The Bioconductor Project: Open-source Statistical Software for the Analysis of Microarray Data

Sandrine Dudoit
Division of Biostatistics
University of California, Berkeley
www.stat.berkeley.edu/~sandrine

EMBO Practical Course on Analysis and Informatics of Microarray Data
Wellcome Trust Genome Campus, Hinxton, Cambridge, UK
March 18, 2003

Differential gene expression

Combining data across arrays

Data on *G* genes for *n* arrays

→ *G x n* genes-by-arrays data matrix

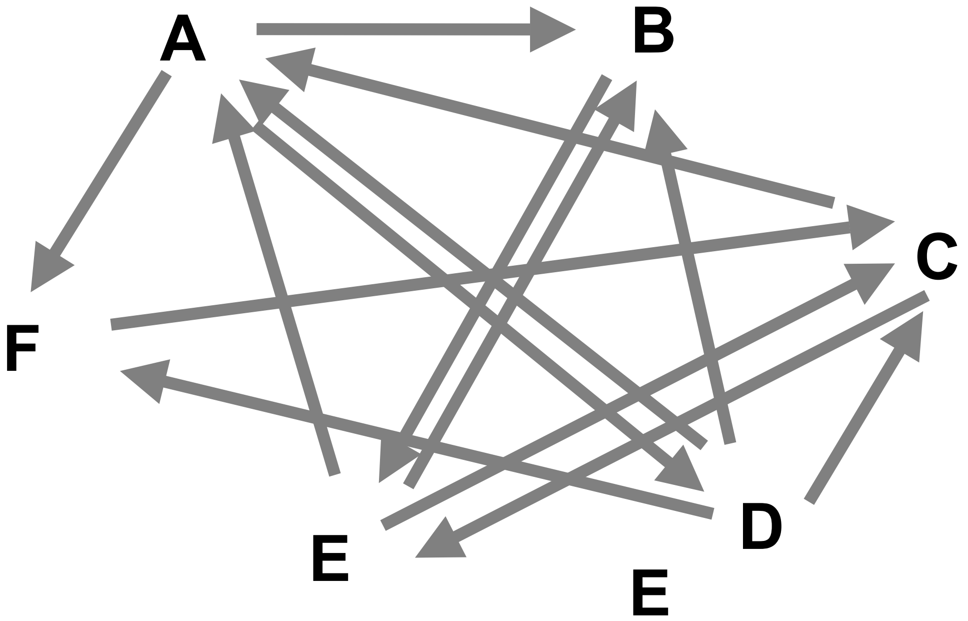
Arrays

		Array1	Array2	Array3	Array4	Array5	...
	Gene1	0.46	0.30	0.80	1.51	0.90	...
	Gene2	-0.10	0.49	0.24	0.06	0.46	...
Genes	Gene3	0.15	0.74	0.04	0.10	0.20	...
	Gene4	-0.45	-1.03	-0.79	-0.56	-0.32	...
	Gene5	-0.06	1.06	1.35	1.09	-1.09	...
	...	...	...	...	...	...	

M = $\log_2$(Red intensity / Green intensity)
expression measure, e.g. RMA.

Combining data across arrays

… but the columns have structure, determined by the experimental design.



Combining data across arrays

- *cDNA array factorial experiment*. Each column corresponds to a pair of mRNA samples with different drug x dose x time combinations.
- *Clinical trial.* Each column corresponds to a patient, with associated clinical outcome, such as survival and response to treatment.
- Linear models and extensions thereof can be used to effectively combine data across arrays for complex experimental designs.

Gene filtering

- A very common task in microarray data analysis is gene-by-gene selection.
- Filter genes based on
  - data quality criteria, e.g. absolute intensity or variance;
  - subject matter knowledge;
  - their ability to differentiate cases from controls;
  - their spatial or temporal expression pattern.
- Depending on the experimental design, some highly specialized filters may be required and applied sequentially.

Gene filtering

- *Clinical trial.* Filter genes based on association with survival, e.g. using a Cox model.
- *Factorial experiment.* Filter genes based on interaction between two treatments, e.g. using 2-way ANOVA.
- *Time-course experiment*. Filter genes based on periodicity of expression pattern, e.g. using Fourier transform.

genefilter package

- The genefilter package provides tools to sequentially apply filters to the rows (genes) of a matrix or of an instance of the **exprSet** class.
- There are two main functions, **filterfun** and **genefilter**, for assembling and applying the filters, respectively.
- Any number of functions for specific filtering tasks can be defined and supplied to **filterfun**.

  E.g. Cox model p-values, coefficient of variation.

`genefilter`: separation of tasks

1. Select/define functions for specific filtering tasks.
2. Assemble the filters using the **`filterfun`** function.
3. Apply the filters using the **`genefilter`** function → a logical vector, **`TRUE`** indicates genes that are retained.
4. Apply that vector to the **`exprSet`** to obtain a microarray object for the subset of interesting genes.

`genefilter`: supplied filters

Filters supplied in the package

- `kOverA` – select genes for which k samples have expression measures larger than A.
- `gapFilter` – select genes with a large IQR or gap (jump) in expression measures across samples.
- `ttest` – select genes according to t-test nominal p-values.
- `Anova` – select genes according to ANOVA nominal p-values.
- `coxfilter` – select genes according to Cox model nominal p-values.

`genefilter`: writing filters

- It is very simple to write your own filters.
- You can use the supplied filtering functions as templates.
- The basic idea is to rely on lexical scope to provide values (bindings) for the variables that are needed to do the filtering.

genefilter: How to?

1. First, build the filters

   ```
   f1 <- anyNA
   f2 <- kOverA(5, 100)
   ```

2. Next, assemble them in a filtering function

   ```
   ff <- filterfun(f1,f2)
   ```

3. Finally, apply the filter

   ```
   wh <- genefilter(marrayDat, ff)
   ```

4. Use `wh` to obtain the relevant subset of the data

   ```
   mySub <- marrayDat[wh,]
   ```

Differential gene expression

- Identify genes whose expression levels are associated with a response or covariate of interest
  - clinical outcome such as survival, response to treatment, tumor class;
  - covariate such as treatment, dose, time.
- Estimation: estimate effects of interest and variability of these estimates.

  E.g. slope, interaction, or difference in means in a linear model.
- Testing: assess the statistical significance of the observed associations.

Multiple hypothesis testing

- Large multiplicity problem: thousands of hypotheses are tested simultaneously!
  - Increased chance of false positives.
  - E.g. chance of at least one p-value < α for G independent tests is $1-(1-\alpha)^G$

    and converges to one as G increases.

    For G=1,000 and $\alpha = 0.01$, this chance is 0.9999568!
  - Individual p-values of 0.01 no longer correspond to significant findings.

- Need to adjust for multiple testing when assessing the statistical significance of the observed associations.

Multiple hypothesis testing

- Define an appropriate Type I error or false positive rate.
- Develop multiple testing procedures that
  - provide strong control of this error rate,
  - are powerful (few false negatives),
  - take into account the joint distribution of the test statistics.
- Report adjusted p-values for each gene which reflect the overall Type I error rate for the experiment.
- Resampling methods are useful tools to deal with the unknown joint distribution of the test statistics.

multtest package

- Multiple testing procedures for controlling
  - Family-Wise Error Rate - FWER: Bonferroni, Holm (1979), Hochberg (1986), Westfall & Young (1993) maxT and minP;
  - False Discovery Rate - FDR: Benjamini & Hochberg (1995), Benjamini & Yekutieli (2001).
- Tests based on t- or F-statistics for one- and two-factor designs.
- Permutation procedures for estimating adjusted p-values.
- Fast permutation algorithm for minP adjusted p-values.
- Documentation: tutorial on multiple testing.

Clustering and classification

Clustering vs. classification

- Cluster analysis (a.k.a. unsupersived learning)
  - the classes are unknown a priori;
  - the goal is to discover these classes from the data.
- Classification (a.k.a. class prediction, supervised learning)
  - the classes are predefined;
  - the goal is to understand the basis for the classification from a set of labeled objects and build a predictor for future unlabeled observations.

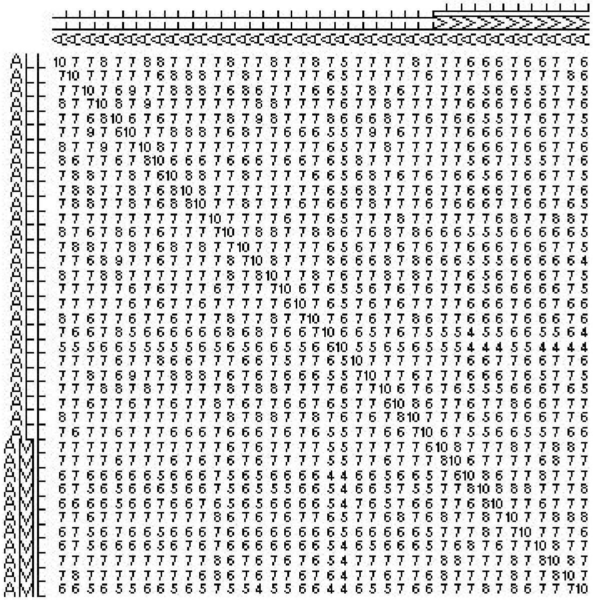
Distances

- Microarray data analysis often involves
  - clustering genes or samples;
  - classifying genes or samples.
- Both types of analyses are based on a measure of distance (or similarity) between genes or samples.
- R has a number of functions for computing and plotting distance and similarity matrices.

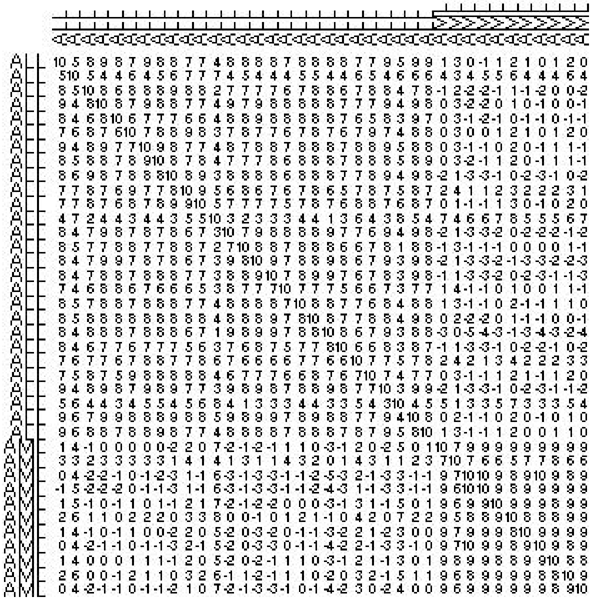
Distances

- Distance functions
  - **`dist`** (**`mva`**): Euclidean, Manhattan, Canberra, binary;
  - **`daisy`** (**`cluster`**).
- Correlation functions
  - **`cor`**, **`cov.wt`**.
- Plotting functions
  - **`image`**;
  - **`plotcorr`** (**`ellipse`**);
  - **`plot.cor`**, **`plot.mat`** (**`sma`**).

Correlation matrices



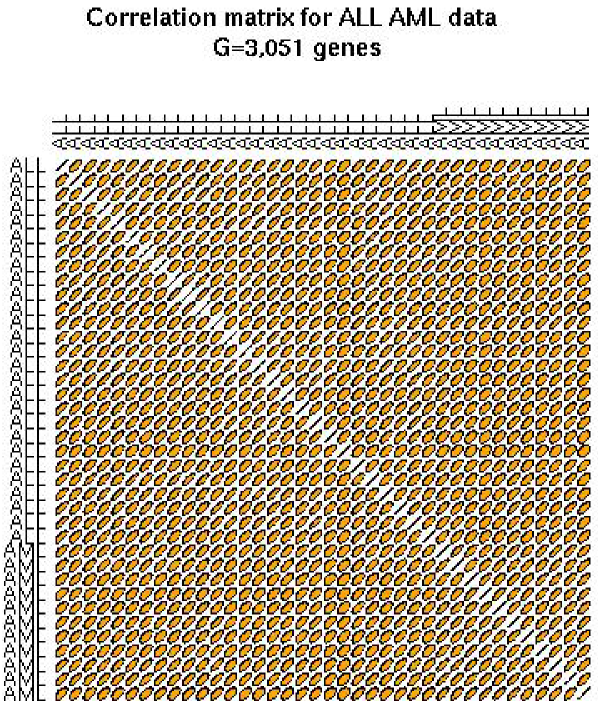
Correlation matrix for ALL AML data
G=3,051 genes

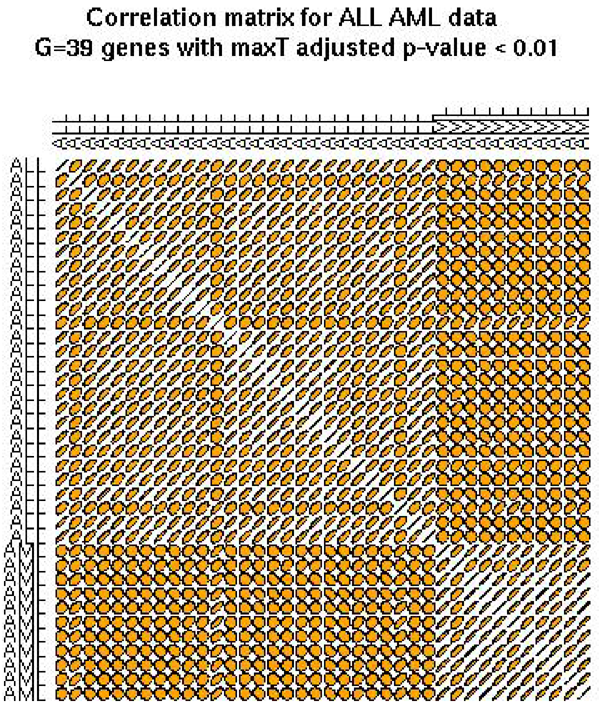


Correlation matrix for ALL AML data
G=39 genes with maxT adjusted p-value < 0.01

`plotcorr` function from **`ellipse`** package

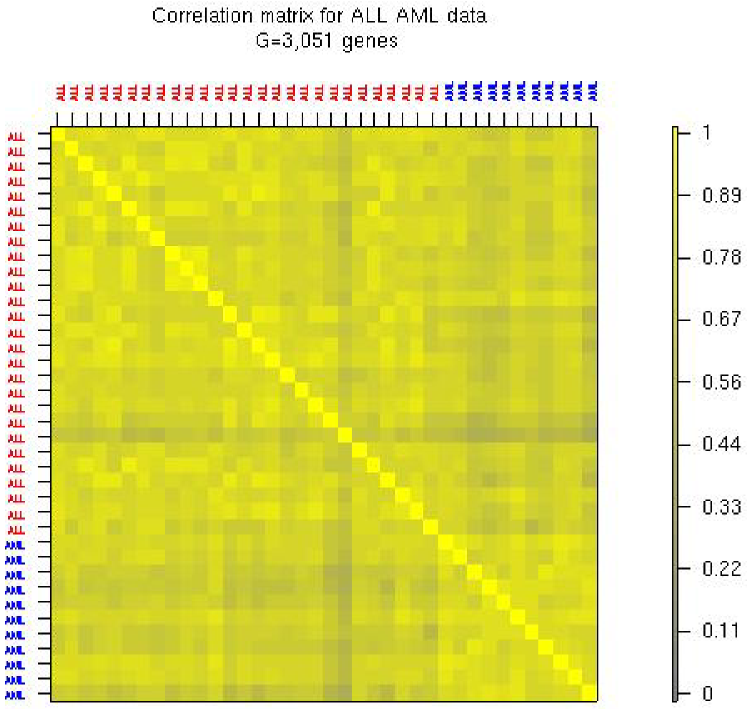
Correlation matrices



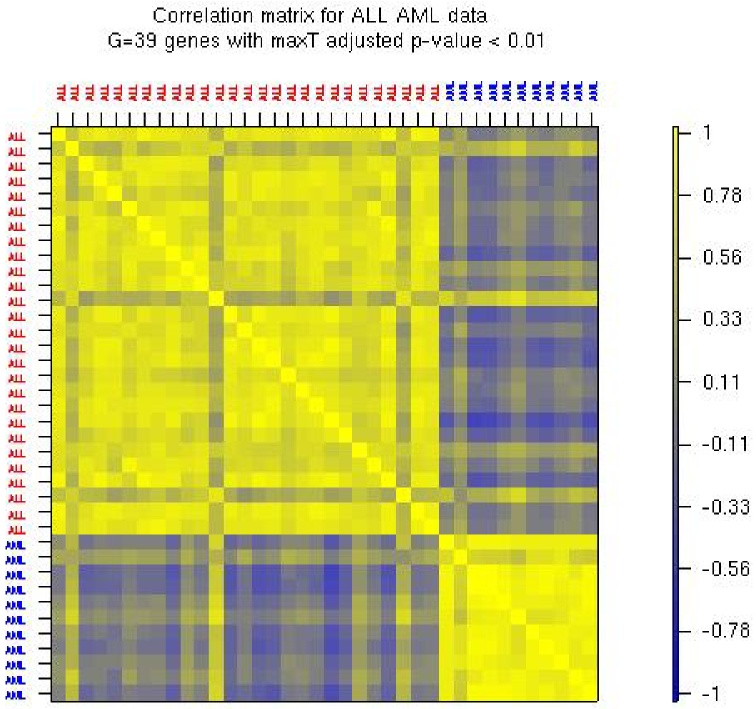


`plotcorr` function from `ellipse` package

Correlation matrices



Correlation matrix for ALL AML data
G=3,051 genes



Correlation matrix for ALL AML data
G=39 genes with maxT adjusted p-value < 0.01

`plot.cor` function from `sma` package

Multidimensional scaling

- Given any n x n dissimilarity matrix D, multidimensional scaling (MDS) is concerned with identifying n points in Euclidean space with a similar distance structure D'.
- The purpose is to provide a lower dimensional representation of the distances which conveys information on the relationships between the n objects, such as the existence of clusters or one-dimensional structure in the data (e.g., seriation).

MDS

- There are different approaches for reducing dimensionality, depending on how we define similarity between the old and new dissimilarity matrices for the n objects, i.e., depending on the objective or stress function S that we seek to minimize.
  - Least-squares scaling $S(D,D')=\left(\sum(d_{ij}-d'_{ij})^2\right)^{1/2}$
  - Samming mapping $S(D,D')=\sum(d_{ij}-d'_{ij})^2/d_{ij}$

    places more emphasis on smaller dissimilarities (and hence should be preferred for clustering methods).
  - Shepard-Kruskal non-metric scaling is based on ranks, i.e., the order of the distances is more important than their actual values.

MDS and PCA

- When the distance matrix D is the Euclidean distance matrix between the rows of an n x m matrix X, there is a duality between principal component analysis (PCA) and MDS.
- The k-dimensional classical solution to the MDS problem is given by the centered scores of the n objects on the first k principal components.
- The classical solution of MDS in k-dimensional space minimizes the sum of squared differences between the entries of the new and old dissimilarity matrices, i.e., is optimal for least-squares scaling.

MDS

- As with PCA, the quality of the representation will depend on the magnitude of the first k eigenvalues.
- The data analyst should choose a value for k that is small enough for ease representation but also corresponds to a substantial “proportion of the distance matrix explained”.

MDS

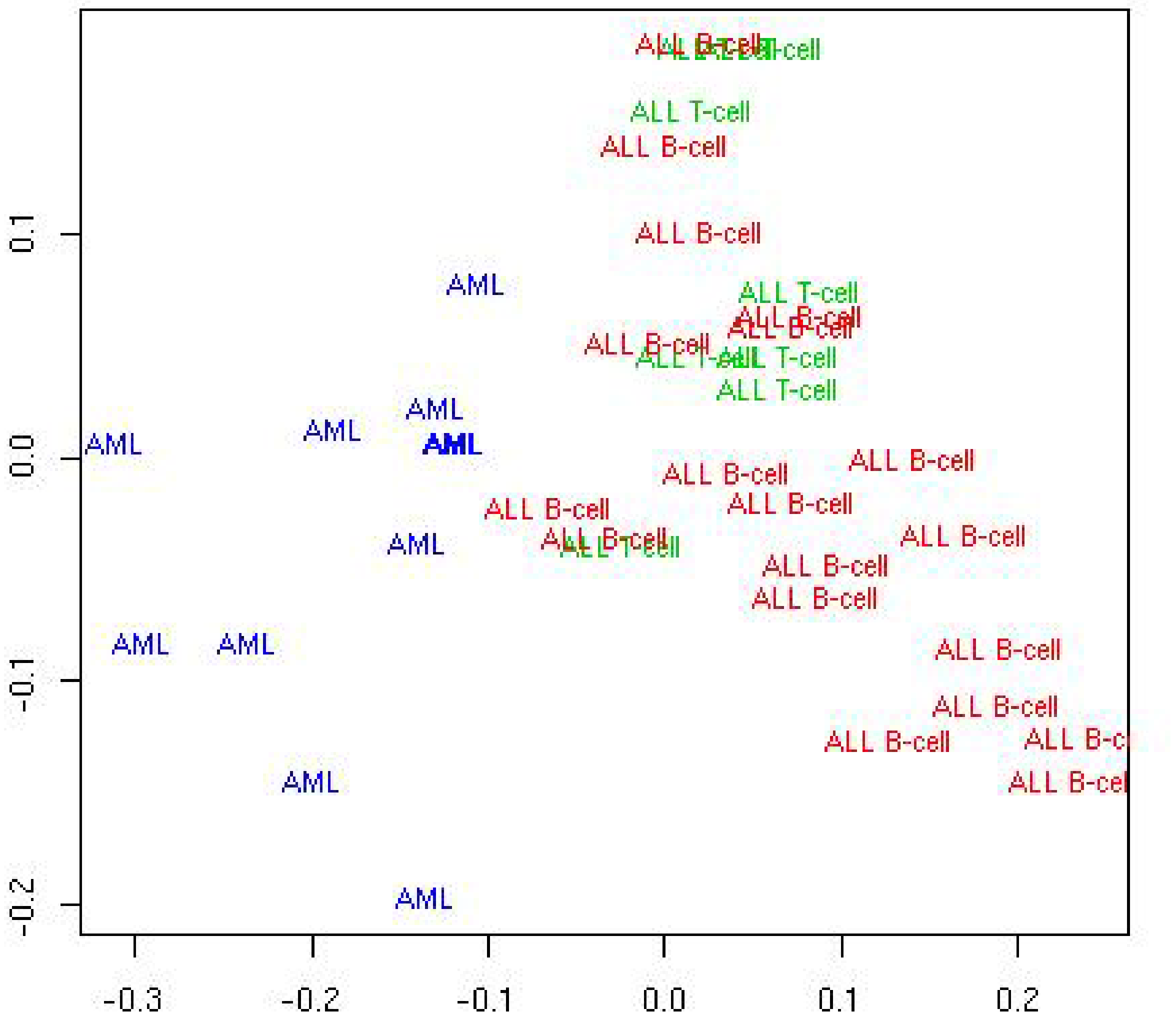
- **N.B.** The MDS solution reflects not only the choice of a distance function, but also the features selected.
- If features were selected to separate the data into two groups (e.g., on the basis of two-sample t-statistics), it should come as no surprise that an MDS plot has two groups. In this instance MDS is not a confirmatory approach.

R MDS software

- **`cmdscale`**: Classical solution to MDS, in package **`mva`**.
- **`sammon`**: Sammon mapping, in package **`MASS`**.
- **`isoMDS`**: Kruskal's non-metric MDS, in package **`MASS`**.

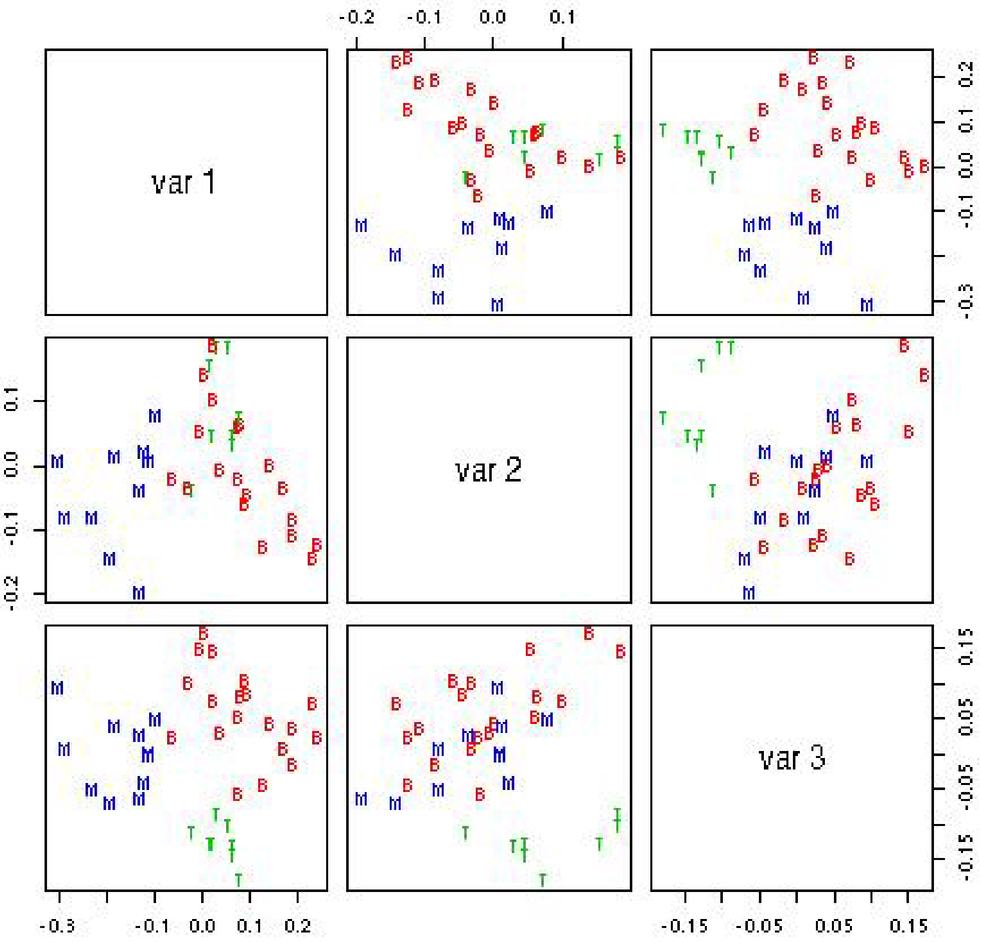
Classical MDS

MDS for ALL AML data, correlation matrix, G=3,051 genes, k=2



Classical MDS

MDS for ALL AML data, correlation matrix, G=3,051 genes, k=3



$$\frac{|\lambda_1|+|\lambda_2|}{\sum|\lambda_i|} = 43\%$$

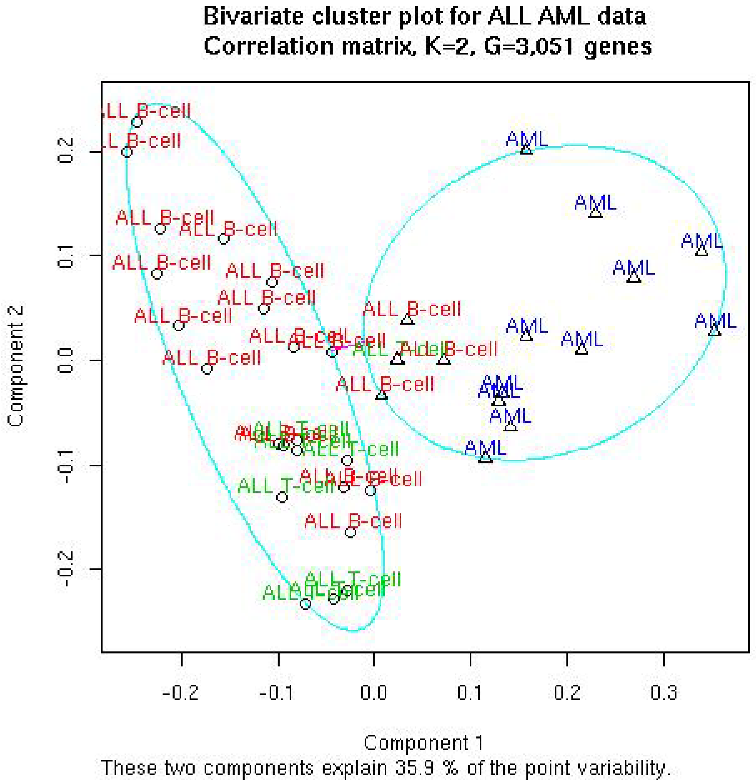
$$\frac{|\lambda_1|+|\lambda_2|+|\lambda_3|}{\sum|\lambda_i|} = 55\%$$

Cluster analysis packages

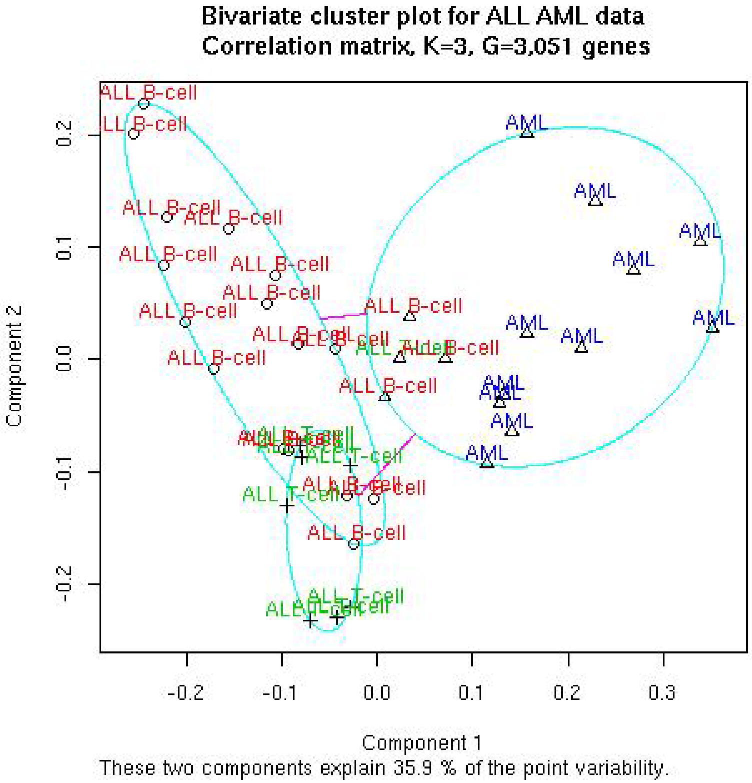
- **`class`**: self organizing maps (**`SOM`**).
- **`cluster`**:
  - AGglomerative NESting (**`agnes`**),
  - Clustering LARe Applications (**`clara`**),
  - DIvisive ANAlysis (**`diana`**),
  - Fuzzy Analysis (**`fanny`**),
  - MONothetic Analysis (**`mona`**),
  - Partitioning Around Medoids (**`pam`**).
- **`e1071`**:
  - fuzzy C-means clustering (**`cmeans`**),
  - bagged clustering (**`bclust`**).
- **`mva`**:
  - hierarchical clustering (**`hclust`**),
  - k-means (**`kmeans`**).
- Specialized summary, plot, and print methods for clustering results.

pam

K=2


Bivariate cluster plot for ALL AML data
Correlation matrix, K=2, G=3,051 genes

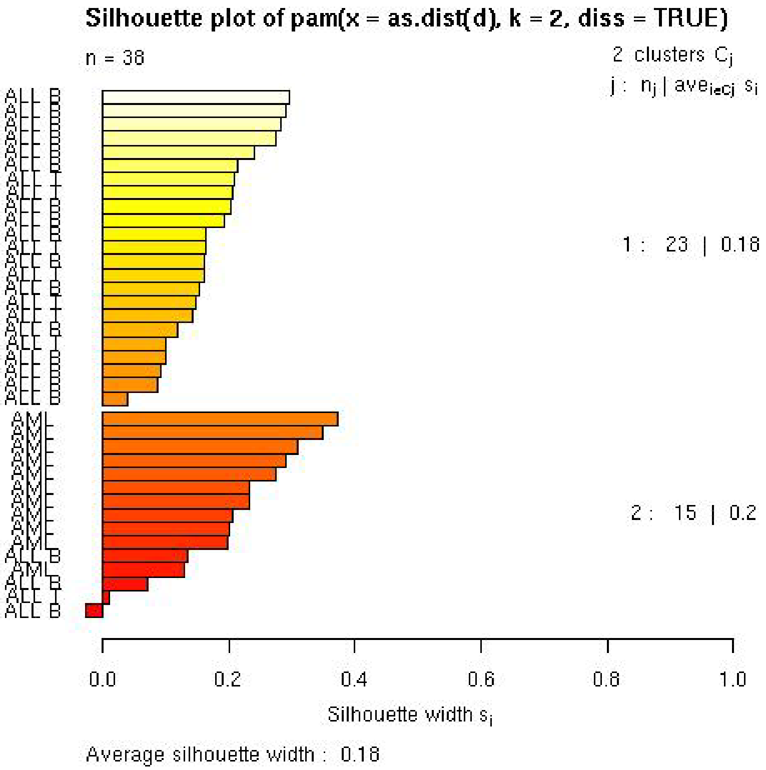
K=3


Bivariate cluster plot for ALL AML data
Correlation matrix, K=3, G=3,051 genes

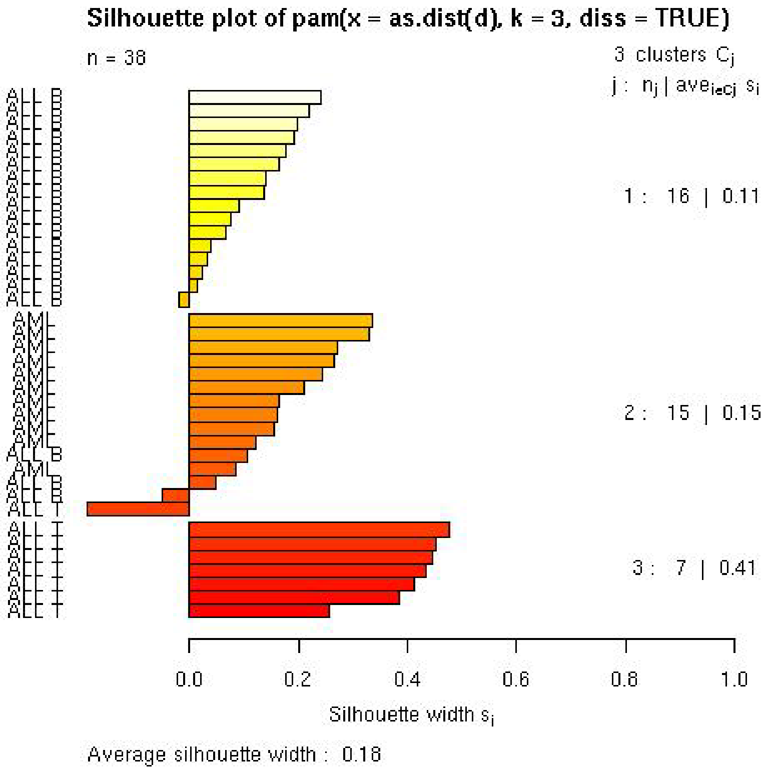
`pam` and `clusplot` functions from `cluster` package

pam

K=2

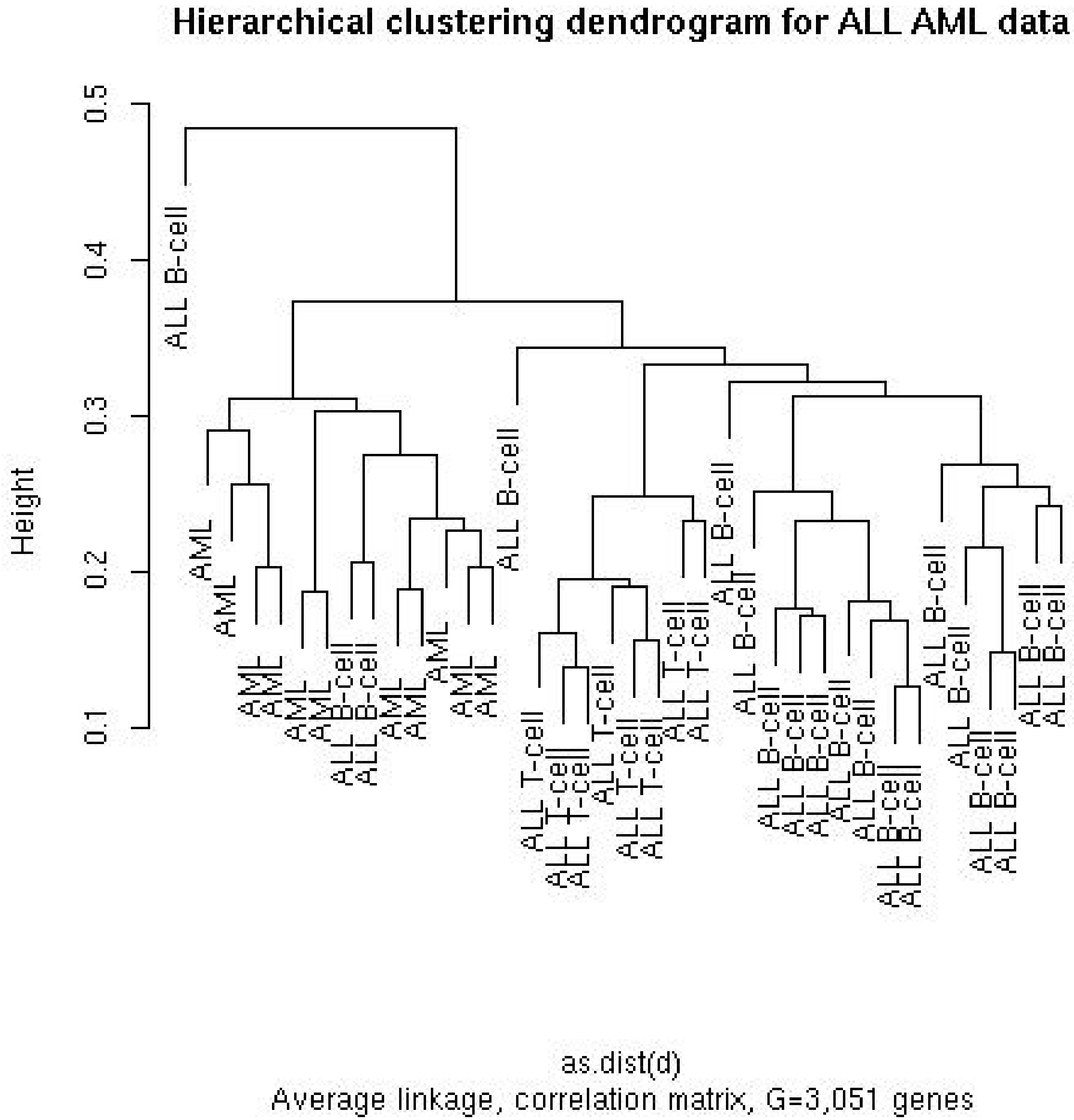


K=3



`pam` and **`plot`** functions from **`cluster`** package

hclust



hclust function from **mva** package

Dendrogram

- **N.B.** While dendrograms are quite appealing because of their apparent ease of interpretation, they can be misleading.
- First, the dendrogram corresponding to a given hierarchical clustering is not unique, since for each merge one needs to specify which subtree should go on the left and which on the right --- there are 2^(n-1) choices.
- The default in the R function **`hclust`** is to order the subtrees so that the tighter cluster is on the left.

Dendrogram

- Second, they *impose* structure on the data, instead of *revealing* structure in these data.
- Such a representation will be valid only to the extent that the pairwise dissimilarities possess the hierarchical structure imposed by the clustering algorithm.

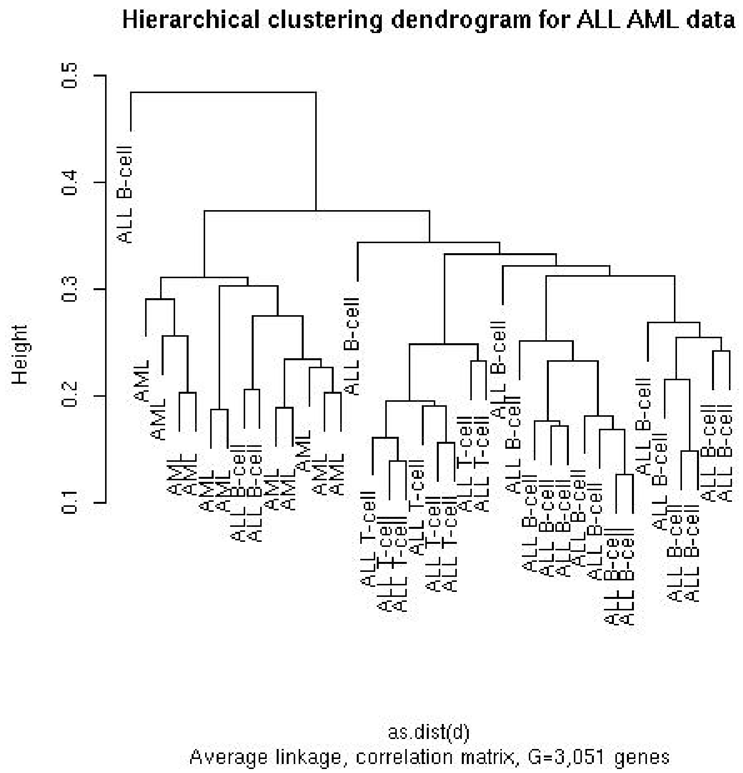
Dendrogram

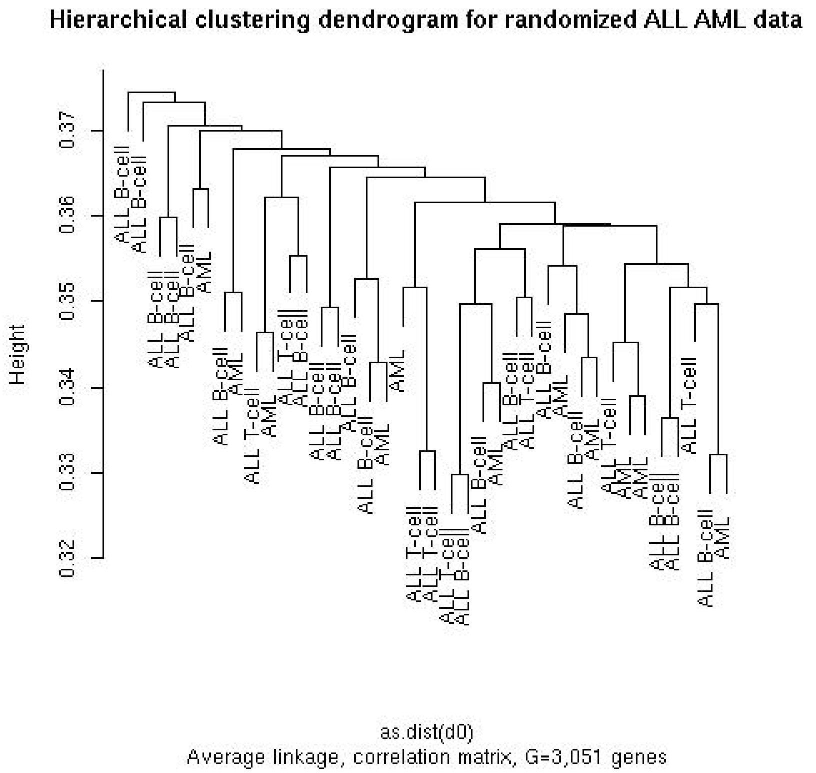
- The cophenetic correlation coefficient can be used to measure how well the hierarchical structure from the dendrogram represents the actual distances.
- This measure is defined as the correlation between the n(n-1)/2 pairwise dissimilarities between observations and their cophenetic dissimilarities from the dendrogram, i.e., the between cluster dissimilarities at which two observations are first joined together in the same cluster.
- Function **`cophenetic`** in **`mva`** package.

Dendrogram

Original data,
coph corr = 0.74

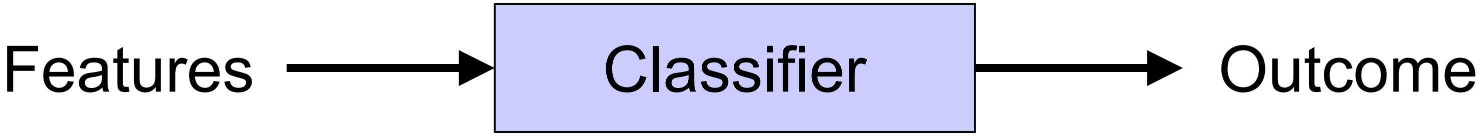
Randomized data
(perm. wi features),
coph corr = 0.57





Classification

- Predict a biological outcome on the basis of observable features.



- **Outcome**: tumor class, type of bacterial infection, survival, response to treatment.
- **Features**: gene expression measures, covariates such as age, sex.

Classification

- Old and extensive literature on classification, in statistics and machine learning.
- Examples of classifiers
  - nearest neighbor classifiers (k-NN);
  - discriminant analysis: linear, quadratic, logistic;
  - neural networks;
  - classification trees;
  - support vector machines.
- Aggregated classifiers: bagging and boosting.
- Comparison on microarray data:

  simple classifiers like k-NN and naïve Bayes perform remarkably well.

Performance assessment

- Classification error rates, or related measures, are usually reported
  - to compare the performance of different classifiers;
  - to support statements such as

    "*clinical outcome X for cancer Y can be predicted accurately based on gene expression measures*".

- Classification error rates can be estimated by resampling, e.g. bootstrap or cross-validation.

Performance assessment

- It is essential to take into account feature selection and other training decisions in the error rate estimation process.

  E.g. number of neighbors in k-NN, kernel in SVMs.

- Otherwise, error estimates can be severely biased downward, i.e., overly optimistic.

Important issues

- Standardization;
- Distance function;
- Feature selection;
- Loss function;
- Class priors;
- Binary vs. polychotomous classification.

Classification packages

- **`class`**:
  - k-nearest neighbor (**`knn`**),
  - learning vector quantization (**`lvq`**).
- **`e1071`**: support vector machines (**`svm`**).
- **`ipred`**: bagging, resampling based estimation of prediction error.
- **`LogitBoost`**: boosting for tree stumps.
- **`MASS`**: linear and quadratic discriminant analysis (**`lda`**, **`qda`**).
- **`mlbench`**: machine learning benchmark problems.
- **`nnet`**: feed-forward neural networks and multinomial log-linear models.
- **`ranForest`**, **`RanForests`**: random forests.
- **`rpart`**: classification and regression trees.
- **`sma`**: diagonal linear and quadratic discriminant analysis, naïve Bayes (**`stat.diag.da`**).