Exploratory data analysis for microarrays

Jörg Rahnenführer

Computational Biology and Applied Algorithmics
Max Planck Institute for Informatics
D-66123 Saarbrücken
Germany

MAX-PLANCK-GESELLSCHAFT

NGFN - Courses in Practical DNA Microarray Analysis

Heidelberg, March 15, 2005

Overview

- **Exploratory data analysis:** Unsupervised learning

- **Example:** Time series

- **Distance measures:** Object (dis-)similarities

- **Cluster algorithms:** Grouping of data

- **Clustering microarray data:** Comparisons and hints

- **Other exploratory methods** for microarray data

Interactive exploratory data analysis

YOU ARE ALL GENES…

Classification tasks for microarrays

- ## <u>Classification of SAMPLES</u>

    Generate gene expression profiles that can

    (i) discriminate between different **known** cell types or conditions, e.g. between tumor and normal tissue,

    (ii) identify different and previously **unknown** cell types or conditions, e.g. new subclasses of an existing class of tumors.

- ## <u>Classification of GENES</u>

    (i) Assign an unknown cDNA sequence to one of a set of **known** gene classes.

    (ii) Partition a set of genes into new (**unknown**) functional classes on the basis of their expression patterns across a number of samples.

Classification

Seminal microarray analysis paper (Golub et al.)

Molecular classification of cancer: **class discovery** and **class prediction** by gene expression monitoring, *Science* 1999, 86:531-537, see *http://www.genome.wi.mit.edu/MPR*

Cancer classification	Class discovery	Class prediction
Machine learning	Unsupervised learning	Supervised learning
Statistics	Cluster analysis	Discriminant analysis

Classification

MESSAGE 1

Discriminant analysis: CLASSES KNOWN

Cluster analysis: CLASSES NOT KNOWN

Classification

- **Difference between discriminant analysis (supervised learning) and cluster analysis (unsupervised learning) is important:**

- **If the class labels are known, many different supervised learning methods are available. They can be used for prediction of the outcome of future objects.**

- **If the class labels are unknown, unsupervised learning methods have to be used. For those, it is *difficult to ascertain the validity of inferences* drawn from the output.**

Cluster analysis

Goal in cluster analysis:

Grouping a collection of objects into subsets or "clusters",

such that those within each cluster are more closely related to one another than objects assigned to different clusters.

Two ingredients are needed to group objects:

<u>Distance measure</u>

A notion of distance or similarity of two objects: When are two objects close to each other?

<u>Cluster algorithm</u>

A procedure to minimize distances of objects within groups and/or maximize distances between groups.

Cluster analysis: Bi-Clustering

- Clustering columns: grouping similar samples

- Clustering rows: Grouping genes with similar trajectories

- Biclustering: Group genes that have similar partial trajectories in a subset of the samples

Literature

Tanay, A., Sharan, R., and Shamir, R. (2002): **Discovering Statistically Signifnicant Biclusters in Gene Expression Data,** *Bioinformatics* 18, Suppl.1, 136-144.

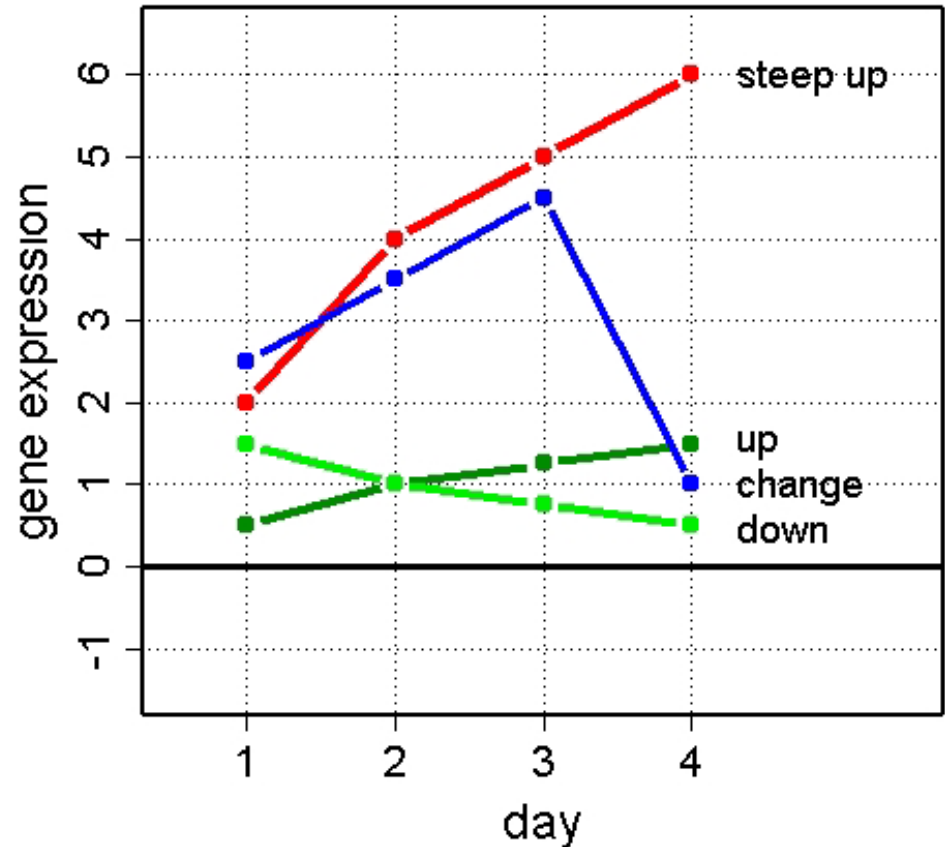The gene expression matrix

probes

genes
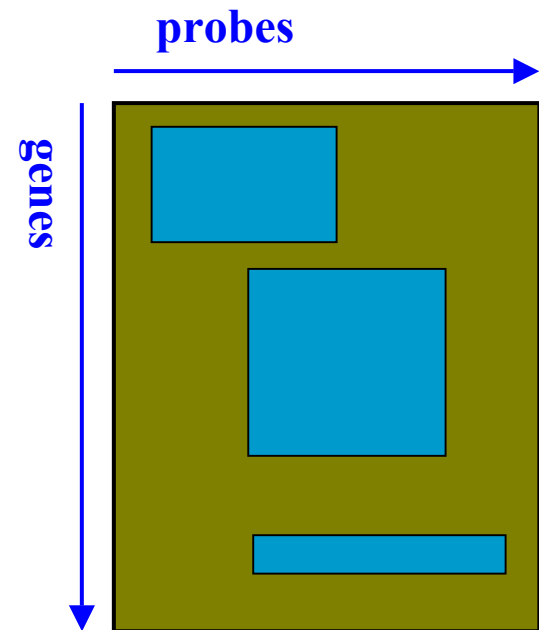
Time series example

Biology

Measurements of gene expression on 4 (consecutive) days.

Statistics

Every gene is coded by a vector of length 4.

- steep up: $x_1 = (2, 4, 5, 6)$
- up: $x_2 = (2/4, 4/4, 5/4, 6/4)$
- down: $x_3 = (6/4, 4/4, 3/4, 2/4)$
- change: $x_4 = (2.5, 3.5, 4.5, 1)$

Distance measures - Time series example

Euclidean distance

The distance between two vectors is the square root of the sum of the squared differences over all coordinates.

$$d_E(x_1, x_2) = \sqrt{(2\text{-}2/4)^2 + (4\text{-}4/4)^2 + (5\text{-}5/4)^2 + (6\text{-}6/4)^2} = 3\sqrt{3/4} \approx 2.598$$

- steep up: x_1 = (2, 4, 5, 6)
- up: x_2 = (2/4, 4/4, 5/4, 6/4)
- down: x_3 = (6/4, 4/4, 3/4, 2/4)
- change: x_4 = (2.5, 3.5, 4.5, 1)

0	2.60	2.75	2.25
2.60	0	1.23	2.14
2.75	1.23	0	2.15
2.25	2.14	2.15	0

Matrix of pairwise distances

Distance measures - Time series example

Manhattan distance

The distance between two vectors is the sum of the absolute (unsquared) differences over all coordinates.

$$d_M(x_1, x_2) = |2\text{-}2/4| + |4\text{-}4/4| + |5\text{-}5/4| + |6\text{-}6/4| = 51/4 = 12.75$$

- steep up: $x_1 = (2, 4, 5, 6)$
- up: $x_2 = (2/4, 4/4, 5/4, 6/4)$
- down: $x_3 = (6/4, 4/4, 3/4, 2/4)$
- change: $x_4 = (2.5, 3.5, 4.5, 1)$

0	12.75	13.25	6.50
12.75	0	2.50	8.25
13.25	2.50	0	7.75
6.50	8.25	7.75	0

Matrix of pairwise distances

Distance measures - Time series example

Correlation distance

Distance between two vectors is $1-\rho$, where ρ is the Pearson correlation of the two vectors.

$$d_C(x_1, x_2) = 1 - \frac{(2-\frac{17}{4})(\frac{2}{4}-\frac{17}{16}) + (4-\frac{17}{4})(\frac{4}{4}-\frac{17}{16}) + (5-\frac{17}{4})(\frac{5}{4}-\frac{17}{16}) + (6-\frac{17}{4})(\frac{6}{4}-\frac{17}{16})}{\sqrt{(2-\frac{17}{4})^2 + (4-\frac{17}{4})^2 + (5-\frac{17}{4})^2 + (6-\frac{17}{4})^2} \sqrt{(\frac{2}{4}-\frac{17}{16})^2 + (\frac{4}{4}-\frac{17}{16})^2 + (\frac{5}{4}-\frac{17}{16})^2 + (\frac{6}{4}-\frac{17}{16})^2}}$$

- steep up: x_1 = (2, 4, 5, 6)

- up: x_2 = (2/4, 4/4, 5/4, 6/4)

- down: x_3 = (6/4, 4/4, 3/4, 2/4)

- change: x_4 = (2.5, 3.5, 4.5, 1)

0	0	2	1.18
0	0	2	1.18
2	2	0	0.82
1.18	1.18	0.82	0

Matrix of pairwise distances

Distance measures - Time series example

Euclidean

0	2.60	2.75	2.25
2.60	0	1.23	2.14
2.75	1.23	0	2.15
2.25	2.14	2.15	0

Manhattan

0	12.75	13.25	6.50
12.75	0	2.50	8.25
13.25	2.50	0	7.75
6.50	8.25	7.75	0

Correlation

0	0	2	1.18
0	0	2	1.18
2	2	0	0.82
1.18	1.18	0.82	0

Comparison: All distances are normalized to the interval [0,10] and then rounded.

	steep up			up			down			change		
steep up	0	0	0	9	9	0	10	10	10	8	4	5
up	9	9	0	0	0	0	4	1	10	7	6	5
down	10	10	10	4	1	10	0	0	0	7	5	4
change	8	4	5	7	6	5	7	5	4	0	0	0

Summary

- **Euclidean** distance measures average difference across coordinates.

- **Manhattan** distance measures average difference across coordinates, in a robust way.

- **Correlation** distance measures difference with respect to trends.

Distance measures - standardization

Standardization

- Data points are normalized with respect to mean and variance:

  Apply transformation $x \mapsto \frac{x - \hat{\mu}}{\hat{\sigma}}$, where $\hat{\mu}$ is an estimator of the mean (usually average across coordinates) and $\hat{\sigma}$ is an estimator of the variation (usually empirical standard deviation).

- After standardization, Euclidean distance and Correlation distance are equivalent(!): $d_E(x_1, x_2)^2 = 2n d_C(x_1, x_2)$

- Standardization makes sense, if you are not interested in the magnitude of the effects, but in the effect itself. Results can be misleading for noisy data.

Distance measures

MESSAGE 2

Appropriate choice of distance measure depends on your intention!

Cluster algorithms

Most popular cluster algorithms:

- **Hierarchical clustering algorithms**

- **K-means**

- **PAM (Partitioning around medoids)**

- **SOM's (Self-Organizing Maps)**

- K-means and SOM's take original data directly as input.

- Hierarchical cluster algorithms and PAM allow the choice of a dissimilarity matrix $\mathbf{d}$, that assigns to each pair of objects $\mathbf{x_i}$ and $\mathbf{x_j}$ a value $d(\mathbf{x_i}, \mathbf{x_j})$ as their distance.

Hierarchical cluster algorithms

- **Hierarchical clustering** was the first algorithm used in microarray research to cluster genes (Eisen et al. (1998)).

- First, each object is assigned to its own cluster. Then, **iteratively, the two most similar clusters are joined**, representing a new node of the clustering tree. The node is computed as average of all objects of the joined clusters. Then, the similarity matrix is updated with this new node replacing the two joined clusters. This process is repeated until only one single cluster remains.

Hierarchical cluster algorithms

- Calculation of distance between two clusters is based on object dissimilarity between the objects from the two clusters:

  - Average linkage: Average distance
  - Single linkage: Smallest distance
  - Complete linkage: Largest distance

- Instead of agglomerative clustering, sometimes divisive clustering is used:

  Iteratively, best possible splits are calculated.

Hierarchical cluster algorithms

- **Visualization** of hierarchical clustering through **dendrogram**:
    - Clusters that are joined are combined by a line.
    - Height of line is average distance between clusters.
    - Cluster with smaller variation is plotted on left side.

- The procedure provides a **hierarchy of clusterings**, with the number of clusters ranging from 1 to the number of objects.

- **BUT**:
    - Parameters for distance matrix: **n(n-1)/2**
    - Parameters for dendrogram: **n-1**.
    - ⟶ Hierarchical clustering does not show the full picture!

Time series example

- **<u>Euclidean distance</u>**
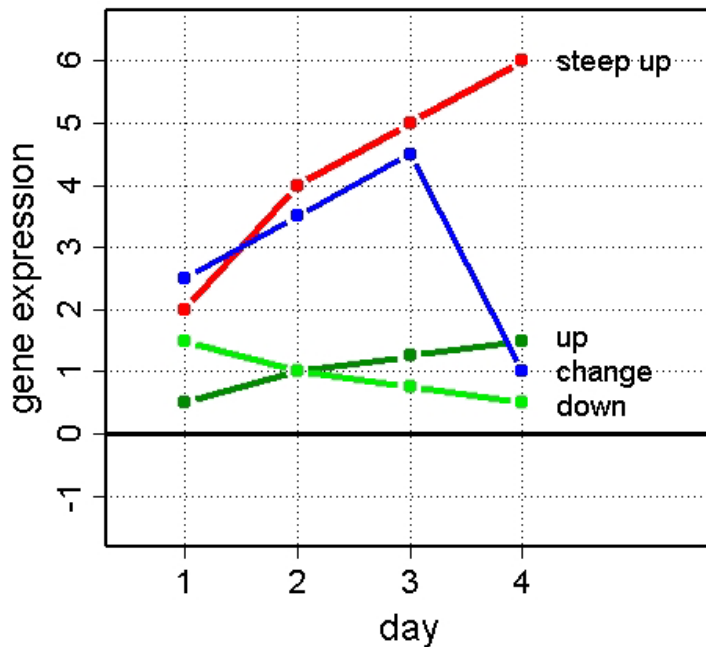 **Similar values are
  clustered together**
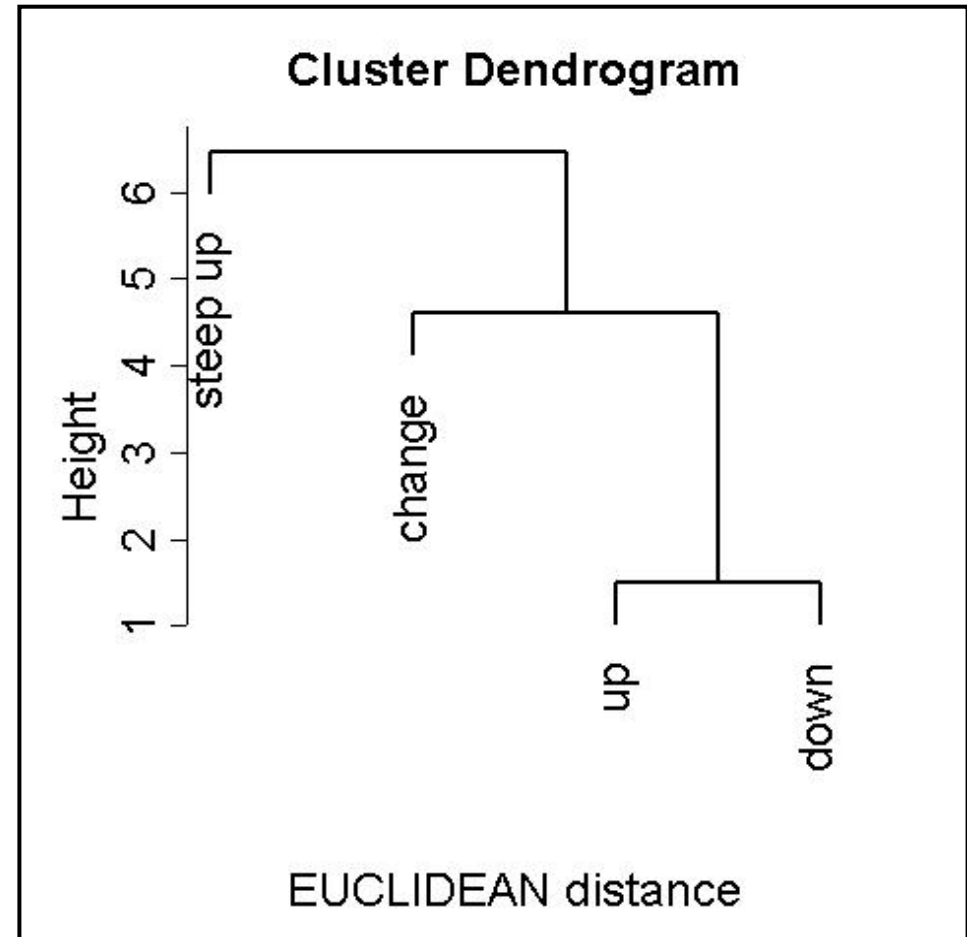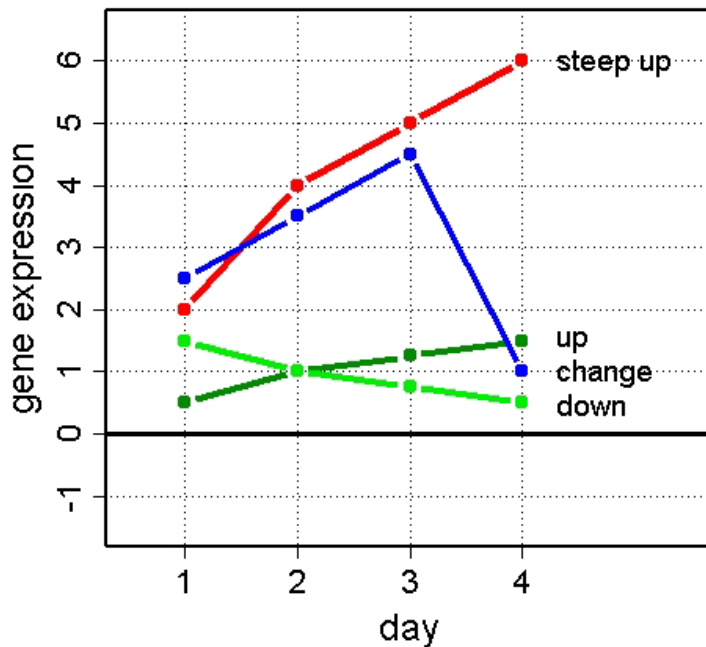
Time series example

- **Manhattan distance**
  **Similar values are
  clustered together
  (robust)**
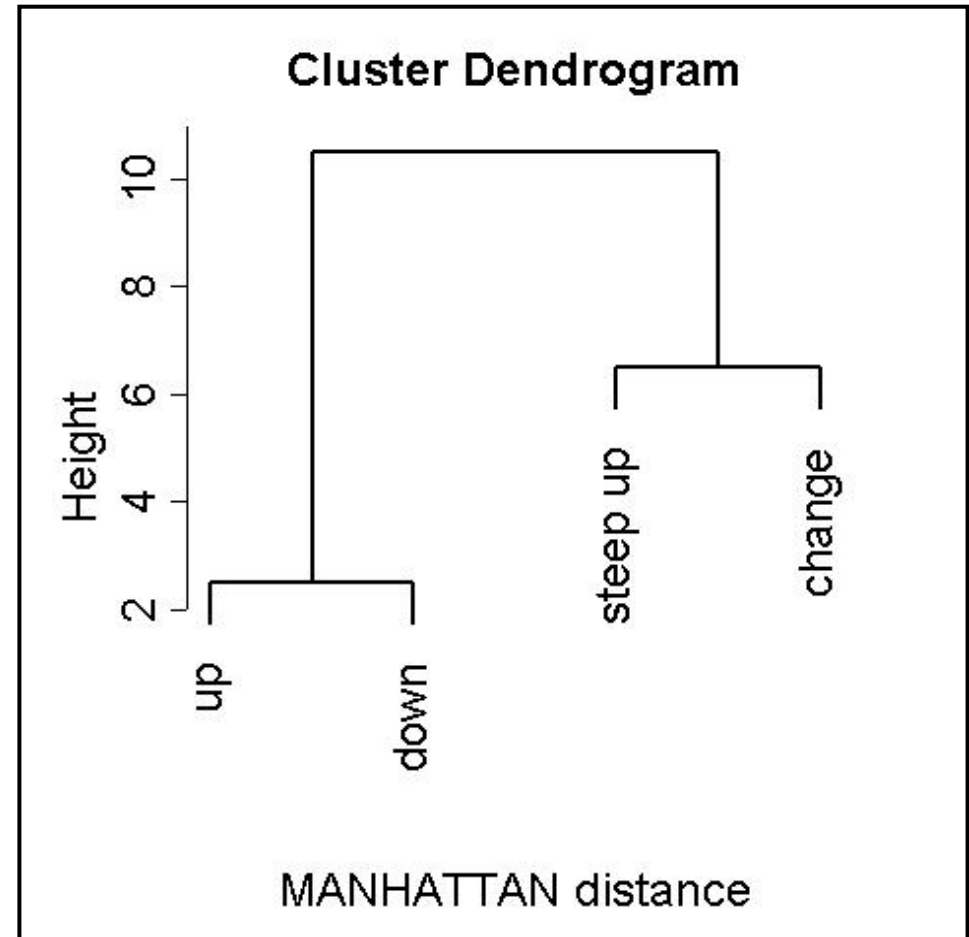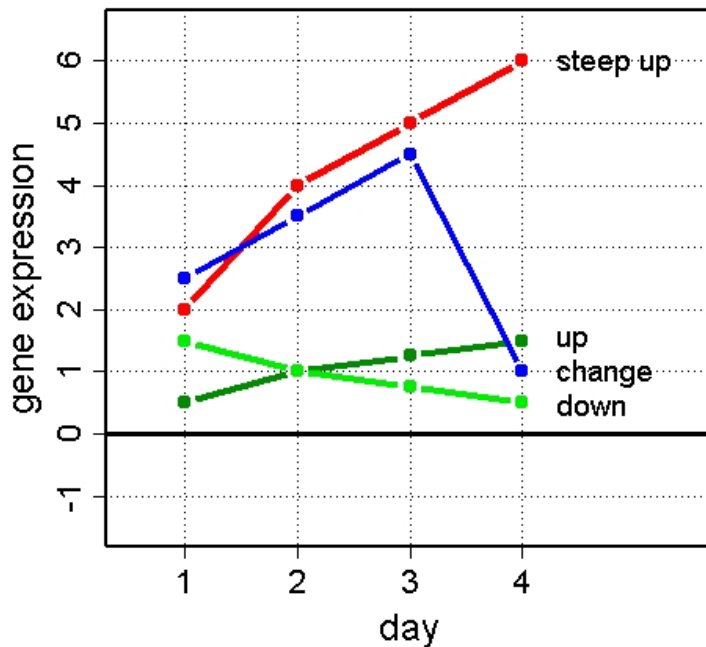
Time series example

- **Correlation distance**
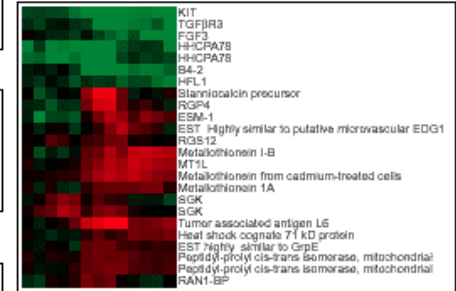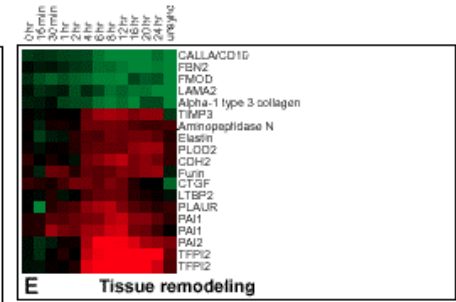 **Similar trends are
  clustered together**
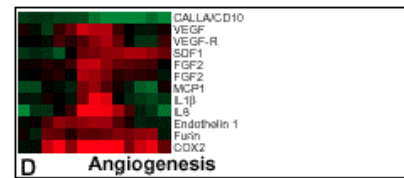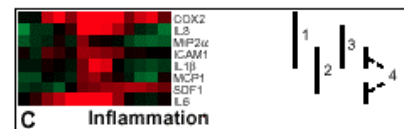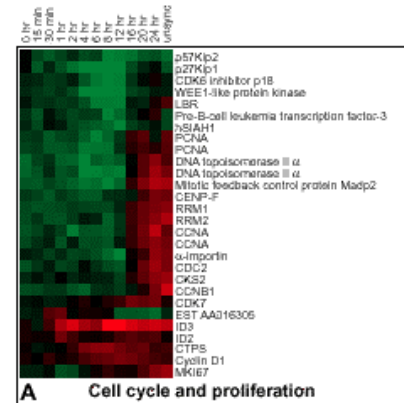

Cluster Dendrogram

CORRELATION distance
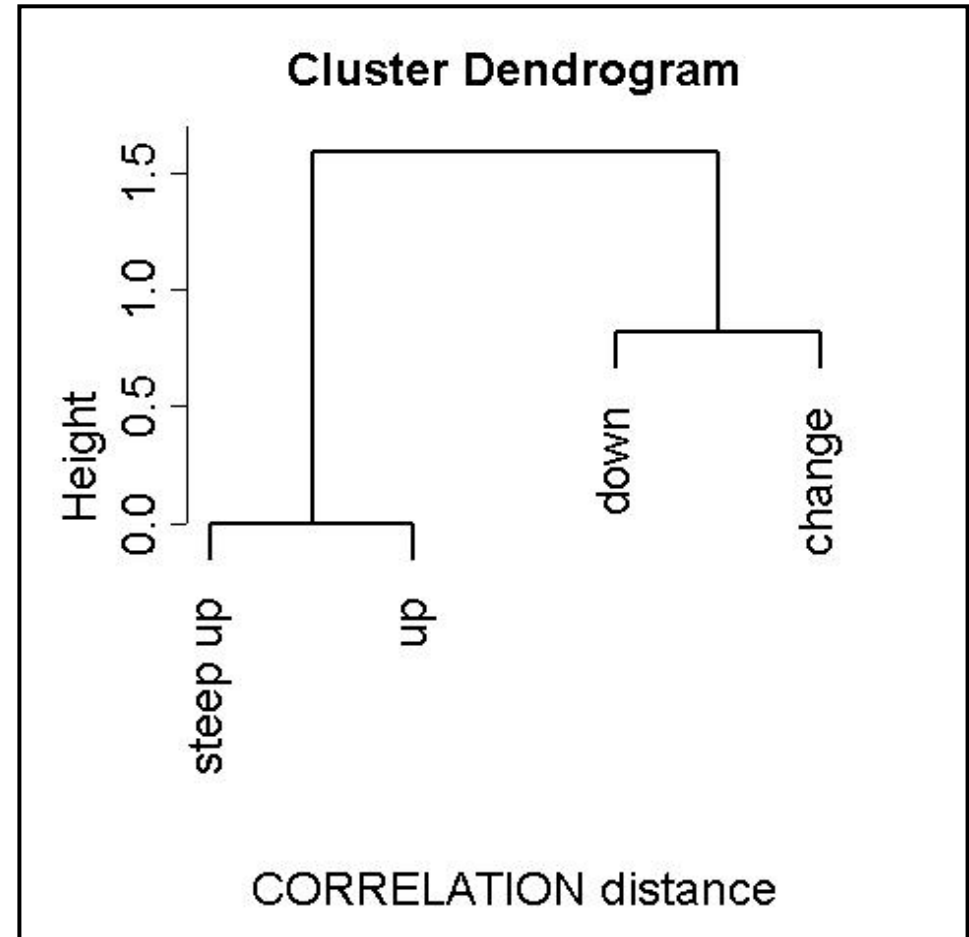
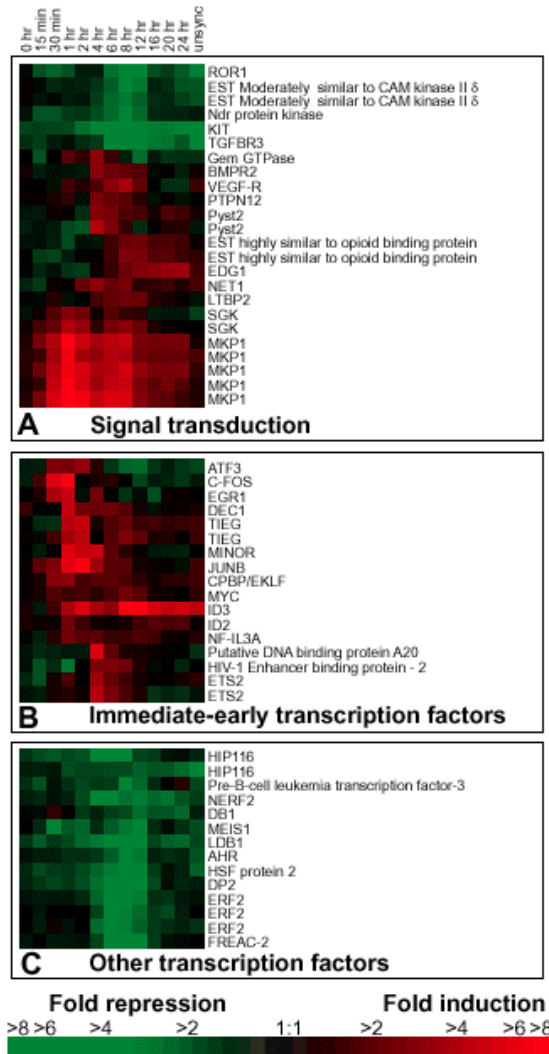# Clustering time series data – literature examples

Iyer et al., Science, Jan 1999:

Genes from functinal classes are clustered together (sometimes!).

Careful interpretation neccessary!

Golub et al.: Leukemia dataset, _http://www.genome.wi.mit.edu/MPR_

3 cancer classes:
25 acute myeloid
leukemia (AML),
47 acute lympho-
blastic leukemia
(ALL), the latter
9 T-cell and 38
B-cell.
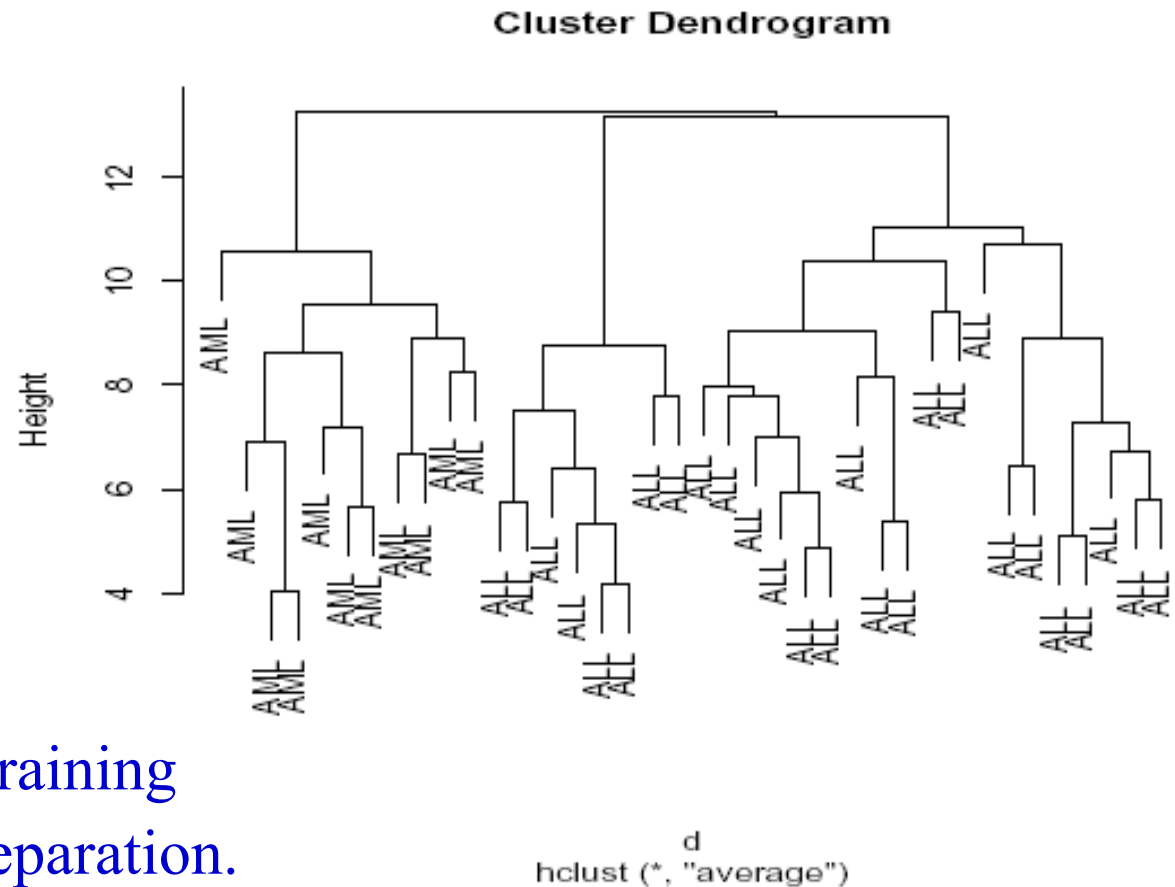
Dendrogram for 38 training
data shows perfect separation.

Cluster Dendrogram

d
hclust (*, "average")

Cluster algorithms – k-means

- **K-means** is a **partitioning algorithm** with a prefixed number **k** of clusters. It tries to minimize the sum of within-cluster-variances.

- The algorithm chooses a random sample of **k** different objects as initial cluster midpoints. Then it alternates between two steps until convergence:

  1. Assign each object to its closest of the k midpoints with respect to **Euclidean distance**.

  2. Calculate k new midpoints as the averages of all points assigned to the old midpoints, respectively.

- K-means is a randomized algorithm, two runs usually produce different results. Thus it has to be applied a few times to the same data set and the result with minimal sum of within-cluster-variances should be chosen.

Cluster algorithms – Self-Organizing maps

- **SOM's** are similar to k-means, but with additional **constraints**.

- Mapping from input space onto one or two-dimensional array of **k** total nodes.

- Iteration steps (20000-50000):
  - Pick data point P at random
  - Move all nodes in direction of P, the closest node most, the further a node is in network topology, the less
  - Decrease amount of movement with iteration steps

Data point   Node (cluster prototypes)
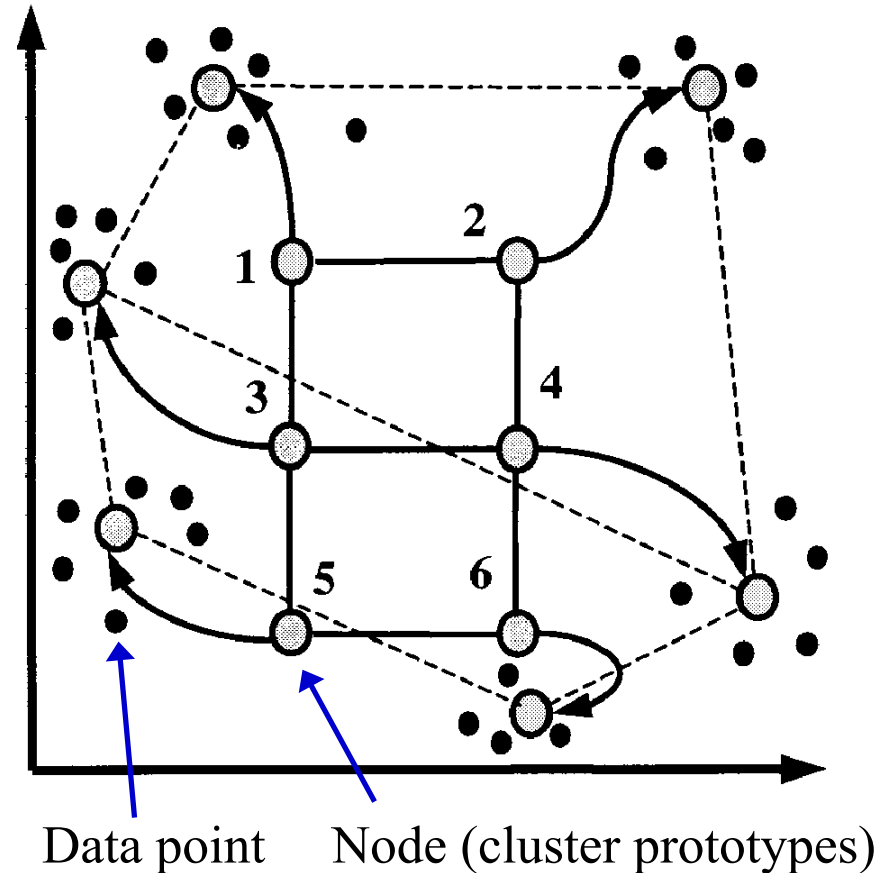
Tamayo et al. (1999): First use of SOM's for gene clustering from microarrays

Cluster algorithms - PAM

- **PAM** (Partitioning around medoids, Kaufman and Rousseeuw (1990)) is a partitioning algorithm, a generalization of k-means.

- For an arbitrary dissimilarity matrix **d** it tries to minimize the sum (over all objects) of distances to the closest of **k** prototypes.

- Objective function: $\sum_{i=1}^{n} \min_{j=1,\ldots,k} d(i, m_j)$ (**d**: Manhattan, Correlation, etc.)

- BUILD phase: Initial 'medoids'.

  SWAP phase: Repeat until convergence:
  - Consider all pairs of objects (i,j), where i is a medoid and j not, and make the i ↔ j swap (if any) which decreases the objective function most.

Comparative study

- **Comparative study** for **tumor classification** with microarrays: Comparison of hierarchical clustering, k-means, PAM and SOM's

- **Data sets**:
    - Golub et al: Leukemia dataset, *http://www.genome.wi.mit.edu/MPR*, 3 cancer classes: 25 acute myeloid leukemia (AML) and 47 acute lymphoblastic leukemia (ALL) (9 T-cell and 38 B-cell), Affymetrix.

    - Ross et al.: NCI60 cancer dataset, *http://genome-www.stanford.edu/nci60*,

      9 cancer classes: 9 breast, 6 central nervous system, 7 colon, 8 leukemia, 8 melanoma, 9 lung, 6 ovarian, 2 prostate, 8 renal, cDNA microarray

- Rahnenführer (2002): **Efficient clustering methods for tumor classification with gene expression arrays,** *Proc. of '26th Ann. Conf. of the Gesellschaft für Klassifikation'*, Mannheim, July 2002.

Comparative study – cluster validity

- If true class labels are known, the validity of the clustering can be verified by comparing true class labels and clustering label.

N… table of observations
n_{ij}… number of observations
in class i and cluster j

$$
\begin{array}{c|c}
\boxed{N} & \cdot \\
\hline
\cdot & n_{..}
\end{array}
=
\begin{array}{cccc|c}
n_{11} & n_{12} & \cdots & n_{1l} & n_{1.} \\
n_{21} & n_{22} & \cdots & n_{2l} & n_{2.} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
n_{k1} & n_{k2} & \cdots & n_{kl} & n_{k.} \\
\hline
n_{.1} & n_{.2} & \cdots & n_{.l} & n_{..}
\end{array}
$$

Rand index:
Probability of randomly drawing 'consistent' pair of observations

$$
Rand = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \left[\sum_i\binom{n_{i.}}{2}\sum_j\binom{n_{.j}}{2}\right]\Big/\binom{n}{2}}{\frac{1}{2}\left[\sum_i\binom{n_{i.}}{2} + \sum_j\binom{n_{.j}}{2}\right] - \left[\sum_i\binom{n_{i.}}{2}\sum_j\binom{n_{.j}}{2}\right]\Big/\binom{n}{2}}
$$

Comparative study – method

Definition 5.1 *BOSC (Bootstrap-scaling):*
Let $n, p, k, B \in \mathbb{N}_{\geq 0}$. Let $(M_{ij})_{i=1..p, j=1..n}$ be a data (gene expression) matrix of dimension $p \times n$ (for p genes and n samples) and $(l_1, \ldots, l_n) \in (1, \ldots, k)^n$ an n-dimensional label vector that assigns every sample to one of k clusters (tumor types).

1. *For $b \in (1, \ldots, B)$:*
   Create replicate M_{ij}^b by randomly drawing from all values $M_{i_0 j_0}$ of the original data matrix that fulfill $i_0 = i$ and $j_0 \in \{j : l_j = l_{j_0}\}$.
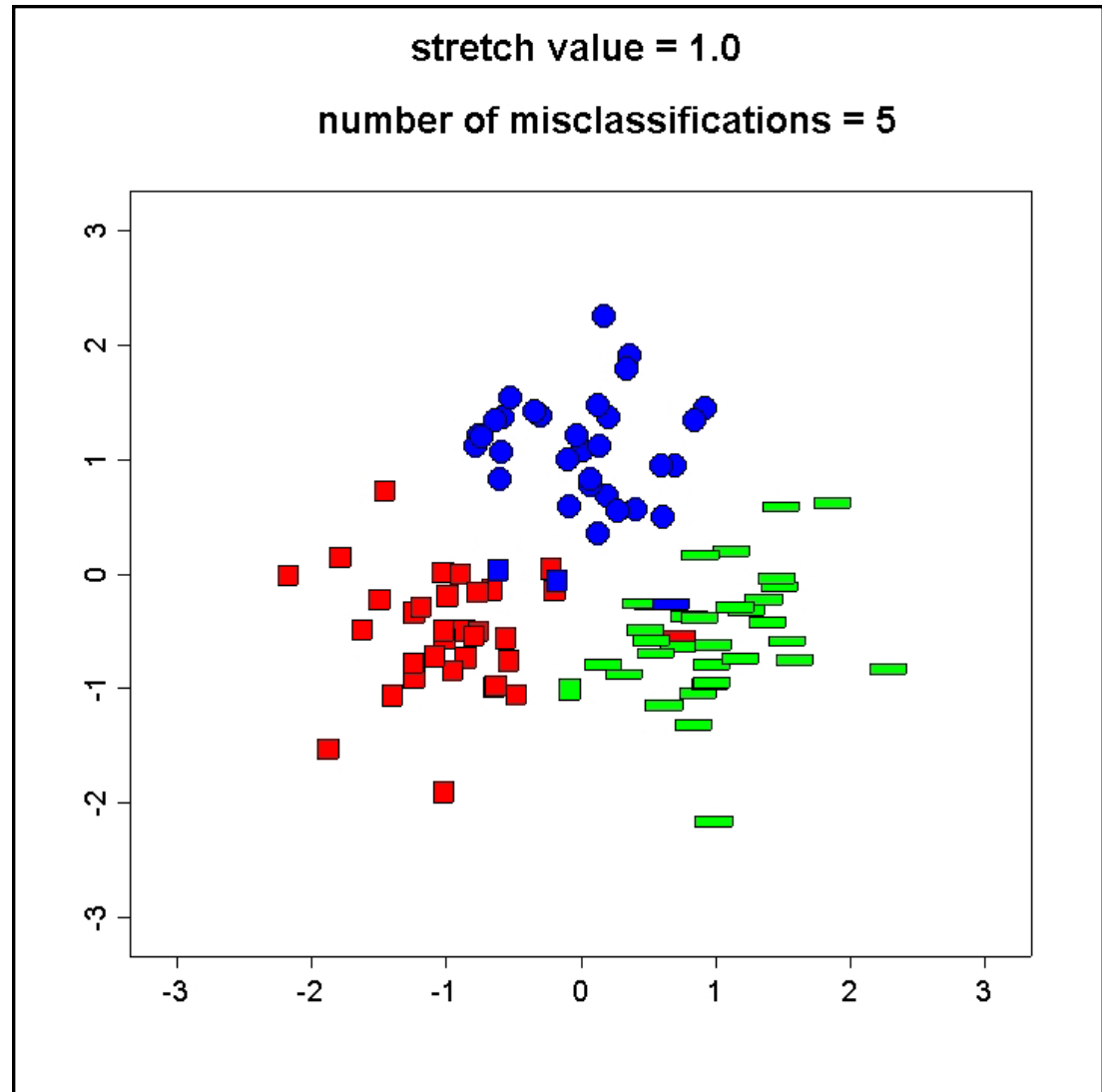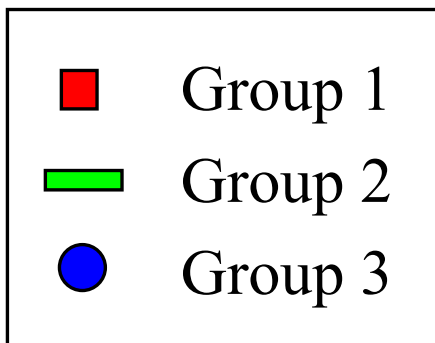
2. *For $s \in (s_1, \ldots, s_S)$ with $0 < s_1 < s_2 < \ldots < s_S < \infty$:*
   Define the modified (stretched) replicate $M_{ij}^b(s)$ as

$$M_{ij}^b(s) := (1 - s)\left(\frac{1}{\#\{j_0 : l_j = l_{j_0}\}} \sum_{j_0 : l_j = l_{j_0}} M_{ij_0} \right) + s\, M_{ij}^b. \qquad (5.1)$$

Comparative study - method

Color ➝ Group
Shape ➝ Cluster

🟥	Group 1
🟩	Group 2
🔵	Group 3

stretch value = 1.0

number of misclassifications = 5

Comparative study - method

Color ➡ Group
Shape ➡ Cluster

🟥	Group 1
🟩	Group 2
🔵	Group 3

stretch value = 1.5

number of misclassifications = 14
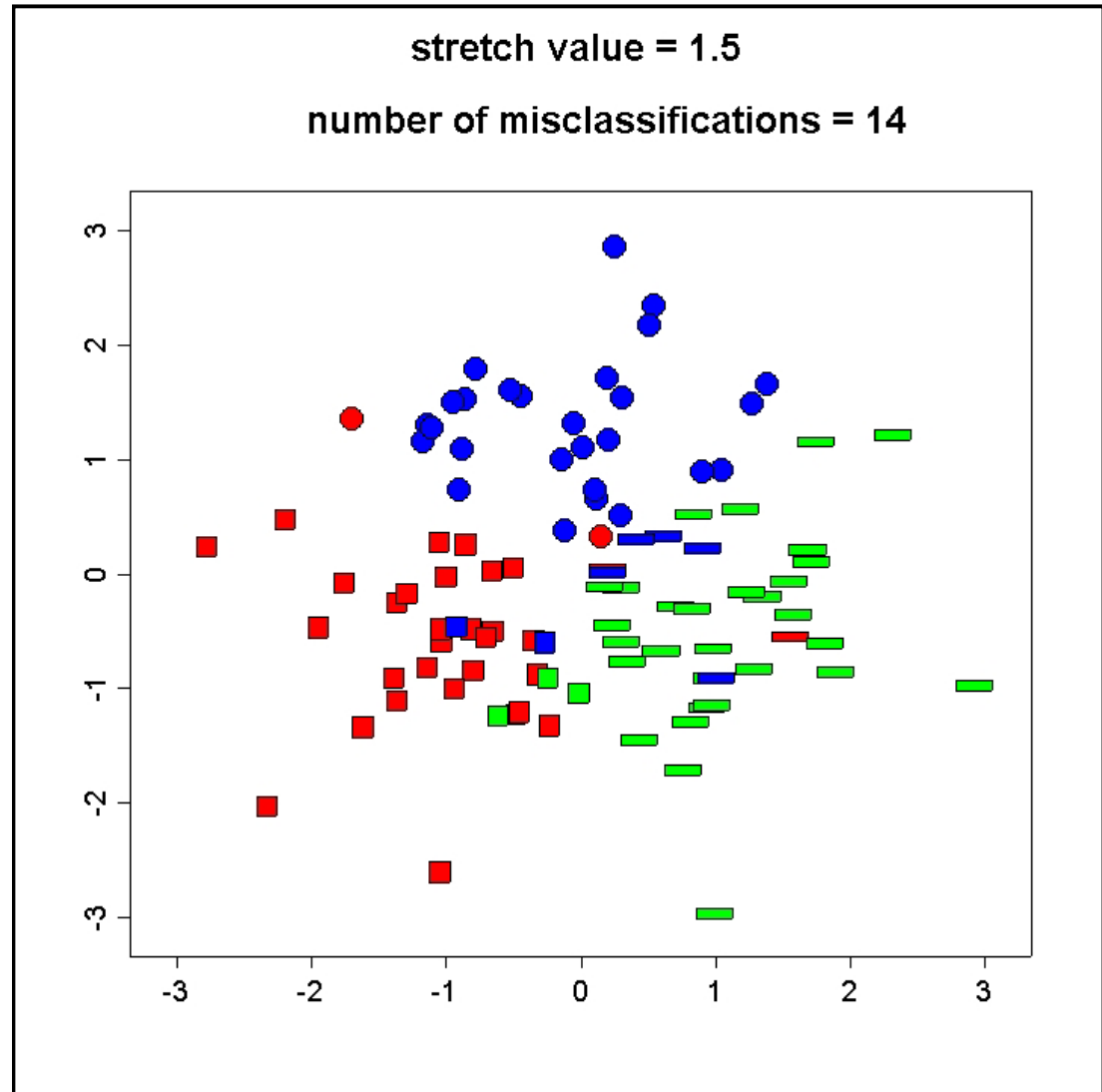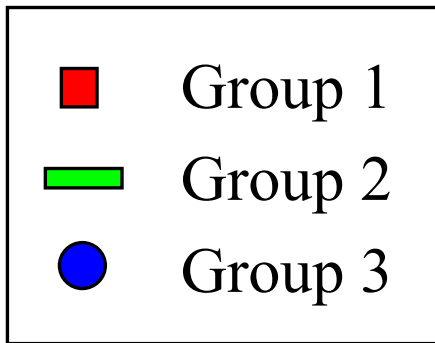
Comparative study - winners

Winners

- —— **K-means**
- – – – **Hierarchical, Correlation**
- ⋯⋯ **PAM, Manhattan**
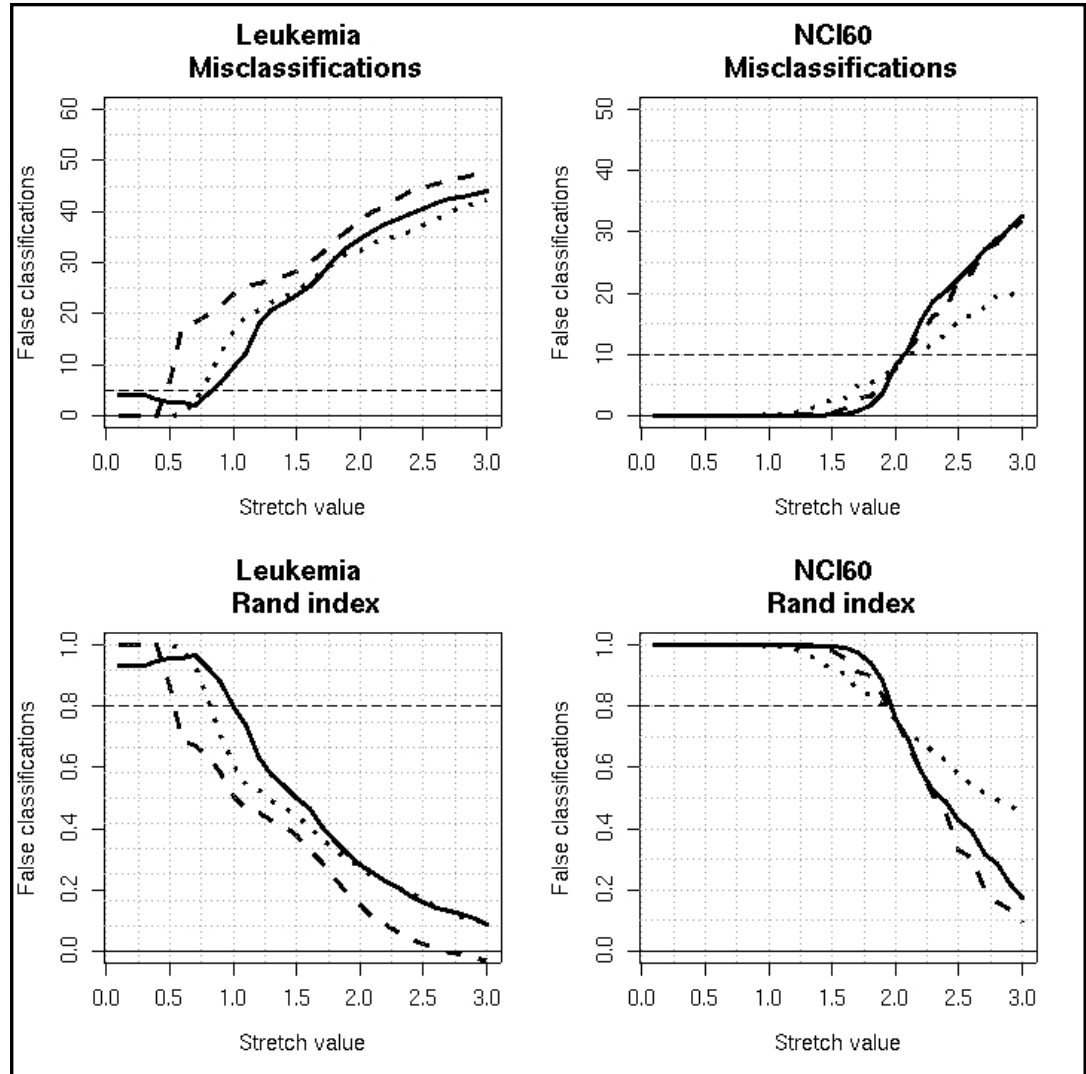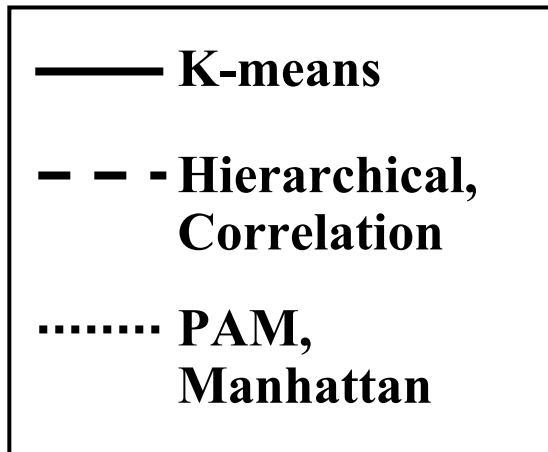
Comparative study - results

- **Superiority of k-means with repeated runs**

  Similar for discriminant analysis: FLDA best (Dudoit et al., 2001)

- **Superiority of PAM with Manhattan distance** for noisy data

- Differences depend on the specific dataset

- **<u>Preselection of genes</u>**

  Various approaches have been proposed for gene selection, especially in *supervised* learning.

  For clustering samples, a practical proceeding is to choose the **top 100-200 genes with respect to variance (across samples)**. This decreases noise and computation time.

Classification

MESSAGE 3

**Simple cluster algorithms work better
in case of little model knowledge!**

**(But: More sophisticated methods might be more
appropriate with more a priori knowledge)**

Recommendations

- **<u>Interest in specific genes:</u>**

  If you search for genes that are co-regulated with a specific gene of you choice, **DO SO!** Don't use clustering, but generate a list of genes close to your gene with respect to some distance.

- **<u>Clustering after feature selection?</u>**

  NO! Do not first select genes based on the outcome of some covariable (e.g. tumor type) for your clustering. You will ALWAYS find differences w.r.t. this covariable, since this is how you selected the genes!

- **<u>Number of clusters</u>**

  No general rule how to select the 'correct' number of clusters.
  Adhoc approach is to **try different numbers** and choose cutoff, for which performance of the clustering algorithm breaks down.
  The quality of a clustering result depends on the concept of a cluster.

Recommendations – silhouette plot

Heuristic approach for estimation of number of clusters

For each observation *i*, define *silhouette width s(i)* as follows:

a(i) := average dissimilarity between i and all other points of its cluster.

For all *other* clusters *C*, let *d(i,C)* := average dissimilarity of *i* to all observations of *C*. Define $b(i) := min_C d(i,C)$.

Silhouette width: $s(i) := (b(i)-a(i)) / max(a(i),b(i))$.

Maximal **average silhouette width** over all observations can be used to select the number of clusters.

Observations with *s(i)* close to *1* can be considered well-clustered, whereas observations with *s(i)<1* are misclassified.

Recommendations – silhouette plot

```
x = matrix(rnorm(10),ncol=2)
row.names(x) = paste("x",1:5,sep="")
y = matrix(rnorm(10)+2,ncol=2)
row.names(y) = paste("y",1:5,sep="")
data = rbind(x,y)


p = pam(data,k=2)
data1 = data[which(p$clustering==1),]
data2 = data[which(p$clustering==2),]


plot(0,10,xlim=c(-2,4),ylim=c(-2,4),xlab="",ylab="",cex.lab=2)
symbols(rbind(c(0,0),c(2,2)),squares=c(0.25,0.25),inches=F,bg="black",add=T)
symbols(data1,circles=rep(0.1,nrow(data1)),inches=F,bg="blue",add=T)
symbols(data2,circles=rep(0.1,nrow(data2)),inches=F,bg="red",add=T)
positions = data
positions[,1] = positions[,1]+0.3
text(positions,rownames(data),cex=2)


X11()
color = c(rep("blue",nrow(data1)),rep("red",nrow(data2)))
plot(silhouette(p),col=rev(color),do.col.sort=F)
```
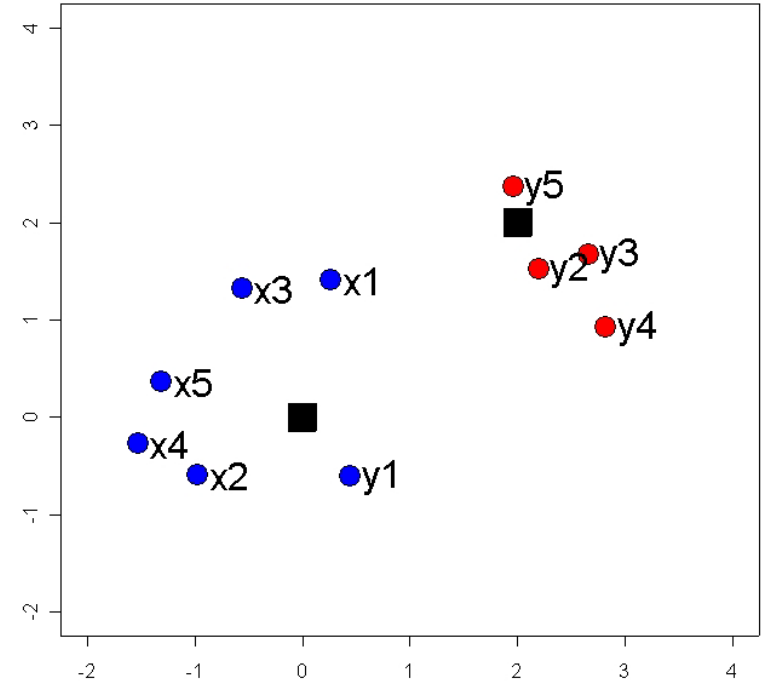
Recommendations – silhouette plot

Silhouette plot of pam(x = data, k = 2)

n = 10

2 clusters C_j

$j : n_j | ave_{i \in C_j} s_i$

1 : 6 | 0.49

2 : 4 | 0.73

Silhouette width s_i

Average silhouette width : 0.58
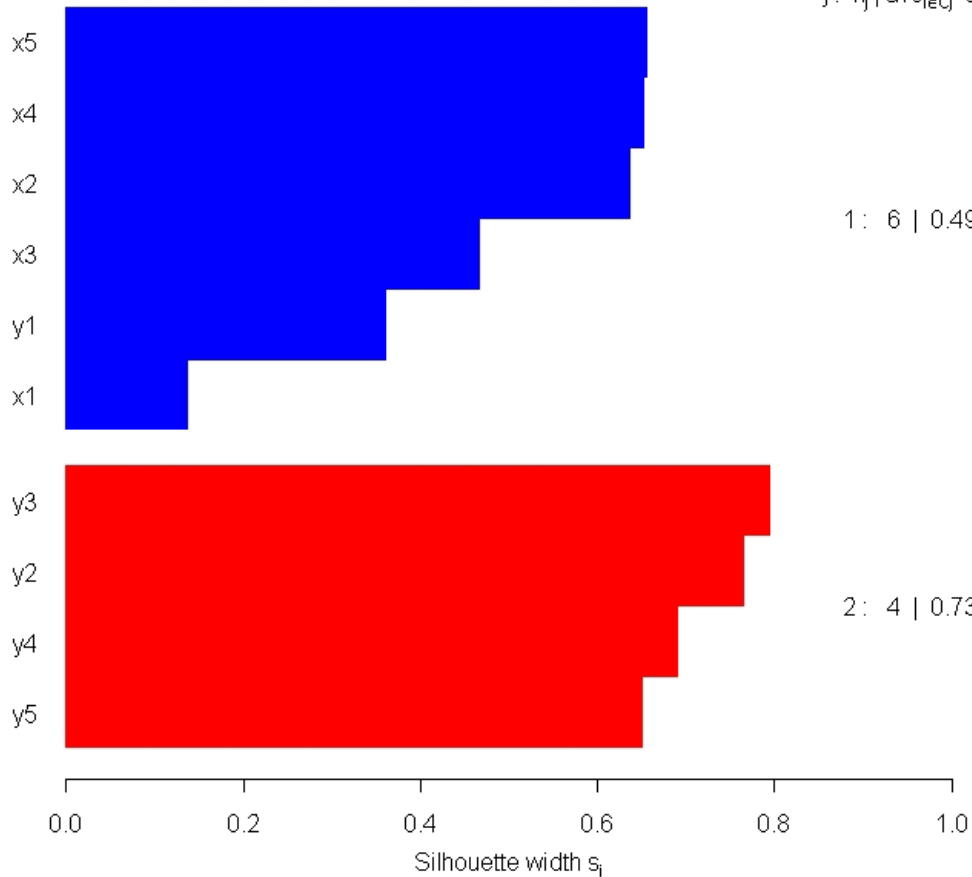
Other exploratory methods

- **PCA: Principal Component Analysis**
  Data are projected on lower dimensional space.
  Iteratively, the direction with largest variance is selected as i-th principal component (orthogonality constraint).
  Can be used as preprocessing step, but low interpretability.

- **Correspondence Analysis**
  Genes and samples are projected into two-dimensional plane to show associations between them.

- **ISIS: A class discovery method**
  Search for class distinctions that are characterized by differential expression of just a small set of genes, not by global similarity of the gene expression profile.

Other exploratory methods - CAST

- **CAST (Cluster Affinity Search Technique)**

  Ben-Dor A, Shamir R, Yakhini Z (1999): **Clustering gene expression patterns.** *J. Comput Biology* 6: 281-97.

  - Input: Similarity matrix and a threshold parameter.

  - Iteratively, clusters are generated one at a time. Genes are added to an open cluster, as long as their average similarity (affinity) exceeds the threshold. Then a new cluster is started.

  - After termination of the assignment process, objects can still be added or removed from clusters.

  - This improves standard hierarchical clustering.

Other exploratory methods - Bi-Clustering

Tanay A, Sharan R, Shamir R (2002):
Discovering Statistically Significant Biclusters in Gene Expression Data.
Bioinformatics 18, Suppl.1, 136-144.

- Graph-theoretic algorithm coupled with statistical modeling.

- Genes and samples both represented as nodes of a bipartite graph and connected with weights according to the expression of the respective gene and sample.

- The heaviest subgraph is determined with an algorithm that runs in polynomial time.

The gene expression matrix
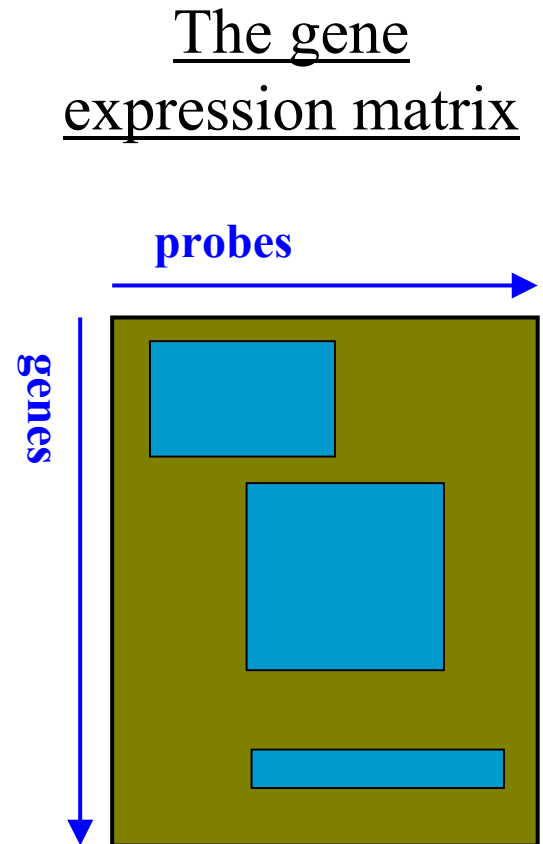
Cluster validity

- **Yeung et al. (Bioinformatics, 2001)**

  Framework for assessing quality of algorithms for gene clustering.
  Apply algorithm to data from all but one condition (sample) and use the remaining condition to assess predictive power of the resulting clusters (leave-one-out scenario).

- **Dudoit and Fridlyand (Genome Biology, 2002)**

  Prediction-based resampling method *Clest* to estimate the number of clusters in a dataset.

- **Smolkin and Ghosh (BMC Bioinformatics, 2003)**

  Cluster stability scores for microarray data in cancer studies, based on subsampling techniques.

R commands and libraries

- **library(mva)**
    - Hierarchical clustering: *hclust()*
    - Kmeans: *kmeans()*
    - Principal components: *princomp()*

- **library(cluster)**
    - PAM: *pam()*
    - Silhouette information: *silhouette()*

- **ISIS package:** http://www.molgen.mpg.de/~heydebre

SUMMARY

MESSAGE 1:

Discriminant analysis: CLASSES <u>KNOWN</u>

Cluster analysis: CLASSES <u>NOT KNOWN</u>

MESSAGE 2:

Appropriate choice of distance measure depends on your intention!

MESSAGE 3:

Simple cluster algorithms work better in case of little model knowledge!

Literature

1. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 1999; 286: 531-37.

2. Alizadeh AA, Eisen MB, Davis RE and 28 others. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 2000; 403: 503-11.

3. Jain A, Dubes RC. Algorithms for Clustering Data. Englewood Cliffs, New Jersey: Prentice Hall; 1988.

4. Azuaje F. Clustering-based approaches to discovering and visualising microarray data patterns. Brief. Bioinformatics 2003; 4: 31-42.

5. Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. PNAS 1998; 95: 14863-68.

6. Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. PNAS 1999; 96: 2907-12.

7. Kaufman L, Rousseeuw P. Finding Groups in Data. New York: John Wiley and Sons; 1990.

8. Ben-Dor A, Shamir R, Yakhini Z. Clustering gene expression patterns. J. Comput Biol. 1999; 6: 281-97.

Literature

9. Cheng Y, Church GM. Biclustering of expression data. Proc Int Conf Intell Syst Mol Biol. 2000; 8:93-103.

10. Tanay A, Sharan R, Shamir R. Discovering statistically significant biclusters in gene expression data. Bioinformatics 2002; Suppl 1: 136-44.

11. Hastie T, Tibshirani R, Eisen MB, Alizadeh A, Levy R, Staudt L, Chan WC, Botstein D, Brown P. 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. Genome Biol. 2000; 1(2): RESEARCH0003.

12. Yeung KY, Haynor DR, Ruzzo WL. Validating clustering for gene expression data. Bioinformatics 2001; 17: 309-18.

13. Rahnenführer J. Efficient clustering methods for tumor classification with microarrays. In: Between Data Science and Applied Data Analysis (Eds: M. Schader, W. Gaul, M. Vichi), Springer, Proc. 26th Ann. Conf. GfKl 2002; 670-679.

14. Dudoit S, Fridlyand J: A prediction-based resampling method to estimate the number of clusters in a dataset. Genome Biology 2002; 3:RESEARCH0036.

15. Smolkin, M, Ghosh, D. Cluster stability scores for microarray data in cancer studies. BMC Bioinformatics 2003, 4:36.

Much more interesting microarray analysis…

Contact: **rahnenfj@mpi-sb.mpg.de**

Jörg Rahnenführer
Computational Biology and Applied Algorithmics
Max Planck Institute for Informatics
D-66123 Saarbrücken, Germany
Phone: (+49) 681-9325 320

Visit us in Saarbrücken!

Saarvoir vivre…