SUPERVISED CLUSTERING OF GENES

Marcel Dettling

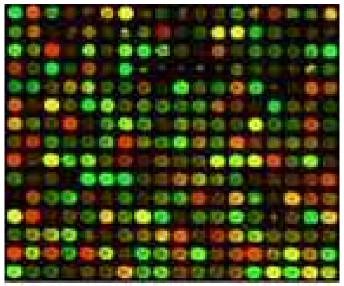
Seminar für Statistik, ETH Zürich

September 20, 2001

joint work with Peter Bühlmann, ETH Zürich

Microarray Gene Expression Data

- Microarray technology & preprocessing steps $\rightsquigarrow$ **gene expression matrix**



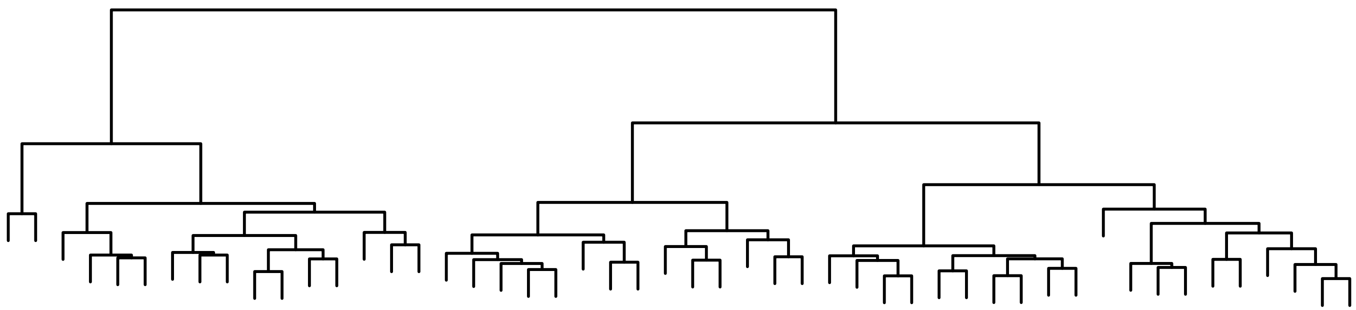
$$\rightsquigarrow \quad (x_{ij}) = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{p1} & x_{p2} & \cdots & x_{pn} \end{pmatrix}$$

- n experiments, showing expression levels of p genes.
- Additional categorical response vector

$$(y_j) = (y_1, y_2, \ldots, y_n)$$

Previous Work in Gene Clustering (I)

Standard approach: Hierarchical Clustering



- **Pursued by:** Weinstein et al. (1997), Eisen et al. (1998), . . .
- **Goal:** Arrange similar genes together
- **Main drawback:** Lack of precision

Previous Work in Gene Clustering (II)

Modified approach by Hastie et al. (2001): Supervised Clustering

Step 1: Hierarchical Clustering $\rightsquigarrow$ many candidate clusters

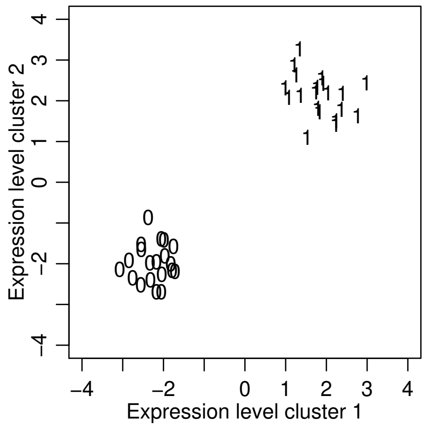
Step 2: Clusters with high correlation to the response are chosen

We think:

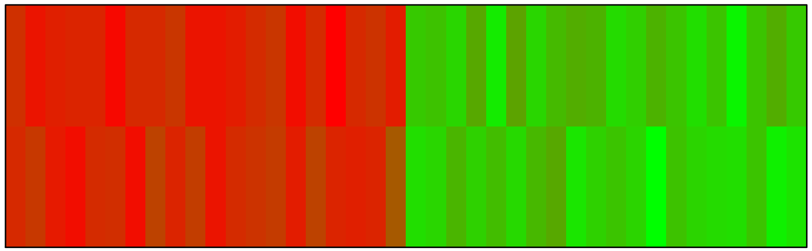
- Hastie et al.'s method is the wrong way to do it!
- Need to include the response variables **directly** in the clustering process

The Goal of Supervised Clustering

Clusters should optimally separate the response categories



Try to make the separation as easy as possible



The Partitioning Problem

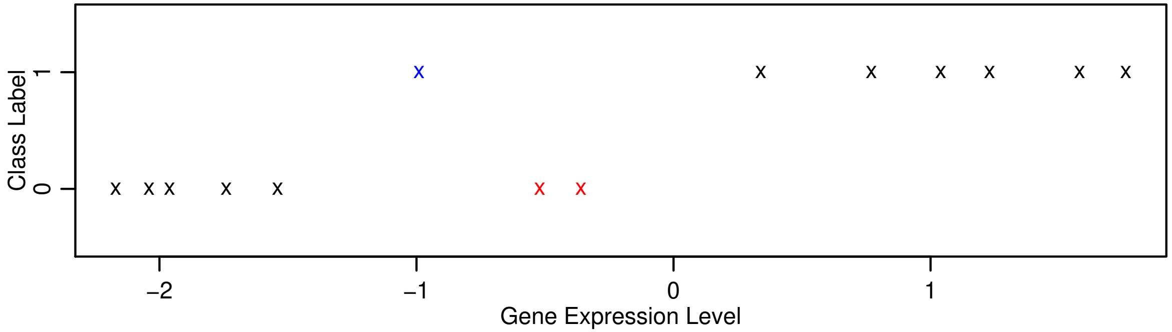
Summarize clusters by a representative value, the average expression profile of several gene expression vectors $x_i = (x_{i1}, \ldots, x_{in})$, $i \in \mathcal{C}$

$$x_{\mathcal{C}} = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} x_i \ \in \mathbb{R}^n$$

- Sensible choice since we want to cluster similar genes
- Numerous "good" clusters do exist $\rightsquigarrow$ Exhaustive search is impracticable
- We rely on a greedy algorithm
- Need for an objective function to measure cluster quality

Clustering by Score and Margin (I)

Basic Idea: Discrimination of the classes is easy if there is no "overlap"



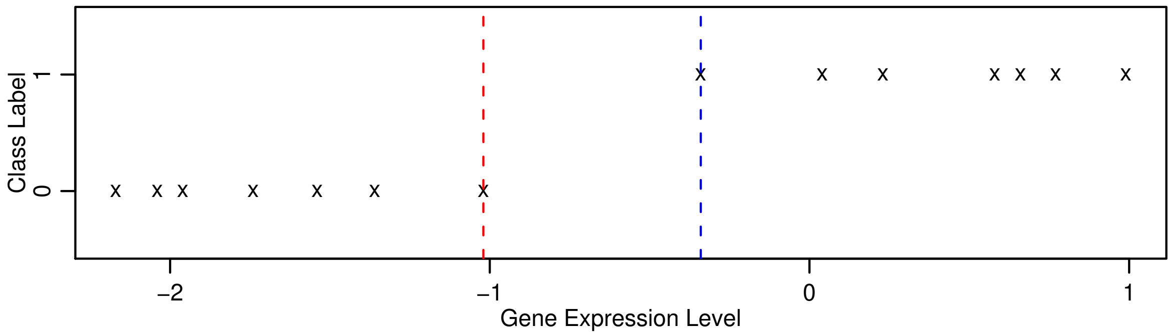
$\Rightarrow$ Severeness of the "overlap" for a particular gene or cluster expression vector $x = (x_1, \ldots, x_n)$ is measured by the score function

$$\text{Score}(x) = \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}_1} 1_{[x_j - x_i \leq 0]},$$

Equals the number of transpositions that are necessary for perfect separation

Clustering by Score and Margin (II)

Refinement by a second criterion: the margin function



⇒ Need margin function due to discreteness of the score function and to achieve uniqueness in decisions among genes/clusters with identical score

$$\text{Margin}(x) = \min_{j \in \mathcal{N}_1}(x_j) - \max_{i \in \mathcal{N}_0}(x_i),$$

While clustering, scores are optimized with first, margins with second priority

The Clustering Algorithm (I)

1. **Starting genes**

   a) If given, compute their average expression profile

   b) If unknown, identify the best individual gene

2. **Forward search**

   a) Average the current cluster with each individual gene $k = 1, \ldots, p$

   $$x_{\mathcal{C}+k} = \frac{1}{|\mathcal{C}| + 1} \left(x_k + \sum_{i \in \mathcal{C}} x_i \right)$$

   b) Identify the winning gene k^*

   c) Accept and repeat until cluster quality worsens

The Clustering Algorithm (II)

3. **Backward search**

   a) Exclude each gene $k \in \mathcal{C}$ and average the remaining ones

   $$x_{\mathcal{C}-k} = \frac{1}{|\mathcal{C}| - 1} \left(\sum_{i \in \mathcal{C} \setminus \{k\}} x_i \right)$$

   b) Identify the winning gene k^*

   c) Exclude and repeat until cluster quality worsens

4. **Stagewise clustering**

   Repeat forward and backward stages until convergence

5. **Multiple clustering**

   Discard the clustered genes and restart the algorithm at step 1

Generalization for Multiclass Problems

- Algorithm can only deal with binary problems, but extension to J-class problems is possible
- "Standard" approach: define an indicator response vector $\vec{y}$

$$\vec{y} = \left(y^{(1)}, \ldots, y^{(J)}\right) \in \{0,1\}^J, \text{ where } y^{(j)} = \begin{cases} 1, & \text{if } y = j, \\ 0, & \text{else.} \end{cases}$$

- Amounts to "one vs. all other classes" $\rightsquigarrow$ J binary problems
- Remark: Other partitions in binary problems may be sensible

The Empirical Study

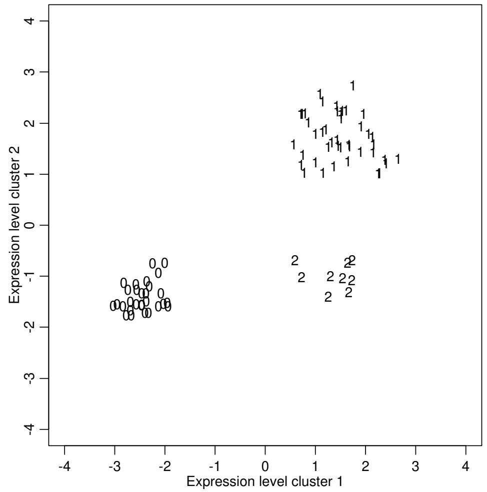
Datasets

- Leukemia data: AML/ALL distinction, 72 experiments with 3’571 genes each
- Lymphoma data: DLBCL/FL/CLL distinction, 62 individuals with 4’026 genes each
- NCI60 data: 8 distinct tumour types, 61 cell lines with 5’244 genes each

Questions

- What is the typical output?
- Is the output more than a noise artefact?
- Evaluation of the predictive potential
- What about the stability of the clusters?

Typical Output of the Supervised Clustering Algorithm



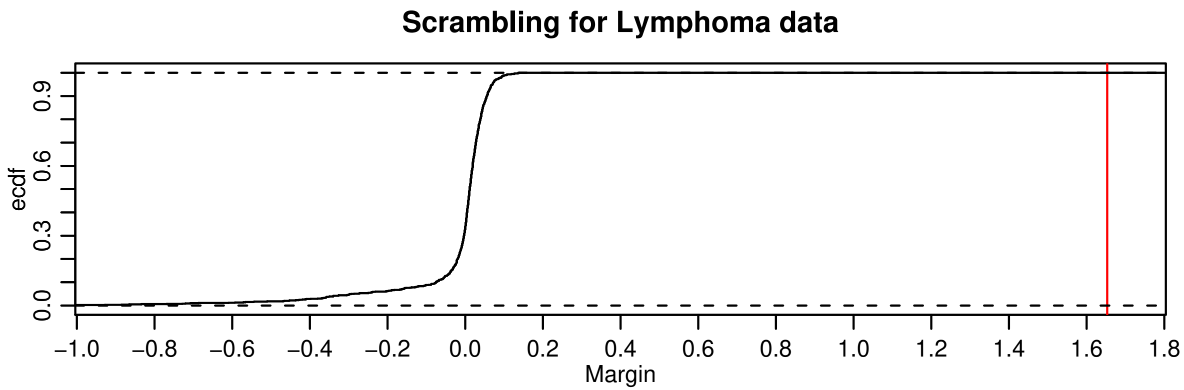
- Cluster size is typically 3–9 genes
- Very clear separation: zero scores and large margins
- Error-free classification of training data already with simple classifiers
- Error-free classification of test data if clusters are shaped on the entire dataset

Are the clusters more than a noise artefact?

- Idea: Compare real clusters to clusters from random noise gene expression data
- Method: Shuffling/Scrambling = “Random permutation of class labels”

$$(y_j)^* = (y_1^*, y_2^*, \ldots, y_n^*)$$

- Consider 1000 permutation trials and explore the empirical cdf of final margins



$\Rightarrow$ Clusters are definitely more than a noise artefact

Predictive Potential

Methods:

- Classifiers: Nearest Neighbors, Logistic Regression, Aggregated Trees
- Based on 100 random division in training and test data on a 2:1 scheme
- Cluster formation and classifier construction restricted to training data

We observe:

- Performance is depending on the number of clusters
- Better performance than with unclustered genes (where comparable)

Results:

$\Rightarrow$ E.g. 27% error rate in NCI dataset instead of ca. 42% as by Dudoit et al. (2001)

Stability of Gene Clusters (I)

Desired property for functional genomics, but difficult for statistical inference

$\rightsquigarrow$ Clusters $\mathcal{C}^*$ are shaped on bootstrapped gene expression data

$$\begin{pmatrix} y_{1^*} \\ x_{1,1^*} \\ \vdots \\ x_{p,1^*} \end{pmatrix}, \begin{pmatrix} y_{2^*} \\ x_{1,2^*} \\ \vdots \\ x_{p,2^*} \end{pmatrix}, \ldots, \begin{pmatrix} y_{n^*} \\ x_{1,n^*} \\ \vdots \\ x_{p,n^*} \end{pmatrix}$$

Repeat the procedure 1000 times and assess a confidence level to each gene i

$$\pi_i = \frac{N_i}{1000} = \frac{1}{1000} \cdot \sum_{\ell=1}^{1000} 1_{[gene\ i\ \in\ \mathcal{C}^{*(\ell)}]}, \quad i = 1, \ldots, p,$$

Stability of Gene Clusters (II)

Results:

- Between 5% and 18% (in different datasets) of genes ever enter a cluster
- Less than 0.5% of genes are used in more than 50 out of 1000 bootstrap trials

$\Rightarrow$ "reasonable" stability

Pairwise confidence levels:

$$\pi_{ij} = \frac{1}{1000} \cdot \sum_{\ell=1}^{L} 1_{[gene\ i\ \in\ \mathcal{C}^{*(\ell)}]} \cdot 1_{[gene\ j\ \in\ \mathcal{C}^{*(\ell)}]}, \quad i,j \in \{1,\ldots,p\}.$$

- Hypothesis testing: $H_0 : \pi_{ij} = \pi_i \pi_j$
- Useful in functional genomics to detect attraction/rejection among genes

$\Rightarrow$ Various cases of strongly significant attraction/rejection are present

Extensions & Open Questions

Averaging with general linear combinations:

$$x_{\mathcal{C}} = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \alpha_i x_i, \text{ with } \sum_i |\alpha_i| = 1$$

Impracticable during the clustering process $\rightsquigarrow$ try to optimize final cluster

$\Rightarrow$ Results in slight changes only and is time consuming

Biasing towards larger clusters:

Makes sense if goal is to have larger clusters, but does not improve them

Open Question:

How to generalize this type of algorithm to non-categorical response?

Conclusions

- Quick and fairly easy algorithm for supervised clustering of genes
- Produces reasonably stable clusters which are more than just noise artefacts and show better classification performance than single genes

  $\Rightarrow$ “It is worth the effort”
- Available as an R-Package from `dettling@stat.math.ethz.ch`