graph

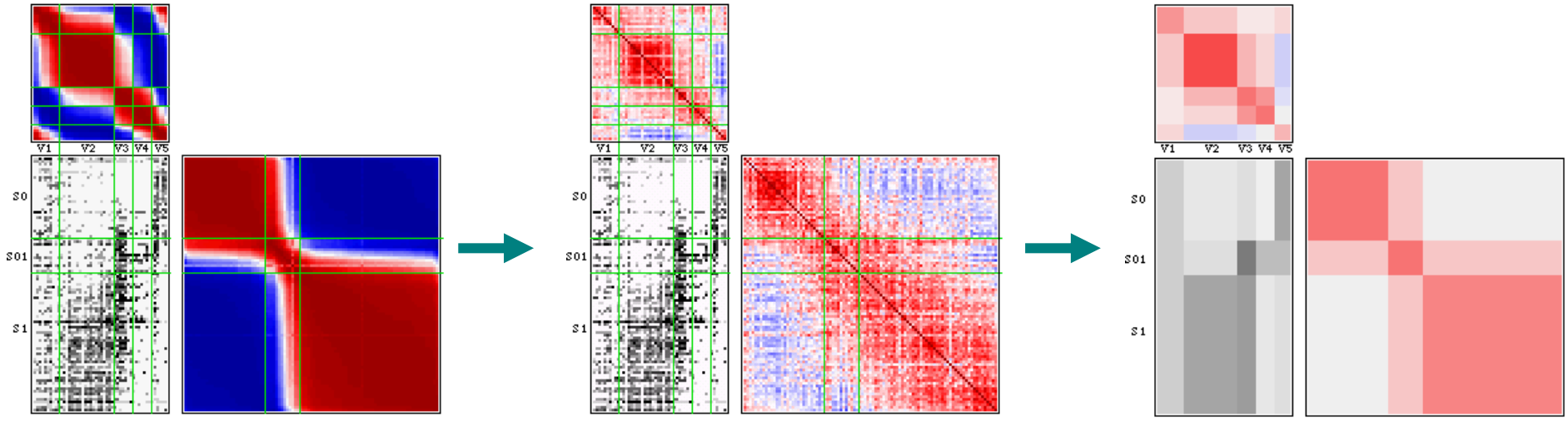
	A	B	C	D	E	F
1	學號	小考1	期中考	小考2	期末考	報告
2	A01	69	92	85	45	62
3	A02	66	90	83	36	90
4	A03	72	92	80	62	70
5	A04	68	90	60	37	95
6	A05	74	60	86	54	70
7	A06	77	90	88	88	95
8	A07	73	88	77	51	95
9	A08	61	90	84	40	82
10	A09	66	88	82	39	80
11	A10	76	75	87	72	80
12	A11	64	90	90	26	95
13	A12	75	90	60	55	70
14	A13	92	90	83	90	95

Sufficient Statistic

	小考1	期中考	小考2	期末考	報告
平均	71.77	86.54	80.38	53.46	83
低平均	65.67	81.83	73.67	53.67	72
高平均	77.83	90.67	86.67	53.67	94.17

70





Generalization and Flexibility

Sedimented MV for patients and symptoms.

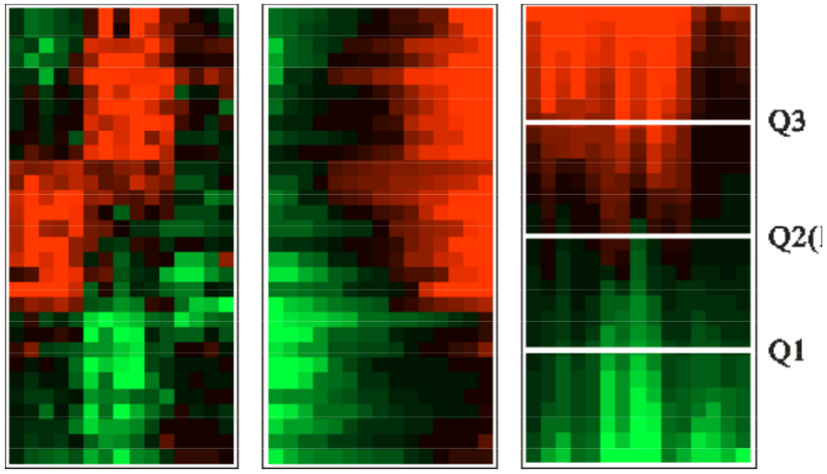


Image source: Chen etal 2004

The sediment MV for patients: express severity structure.

The sediment MV for symptoms: this is a side-by-side bar-chart and box-plot which displays the distribution structure for all symptoms simultaneously.

Sectional MV for the permuted correlation coefficient map.

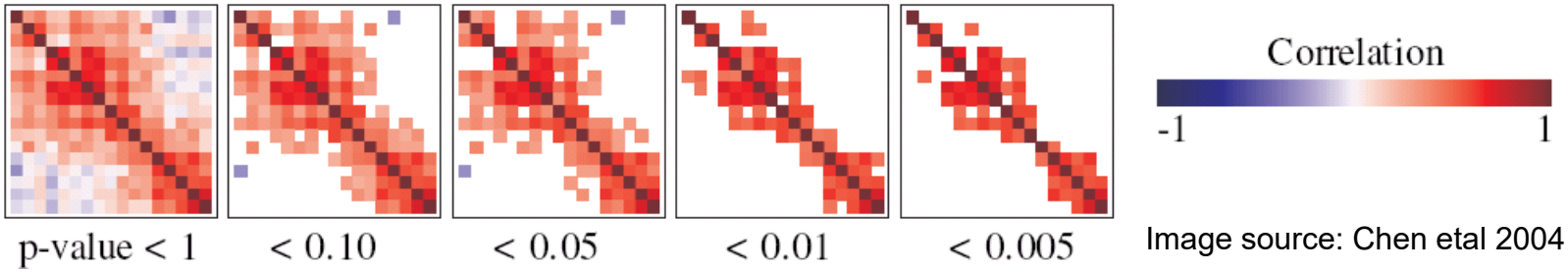
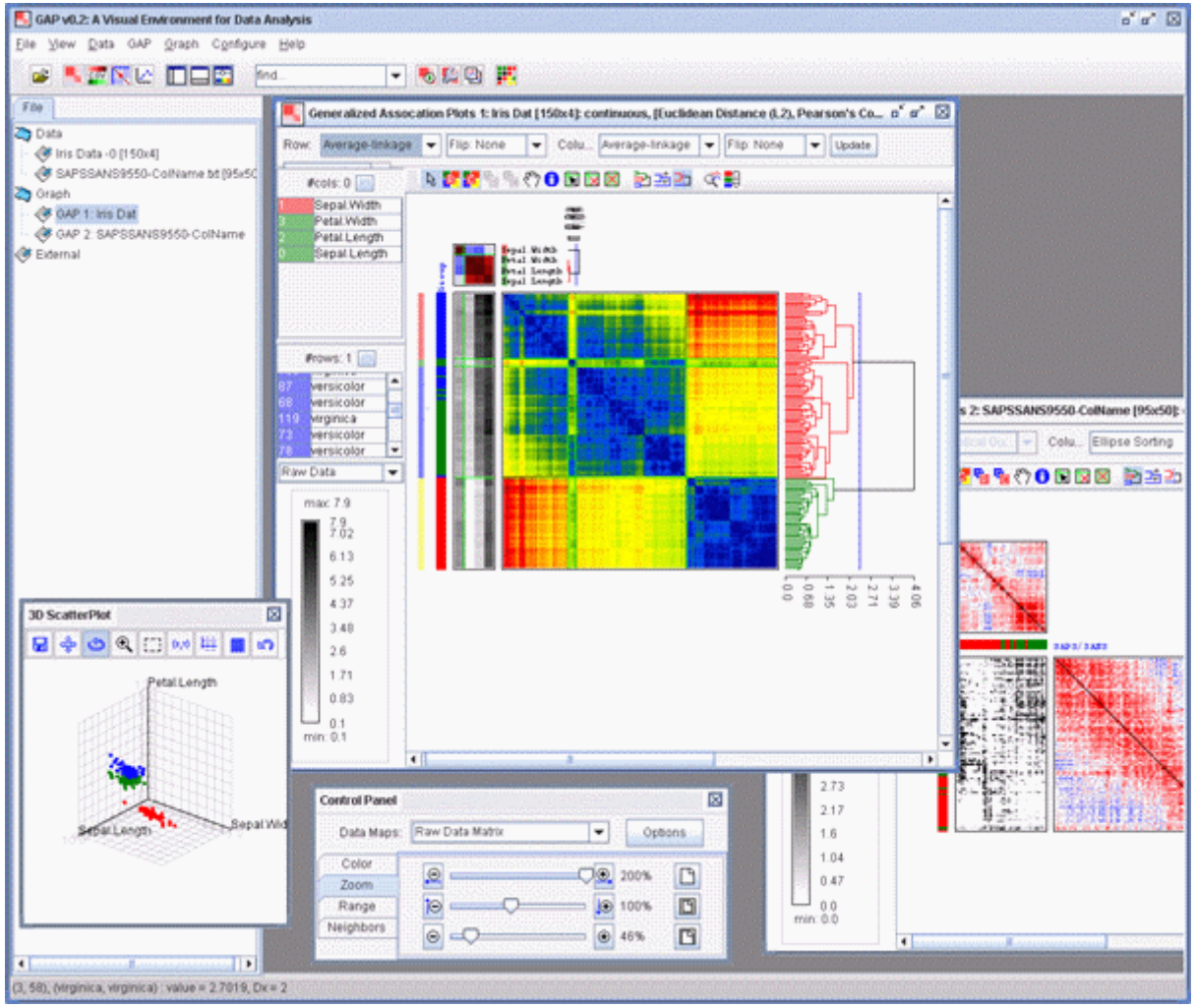


Image source: Chen etal 2004

GAP Software

http://www.hmwu.idv.tw/GAPSoftware/

MacOS: http://gap.stat.sinica.edu.tw/Software/download.htm



See also:
R `seriation` package

Visualization of Data Matrices

Simple ← Information Visualization of Data Matrices → Difficult

Continuous (Gene/Time)	Ordinal (Patient/Symptom)	Binary (Mouse/Tumor)	Categorical (Subject/SNP)

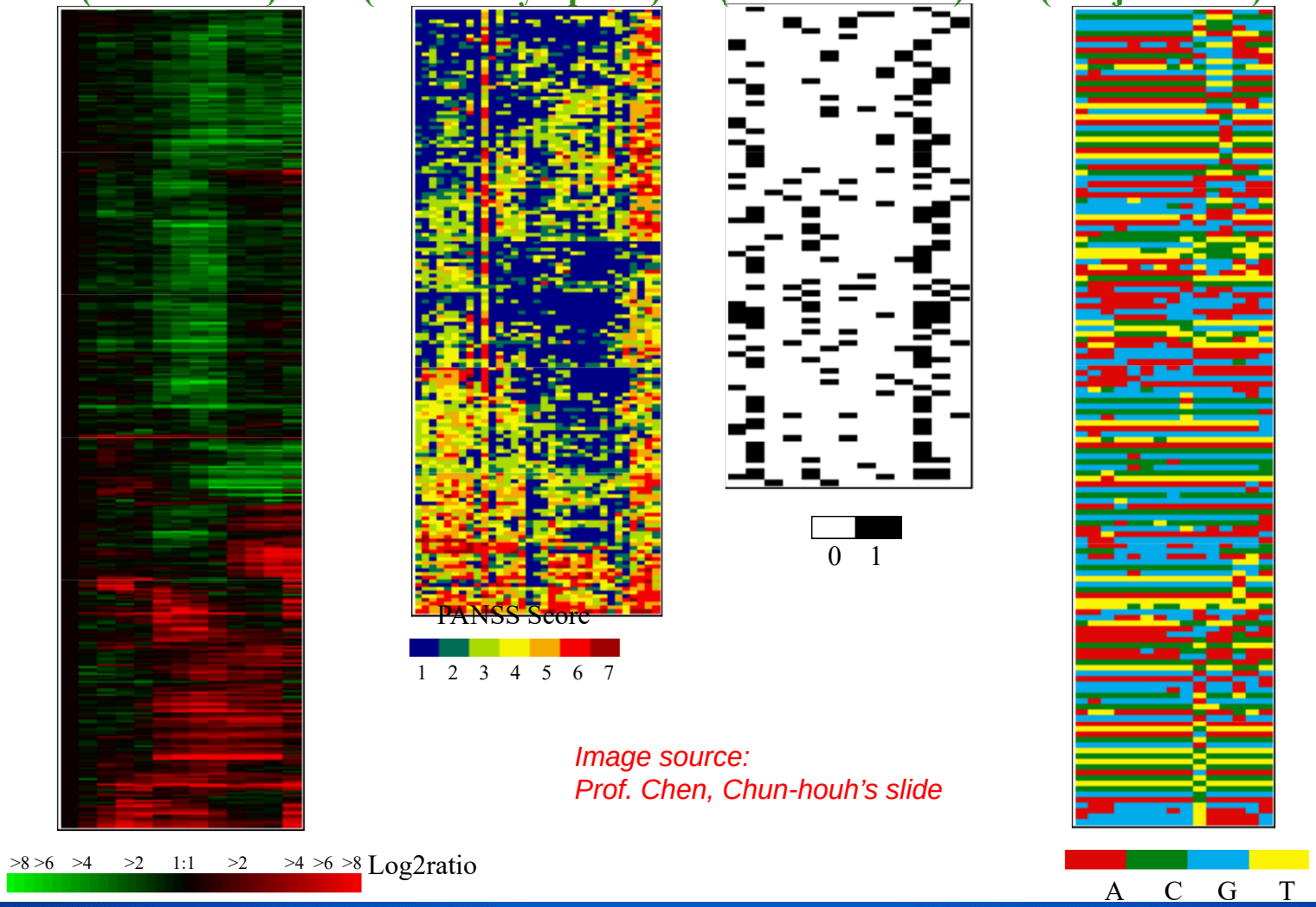
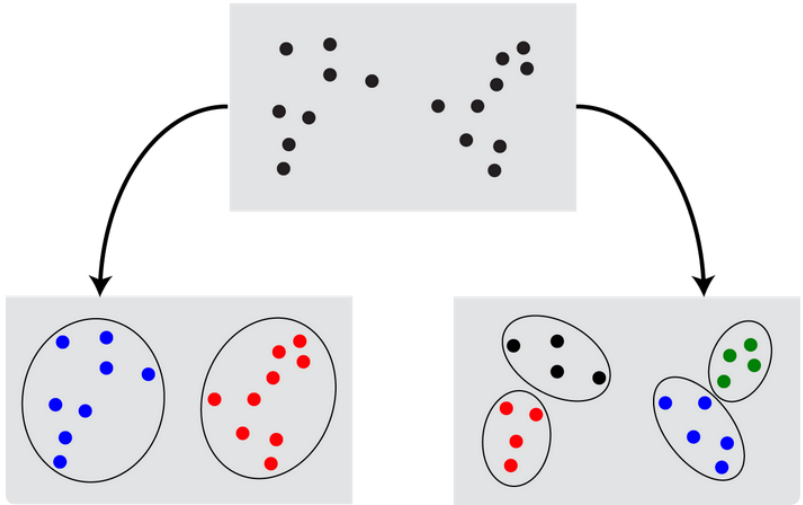


Image source:
Prof. Chen, Chun-houh's slide

Choosing the Number of Clusters (1/2)

Are these data better described by 2 or 4 clusters?

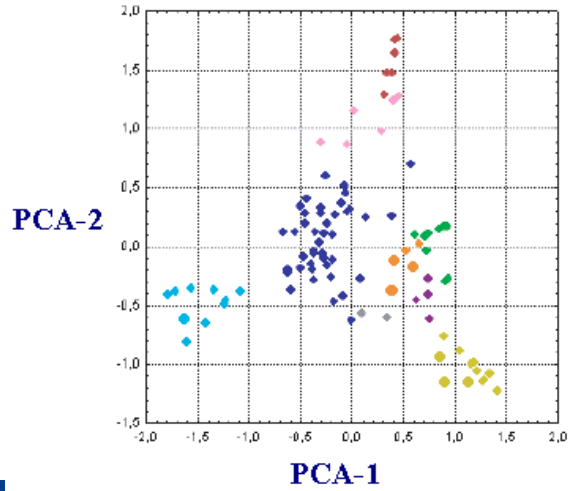


Source: http://alexhwilliams.info/itsneuronalblog/2015/09/11/clustering1/

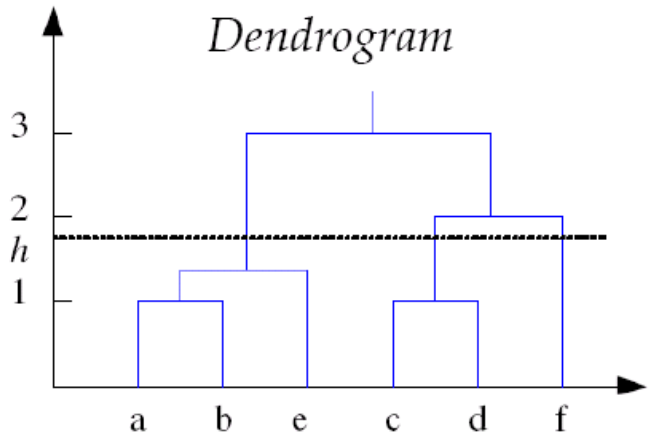
(1) K is defined by the application.

Calinski and Harabasz (1974): CH(k)
Hartigan (1975): $H(k)$
Krzanowski and Lai (1985): KL(k)
Kaufman and Rousseeuw (1990): $s(i)$

(2) Plot the data in two PCA dimensions.



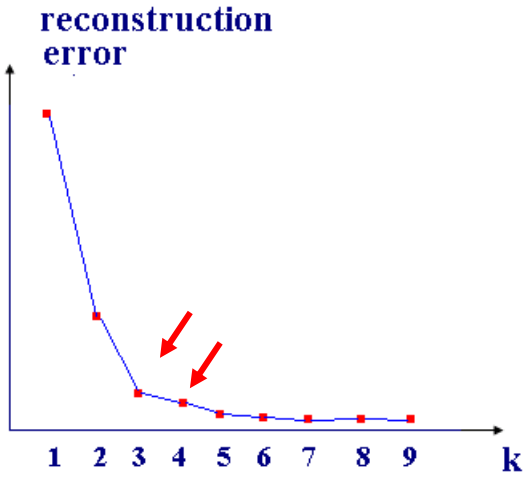
(3) Hierarchical clustering: look at the difference between levels in the tree.



Choosing the Number of Clusters (2/2)

(4) Plot the **reconstruction error** or log likelihood as a function of k, and look for the elbow.

(e.g., k-means: within-cluster sum of squares)

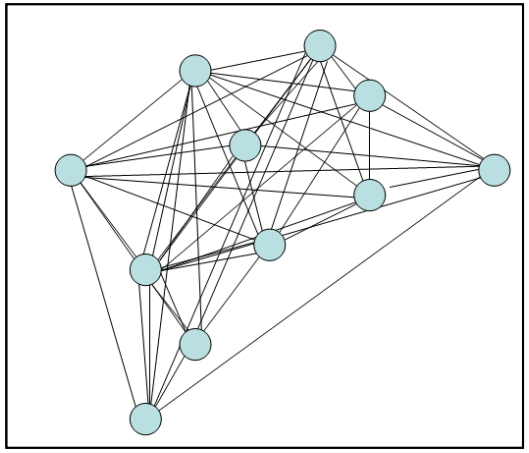


Scree Plot

```
> kmeans(iris[, 1:4], 3)
K-means clustering with 3 clusters of sizes 62,
50, 38
...
Within cluster sum of squares by cluster:
[1] 39.82097 15.15100 23.87947
 (between_SS / total_SS =  88.4 %)
...
```

clusGap {cluster}

Gap Statistic for Estimating the Number of Clusters



Computational Implementation

Within-Cluster Sum of Squares

$$D_r = \sum_{i \in C_r} \sum_{j \in C_r} \| x_i - x_j \|^2$$

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

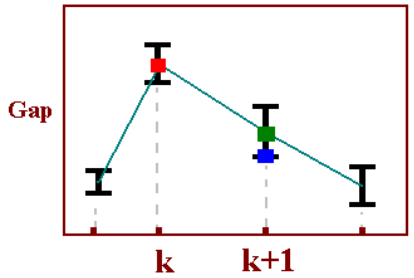
J. R. Statist. Soc. B (2001)
63, *Part* 2, *pp.* 411–423

Estimating the number of clusters in a data set via the gap statistic

Robert Tibshirani, Guenther Walther and Trevor Hastie

Stanford University, USA

$$\mathrm{Gap}_n(k) = E_n^*\{\log(W_k)\} - \log(W_k)$$

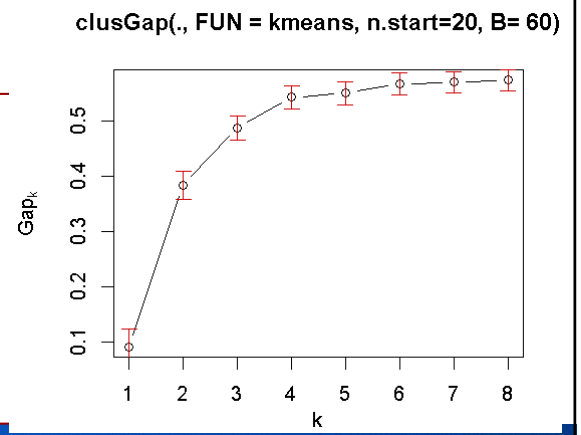


choose the number of clusters via

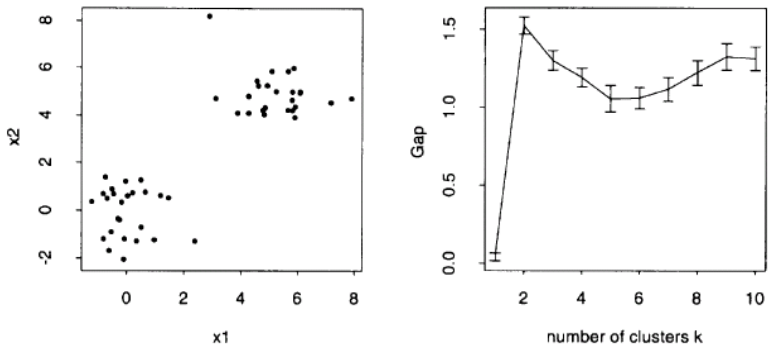
$$\hat{k} = \text{ smallest } k \text{ such that}$$

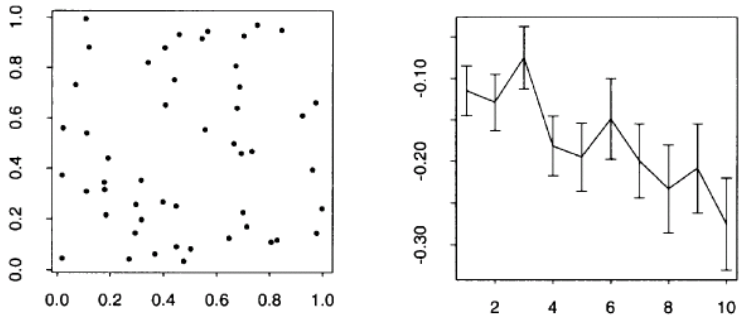
$$\mathrm{Gap}(k) \geq \mathrm{Gap}(k+1) - s_{k+1}$$

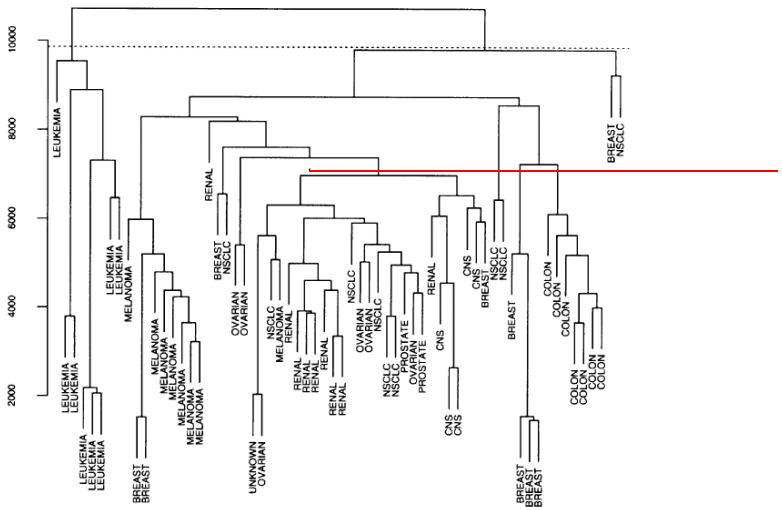
```
> library(cluster)
> x <- iris[, 1:4]
> gskmn <- clusGap(x, FUN = kmeans, nstart = 20, K.max = 8, B = 60)
Clustering k = 1,2,..., K.max (= 8): .. done
Bootstrapping, b = 1,2,..., B (= 60)  [one "." per sample]:
.................................................. 50
.......... 60
> plot(gskmn, main = "clusGap(., FUN = kmeans, n.start=20, B= 60)")
```



gap statistics

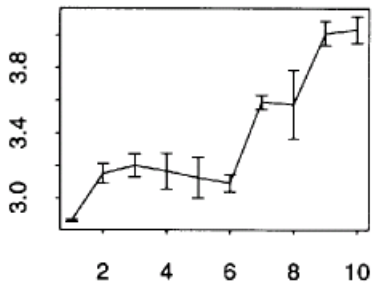






application to hierarchical clustering and DNA microarray data

6834×64 matrix



http://www-genome.stanford.edu/nci60

See Also: **`NbClust`: Determining the Best Number of Clusters in a Data Set**

It provides 30 indexes for determining the optimal number of clusters in a data set and offers the best clustering scheme from different results to the user.

Cluster Validation

Assess the quality and reliability of the cluster sets.

- **Quality:**
  clusters can be measured in terms of homogeneity and separation.

- **Reliability:**
  cluster structure is not formed by chance.

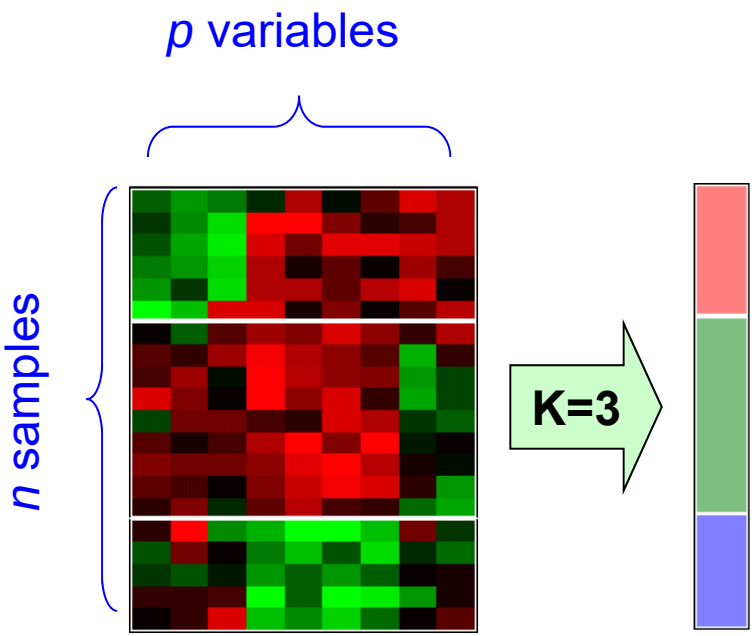
- **Ground Truth**:
  from domain knowledge.

NOTE:

Help to decide the number of clusters in the data.

Validation Indices for Hard Clustering

- K-means, SOM, Hierarchical Clustering,...



Class Labels

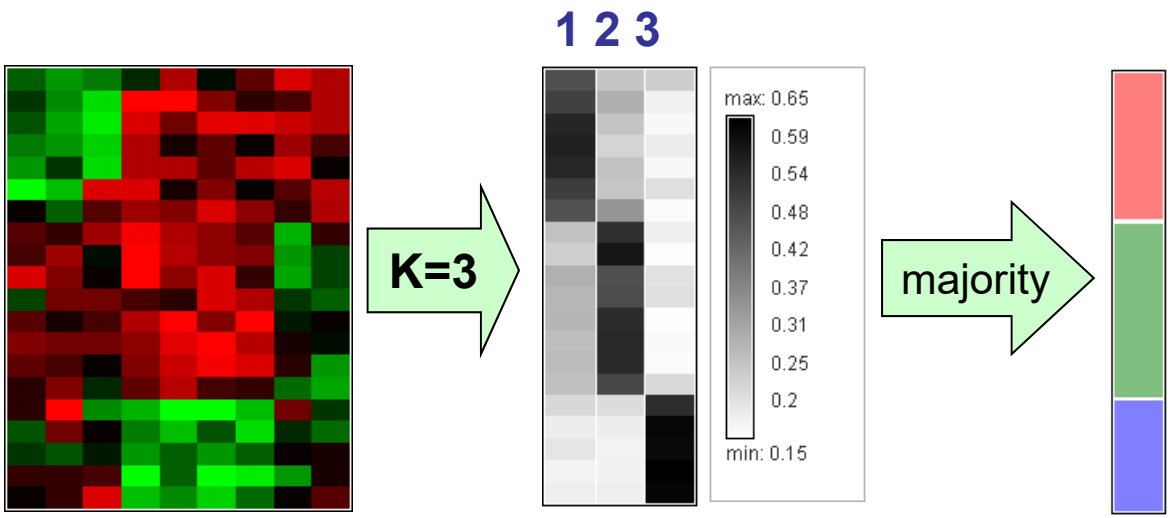
Cluster Validation Indices

Data-driven ✓

Biological ✓

Validation Indices for Soft Clustering

- Fuzzy c-means, Model-based clustering, …



Cluster Validation Indices

Data-driven ✓ Biological ✓

Class Membership **Class Labels**

Cluster Validation Indices

Data-driven ✓ Biological ?

Wu, H. M.* (2011), On Biological Validity Indices for Soft Clustering Algorithms for Gene Expression Data, Computational Statistics and Data Analysis, 55(5), 1969-1979.

Literatures on Cluster Validation (1/2)

2004

- Daxin Jiang, Chun Tang and Aidong Zhang, (2004), Cluster analysis for gene expression data: a survey, IEEE Transactions on Knowledge and Data Engineering 16(11), 1370- 1386. [web]
- Kimberly D. Siegmund, Peter W. Laird and Ite A. Laird-Offringa, (2004), A comparison of cluster analysis methods using DNA methylation data, Bioinformatics 20(12), 1896-1904.
- Tilman Lange, Volker Roth, Mikio L. Braun, and Joachim M. Buhmann, Stability-Based Validation of Clustering Solutions, Neural Comp. 2004 16: 1299-1323.

2003

- Datta S, Datta S. Comparisons and validation of statistical clustering techniques for microarray gene expression data. Bioinformatics. 2003 Mar 1;19(4):459-66.
- N. Bolshakova and F. Azuaje, (2003), Cluster validation techniques for genome expression data, Signal Processing 83(4), 825-833.

2001

- K. Y. Yeung, D. R. Haynor and W. L. Ruzzo, (2001), Validating clustering for gene expression data, Bioinformatics 17(4), 309-318. [web]
- Maria Halkidi, Yannis Batistakis, Michalis Vazirgiannis,(2001), On Clustering Validation Techniques, Journal of Intelligent Information Systems, 17(2), 107 - 145.
- Kerr MK, Churchill GA. Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. Proc Natl Acad Sci U S A. 2001 Jul 31;98(16):8961-5.
- Levine E, Domany E. Resampling method for unsupervised estimation of cluster validity. Neural Comput. 2001 Nov;13(11):2573-93.
- Maria Halkidi, Michalis Vazirgiannis, Clustering Validity Assessment: Finding the Optimal Partitioning of a Data Set, icdm, p. 187, First IEEE International Conference on Data Mining (ICDM'01), 2001

~2000

- Zhang K, Zhao H. Assessing reliability of gene clusters from gene expression data. Funct Integr Genomics. 2000 Nov;1(3):156-73.
- Xie, X.L. Beni, G. (1991), A validity measure for fuzzy clustering, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 13(8), 841-847.
- Peter Rousseeuw, (1987), Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 20(1), 53-65.
- Lawrence Hubert and Phipps Arabie (1985), Comparing partitions, Journal of Classification 2(1), 193-218.
- Wallace, D. L. 1983. A method for comparing two hierarchical clusterings: comment. Journal of the American Statistical Association 78:569-576.
- E. B. Fowlkes; C. L. Mallows, (1983), A Method for Comparing Two Hierarchical Clusterings, Journal of the American Statistical Association, 78(383), 553-569.
- William M. Rand, (1971), Objective Criteria for the Evaluation of Clustering Methods, Journal of the American Statistical Association 66(336), 846-850.

Literatures on Cluster Validation (2/2)

2007

- Marcel Brun, Chao Sima, Jianping Hua, James Lowey, Brent Carroll, Edward Suh and Edward R. Dougherty, (2007), Model-based evaluation of clustering validation measures, Pattern Recognition 40(3), 807-824.
- Francisco R. Pinto, João A. Carriço, Mário Ramirez and Jonas S Almeida, (2007), Ranked Adjusted Rand: integrating distance and partition information in a measure of clustering agreement, BMC Bioinformatics, 8:44.

2006

- Susmita Datta and Somnath Datta, (2006), Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes, BMC Bioinformatics 2006, 7:397. [web]
- Anbupalam Thalamuthu, Indranil Mukhopadhyay, Xiaojing Zheng and George C. Tseng, (2006), Evaluation and comparison of gene clustering methods in microarray analysis, Bioinformatics 22(19), 2405-2412.
- Giorgio Valentini , (2006), Clusterv: a tool for assessing the reliability of clusters discovered in DNA microarray data, Bioinformatics, 22(3), 369-370.
- Susmita Datta and Somnath Datta, (2006), Evaluation of clustering algorithms for gene expression data, BMC Bioinformatics 2006, 7(Suppl 4):S17. [web]

2005

- Tibshirani, Robert; Walther, Guenther (2005), Cluster Validation by Prediction Strength, Journal of Computational & Graphical Statistics 14(3), pp. 511-528(18)
- Julia Handl, Joshua Knowles and Douglas B. Kell, (2005), Computational cluster validation in post-genomic data analysis, Bioinformatics 21(15), 3201-3212. [web] [supp]
- Nadia B,Francisco A,Padraig C. (2005), An integrated tool for microarray data clustering and cluster validity assessment, Bioinformatics 21:451. [Web]
- Julia Handl and Joshua Knowles. (2005) Exploiting the trade-off -- the benefits of multiple objectives in data clustering. Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Pages 547-560. LNCS 3410. Copyright Springer-Verlag. PDF.
- Nikhil R Garge, Grier P Page, Alan P Sprague , Bernard S Gorman and David B Allison, Reproducible Clusters from Microarray Research: Whither? BMC Bioinformatics 2005, 6(Suppl 2):S10. [web]

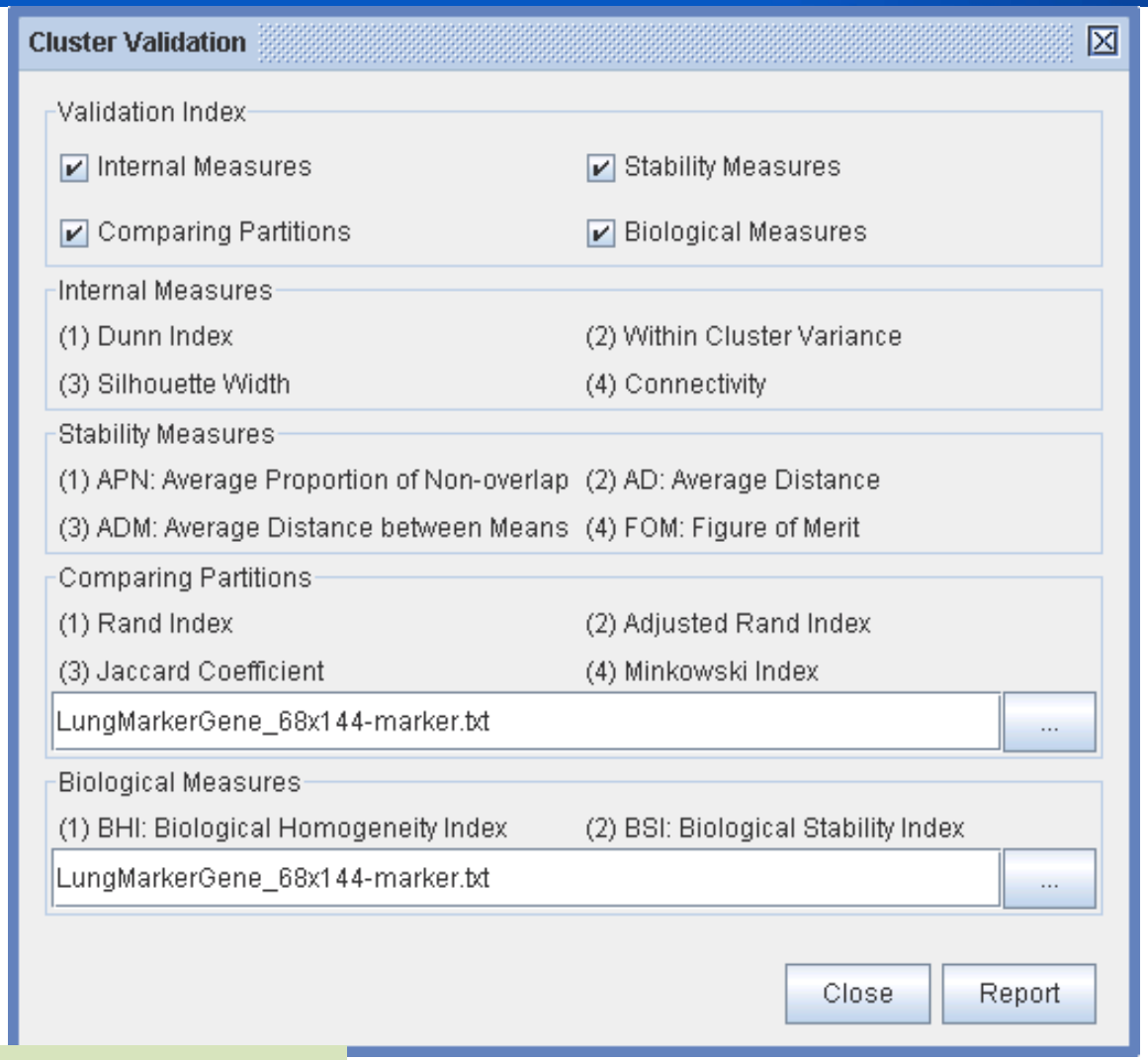
Cluster Validation Index

Internal Measures

Stability Measures

Comparing Partitions

Biological Measures



See: ***clValid***: an R package for cluster validation.

Statistical Evaluation: Internal Measures

Compactness | Homogeneity | Separation

Connectivity

$$Conn(\mathcal{C}) = \sum_{i=1}^{N} \sum_{j=1}^{L} d_{i,nn_{i(j)}}$$

$$d_{i,nn_{i(j)}} = \begin{cases} 0, & \text{for } i \text{ and } j \text{ are in the same cluster,} \\ 1/j, & \text{otherwise.} \end{cases}$$

Dunn index *(Dunn, 1974)*

$$D(\mathcal{C}) = \frac{\min\limits_{C_k, C_l \in \mathcal{C},\, C_k \neq C_l} \left(\min\limits_{i \in C_k,\, j \in C_l} \text{dist}(i,j) \right)}{\max\limits_{C_m \in \mathcal{C}} \text{diam}(C_m)}$$

Within-cluster Variance

$$V(\mathcal{C}) = \sqrt{\frac{1}{N} \sum_{C_k \in \mathcal{C}} \sum_{i \in C_k} \text{dist}(i, \mu_k)}$$

Silhouette Width *(Rousseeuw, 1987)*

$$S(\mathcal{C}) = \sum_{i=1}^{N} \frac{S(i)}{N}, \quad S(i) = \frac{b_i - a_i}{\max(b_i, a_i)}$$

the average distance between i and the observations in the closet other cluster

the average distance between i and all other observations in the same cluster.

Statistical Evaluation: Stability

- Average Proportion of Non-overlap (APN)
- Average Distance (AD)
- Average Distance between Means (ADM)
- Prediction Strength: Figure of Merit (FOM)

	A	B	C	D	E	F	G	H	I
1	-1.37	-2.30	-1.80	-0.55	2.45	-0.13	1.49	3.03	2.48
2	-0.68	-2.11	-3.42	4.67	4.57	1.75	0.61	0.92	2.52
3	-1.19	-2.49	-3.66	3.14	1.70	3.29	3.33	2.92	2.48
4	-1.93	-2.28	-3.16	2.51	0.32	1.49	0.21	2.20	1.03
5	-2.21	-0.79	-3.29	2.55	2.44	1.45	2.68	3.03	0.19
6	-4.14	-2.91	-1.64	3.21	0.37	1.93	0.14	1.27	2.67
7	0.21	-1.36	-0.44	2.22	1.85	3.11	2.03	0.67	2.40
8	1.13	0.79	2.25	3.65	2.52	2.09	1.13	-2.59	0.67
9	0.95	2.33	-0.07	3.89	2.72	2.13	1.75	-2.17	-0.90
10	3.04	1.85	0.21	7.07	2.01	3.05	0.76	-2.58	-1.04
11	-1.02	1.65	1.53	0.95	0.60	3.12	2.52	-0.77	-1.40
12	1.21	0.24	1.04	2.50	3.69	1.81	3.98	-0.33	0.11
13	1.74	1.60	1.70	2.02	3.45	4.46	2.69	0.41	-0.09
14	1.34	1.06	0.06	1.81	2.90	3.64	3.04	0.49	-2.33
15	0.57	1.81	-0.47	1.40	2.70	0.99	0.82	-1.61	-2.56
16	0.61	4.22	-2.03	-2.61	-4.00	-4.64	-2.92	1.55	-0.71
17	-1.13	1.64	0.01	-1.77	-2.85	-1.24	-3.41	-0.59	-1.64
18	-0.86	-1.17	-0.41	-2.20	-1.30	-2.37	-1.41	0.08	0.25
19	0.75	0.66	1.04	-4.26	-1.41	-3.99	-3.53	-2.17	0.34
20	0.15	0.68	3.18	-2.86	-2.01	-3.18	-1.58	0.10	1.28

Full data (nxp)

Compare two clusterings

Repeat: 1,…p

	A	B	C	D	E	F	G	H
1	-1.37	-2.30	-1.80	-0.55	2.45	-0.13	1.49	3.03
2	-0.68	-2.11	-3.42	4.67	4.57	1.75	0.61	0.92
3	-1.19	-2.49	-3.66	3.14	1.70	3.29	3.33	2.92
4	-1.93	-2.28	-3.16	2.51	0.32	1.49	0.21	2.20
5	-2.21	-0.79	-3.29	2.55	2.44	1.45	2.68	3.03
6	-4.14	-2.91	-1.64	3.21	0.37	1.93	0.14	1.27
7	0.21	-1.36	-0.44	2.22	1.85	3.11	2.03	0.67
8	1.13	0.79	2.25	3.65	2.52	2.09	1.13	-2.59
9	0.95	2.33	-0.07	3.89	2.72	2.13	1.75	-2.17
10	3.04	1.85	0.21	7.07	2.01	3.05	0.76	-2.58
11	-1.02	1.65	1.53	0.95	0.60	3.12	2.52	-0.77
12	1.21	0.24	1.04	2.50	3.69	1.81	3.98	-0.33
13	1.74	1.60	1.70	2.02	3.45	4.46	2.69	0.41
14	1.34	1.06	0.06	1.81	2.90	3.64	3.04	0.49
15	0.57	1.81	-0.47	1.40	2.70	0.99	0.82	-1.61
16	0.61	4.22	-2.03	-2.61	-4.00	-4.64	-2.92	1.55
17	-1.13	1.64	0.01	-1.77	-2.85	-1.24	-3.41	-0.59
18	-0.86	-1.17	-0.41	-2.20	-1.30	-2.37	-1.41	0.08
19	0.75	0.66	1.04	-4.26	-1.41	-3.99	-3.53	-2.17
20	0.15	0.68	3.18	-2.86	-2.01	-3.18	-1.58	0.10

left-out column ℓ

sample

Remaining data (nx(p-1))

Figure of Merit (FOM)

left-out column ℓ

	A	B	C	D	E	F	H	I
1	-1.37	-2.30	-1.80	-0.55	2.45	-0.13	3.03	2.48
2	-0.68	-2.11	-3.42	4.67	4.57	1.75	0.92	2.52
3	-1.19	-2.49	-3.66	3.14	1.70	3.29	2.92	2.48
4	-1.93	-2.28	-3.16	2.51	0.32	1.49	2.20	1.03
5	-2.21	-0.79	-3.29	2.55	2.44	1.45	3.03	0.19
6	-4.14	-2.91	-1.64	3.21	0.37	1.93	1.27	2.67
7	0.21	-1.36	-0.44	2.22	1.85	3.11	0.67	2.40
8	1.13	0.79	2.25	3.65	2.52	2.09	-2.59	0.67
9	0.95	2.33	-0.07	3.89	2.72	2.13	-2.17	-0.90
10	3.04	1.85	0.21	7.07	2.01	3.05	-2.58	-1.04
11	-1.02	1.65	1.53	0.95	0.60	3.12	-0.77	-1.40
12	1.21	0.24	1.04	2.50	3.69	1.81	-0.33	0.11
13	1.74	1.60	1.70	2.02	3.45	4.46	0.41	-0.09
14	1.34	1.06	0.06	1.81	2.90	3.64	0.49	-2.33
15	0.57	1.81	-0.47	1.40	2.70	0.99	-1.61	-2.56
16	0.61	4.22	-2.03	-2.61	-4.00	-4.64	1.55	-0.71
17	-1.13	1.64	0.01	-1.77	-2.85	-1.24	-0.59	-1.64
18	-0.86	-1.17	-0.41	-2.20	-1.30	-2.37	0.08	0.25
19	0.75	0.66	1.04	-4.26	-1.41	-3.99	-2.17	0.34
20	0.15	0.68	3.18	-2.86	-2.01	-3.18	0.10	1.28

Full data (NxM)

Clustering based on Remaining data

	A	B	C	D	E	F	G	H
1	-1.37	-2.30	-1.80	-0.55	2.45	-0.13	1.49	3.03
2	-0.68	-2.11	-3.42	4.67	4.57	1.75	0.61	0.92
3	-1.19	-2.49	-3.66	3.14	1.70	3.29	3.33	2.92
4	-1.93	-2.28	-3.16	2.51	0.32	1.49	0.21	2.20
5	-2.21	-0.79	-3.29	2.55	2.44	1.45	2.08	3.03
6	-4.14	-2.91	-1.64	3.21	0.37	1.93	0.14	1.27
7	0.21	-1.36	-0.44	2.22	1.85	3.11	2.03	0.67
8	1.13	0.79	2.25	3.65	2.52	2.09	1.13	-2.59
9	0.95	2.33	-0.07	3.89	2.72	2.13	1.75	-2.17
10	3.04	1.85	0.21	7.07	2.01	3.05	0.76	-2.58
11	-1.02	1.65	1.53	0.95	0.60	3.12	2.52	-0.77
12	1.21	0.24				1.81	3.98	-0.33
13	1.74	1.60				4.46	2.69	0.41
14	1.34	1.06				3.64	3.04	0.49
15	0.57	1.81				0.99	0.82	-1.61
16	0.61	4.22	-2.03	-2.61	-4.00	-4.64	-2.92	1.55
17	-1.13	1.64	0.01	-1.77	-2.85	-1.24	-3.41	-0.59
18	-0.86	-1.17	-0.41	-2.20	-1.30	-2.37	-1.41	0.08
19	0.75	0.66	1.04	-4.26	-1.41	-3.99	-3.53	-2.17
20	0.15	0.68	3.18	-2.86	-2.01	-3.18	-1.58	0.10

$C_k(\ell)$

	E
1	2.45
2	4.57
3	1.70
4	0.32
5	2.44
6	0.37
7	1.85
8	2.52
9	2.72
10	2.01
11	0.60
12	3.69
13	3.45
14	2.90
15	2.70
16	-4.00
17	-2.85
18	-1.30
19	-1.41
20	-2.01

$x_{i,\ell}$

$\mu_{C_k(\ell)}$

Remaining data (Nx(M-1))

$$\mathrm{dist}\left(x_{i,\ell}, \mu_{C_k(\ell)}\right)$$

$$FOM(\ell, \mathcal{C}) = \sqrt{\frac{1}{N}\sum_{k=1}^{K}\sum_{i \in C_k(\ell)} \mathrm{dist}\left(x_{i,\ell}, \mu_{C_k(\ell)}\right)} \times \sqrt{\frac{N}{N-K}}$$

$$FOM(\mathcal{C}) = \frac{1}{M}\sum_{\ell=1}^{M} FOM(\ell, \mathcal{C})$$

K. Y. Yeung, D. R. Haynor and W. L. Ruzzo, (2001), Validating clustering for gene expression data, Bioinformatics 17(4), 309-318.

Agreement with Reference Partition

- Rand index
- Jaccard coefficient
- Minkowski Measure
- Adjusted Rand index

$$\mathcal{U} = \{U_1, \cdots, U_K\} \qquad \mathcal{V} = \{V_1, \cdots, V_S\}$$

$$A = \begin{array}{c|cccc} & O_1 & O_1 & \cdots & O_n \\ \hline O_1 & & & & \\ O_2 & & & & \\ \cdots & & & & \\ O_n & & & & \end{array} \qquad B = \begin{array}{c|cccc} & O_1 & O_1 & \cdots & O_n \\ \hline O_1 & & & & \\ O_2 & & & & \\ \cdots & & & & \\ O_n & & & & \end{array}$$

$$A_{ij} = \mathrm{I}(O_i \in U_k, O_j \in U_k) \qquad B_{ij} = \mathrm{I}(O_i \in V_s, O_j \in V_s)$$

$$n_{11} = \#\{(O_i, O_j); \mathrm{I}(A_{ij} = 1, B_{ij} = 1)\}$$
$$n_{10} = \#\{(O_i, O_j); \mathrm{I}(A_{ij} = 1, B_{ij} = 0)\}$$
$$n_{01} = \#\{(O_i, O_j); \mathrm{I}(A_{ij} = 0, B_{ij} = 1)\}$$
$$n_{00} = \#\{(O_i, O_j); \mathrm{I}(A_{ij} = 0, B_{ij} = 0)\}$$

- Rand index, [0, 1], maximum:

$$R(\mathcal{U}, \mathcal{V}) = \frac{n_{11} + n_{00}}{n_{00} + n_{01} + n_{10} + n_{11}}$$

- Jaccard coefficient, [0, 1], maximum:

$$J(\mathcal{U}, \mathcal{V}) = \frac{n_{11}}{n_{11} + n_{10} + n_{01}}$$

- Minikowski measure:

$$M(\mathcal{U}, \mathcal{V}) = \sqrt{\frac{n_{10} + n_{01}}{n_{11} + n_{01}}}$$

Adjusted Rand index

May be the most widely used Cluster Validation Index!

$A =$

Cluster	V_1	V_1	$\cdots$	V_C	Sums
U_1	n_{11}	n_{12}		n_{1C}	$n_{1\cdot}$
U_2	n_{21}	n_{22}		n_{2C}	$n_{2\cdot}$
$\cdots$			$\cdots$		$\cdots$
U_R	n_{R1}	n_{R2}		n_{RC}	$n_{R\cdot}$
Sums	$n_{\cdot 1}$	$n_{\cdot 1}$	$\cdots$	$n_{\cdot C}$	n

$$R(\mathcal{U},\mathcal{V}) = 1 + \frac{\sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}^2 - \frac{1}{2}\left(\sum_{i=1}^{R} n_{i\cdot}^2 + \sum_{j=1}^{C} n_{\cdot j}^2\right)}{\binom{n}{2}}$$

$$R(\mathcal{U},\mathcal{V})_{adj} = \frac{\sum_{i=1}^{R}\sum_{j=1}^{C}\binom{n_{ij}}{2} - \sum_{i=1}^{R}\sum_{j=1}^{C}\binom{n_{i\cdot}}{2}\binom{n_{\cdot j}}{2}\Big/\binom{n}{2}}{\frac{1}{2}\left[\sum_{i=1}^{R}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{C}\binom{n_{\cdot j}}{2}\right] - \sum_{i=1}^{R}\sum_{j=1}^{C}\binom{n_{i\cdot}}{2}\binom{n_{\cdot j}}{2}\Big/\binom{n}{2}}$$

Lawrence Hubert and Phipps Arabie (1985), Comparing partitions, Journal of Classification 2(1), 193-218.

clValid {clValid}: Validate Cluster Results

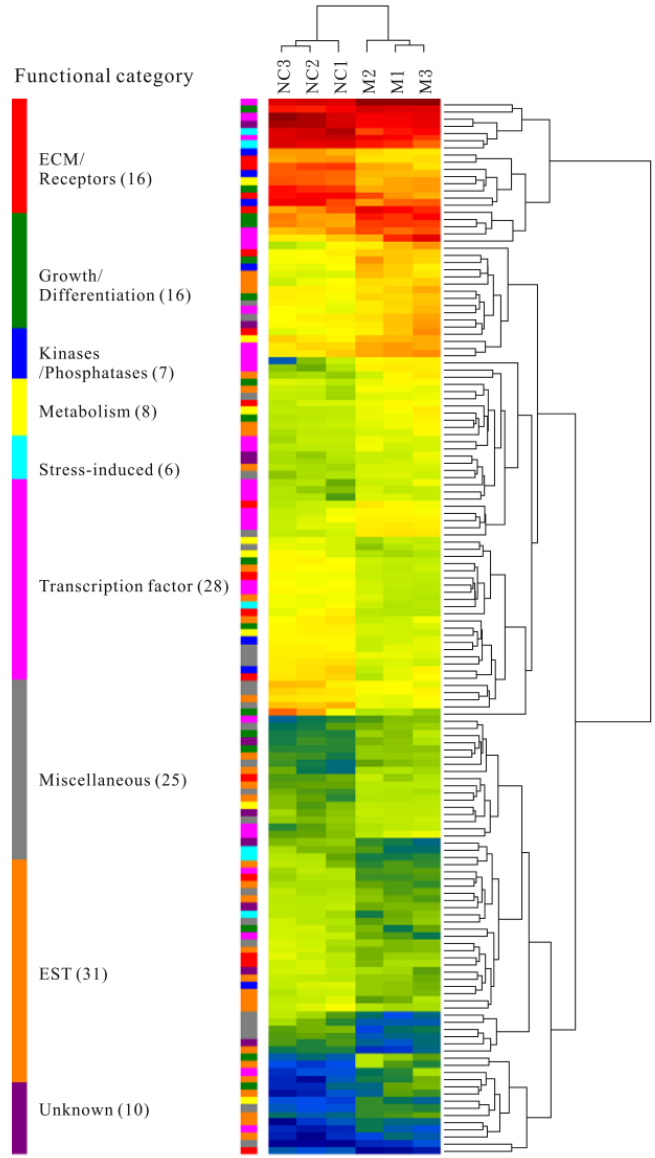
```
clValid(obj, nClust, clMethods = "hierarchical",
  validation = "stability", maxitems = 600,
  metric = "euclidean", method = "average",
  neighbSize = 10, annotation = NULL,
  GOcategory = "all",
  goTermFreq=0.05, dropEvidence=NULL,
  verbose=FALSE, ...)
```

mouse {clValid} R Documentation

Mouse Mesenchymal Cells

Description

Data from an Affymetrix microarray experiment (moe430a) comparing comparing gene expression of mesenchymal cells from two distinct lineages, **neural crest** (NC) and **mesoderm (M)** derived. The dataset consists of 147 genes and ESTs which were determined to be significantly differentially expressed between the two cell lineages, with at least a 1.5 fold increase or decrease in expression. There are three samples for each of the neural crest and mesoderm derived cells.



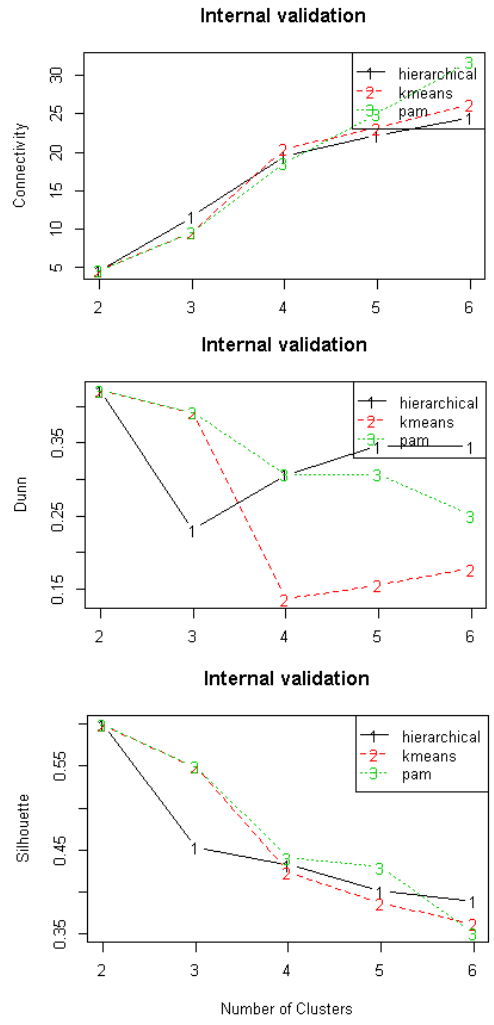
Example: Mouse Data

```
clMethods =  c("hierarchical", "kmeans", "diana", "fanny", "som", "model",
"sota", "pam", "clara", "agnes")
validation  = c("internal", "stability", "biological")
```

```
> library(clValid)
> data(mouse)
> head(mouse)
           ID       M1       M2       M3      NC1      NC2      NC3                    FC
1   1448995_at 4.706812 4.528291 4.325836 5.568435 6.915079 7.353144 Growth/Differentiation
2 1436392_s_at 3.867962 4.052354 3.474651 4.995836 5.056199 5.183585   Transcription factor
3 1437434_a_at 2.875112 3.379619 3.239800 3.877053 4.459629 4.850978          Miscellaneous
4   1428922_at 5.326943 5.498930 5.629814 6.795194 6.535522 6.622577          Miscellaneous
5 1452671_s_at 5.370125 4.546810 5.704810 6.407555 6.310487 6.195847          ECM/Receptors
6   1448147_at 3.471347 4.129992 3.964431 4.474737 5.185631 5.177967 Growth/Differentiation

> # internal validation
> express <- mouse[1:25,c("M1", "M2", "M3", "NC1", "NC2",  "NC3")]
> rownames(express) <- mouse$ID[1:25]
> head(express)
                   M1       M2       M3      NC1      NC2      NC3
1448995_at   4.706812 4.528291 4.325836 5.568435 6.915079 7.353144
1436392_s_at 3.867962 4.052354 3.474651 4.995836 5.056199 5.183585
1437434_a_at 2.875112 3.379619 3.239800 3.877053 4.459629 4.850978
1428922_at   5.326943 5.498930 5.629814 6.795194 6.535522 6.622577
1452671_s_at 5.370125 4.546810 5.704810 6.407555 6.310487 6.195847
1448147_at   3.471347 4.129992 3.964431 4.474737 5.185631 5.177967
>
> intern <- clValid(express, 2:6, clMethods=c("hierarchical", "kmeans", "pam"),
+                   validation="internal")
```

Mouse Data: Internal Measure

```
> summary(intern) # view results

Clustering Methods:
 hierarchical kmeans pam

Cluster sizes:
 2 3 4 5 6

Validation Measures:
                                 2       3       4       5       6

hierarchical Connectivity   4.6159 11.5865 19.5075 22.2075 24.5044
             Dunn           0.4217  0.2315  0.3068  0.3456  0.3456
             Silhouette     0.5997  0.4529  0.4324  0.4007  0.3891
kmeans       Connectivity   4.6159  9.5607 20.4774 23.1774 26.2242
             Dunn           0.4217  0.3924  0.1360  0.1556  0.1778
             Silhouette     0.5997  0.5495  0.4235  0.3871  0.3618
pam          Connectivity   4.6159  9.5607 18.5925 25.0631 31.8381
             Dunn           0.4217  0.3924  0.3068  0.3068  0.2511
             Silhouette     0.5997  0.5495  0.4401  0.4297  0.3506

Optimal Scores:

             Score  Method       Clusters
Connectivity 4.6159 hierarchical 2
Dunn         0.4217 hierarchical 2
Silhouette   0.5997 hierarchical 2
> par(mfrow=c(1, 3))
> plot(intern)
```

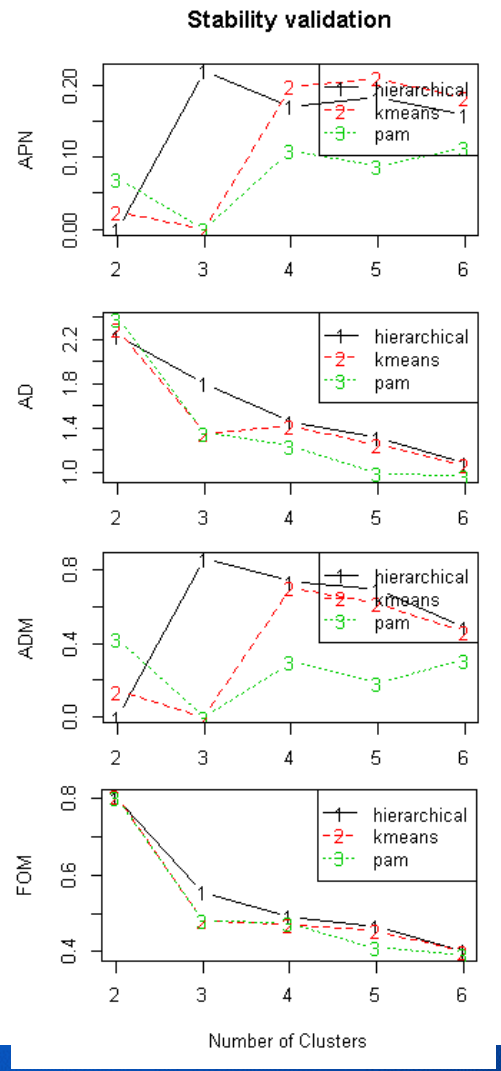


Mouse Data: Stability Measure

```
> stab <- clValid(express, 2:6, clMethods=c("hierarchical","kmeans","pam"),
+                 validation="stability")
```

```
> summary(stab)
Clustering Methods:
 hierarchical kmeans pam
Cluster sizes:
 2 3 4 5 6
Validation Measures:
                      2      3      4      5      6
hierarchical APN  0.0000 0.2200 0.1702 0.1836 0.1600
             AD   2.2328 1.8011 1.4589 1.3174 1.0895
             ADM  0.0000 0.8618 0.7372 0.6943 0.4905
             FOM  0.8020 0.5545 0.4917 0.4649 0.4021
kmeans       APN  0.0241 0.0000 0.1983 0.2100 0.1828
             AD   2.2861 1.3561 1.4182 1.2525 1.0671
             ADM  0.1399 0.0000 0.7050 0.6193 0.4655
             FOM  0.8055 0.4809 0.4689 0.4539 0.4007
pam          APN  0.0698 0.0000 0.1093 0.0867 0.1129
             AD   2.3867 1.3561 1.2413 0.9942 0.9642
             ADM  0.4236 0.0000 0.3008 0.1845 0.3098
             FOM  0.8022 0.4809 0.4731 0.4094 0.3926

Optimal Scores:
    Score  Method       Clusters
APN 0.0000 hierarchical 2
AD  0.9642 pam          6
ADM 0.0000 hierarchical 2
FOM 0.3926 pam          6
```

```
> par(mfrow=c(4, 1))
> plot(stab)
```



Biological Evaluation

- Biological Homogeneity Index (BHI)
- Biological Stability Index (BSI)

ProbeSet	Clustering	GO-BP Category
38389_at	1	0
1662_r_at	1	0
32607_at	1	0
1582_at	1	0
34699_at	1	0
37890_at	2	0
36008_at	2	1 2 3
36591_at	2	1 2 3 8 10
32081_at	2	1 2 3 4 5 6 7 9 10
668_s_at	2	1 2 3
41535_at	2	1 2 3 4
37666_at	2	1 2 3
40310_at	2	1 2 3 4 5 8 9
34256_at	3	1 2 3
38790_at	3	1
39175_at	3	1 2 3
35819_at	3	1 8
37639_at	3	1 2 3
31508_at	3	1 9
31505_at	4	1 2 3
1882_g_at	4	1 2 3 4 6
33154_at	4	1 2 3
837_s_at	4	1 2 3
35194_at	4	1
38422_s_at	4	1 2 3 4 5
33131_at	4	1 2 3 4 6 7

Example:
GO (Gene Ontology)
Multiple Functional Categories

Susmita Datta and Somnath Datta, (2006), Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes, BMC Bioinformatics 7:397.

Biological Evaluation

Susmita Datta and Somnath Datta, (2006), Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes, *BMC Bioinformatics* 7:397.

Biological Homogeneity Index (**BHI**)

Biological Stability Index (**BSI**)

ProbeSet	hard clustering	GO-BP Category
38389_at	1	0
1662_r_at	1	0
32607_at	1	0
37890_at	2	0
36008_at	2	1 2 3
36591_at	2	1 2 3 8 10
32081_at	2	1 2 3 4 5 6 7 9 10
34256_at	3	1 2 3
38790_at	3	1
39175_at	3	1 2 3
35819_at	3	1 8
37639_at	3	1 2 3
⋮		

Multiple Functional Categories

Generalized indices:

SBHI, SBSI

ProbeSet	soft clustering 1	2	3	GO-BP Category
38389_at	0.0543	0.1044	0.8413	0
1662_r_at	0.0175	0.0356	0.9469	0
32607_at	0.0011	0.0023	0.9967	0
37890_at	0.2707	0.7053	0.0240	0
36008_at	0.0834	0.9027	0.0139	1 2 3
36591_at	0.1012	0.8759	0.0229	1 2 3 8 10
32081_at	0.2113	0.7547	0.0339	1 2 3 4 5 6 7 9 10
34256_at	0.9087	0.0830	0.0084	1 2 3
38790_at	0.8908	0.0947	0.0144	1
39175_at	0.8728	0.1074	0.0198	1 2 3
35819_at	0.7601	0.1890	0.0509	1 8
37639_at	0.9062	0.0848	0.0090	1 2 3
⋮				

Biological Homogeneity Index (BHI)

a set of biological classes $\mathcal{B} = \{B_1, \ldots, B_F\}$

a statistical hard clustering partition $\mathcal{H} = \{H_1, \ldots, H_K\}$

B^i: the functional class containing gene i

B^j: the function class containing gene j

$$\mathrm{BHI}(\mathcal{H}, \mathcal{B}) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n_k(n_k - 1)} \sum_{i \neq j; i,j \in H_k} \mathrm{I}\left(B^i = B^j\right)$$

$$n_k = n(H_k \cap \mathcal{B})$$

Properties of BHI

- BHI: the average proportion of gene pairs belonging to the same functional classes that are clustered together.
- BHI: ranges from 0 to 1.
- BHI: larger values corresponding to more biologically homogeneous clusters.

Soft BHI (SBHI)

$\mathcal{U} = u_{ik}$: fuzzy membership matrix

soft clustering: $u_{ik} = P(y_i \in C_k)$

hard clustering : $u_{ik} = \mathrm{I}(y_i \in C_k)$

a statistical soft clustering partition $\mathcal{S} = \{S_1, \ldots, S_K\}$

$$\boxed{\mathrm{SBHI}(\mathcal{S}, \mathcal{B}) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{\eta_k} \sum_{i \neq j} u_{ik}\ u_{jk}\ \mathrm{I}(B^i = B^j)}$$

$$\eta_k = \sum_{i \neq j} u_{ik} u_{jk} \mathrm{I}(\text{genes } i, j \text{ are annotated})$$

Evaluation of Stability

	A	B	C	D	E	F	G	H	I
1	-1.37	-2.30	-1.80	-0.55	2.45	-0.13	1.49	3.03	2.48
2	-0.68	-2.11	-3.42	4.67	4.57	1.75	0.61	0.92	2.52
3	-1.19	-2.49	-3.66	3.14	1.70	3.29	3.33	2.92	2.48
4	-1.93	-2.28	-3.16	2.51	0.32	1.49	0.21	2.20	1.03
5	-2.21	-0.79	-3.29	2.55	2.44	1.45	2.68	3.03	0.19
6	-4.14	-2.91	-1.64	3.21	0.37	1.93	0.14	1.27	2.67
7	0.21	-1.36	-0.44	2.22	1.85	3.11	2.03	0.67	2.40
8	1.13	0.79	2.25	3.65	2.52	2.09	1.13	-2.59	0.67
9	0.95	2.33	-0.07	3.89	2.72	2.13	1.75	-2.17	-0.90
10	3.04	1.85	0.21	7.07	2.01	3.05	0.76	-2.58	-1.04
11	-1.02	1.65	1.53	0.95	0.60	3.12	2.52	-0.77	-1.40
12	1.21	0.24	1.04	2.50	3.69	1.81	3.98	-0.33	0.11
13	1.74	1.60	1.70	2.02	3.45	4.46	2.69	0.41	-0.09
14	1.34	1.06	0.06	1.81	2.90	3.64	3.04	0.49	-2.33
15	0.57	1.81	-0.47	1.40	2.70	0.99	0.82	-1.61	-2.56
16	0.61	4.22	-2.03	-2.61	-4.00	-4.64	-2.92	1.55	-0.71
17	-1.13	1.64	0.01	-1.77	-2.85	-1.24	-3.41	-0.59	-1.64
18	-0.86	-1.17	-0.41	-2.20	-1.30	-2.37	-1.41	0.08	0.25
19	0.75	0.66	1.04	-4.26	-1.41	-3.99	-3.53	-2.17	0.34
20	0.15	0.68	3.18	-2.86	-2.01	-3.18	-1.58	0.10	1.28

Full data (nxp)

Compare these two partitions

Repeat: 1,…p

	A	B	C	D	E	F	G	H
1	-1.37	-2.30	-1.80	-0.55	2.45	-0.13	1.49	3.03
2	-0.68	-2.11	-3.42	4.67	4.57	1.75	0.61	0.92
3	-1.19	-2.49	-3.66	3.14	1.70	3.29	3.33	2.92
4	-1.93	-2.28	-3.16	2.51	0.32	1.49	0.21	2.20
5	-2.21	-0.79	-3.29	2.55	2.44	1.45	2.68	3.03
6	-4.14	-2.91	-1.64	3.21	0.37	1.93	0.14	1.27
7	0.21	-1.36	-0.44	2.22	1.85	3.11	2.03	0.67
8	1.13	0.79	2.25	3.65	2.52	2.09	1.13	-2.59
9	0.95	2.33	-0.07	3.89	2.72	2.13	1.75	-2.17
10	3.04	1.85	0.21	7.07	2.01	3.05	0.76	-2.58
11	-1.02	1.65	1.53	0.95	0.60	3.12	2.52	-0.77
12	1.21	0.24	1.04	2.50	3.69	1.81	3.98	-0.33
13	1.74	1.60	1.70	2.02	3.45	4.46	2.69	0.41
14	1.34	1.06	0.06	1.81	2.90	3.64	3.04	0.49
15	0.57	1.81	-0.47	1.40	2.70	0.99	0.82	-1.61
16	0.61	4.22	-2.03	-2.61	-4.00	-4.64	-2.92	1.55
17	-1.13	1.64	0.01	-1.77	-2.85	-1.24	-3.41	-0.59
18	-0.86	-1.17	-0.41	-2.20	-1.30	-2.37	-1.41	0.08
19	0.75	0.66	1.04	-4.26	-1.41	-3.99	-3.53	-2.17
20	0.15	0.68	3.18	-2.86	-2.01	-3.18	-1.58	0.10

left-out column ℓ

sample

Remaining data (nx(p-1))

Summering these comparisons

Biological Stability Index (BSI)

$C^{i,0}$: the statistical cluster containing observation i based on all the data.

$C^{j,\ell}$: the statistical cluster containing observation j when column ℓ is removed.

$$\mathrm{BSI}(\mathcal{H},\mathcal{B}) = \frac{1}{F}\sum_{f=1}^{F}\frac{1}{n(B_f)(n(B_f)-1)}\frac{1}{M}\sum_{\ell=1}^{M}\sum_{i\neq j; i,j\in B_f}\frac{n(C^{i,0}\cap C^{j,\ell})}{n(C^{i,0})}$$

$n(B_f)$ number of genes in the biological category B_f

Properties of BSI

- **BSI:** the average overlap of the statistical clusters obtained based on all of the samples with that of those obtained based on leave-one-out samples for genes that were placed in the same functional class.
- **BSI:** from 0 to 1.
- **BSI:** larger values corresponding to more stable clusters of functionally annotated genes.

Soft BSI (SBSI)

$$\mathrm{SBSI}(\mathcal{S},\mathcal{B}) = \frac{1}{F}\sum_{f=1}^{F}\frac{1}{n(B_f)(n(B_f)-1)}\frac{1}{M}\sum_{\ell=1}^{M}\sum_{i\neq j; i,j\in B_f}\frac{\eta(C^{i,0}\cap C^{j,\ell})}{\eta(C^{i,0})}$$

$$\eta(C^{i,0}) = \sum_{k=1}^{K}\sum_{t=1}^{n} u_{ik}^{(0)} u_{tk}^{(0)}$$

$$\eta(C^{i,0}\cap C^{j,\ell}) = \sum_{k=1}^{K}\sum_{h=1}^{K}\sum_{t=1}^{n} u_{ik}^{(0)} u_{tk}^{(0)} u_{jh}^{(\ell)} u_{th}^{(\ell)}$$

$u_{ik}^{(0)}$ is the membership for gene i to cluster k based on all of the data

SBHI & SBSI

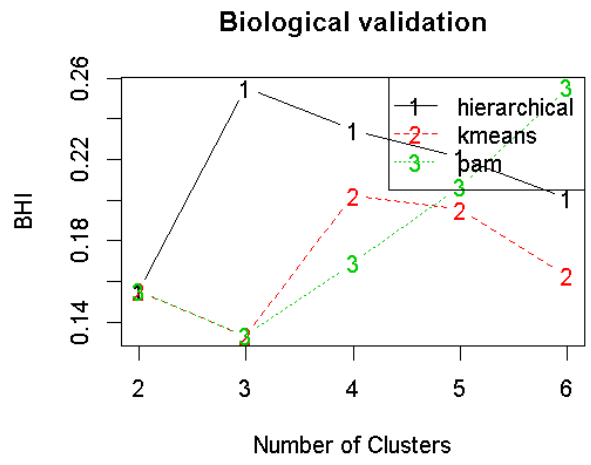
- **SBHI & SBSI** can be reduced to the BHI and the BSI.
- **SBHI & SBSI** are close to BHI & BSI when the clustering of a data set is close to being crisp.
- **SBHI & SBSI** consider multiple statistical cluster information for each gene and are thus more *reasonable*.

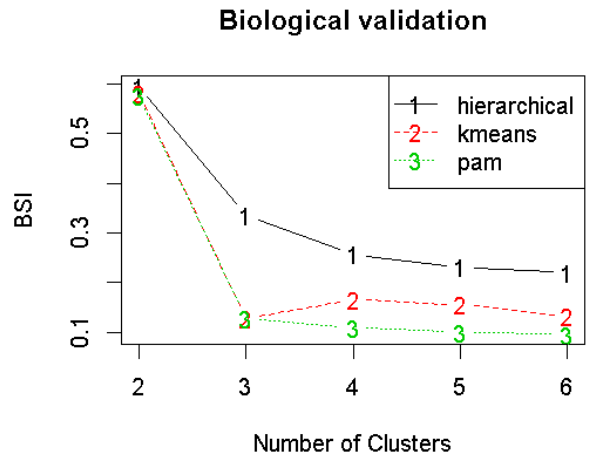
Mouse Data: Biological Measure

```
> # biological measures (functional classes predetermined)
> fc <- tapply(rownames(express), mouse$FC[1:25], c)
> fc
$`ECM/Receptors`
[1] "1452671_s_at" "1423110_at"   "1439381_x_at" "1450857_a_at"

$EST
[1] "1435327_at"   "1418382_at"   "1439373_x_at"

$`Growth/Differentiation`
[1] "1448995_at" "1448147_at" "1421180_at" "1416855_at"

$`Kinases/Phosphatases`
[1] "1439148_a_at" "1424474_a_at"

$Metabolism
[1] "1449059_a_at"

$Miscellaneous
[1] "1437434_a_at" "1428922_at"   "1449623_at"   "1418486_at"   "1415993_at"
"1452003_at"

$`Stress-induced`
[1] "1422557_s_at" "1456434_x_at" "1416481_s_at"

$`Transcription factor`
[1] "1436392_s_at" "1424186_at"

$Unknown
NULL
```

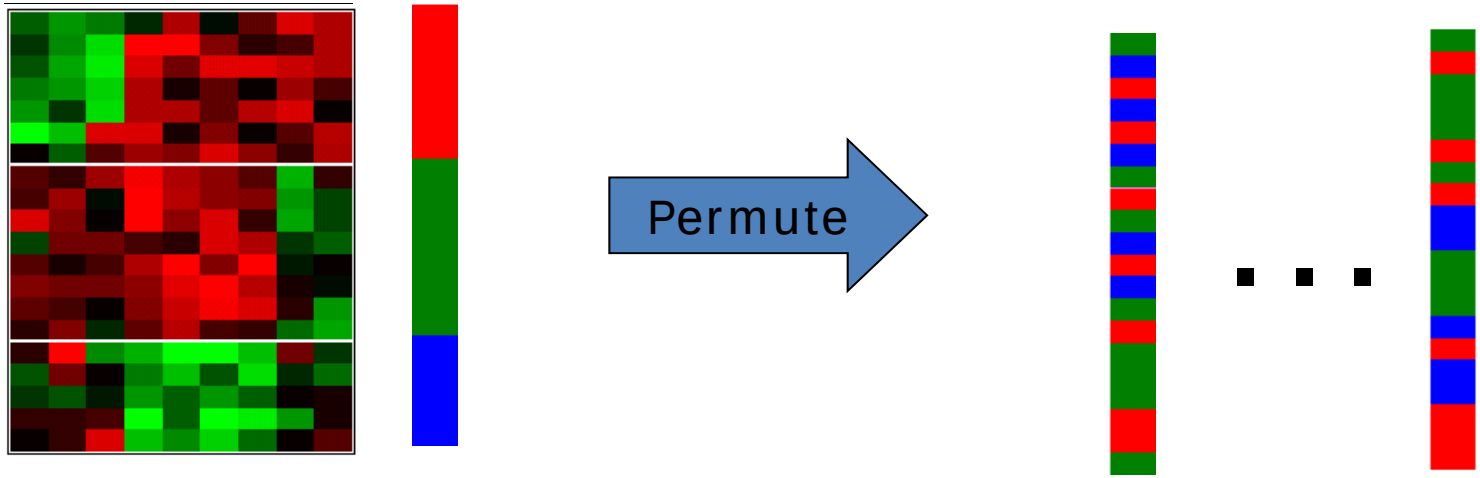
Mouse Data: Biological Measure

```
> fc <- fc[-match( c("EST","Unknown"), names(fc))]
> bio <- clValid(express, 2:6, clMethods=c("hierarchical", "kmeans", "pam"),
+                validation="biological", annotation=fc)
> summary(bio)

Clustering Methods:
 hierarchical kmeans pam

Cluster sizes:
 2 3 4 5 6

Validation Measures:
                       2      3      4      5      6

hierarchical BHI  0.1552 0.2552 0.2345 0.2210 0.2008
             BSI  0.5952 0.3361 0.2574 0.2314 0.2202
kmeans       BHI  0.1552 0.1335 0.2024 0.1952 0.1627
             BSI  0.5818 0.1286 0.1658 0.1557 0.1331
pam          BHI  0.1552 0.1335 0.1690 0.2067 0.2556
             BSI  0.5751 0.1286 0.1094 0.1003 0.0956

Optimal Scores:

    Score  Method       Clusters
BHI 0.2556 pam          6
BSI 0.5952 hierarchical 2

> par(mfrow=c(2, 1))
> plot(bio)
```





Significance of the Indices: Permutation Test

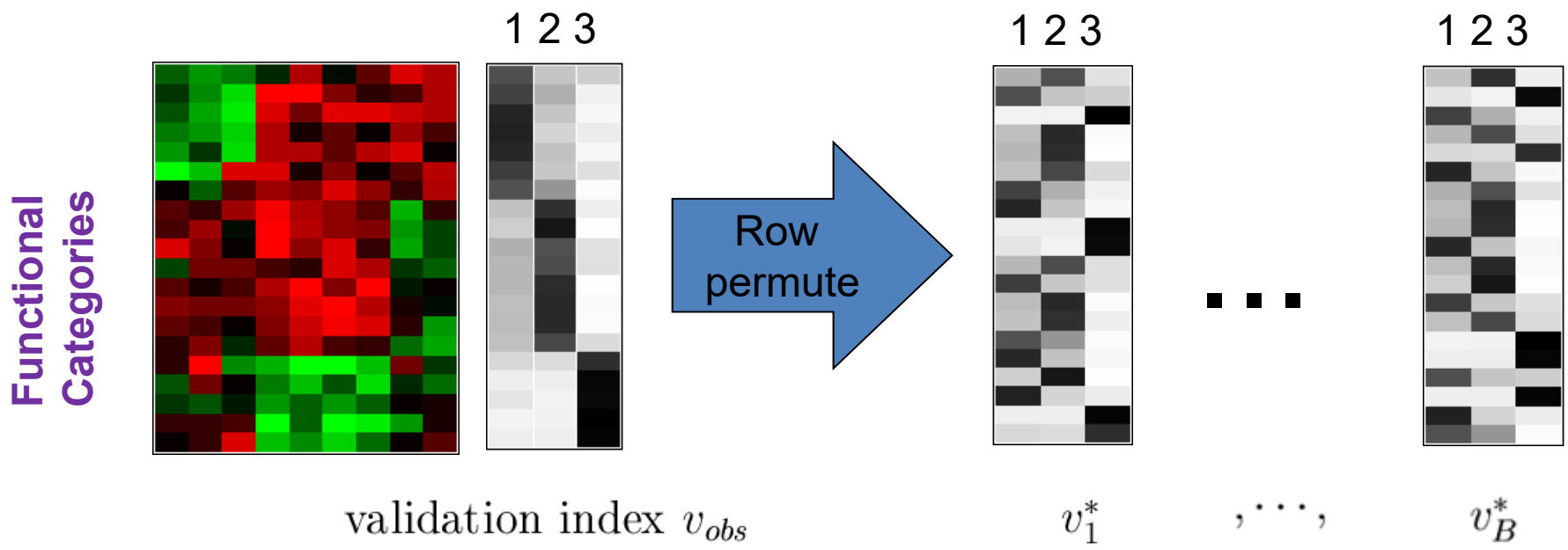


validation index v_{obs}

$v_1^*, \cdots, v_B^*$

$$p = \frac{1}{B} \sum_{i=1}^{B} I(v_i^* \geq v_{obs})$$

Permutation Test for Significance of the Indices

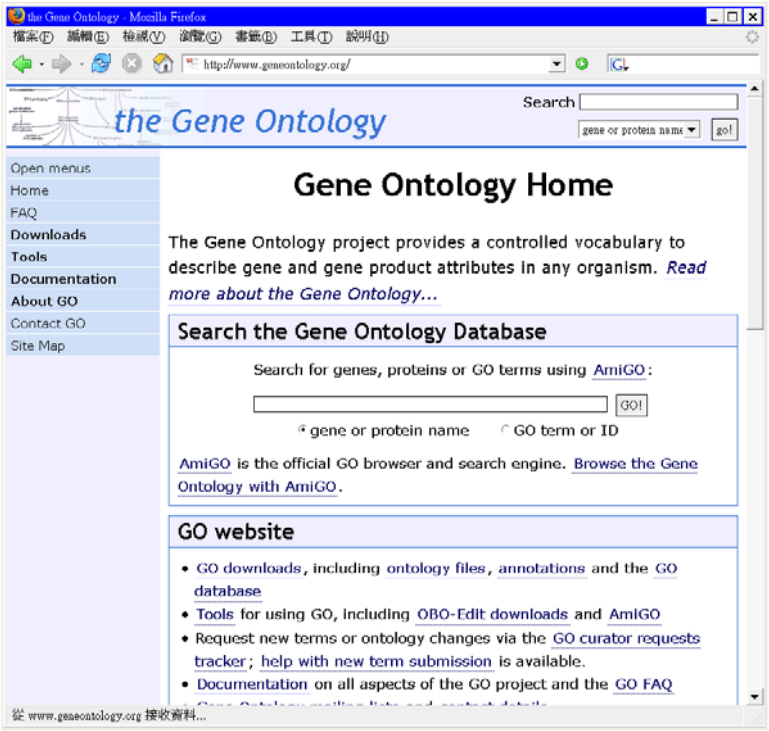


$$p = \frac{1}{B}\sum_{i=1}^{B} I(v_i^* \geq v_{obs})$$

Obtain Functional Categories (Annotation)

MIPS: the Munich Information Center for Protein Sequences

- http://mips.gsf.de/
- MIPS: a database for protein sequences and complete genomes, Nucleic Acids Research, 27:44-48, 1999



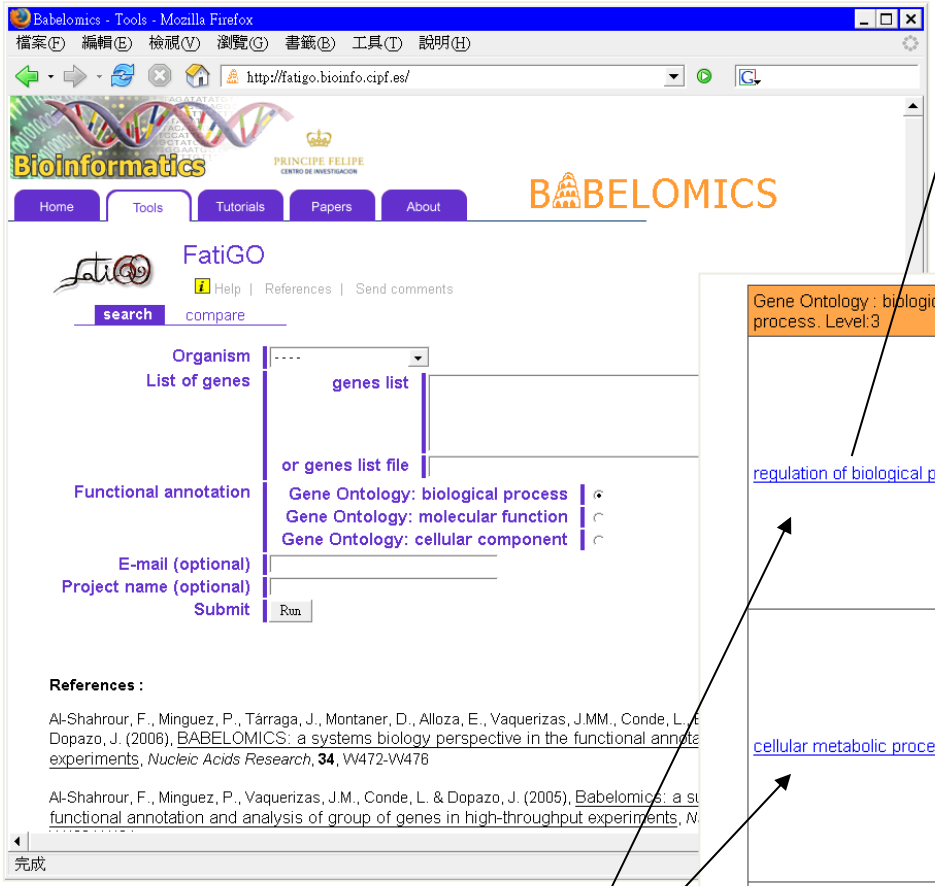
GO: Gene Ontology

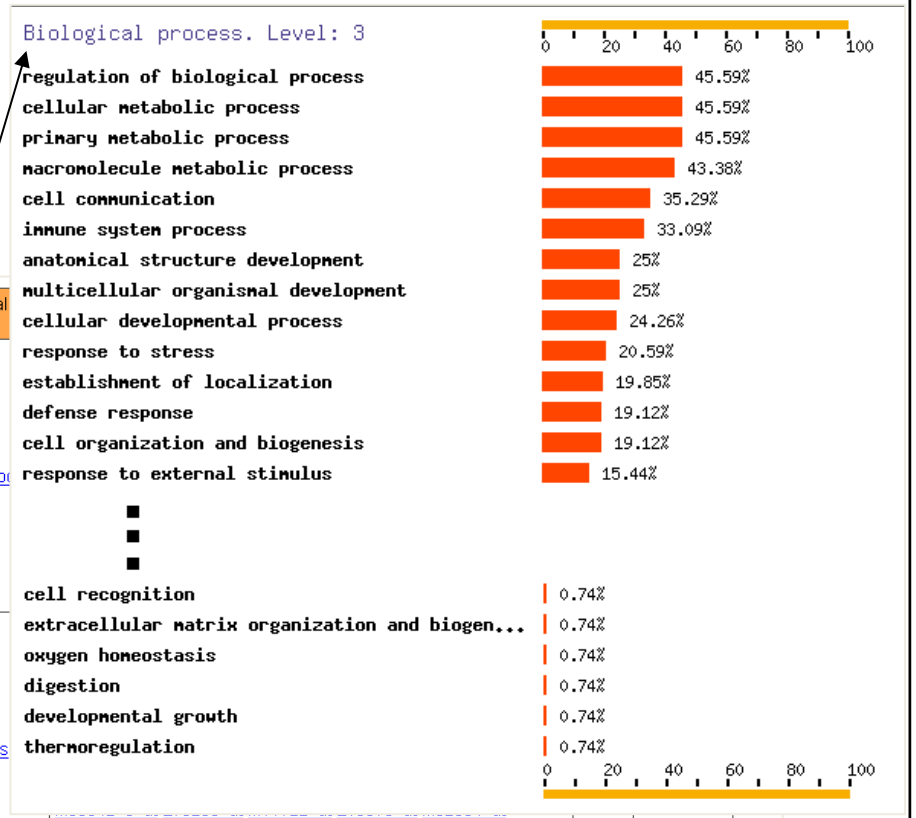
- A GO annotation is a Gene Ontology term associated with a gene product.
- http://www.geneontology.org/
- The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. Nature Genet. (2000) 25: 25-29.

- FatiGO (Al-Shahrour et al., 2004)
- FunCat (Ruepp et al., 2004)

FatiGO

http://babelomics.bioinfo.cipf.es/index.html





The ontologies are used to categorize gene products.

- Biological process ontology
- Molecular function ontology
- Cellular component ontology