Annotation and Analysis

Sandrine Dudoit, Robert Gentleman, and Rafael Irizarry

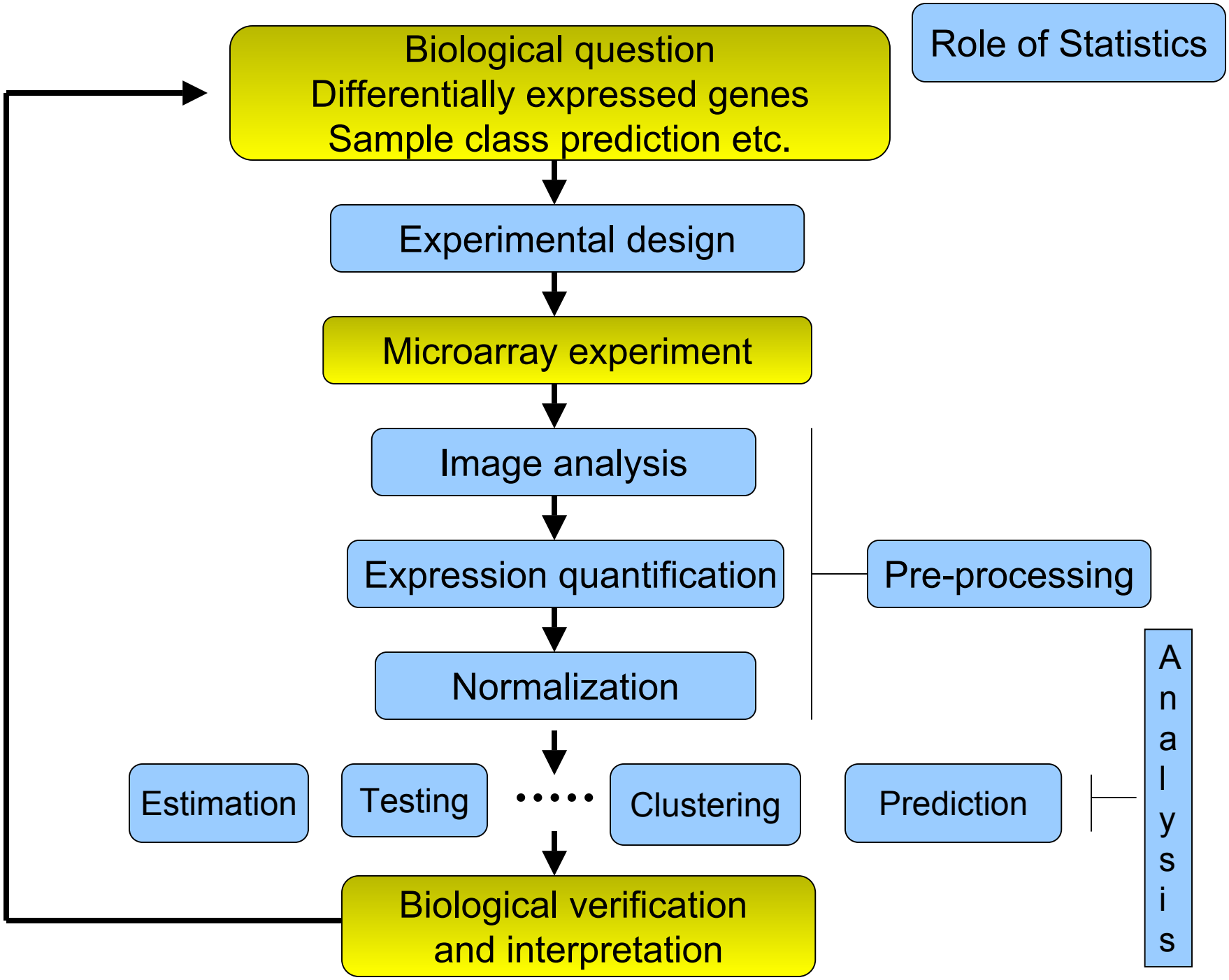
Bioconductor Workshop

JHMI Microarray Core Facility

October 28-29, 2002

Acknowledgements

- **Bioconductor core team**
- **Ben Bolstad**, Biostatistics, UC Berkeley
- **Vincent Carey**, Biostatistics, Harvard
- **Francois Collin**, GeneLogic
- **Leslie Cope**, JHU
- **Laurent Gautier**, Technical University of Denmark, Denmark
- **Yongchao Ge**, Statistics, UC Berkeley
- **Robert Gentleman**, Biostatistics, Harvard
- **Jeff Gentry**, Dana-Farber Cancer Institute
- **John Ngai Lab**, MCB, UC Berkeley
- **Juliet Shaffer**, Statistics, UC Berkeley
- **Terry Speed**, Statistics, UC Berkeley
- **Yee Hwa (Jean) Yang**, Biostatistics, UCSF
- **Jianhua (John) Zhang**, Dana-Farber Cancer Institute
- Spike-in and dilution datasets:
  - **Gene Brown's group**, Wyeth/Genetics Institute
  - **Uwe Scherf's group**, Genomics Research & Development, GeneLogic.
- **GeneLogic** and **Affymetrix** for permission to use their data.


Role of Statistics
Biological question
Differentially expressed genes
Sample class prediction etc.
Experimental design
Microarray experiment
Image analysis
Expression quantification
Normalization
Pre-processing
Estimation
Testing
.....
Clustering
Prediction
Analysis
Biological verification
and interpretation

Bioconductor packages

Release 1.0, May 2nd, 2002

- General infrastructure:
  `Biobase`, `rhdf5`, `tkWidgets`.
- Annotation:
  `annotate`, `AnnBuilder` → data packages.
- Graphics:
  `geneplotter`.
- Pre-processing for Affymetrix oligonucleotide chip data:
  `affy`.
- Pre-processing for cDNA microarray data:
  `marrayClasses`, `marrayInput`, `marrayNorm`, `marrayPlots`.
- Differential gene expression:
  `edd`, `genefilter`, `multtest`, `ROC`.

References

- Consult the slides from the Short Course, *Statistical Methods and Software for the Analysis of DNA Microarray Experiments* (Summer 2002),

  www.bioconductor.org/workshops/Summer02Course/

  for a more detailed discussion of pre-processing, experimental design, multiple testing, distances, cluster analysis, and classification

Outline

- **`annotate`** and **`AnnBuilder`** packages
- **`genefilter`** package
- **`multtest`** package
- R clustering and classification packages

Annotation packages

- One of the largest challenges in analyzing genomic data is associating the experimental data with the available metadata, e.g. sequence, gene annotation, chromosomal maps, literature.
- The `annotate` and `AnnBuilder` packages provides some tools for carrying this out.
- These are very likely to change, evolve and improve, so please check the current documentation - things may already have changed!

Annotation packages

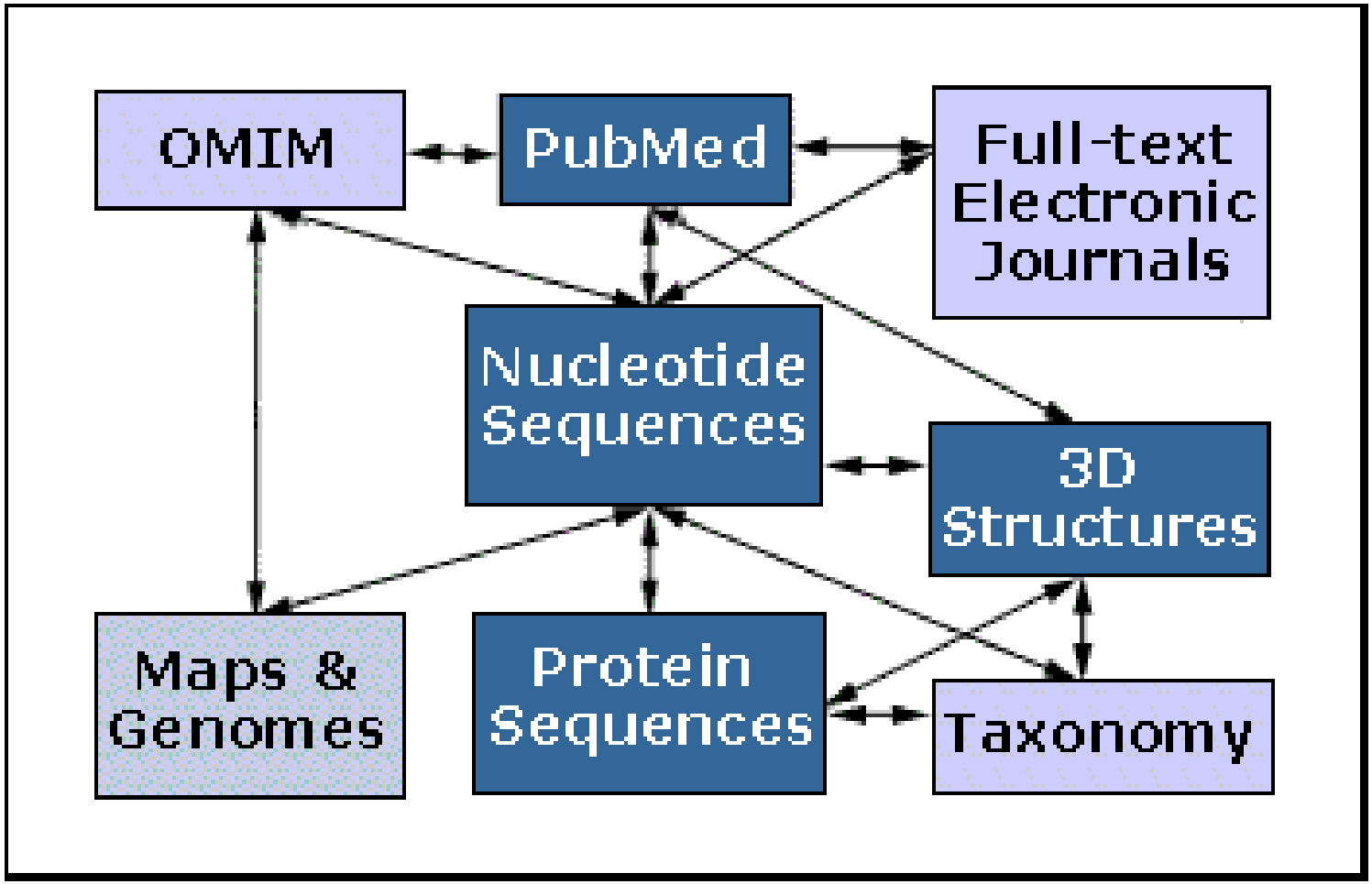
- Annotation data packages;
- Matching IDs using environments;
- Searching and processing queries from WWW databases
  - LocusLink,
  - GenBank,
  - PubMed;
- HTML reports.

WWW resources

- Nucleotide databases: e.g. GenBank.
- Gene databases: e.g. LocusLink, UniGene.
- Protein sequence and structure databases: e.g. SwissProt, Protein DataBank (PDB).
- Literature databases: e.g. PubMed, OMIM.
- Chromosome maps: e.g. NCBI Map Viewer.
- Pathways: e.g. KEGG.
- Entrez is a search and retrieval system that integrates information from databases at NCBI (National Center for Biotechnology Information).

NCBI Entrez

www.ncbi.nlm.nih.gov/Entrez



`annotate`: matching IDs

Important tasks

- Associate manufacturers probe identifiers (e.g. Affymetrix IDs) to other available identifiers (e.g. gene symbol, PubMed PMID, LocusLink LocusID, GenBank accession number).
- Associate probes with biological data such as chromosomal position, pathways.
- Associate probes with published literature data via PubMed.

`annotate`: matching IDs

Affymetrix identifier HGU95A chips	**"41046_s_at"**
LocusLink, LocusID	**"9203"**
GenBank accession #	**"X95808"**
Gene symbol	**"ZNF261"**
PubMed, PMID	**"10486218"** **"9205841"** **"8817323"**
Chromosomal location	**"X", "Xq13.1"**

Annotation data packages

- The Bioconductor project has started to deploy packages that contain only data.

  E.g. **`hgu95a`** package for Affymetrix HGU95A GeneChips series, also, **`hgu133a`**, **`hu6800`**, **`mgu74a`**, **`rgu34a`**.
- These data packages are built using **`AnnBuilder`**.
- These packages contain many different mappings to interesting data.
- They are available from the Bioconductor website and also using **`update.packages`**.

Annotation data packages

- Maps to GenBank accession number, LocusLink LocusID, gene symbol, gene name, UniGene cluster.
- Maps to chromosomal location: chromosome, cytoband, physical distance (bp), orientation.
- Maps to KEGG pathways, enzymes, Gene Ontology Consortium (GO).
- Maps to PubMed PMID.
- These packages will be updated and expanded regularly as new or updated data become available.

hu6800 data package

R: A data package for hu6800 - Netscape 6

File Edit View Search Go Bookmarks Tasks Help

file:///C:/Sandrine/Programs/rw1051/library/hu6800/html/00Index.html

Home Netscape Search Shop Bookmarks Net2Phone

A data package for hu6800

hu6800	A funciton to return a vector of rda file names
hu6800ACCNUM	Annotation data file for hu6800 on ACCNUM
hu6800AFFYCOUNTS	Annotation data file for GOByNum on AFFYCOUNTS
hu6800CHR	Annotation data file for hu6800 on CHR
hu6800CHRLOC	Annotation data file for hu6800 on CHRLOC
hu6800CHRORI	Annotation data file for hu6800 on CHRORI
hu6800ENZYME	Annotation data file for hu6800 on ENZYME
hu6800ENZYME2AFFY	Annotation data file for hu6800 on ENZYME2AFFY
hu6800GENENAME	Annotation data file for hu6800 on GENENAME
hu6800GO	Annotation data file for hu6800 on GO
hu6800GO2AFFY	Annotation data file for GOByNum on GO2AFFY
hu6800GO2ALLAFFY	Annotation data file for GOByNum on GO2ALLAFFY
hu6800GRIF	Annotation data file for hu6800 on GRIF
hu6800LOCUSID	Annotation data file for hu6800 on LOCUSID
hu6800MAP	Annotation data file for hu6800 on MAP
hu6800PATH	Annotation data file for hu6800 on PATH
hu6800PATH2AFFY	Annotation data file for hu6800 on PATH2AFFY
hu6800PMID	Annotation data file for hu6800 on PMID
hu6800PMID2AFFY	Annotation data file for hu6800 on PMID2AFFY
hu6800SUMFUNC	Annotation data file for hu6800 on SUMFUNC
hu6800SYMBOL	Annotation data file for hu6800 on SYMBOL
hu6800UNIGENE	Annotation data file for hu6800 on UNIGENE

Document: Done (0.24 secs)

`annotate`: matching IDs

- Much of what `annotate` does relies on matching symbols.
- This is basically the role of a hash table in most programming languages.
- In R, we rely on environments (they are similar to hash tables).
- The annotation data packages provide R environment objects containing key and value pairs for the mappings between two sets of probe identifiers.
- Keys can be accessed using the R `ls` function.
- Matching values in different environments can be accessed using the `get` or `multiget` functions.

`annotate`: matching IDs

E.g. `hgu95a` package.

- To load package `library(hgu95a)`
- For info on the package and list of mappings available

```
? hgu95a
hgu95a()
```

- For info on a particular mapping

```
? hgu95aPMID
```

annotate: matching IDs

```
> library(hgu95a)
> get("41046_s_at", env = hgu95aACCNUM)
[1] "X95808"
> get("41046_s_at", env = hgu95aLOCUSID)
[1] "9203"
> get("41046_s_at", env = hgu95aSYMBOL)
[1] "ZNF261"
> get("41046_s_at", env = hgu95aGENENAME)
[1] "zinc finger protein 261"
> get("41046_s_at", env = hgu95aSUMFUNC)
[1] "Contains a putative zinc-binding
  motif (MYM)|Proteome"
> get("41046_s_at", env = hgu95aUNIGENE)
[1] "Hs.9568"
```

annotate: matching IDs

```
> get("41046_s_at", env = hgu95aCHR)
[1] "X"
> get("41046_s_at", env = hgu95aCHRLOC)
[1] "66457019@X"
> get("41046_s_at", env = hgu95aCHRORI)
[1] "-@X"
> get("41046_s_at", env = hgu95aMAP)
[1] "Xq13.1”
> get("41046_s_at", env = hgu95aPMID)
[1] "10486218" "9205841"  "8817323"
> get("41046_s_at", env = hgu95aGO)
[1] "GO:0003677" "GO:0007275"
```

`annotate`: database searches and report generation

- Provide tools for searching and processing information from various biological databases.
- Provide tools for regular expression searching of PubMed abstracts.
- Provide nice HTML reports of analyses, with links to biological databases.

`annotate`: WWW queries

- Functions for querying WWW databases from R rely on the **`browseURL`** function

```
browseURL("www.r-project.org")
```

`annotate`: GenBank query

www.ncbi.nlm.nih.gov/Genbank/index.html

- Given a vector of GenBank accession numbers or NCBI UIDs, the **`genbank`** function
  - opens a browser at the URLs for the corresponding GenBank queries;
  - returns an **`XMLdoc`** object with the same data.

```
genbank("X95808",disp="browser")
```

http://www.ncbi.nih.gov/entrez/query.fcgi?tool=bioconductor&cmd=Search&db=Nucleotide&term=X95808

```
genbank(1430782,disp="data",
   type="uid")
```

`annotate`: LocusLink query

www.ncbi.nlm.nih.gov/LocusLink/

- **`locuslinkByID`**: given one or more LocusIDs, the browser is opened at the URL corresponding to the first gene.

  `locuslinkByID("9203")`

  http://www.ncbi.nih.gov/LocusLink/LocRpt.cgi?l=9203

- **`locuslinkQuery`**: given a search string, the results of the LocusLink query are displayed in the browser.

  `locuslinkQuery("zinc finger")`

  http://www.ncbi.nih.gov/LocusLink/list.cgi?Q=zinc finger&ORG=Hs&V=0

`annotate`: PubMed query

www.ncbi.nlm.nih.gov

- For any gene there is often a large amount of data available from PubMed.
- The `annotate` package provides the following tools for interacting with PubMed
  - `pubMedAbst`: a class structure for PubMed abstracts in R.
  - `pubmed`: the basic engine for talking to PubMed.
- WARNING: be careful you can query them too much and be banned!

`annotate`: `pubMedAbst` class

Class structure for storing and processing PubMed abstracts in R

- **`authors`**
- **`abstText`**
- **`articleTitle`**
- **`journal`**
- **`pubDate`**
- **`abstUrl`**

`annotate`: high level tools for PubMed query

- `pm.getabst`: download the specified PubMed abstracts (stored in XML) and create a list of **`pubMedAbst`** objects.
- `pm.titles`: extract the titles from a set of PubMed abstracts.
- `pm.abstGrep`: regular expression matching on the abstracts.

annotate: PubMed example

```
pmid <-get("41046_s_at", env=hgu95aPMID)
pubmed(pmid, disp="browser")
```

http://www.ncbi.nih.gov/entrez/query.fcgi?tool=bioconductor&cmd=Retrieve&db=PubMed&list_uids=10486218%2c9205841%2c8817323

```
absts <- pm.getabst("41046_s_at",
  base="hgu95a")
pm.titles(absts)
pm.abstGrep("retardation",absts[[1]])
```

annotate: PubMed example

RGui - [R Console]

File Edit Misc Packages Windows Help

```
Slot "articleTitle":
[1] "Prediction of the coding sequences of unidentified human genes. VII. The complete sequences of 100 new cDNA clones from brain which can$

Slot "journal":
[1] "DNA Res"

Slot "pubDate":
[1] "Apr 1997"

Slot "abstUrl":
[1] "No URL Provided"


[[3]]
An object of class "pubMedAbst"
Slot "authors":
[1] "S M SM van der Maarel" "I H IH Scholten"       "I I Huber"             "C C Philippe"          "R F RF Suijkerbuijk"
[6] "S S Gilgenkrantz"      "J J Kere"              "F P FP Cremers"        "H H HH Ropers"

Slot "abstText":
[1] "In several families with non-specific X-linked mental retardation (XLMR) linkage analyses have assigned the underlying gene defect to t$

Slot "articleTitle":
[1] "Cloning and characterization of DXS6673E, a candidate gene for X-linked mental retardation in Xq13.1."

Slot "journal":
[1] "Hum Mol Genet"

Slot "pubDate":
[1] "Jul 1996"

Slot "abstUrl":
[1] "No URL Provided"


> pm.titles(absts)
[[1]]
[1] "Cloning and mapping of members of the MYM family."                                                                          $
[2] "Prediction of the coding sequences of unidentified human genes. VII. The complete sequences of 100 new cDNA clones from brain which can$
[3] "Cloning and characterization of DXS6673E, a candidate gene for X-linked mental retardation in Xq13.1."                      $

> pm.abstGrep("retardation",absts[[1]])
[1]  TRUE FALSE  TRUE
>
```

R 1.5.1 - A Language and Environment

`annotate`: data rendering

- A simple interface, `ll.htmlpage`, can be used to generate an HTML report of your results.
- The page consists of a table with one row per gene, with links to LocusLink.
- Entries can include various gene identifiers and statistics.

BioConductor Gene Listing

Golub et al. data, genes with permutation maxT adjusted p-value < 0.01

Locus Link Genes

LocusID	Gene name	Chromosome	ALL mean	AML mean	t-statistic	raw p-value	adj p-value
7791	X95735_at	7	-0.295	1.59	-10.6	2e-05	2e-05
1471	M27891_at	20	-0.81	2.08	-9.78	2e-05	2e-05
2184	M55150_at	15	0.488	1.24	-8.03	2e-05	0.00014
4067	M16038_at	8	-0.284	1.1	-7.98	2e-05	0.00016
334	L09209_s_at	11	-0.162	1.36	-7.97	2e-05	2e-04
6929	M31523_at	19	0.855	-0.391	7.55	2e-05	5e-04
5928	X74262_at	1	0.869	-0.565	7.42	2e-05	0.00078
7155	Z15115_at	3	1.94	0.945	7.35	2e-05	0.001
26999	L47738_at	5	0.734	-0.779	7.31	2e-05	0.00114
4602	U22376_cds2_s_at	6	1.86	0.294	7.28	2e-05	0.00116
65108	HG1612-HT1612_at	1	1.91	0.888	7.11	2e-05	0.0017
34	M91432_at	1	0.431	-0.771	7.08	2e-05	0.0018
5925	L41870_at	13	-0.438	-1.3	7.08	2e-05	0.0018
546	U72936_s_at	NA	-0.097	-1.07	7.07	2e-05	0.0018
7430	X51521_at	6	1.92	1.07	7.06	2e-05	0.00186
4056	U50136_rna1_at	5	0.71	1.51	-6.97	2e-05	0.00232
54741	Y12670_at	1	-0.167	0.892	-6.96	2e-05	0.00238
7203	X74801_at	1	0.611	-0.183	6.95	2e-05	0.00238
3576	Y00787_s_at	4	-0.371	2.32	-6.87	2e-05	0.00288
6709	J05243_at	9	0.413	-0.982	6.86	2e-05	0.00288
1725	U26266_s_at	19	-0.209	-1.16	6.85	4e-05	0.00294
3205	U82759_at	7	-0.64	0.504	-6.82	2e-05	0.00306
945	M23197_at	19	-0.881	0.354	-6.79	2e-05	0.0033
1509	M63138_at	11	1.21	2.12	-6.77	2e-05	0.00344
6955	M12959_s_at	14	1.13	0.132	6.76	2e-05	0.00352
967	X62654_rna1_at	12	0.0513	1.33	-6.76	2e-05	0.00352
5341	X07743_at	2	-0.959	0.535	-6.74	2e-05	0.00378
140465	M31211_s_at	12	0.108	-0.953	6.71	2e-05	0.00404
7336	U62136_at	8	-0.163	-0.92	6.68	2e-05	0.00428
3660	X15949_at	4	-0.541	-1.33	6.61	2e-05	0.00492
9655	U22944_at	14	0.926	0.368	6.61	2e-05	0.00492

100%

`ll.htmlpage` function from `annotate` package

genelist.html

`annotate`: `chromLoc` class

Location information for <u>one gene</u>

- **`chrom`**: chromosome name.
- **`position`**: starting position of the gene in bp.
- **`strand`**: chromosome strand +/-.

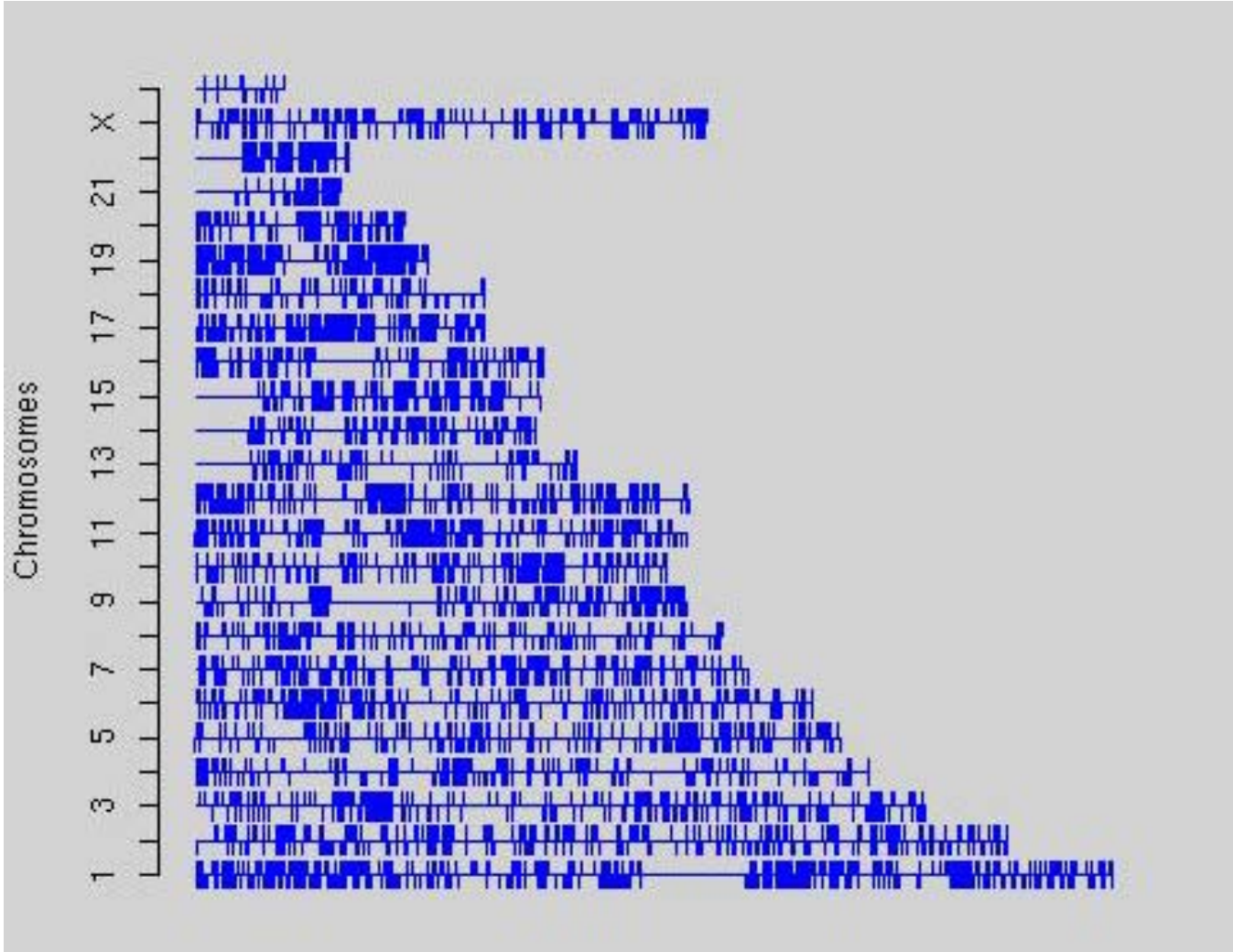
`annotate`: `chromLocation` class

Location information for a <u>set of genes</u>

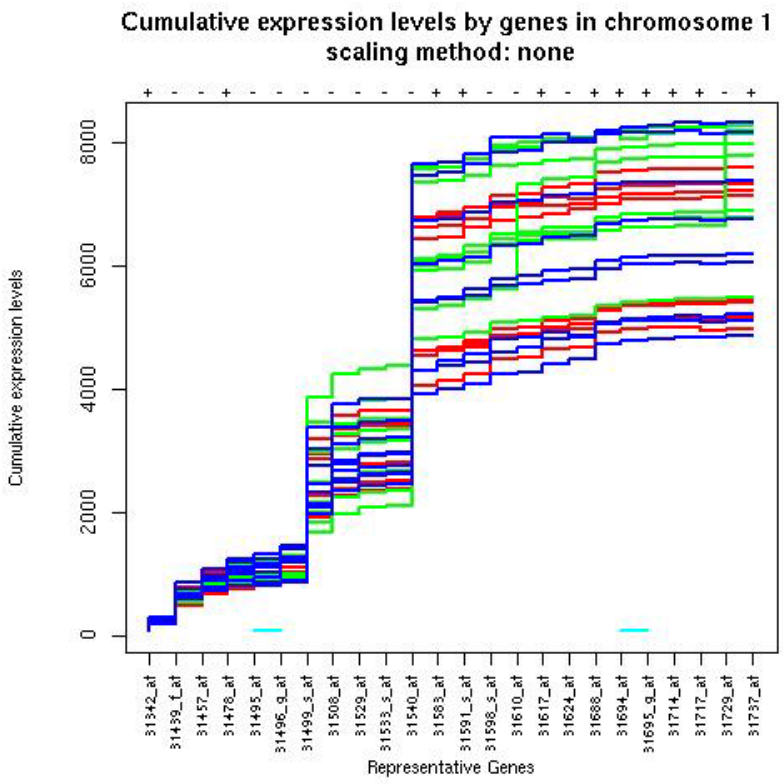
- **`species`**: species that the genes correspond to.
- **`datSource`**: source of the gene location data.
- **`nChrom`**: number of chromosomes for the species.
- **`chromNames`**: chromosome names.
- **`chromLocs`**: starting position of the genes in bp.
- **`chromLengths`**: length of each chromosome in bp.
- **`geneToChrom`**: hash table translating gene IDs to location.

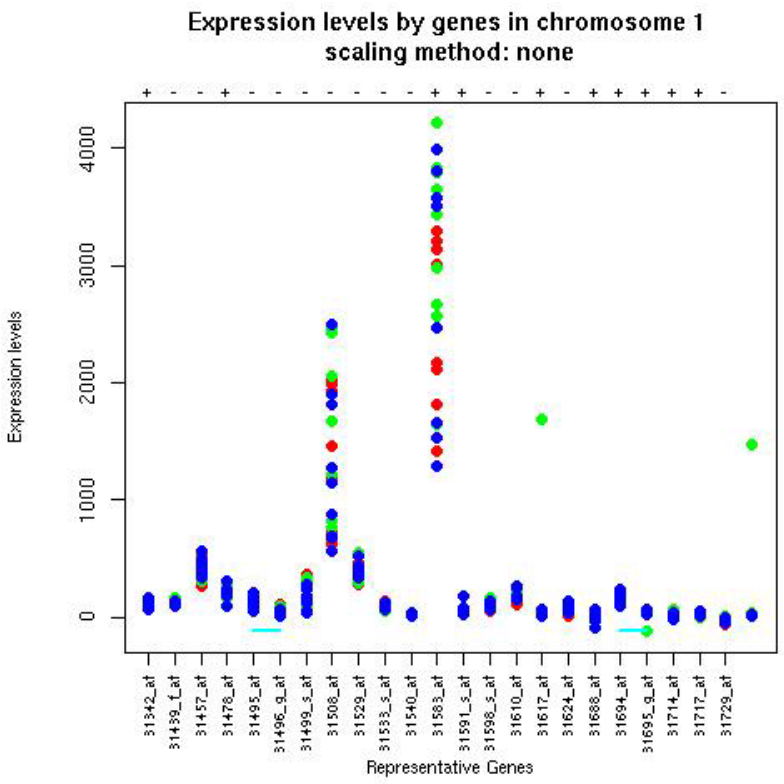
Function **`buildChromClass`**

geneplotter: cPlot

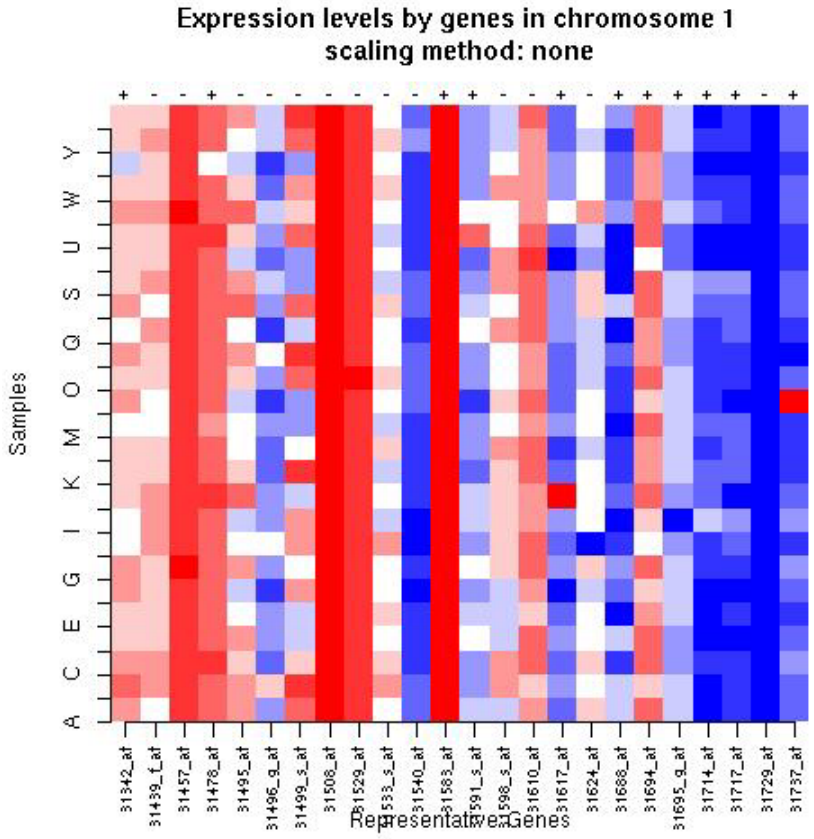


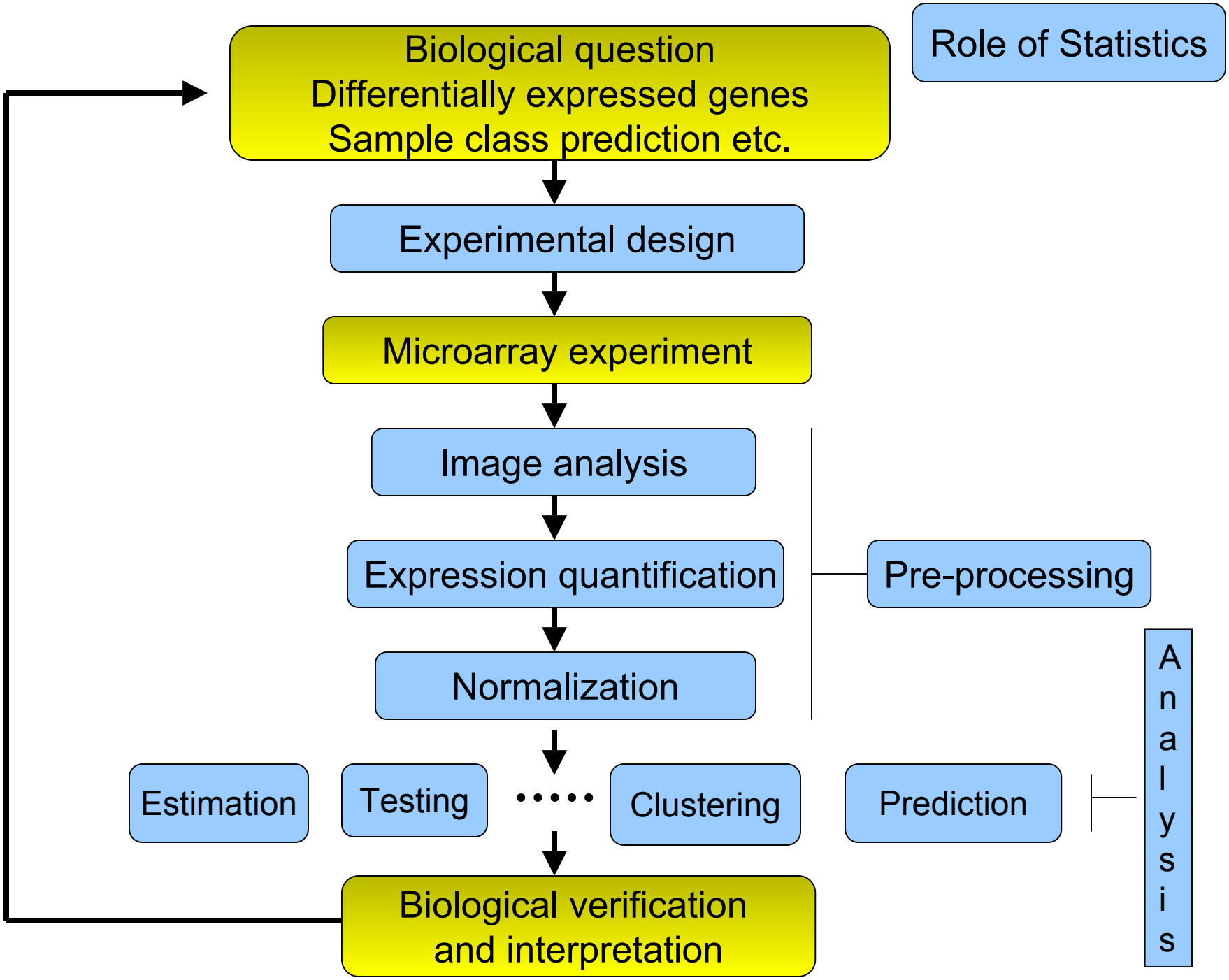
geneplotter: alongChrom





geneplotter: alongChrom




Role of Statistics
Biological question
Differentially expressed genes
Sample class prediction etc.
Experimental design
Microarray experiment
Image analysis
Expression quantification
Normalization
Pre-processing
Estimation
Testing
.....
Clustering
Prediction
Analysis
Biological verification
and interpretation

Combining data across arrays

Data on *G* genes for *n* arrays

→ *G x n* genes-by-arrays data matrix

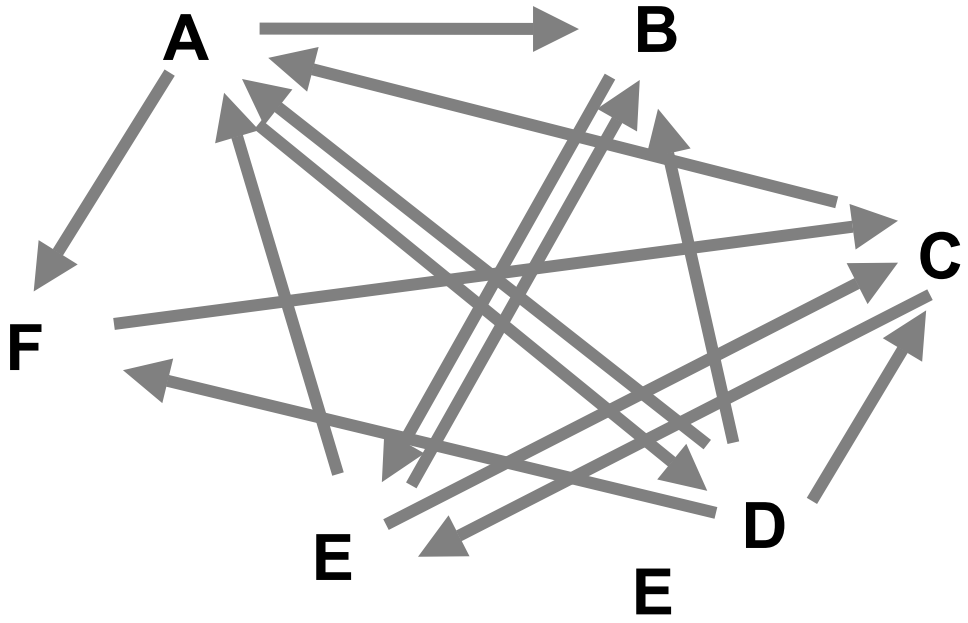
Arrays

Genes	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	...
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.

Biobase: exprSet class

exprs — Matrix of expression measures, genes x samples

se.exprs — Matrix of SEs for expression measures

phenoData — Sample level covariates, instance of class **phenoData**

annotation — Name of annotation data

description — Object of class MIAME

notes — Any notes

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.
- 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(exprs(DATA), ff)
```

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

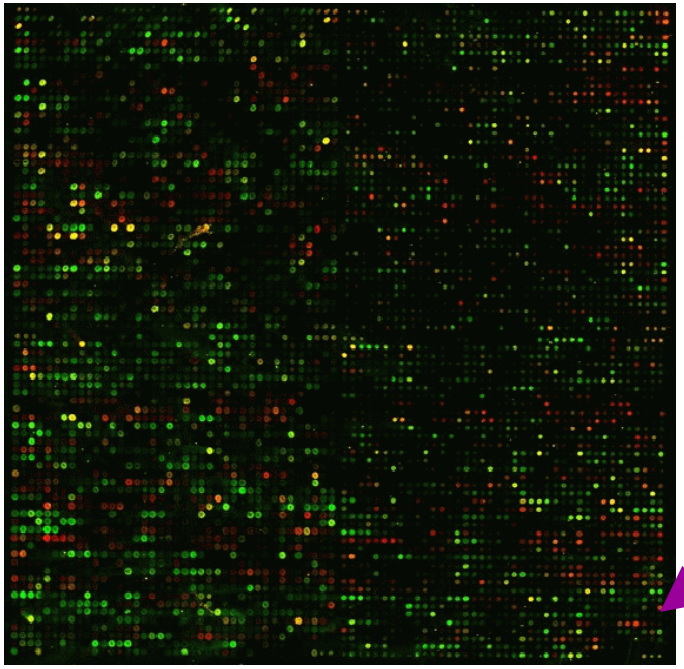
```
mySub <- DATA[wh,]
```

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



p-value = 0.0001 ☺
or
p-value = 5000 x 0.0001 ☹

Multiple hypothesis testing

- When testing for each gene the null hypothesis of no differential expression, e.g. using a t- or F-statistic, two types of errors can be committed.
- Type I error or false positive
  - say that a gene is differentially expressed when it is not,
  - reject a *true null* hypothesis.
- Type II error or false negative
  - fail to identify a truly differentially expressed gene,
  - fail to reject a *false null* hypothesis.

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.

Multiple hypothesis testing

	Non-rejected hypotheses	Rejected hypotheses	
True null hypotheses	**U**	**V** **Type I error**	G_0
False null hypotheses	**T** **Type II error**	**S**	G_1
	G-R	**R**	**G**

From Benjamini & Hochberg (1995)

Type I error rates

- Per-family error rate (PFER). Expected number of false positives, i.e.,

$$\text{PFER} = E(V).$$

- Per-comparison error rate (PCER). Expected value of (# false positives / # of hypotheses), i.e.,

$$\text{PCER} = E(V)/G.$$

- Family-wise error rate (FWER). Probability of at least one false positive, i.e.,

$$\text{FWER} = p(V > 0).$$

Type I error rates

- False discovery rate (FDR). The FDR of Benjamini & Hochberg (1995) is the expected proportion of false positives among the rejected hypotheses, i.e.,

$$\mathrm{FDR} = E(Q),$$

where by definition

$$Q = \begin{cases} V/R, & \text{if } R > 0, \\ 0, & \text{if } R = 0. \end{cases}$$

Strong control

- N.B. Expectations and probabilities above are conditional on which hypotheses are true.
- Strong control. Control of the Type I error rate under any combination of true and false hypotheses.
- Weak control. Control of the Type I error rate under only the complete null hypothesis, i.e., when all null hypotheses are true.
- Strong control is essential in microarray experiments.

Comparison of error rates

- In general, for a given multiple testing procedure,

$$\mathrm{PCER} \leq \mathrm{FWER} \leq \mathrm{PFER}$$

and

$$\mathrm{FDR} \leq \mathrm{FWER}$$

with FDR = FWER under the complete null.

- Thus, for a fixed criterion α for controlling the Type I error rates, the order reverses for the number of rejected hypotheses R: procedures controlling the FWER are generally more conservative than those controlling either the FDR or PCER.

Adjusted p-values

- Given any test procedure, the adjusted p-value for a single gene g can be defined as the nominal level of the entire test procedure at which gene g would just be declared differentially expressed.
- Adjusted p-values reflect for each gene the overall experiment Type I error rate when genes with a smaller p-value are declared differentially expressed.
- Can be estimated by resampling,

  e.g. permutation or bootstrap.

Multiple testing procedures

- Strong control of FWER
  - Bonferroni: single-step;
  - Holm (1979): step-down;
  - Hochberg (1986)*: step-up;
  - Westfall & Young (1993): step-down maxT and minP, exploit *joint* distribution of test statistics.

- Strong control of FDR
  - Benjamini & Hochberg (1995)*: step-up;
  - Benjamini & Yekutieli (2001): step-up.

**some distributional assumptions required.*

Multiple testing procedures

- Golub et al. (1999): neighborhood analysis
  - **weak control only**, problematic definition of error rate.

- Tusher et al. (2001): SAM
  - t- or F-like statistics;
  - similar to univariate test with asymmetric cut-offs;
  - permutation procedure controlling PCER;
  - the SAM estimate of the FDR is $E_0(V)/R$ --- can be greater than one.

`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.

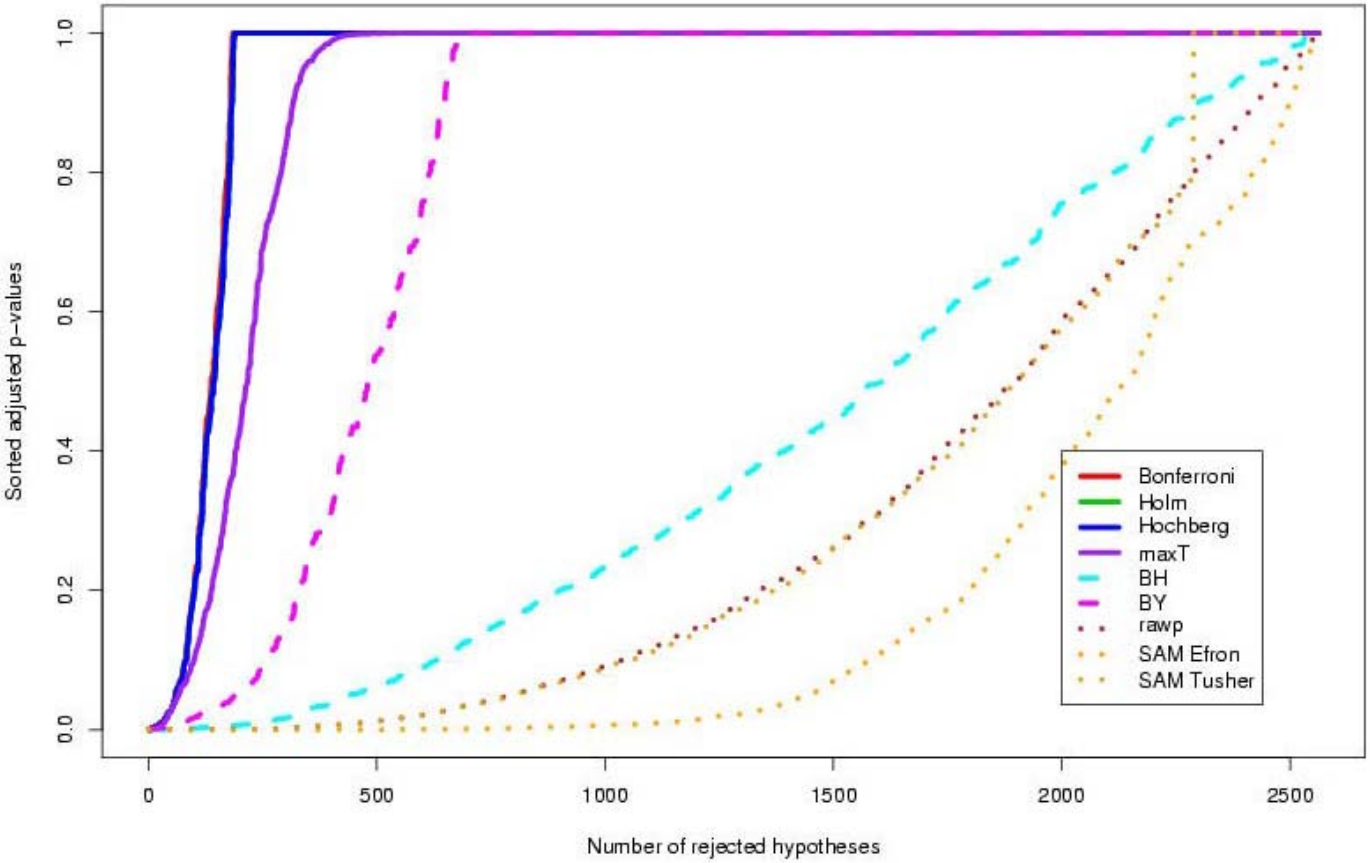
Reporting the results of multiple testing procedures

Plots for adjusted p-values

- allow investigators to examine various false positive rates (FWER, FDR or PCER) associated with different gene lists;
- do not require researchers to preselect a particular definition of Type I error rate or α-level;
- provide tools for deciding on an appropriate combination of number of genes and tolerable false positive rate for a particular experiment and available resources.

`multtest` package

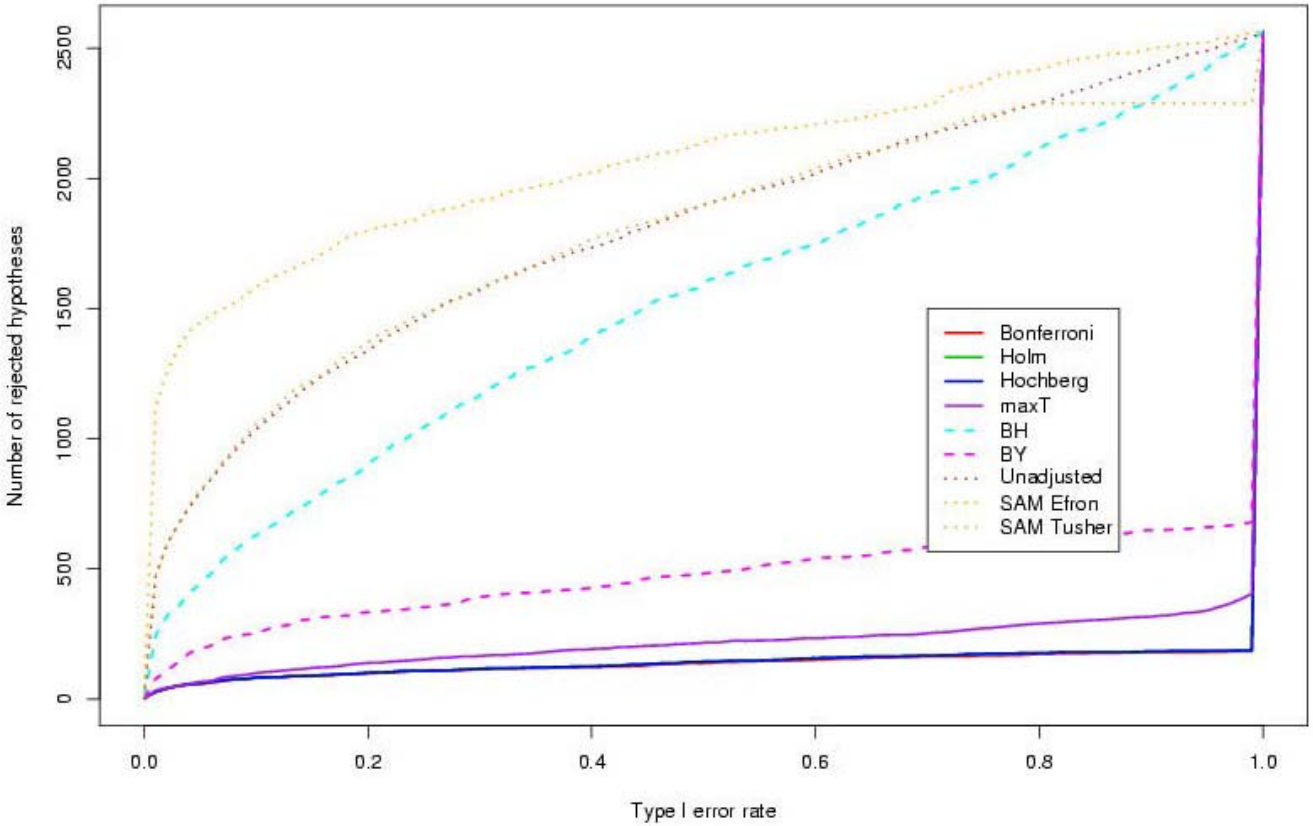
Sorted adjusted p-values for different multiple testing procedures
Golub et al. (1999) ALL AML data



- FWER control
  solid lines
- FDR control
  dashed lines
- PCER control
  dotted lines

`multtest` package

Number of rejected hypotheses vs. false positive rate
Golub et al. (1999) ALL AML data



- FWER control
  solid lines
- FDR control
  dashed lines
- PCER control
  dotted lines

Reporting the results of multiple testing procedures

- Select a number r of genes which you feel comfortable following up and read from the plot the corresponding nominal false positive rates (PCER, FDR, FWER) under various types of error control and testing procedures.
- Find the number of hypotheses that would be rejected using a procedure controlling the FWER at a fixed level, and identify how many others would be rejected using procedures controlling the FDR and PCER at that level.
- Find the number of hypotheses that would be rejected under one procedure, and read the level required to achieve that number under other methods.

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. supervised learning, class prediction
  - 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.

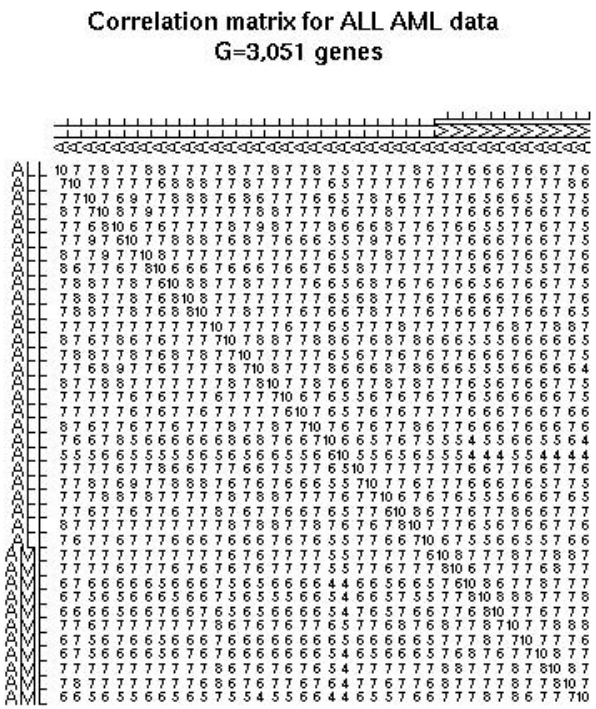
Distances

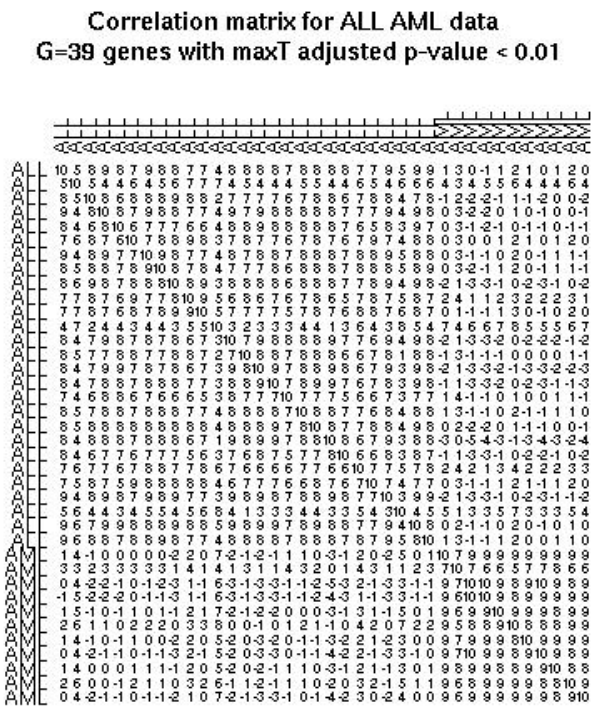
- Microarray data analysis often involves
  - clustering genes or samples;
  - classifying genes or samples.
- Both types of analyses are based on a measure of distance (or similarity) between genes or samples.
- R has a number of functions for computing and plotting distance and similarity matrices.

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`**).

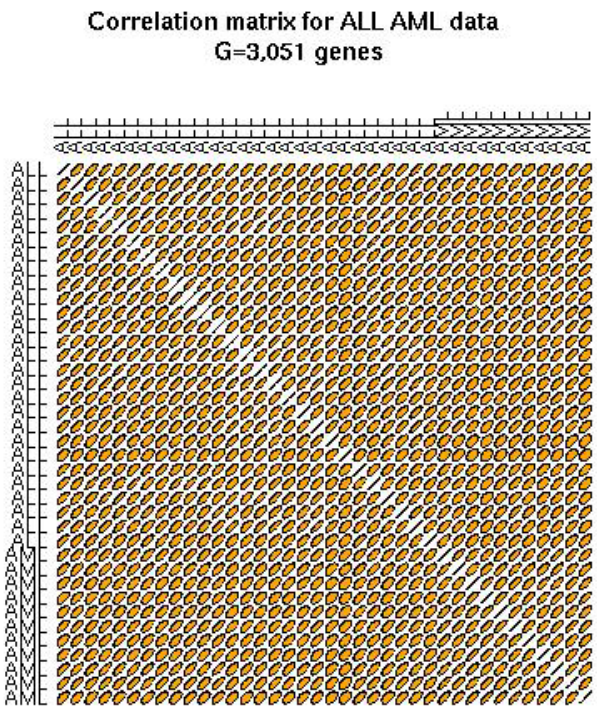
Correlation matrices

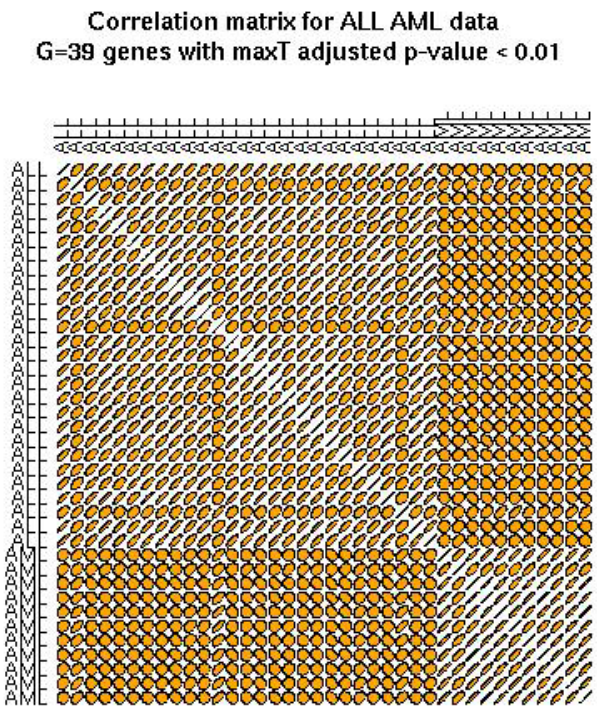




`plotcorr` function from **`ellipse`** package

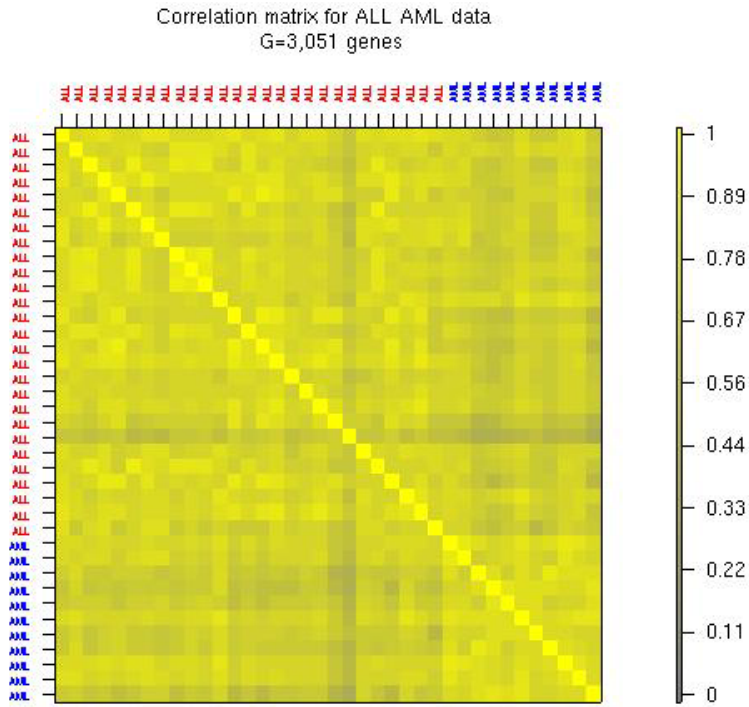
Correlation matrices

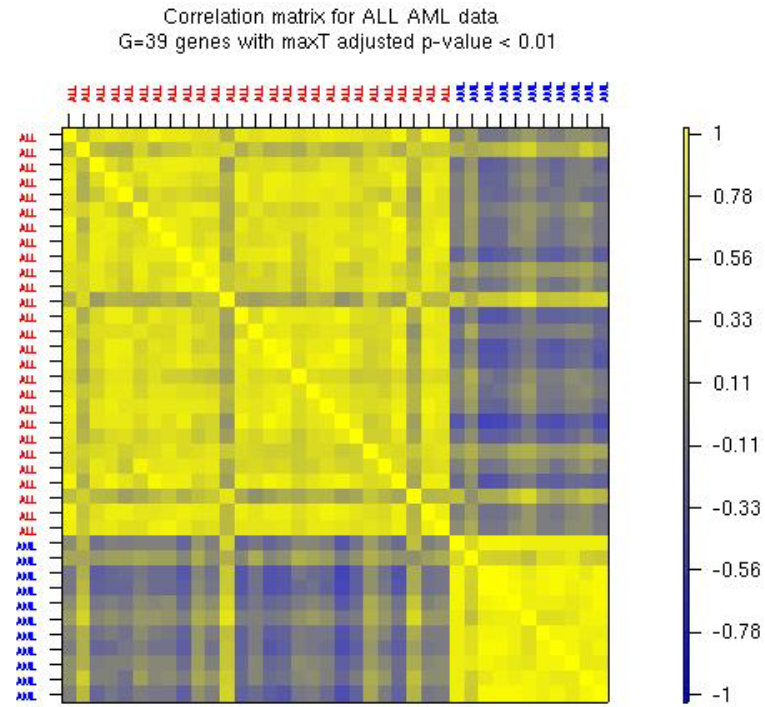




`plotcorr` function from `ellipse` package

Correlation matrices





`plot.cor` function from `sma` package

Multidimensional scaling

- Given any n x n dissimilarity matrix D = (d_{ij}), multidimensional scaling (MDS) is concerned with identifying n points in Euclidean space with a similar distance structure D'=(d_{ij}').
- The purpose is to provide a low(er) 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”.

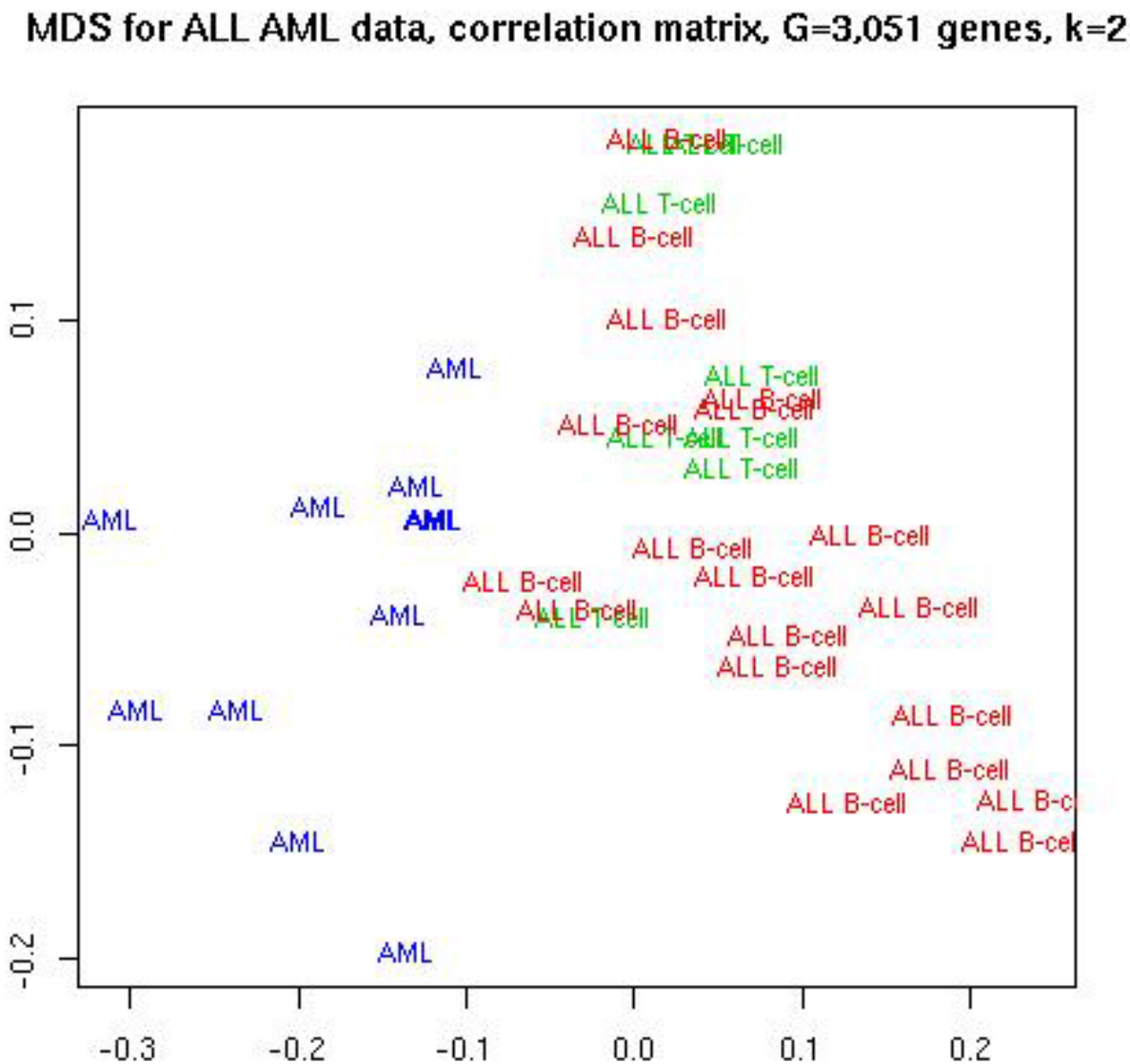
MDS

- **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.

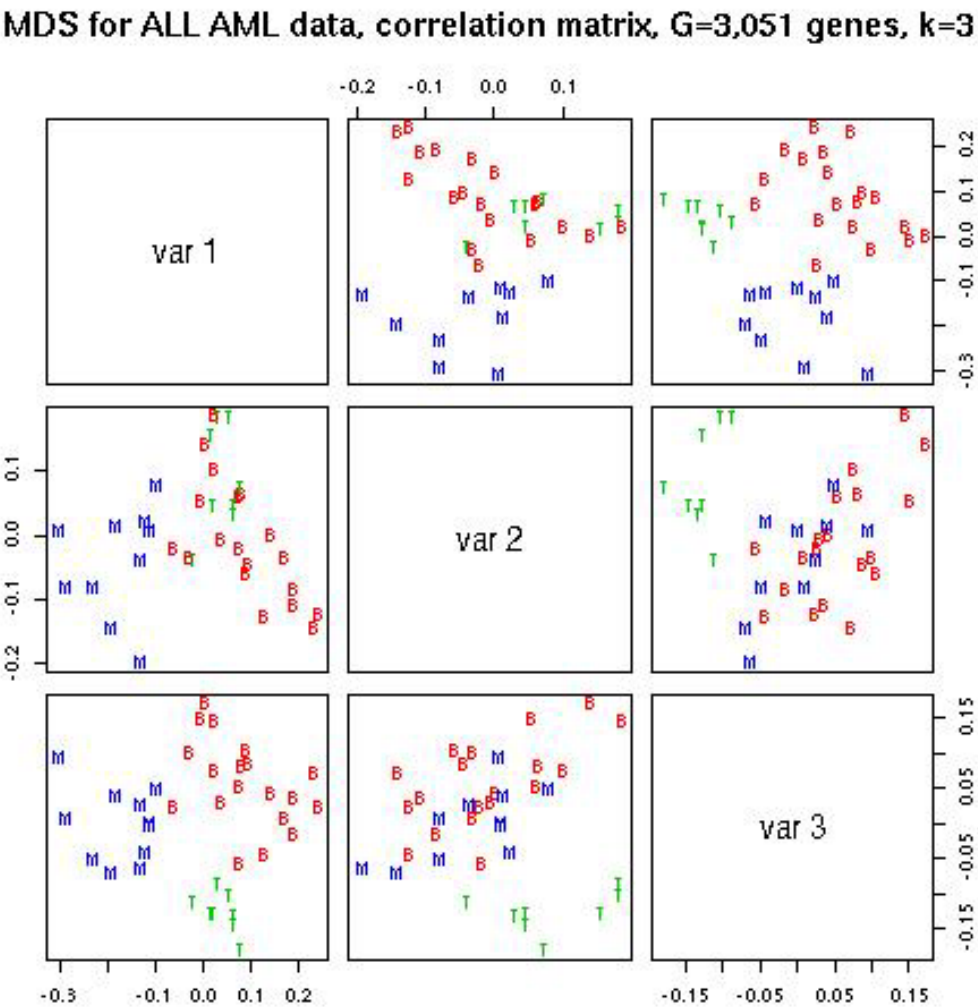
R MDS software

- **`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



Classical MDS



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

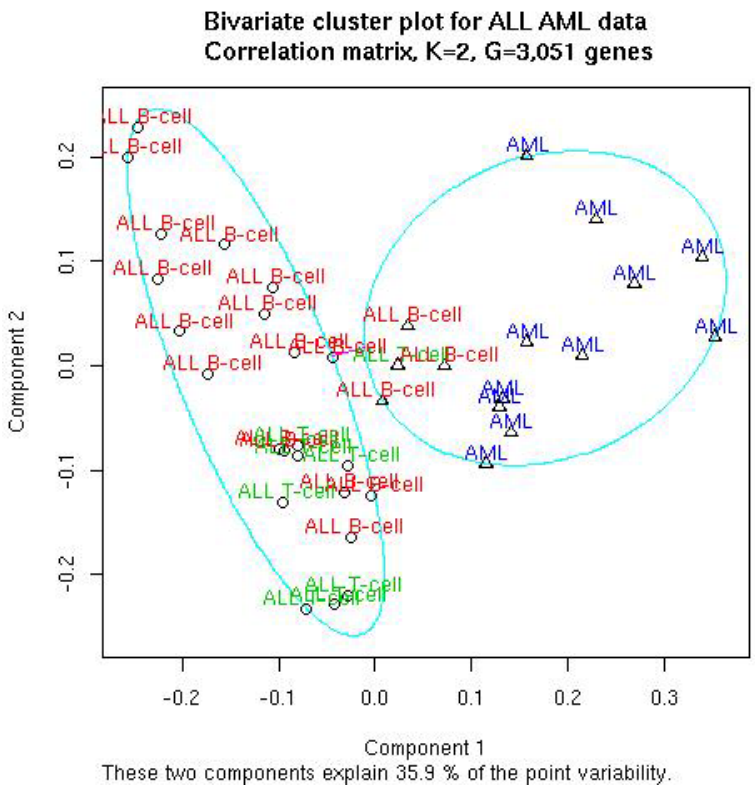
$$\frac{|\lambda_1| + |\lambda_2| + |\lambda_3|}{\sum |\lambda_i|} = 55\%$$

Cluster analysis packages

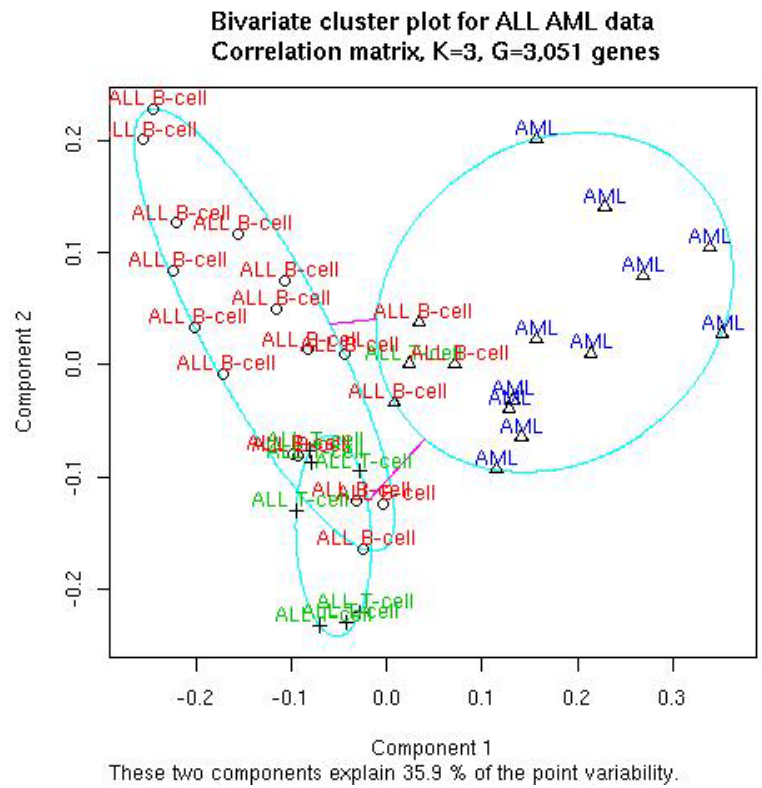
- **`class`**: self organizing maps (**`SOM`**).
- **`cluster`**:
  - AGglomerative NESting (**`agnes`**),
  - Clustering LARe Applications (**`clara`**),
  - DIvisive ANAlysis (**`diana`**),
  - Fuzzy Analysis (**`fanny`**),
  - MONothetic Analysis (**`mona`**),
  - Partitioning Around Medoids (**`pam`**).
- **`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



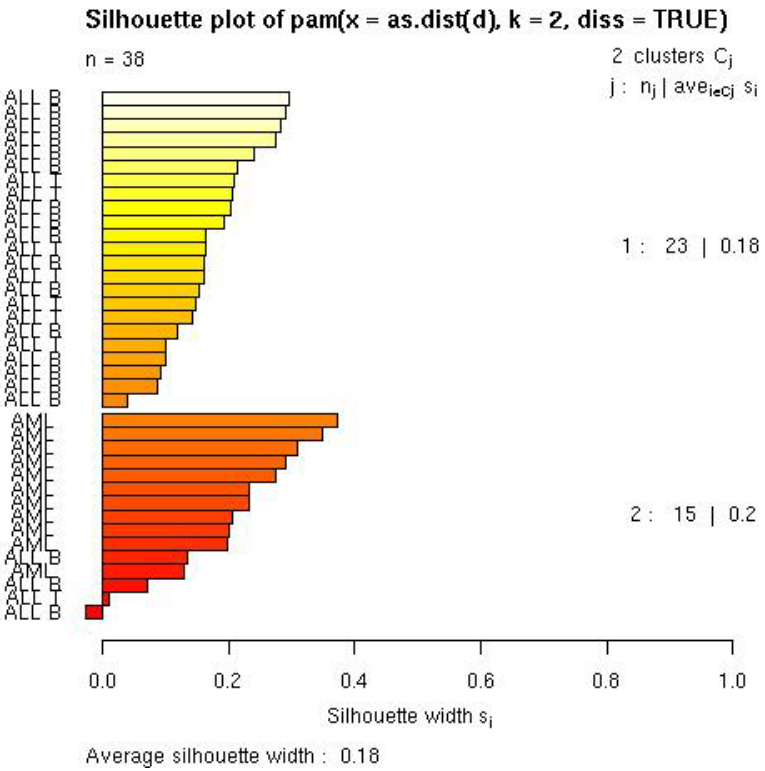
K=3



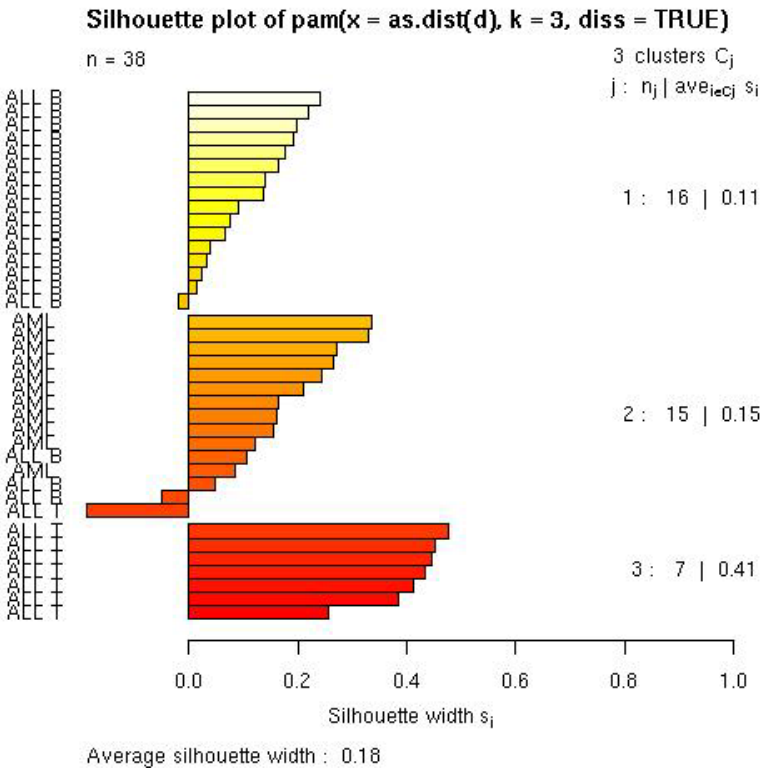
pam and **clusplot** functions from **cluster** package

pam

K=2

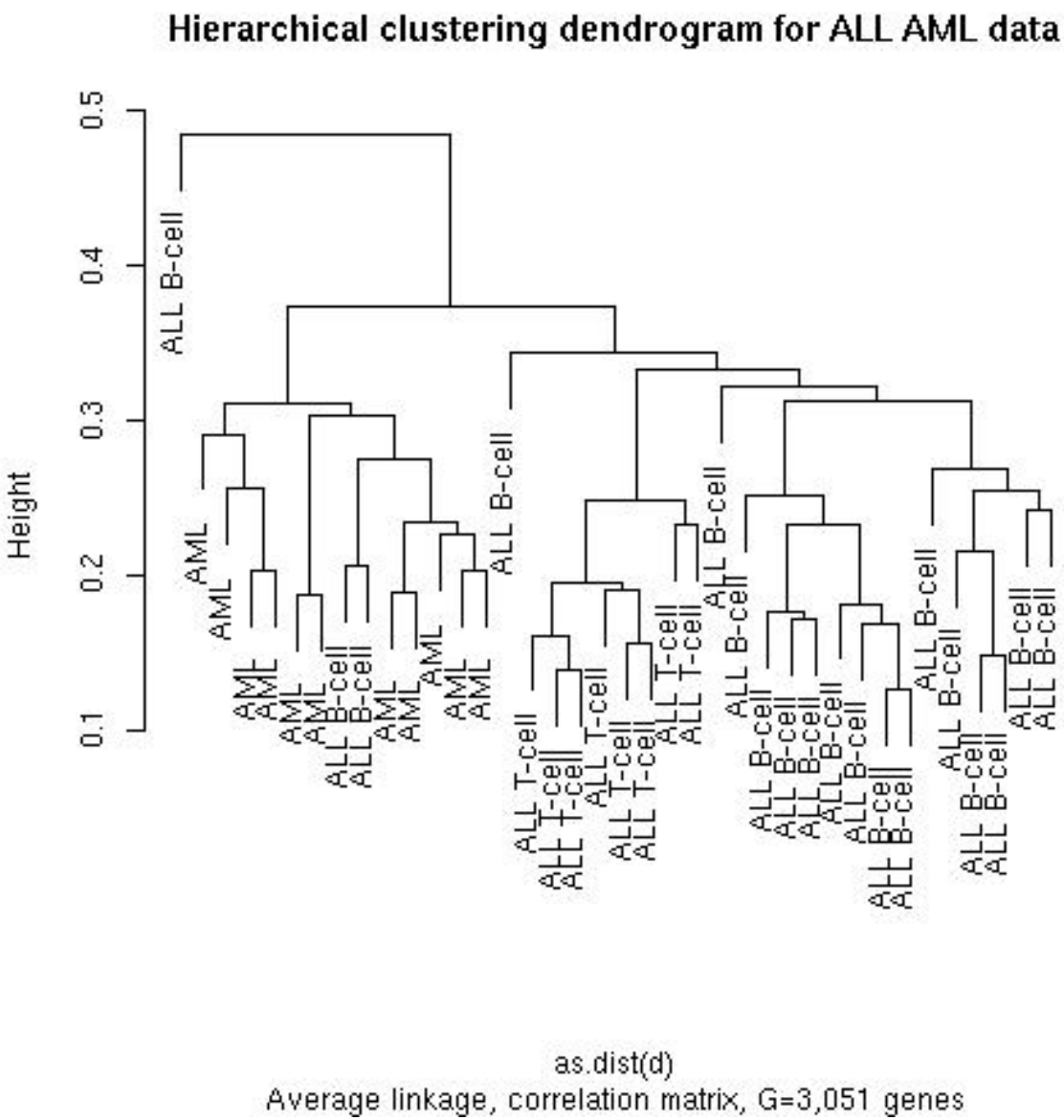


K=3



`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

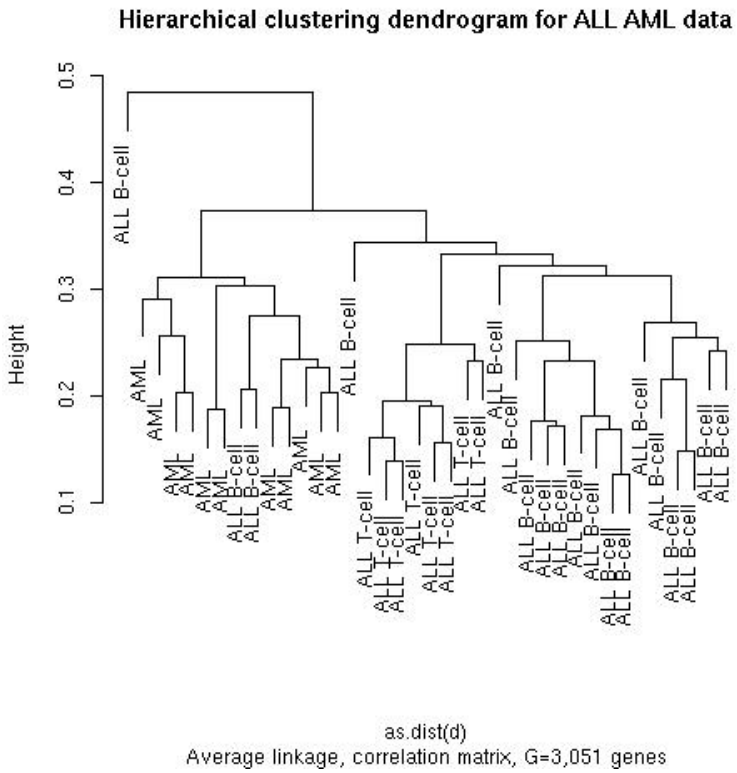
- Second, they *impose* structure on the data, instead of *revealing* structure in these data.
- Such a representation will be valid only to the extent that the pairwise dissimilarities possess the hierarchical structure imposed by the clustering algorithm.

Dendrogram

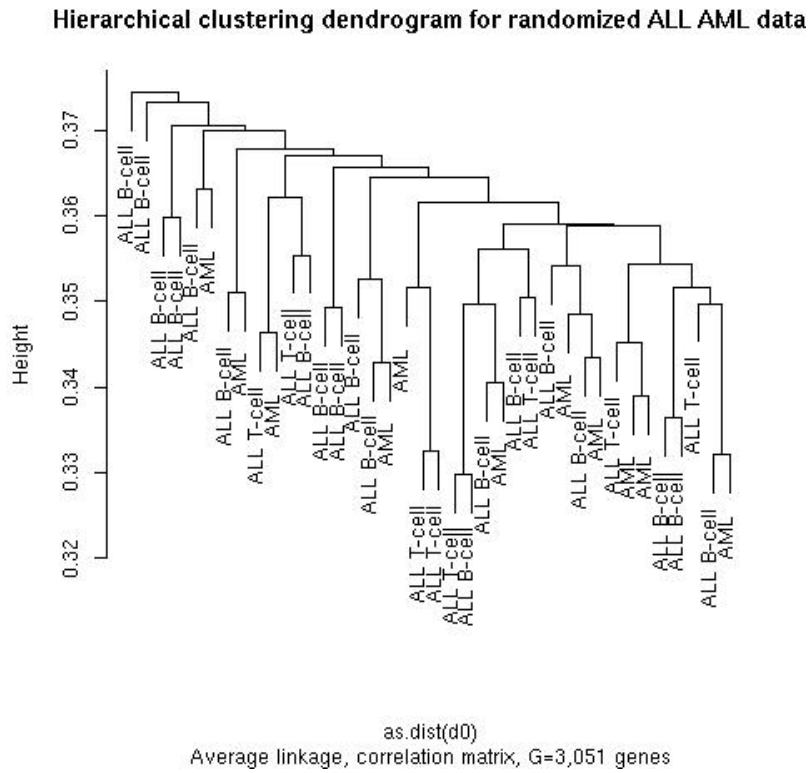
- The cophenetic correlation coefficient can be used to measure how well the hierarchical structure from the dendrogram represents the actual distances.
- This measure is defined as the correlation between the n(n-1)/2 pairwise dissimilarities between observations and their cophenetic dissimilarities 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`**).