Bioconductor Tutorial

Statistical Methods and Software for the Analysis of DNA Microarray Data

Katie Pollard & Todd Lowe
University of California, Santa Cruz

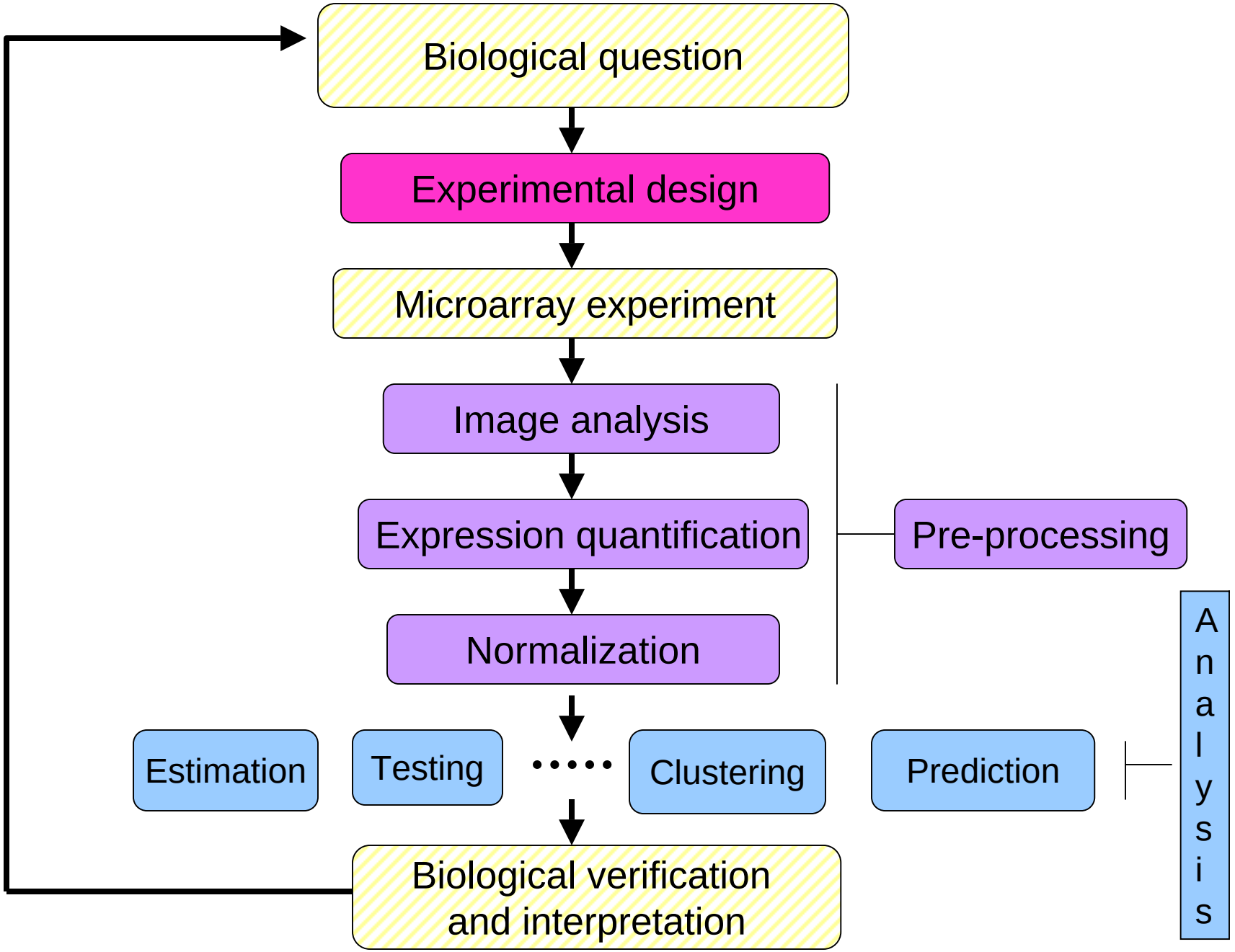
Sandrine Dudoit
University of California, Berkeley

August 15, 2003

Outline

- Overview of the Bioconductor Project.
- Pre-processing two-color spotted microarray data
  - image analysis,
  - normalization.
- Differential gene expression.
- Annotation.
- Visualization.
- Clustering and classification.

Overview of the Bioconductor Project


Biological question
Experimental design
Microarray experiment
Image analysis
Expression quantification
Normalization
Pre-processing
Estimation
Testing
Clustering
Prediction
Analysis
Biological verification and interpretation

Bioconductor

- Bioconductor is an open source and open development software project for the analysis and comprehension of biomedical and genomic data.
- The project was started in the Fall of 2001 by Robert Gentleman, at the Biostatistics Unit of the Dana Farber Cancer Institute.
- R and the R package system are used to design and distribute software.
- Software, data, and documentation are available from www.bioconductor.org.

Bioconductor

- Mechanisms for facilitating the design and deployment of portable, extensible, and scalable software.
- Support for interoperability with software written in other languages.
- Tools for integrating biological metadata from the internet in the analysis of experimental metadata.
- Access to a broad range of statistical and numerical methods.
- High-quality visualization and graphics tools that support interactivity.
- An effective, extensible user interface.
- Tools for producing innovative, high-quality documentation and training materials.
- Methodology that supports the creation, testing, and distribution of software and data modules.

Bioconductor

Scenario:

- Pre-processing of spotted array data with **`marrayNorm`**.
- List of differentially expressed genes from **`multtest`**, **`limma`**, or **`genefilter`**.
- Use the **`annotate`** package
  - to retrieve and search PubMed abstracts for these genes;
  - to generate an HTML report with links to LocusLink for each gene.

Data

- Issues:
  - complexity;
  - size;
  - evolution.

- We distinguish between biological metadata and experimental metadata.

Experimental metadata

- Gene expression measures
  - scanned images, i.e., raw data;
  - image quantitation data, i.e., output from image analysis;
  - normalized expression measures, i.e., log ratios M or Affy measures.
- Reliability information for the expression measures.
- Information on the probe sequences printed on the arrays (array layout).
- Information on the target samples hybridized to the arrays.
- See Minimum Information About a Microarray Experiment – MIAME – standards and new **MAGEML** package.

Biological metadata

- Biological attributes that can be applied to the experimental data.
- E.g. for genes
  - chromosomal location;
  - gene annotation (LocusLink, GO);
  - relevant literature (PubMed).
- Biological metadata sources are large, complex, evolving rapidly, and typically distributed via the WWW.
- Cf. **`annotate`** and **`AnnBuilder`** packages.

`marray` packages

Image quantitation data, e.g. .gpr, .Spot, .gal

Class **`marrayRaw`**

`maNorm`
`maNormMain`
`maNormScale`

Class **`marrayNorm`**

`as(swirl.norm, "exprSet")`

Class **`exprSet`**

Save data to file using **`write.exprs`** or continue analysis using other Bioconductor packages

`marrayLayout` class

Array layout parameters

`maNspots`	Total number of spots
`maNgr` `maNgc`	Dimensions of grid matrix
`maNsr` `maNsc`	Dimensions of spot matrices
`maSub`	Current subset of spots
`maPlate`	Plate IDs for each spot
`maControls`	Control status labels for each spot
`maNotes`	Any notes

`marrayRaw` class

Pre-normalization intensity data for a batch of arrays

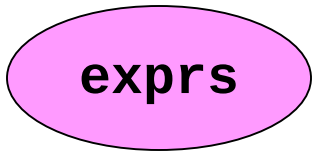
maRf	**maGf**	Matrix of red & green foreground intensities
maRb	**maGb**	Matrix of red & green background intensities
maW		Matrix of spot quality weights
maLayout		Array layout parameters - **marrayLayout**
maGnames		Description of spotted probe sequences - **marrayInfo**
maTargets		Description of target samples - **marrayInfo**
maNotes		Any notes

`marrayNorm` class

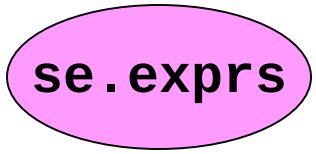
Post-normalization intensity data for a batch of arrays

`maA`		Matrix of average log intensities, A
`maM`		Matrix of normalized intensity log ratios, M
`maMloc`	**`maMscale`**	Matrix of location and scale normalization values
`maW`		Matrix of spot quality weights
`maLayout`		Array layout parameters - **`marrayLayout`**
`maGnames`		Description of spotted probe sequences - **`marrayInfo`**
`maTargets`		Description of target samples - **`marrayInfo`**
`maNormCall`		Function call
`maNotes`		Any notes

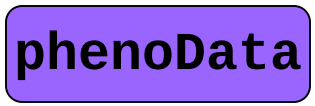
`exprSet` class



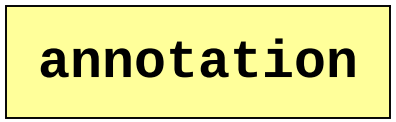
Matrix of expression measures, genes x samples



Matrix of SEs for expression measures, genes x samples

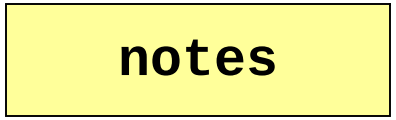


Sample level covariates, instance of class **`phenoData`**



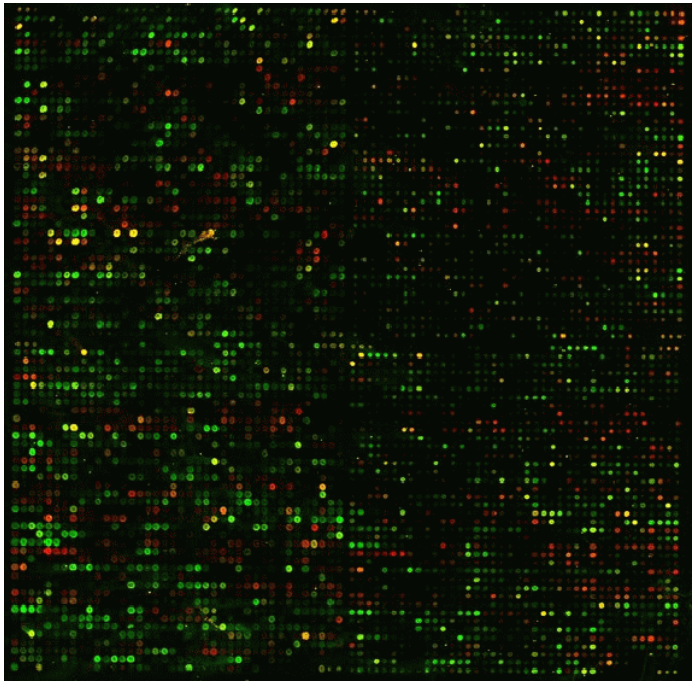
Name of annotation data

MIAME information



Any notes

Pre-processing Two-color Spotted Microarray Data



Raw data

- Pairs of 16–bit TIFFs, one for each dye.
- E.g. Human cDNA arrays:
  - ~43K spots;
  - ~ 20Mb per channel;
  - ~ 2,000 x 5,500 pixels per image;
  - spot separation: ~ 136um.
- For a “typical” array, the spot area has
  - mean = 43 pixels,
  - med = 32 pixels,
  - SD = 26 pixels.

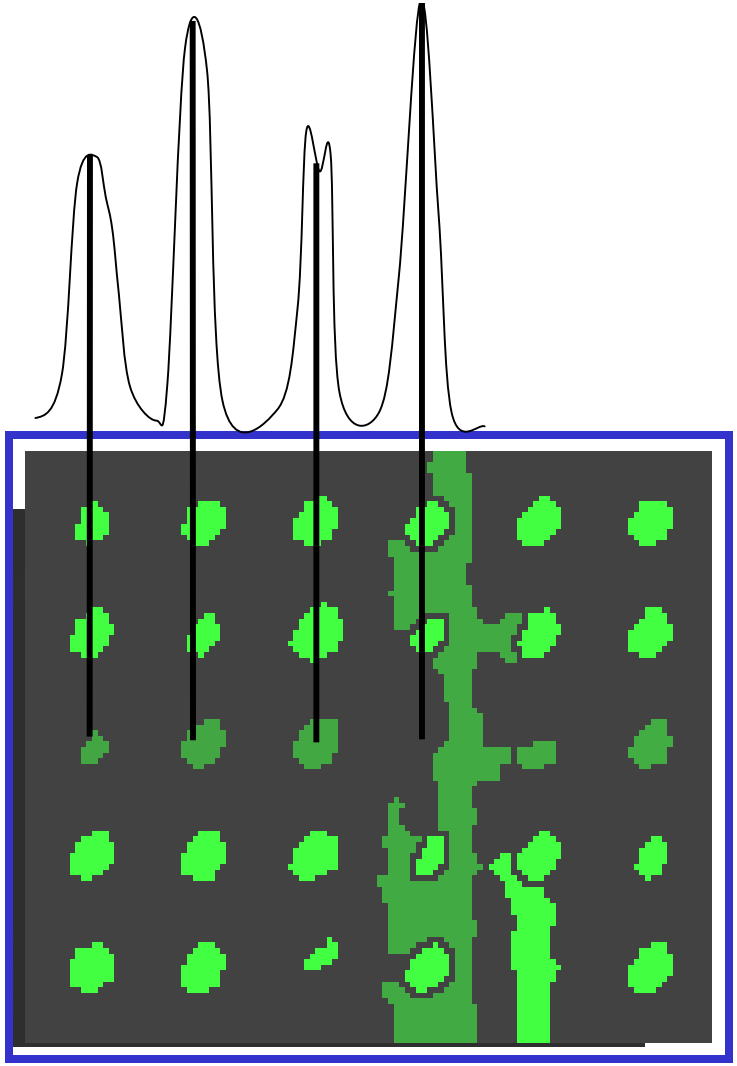
Image analysis

1. Addressing. Estimate location of spot centers.

2. Segmentation. Classify pixels as foreground (signal) or background.

3. Information extraction. For each spot on the array and each dye

- foreground intensities;
- background intensities;
- quality measures.



→ **R and G for each spot on the array.**

`Spot` image analysis software

- Software package **`Spot`**, built on the R language and environment for statistical computing and graphics.
- Batch automatic addressing.
- Segmentation. Seeded region growing (Adams & Bischof 1994): adaptive segmentation method, no restriction on the size or shape of the spots.
- Information extraction
  - Foreground. Mean of pixel intensities within a spot.
  - Background. Morphological opening: non-linear filter which generates an image of the estimated background intensity for the entire slide.
- Spot quality measures.

Normalization

- After image processing, we have measures of the red and green fluorescence intensities, **R** and **G**, for each spot on the array.

- Normalization is needed to ensure that differences in intensities are indeed due to differential expression, and not some printing, hybridization, or scanning artifact.

- Normalization is necessary before any analysis which involves within or between slides comparisons of intensities, e.g., clustering, testing.

Normalization

- Identify and remove the effects of systematic variation in the measured fluorescence intensities, other than differential expression, for example
  - different labeling efficiencies of the dyes (dye swap experiments help);
  - different amounts of Cy3- and Cy5-labelled mRNA;
  - different scanning parameters;
  - print-tip, spatial, time-of-printing, or plate effects, etc.

Normalization

- Within-slide normalization: Correct for systematic differences in intensities between co-hybridized samples.
  - Location normalization - additive on log-scale.
  - Scale normalization - multiplicative on log-scale.
  - Which spots to use?
  - Paired-slides (dye-swap experiments): self-normalization.
- Between-slides normalization: Correct for systematic differences in intensities between samples hybridized to different slides.

Normalization

- Two-channel normalization: normalization of log-ratios.
  - For analysis of relative expression levels, e.g., gene expressed at a higher level in target sample A than in sample B.
- Single-channel normalization: normalization of individual red and green log-intensities.
  - For analysis of absolute expression levels, e.g., testing for expression or lack thereof of certain genes in a target sample A.
  - Two-channel within-slide normalization followed by between-slides normalization, cf. normalization methods for Affymetrix data.

Normalization

- The need for normalization can be seen most clearly in self-self hybridizations, where the same mRNA sample is labeled with the Cy3 and Cy5 dyes.
- The imbalance in the red and green intensities is usually not constant across the spots within and between arrays, and can vary according to overall spot intensity, location, plate origin, etc.
- These factors should be considered in the normalization.

Pre-processing software

- Reading in intensity data, diagnostic plots, normalization, computation of expression measures.
- The packages start with very different data types, but produce similar objects of class **`exprSet`**.
- One can then use other Bioconductor packages, e.g., **`genefilter`**, **`geneplotter`**.

- **`affy`**: Affymetrix oligonucleotide chips.
- **`marray`**, **`limma`**: Spotted DNA microarrays.
- **`vsn`**: Variance stabilization for both types of arrays.

Differential Gene Expression

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.

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.

`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.

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.

`limma`: Linear models for microarray data

- Fitting of gene-wise linear models to estimate log-ratios between two or more target samples simultaneously: **`lm.series`**, **`rlm.series`**, **`glm.series`** (handles replicate spots).
- **`ebayes`**: moderated t-statistics and log-odds of differential expression by empirical Bayes shrinkage of the standard errors towards a common value.

Multiple hypothesis testing

- Large multiplicity problem: thousands of hypotheses are tested simultaneously!
- Need to adjust for multiple testing when assessing the statistical significance of the observed associations.
- Define an appropriate Type I error or false positive rate.
- 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 the null distribution (used to calculate adjusted p-values).
- Similar bootstrap procedures coming soon!
- Fast permutation algorithm for minP adjusted p-values.
- Documentation: tutorial on multiple testing.

Annotation

Annotation

- One of the largest challenges in analyzing genomic data is associating the experimental data with the available biological metadata, e.g., sequence, gene annotation, chromosomal maps, literature.

- Bioconductor provides two main packages for this purpose:
  - **`annotate`** (end-user);
  - **`AnnBuilder`** (developer).

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).

`annotate`: matching IDs

Important tasks

- Associate manufacturers or in-house probe identifiers to other available identifiers.

  E.g.

  Affymetrix IDs → LocusLink LocusID

  Affymetrix IDs → GenBank accession number.
- Associate probes with biological data such as chromosomal position, pathways.
- Associate probes with published literature data via PubMed (need PMID).

`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 provides annotation data packages, that contain many different mappings to interesting data
  - Mappings between Affy IDs and other probe IDs: **`hgu95av2`** for HGU95Av2 GeneChip series, also, **`hgu133a`**, **`hu6800`**, **`mgu74a`**, **`rgu34a`**, **`YG`**.
  - Affy CDF data packages.
  - Probe sequence data packages.
- These packages are updated and expanded regularly as new data become available.
- They can be downloaded from the Bioconductor website and also using **`installDataPackage`**.
- **`DPExplorer`**: a widget for interacting with data packages.
- **`AnnBuilder`**: tools for building annotation data packages.

`annotate`: querying databases

The **`annotate`** package provides tools for

- Searching and processing information from various WWW biological databases
  - GenBank,
  - LocusLink,
  - PubMed.
- Regular expression searching of PubMed abstracts.
- Generating 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
- Other tools: **HTMLPage** class, **getTDRows**, **getQueryLink**, **getQuery4UG**, **getQuery4LL**, **makeAnchor** .
- The **XML** package is used to parse query results.

Visualization

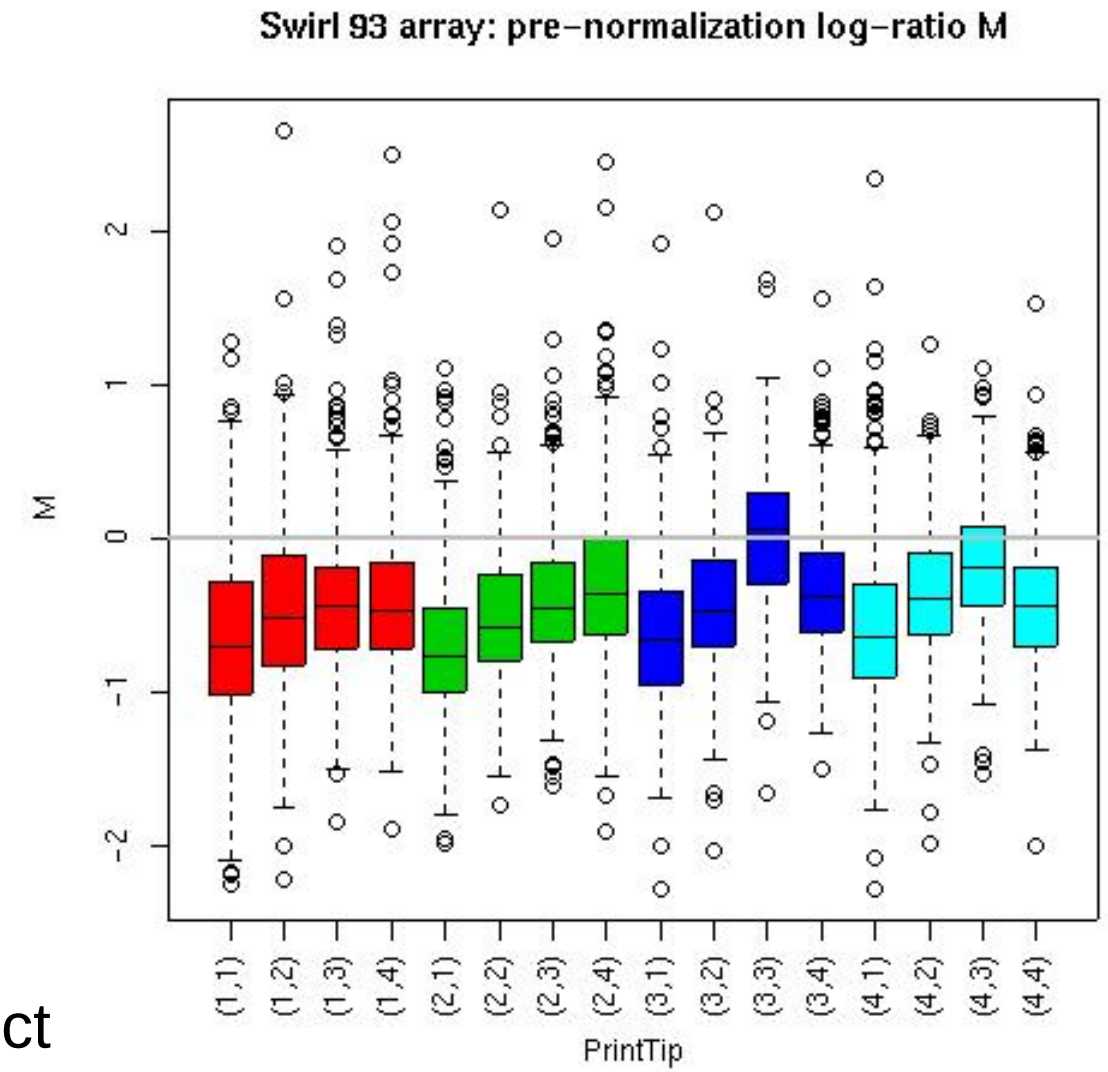
Diagnostic plots

- RGB overlay of Cy3 and Cy5 images.
- Diagnostics plots of spot statistics,

  e.g. red and green log-intensities, intensity log-ratios M, average log-intensities A, spot area
  - Boxplots, dotplots;
  - 2D spatial images;
  - Scatter-plots, e.g., MA-plots;
  - Histograms/density plots.
- Stratify plots according to layout parameters, e.g., print-tip-group, plate, time-of-printing.

Boxplots by print-tip-group

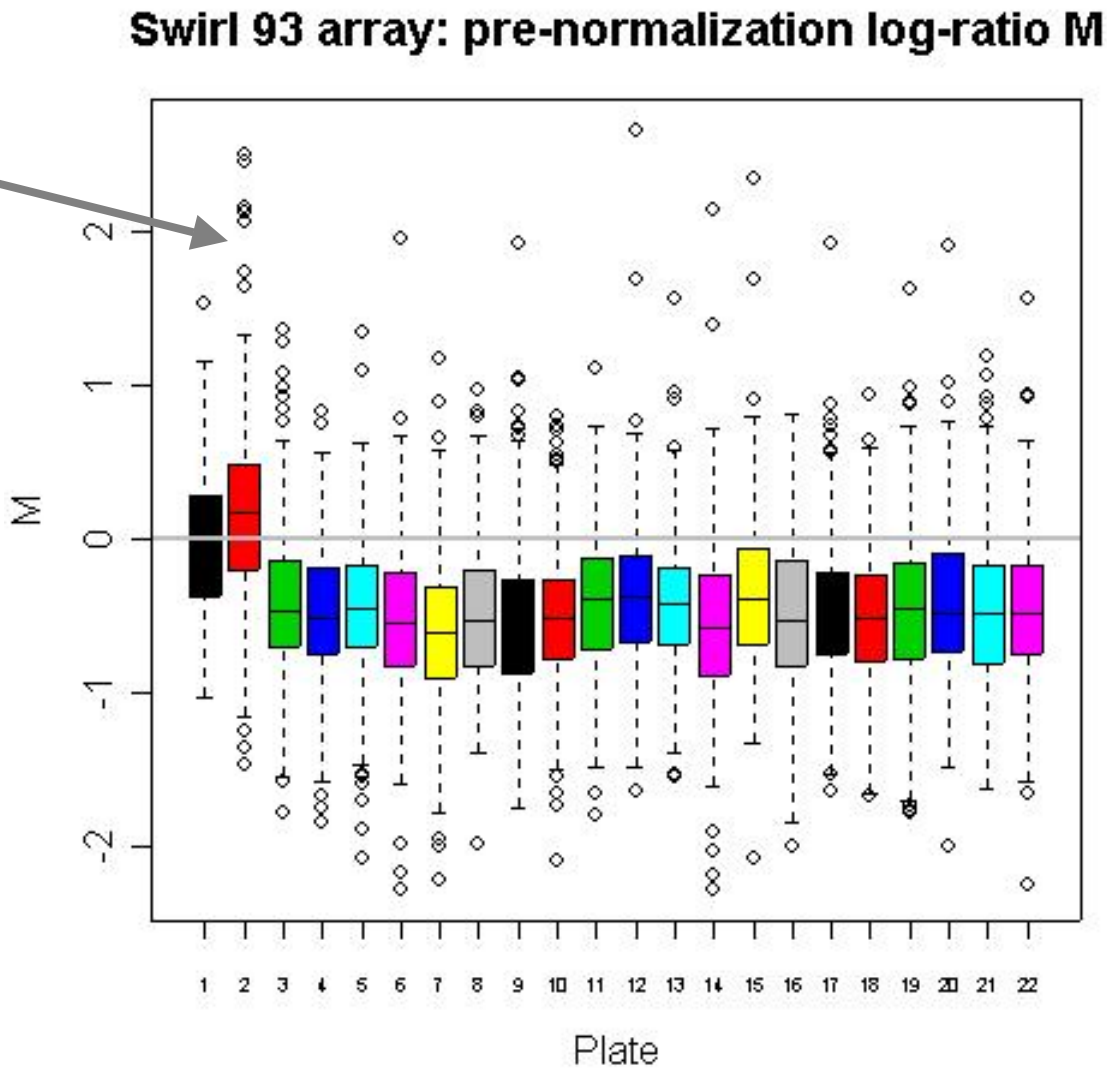
Intensity log-ratio, M



Print-tip effect

Boxplots by plate

Control plate

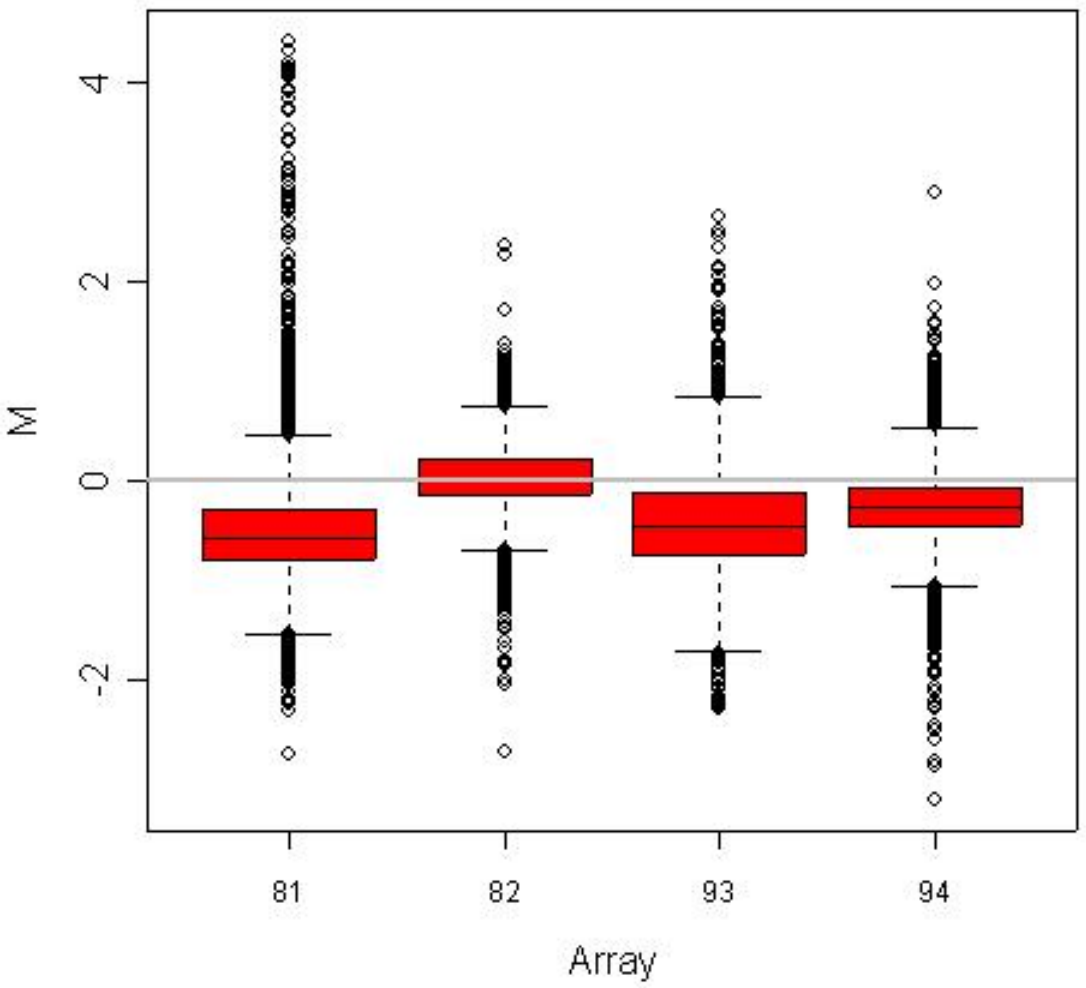


Intensity log-ratio, M

384-well plate effect

Boxplots by array

Intensity log-ratio, M



Array effect

Single-slide data display

- Usually: R vs. G

  $\log_2 R$ vs. $\log_2 G$.

- Preferred

  $M = \log_2 R - \log_2 G$

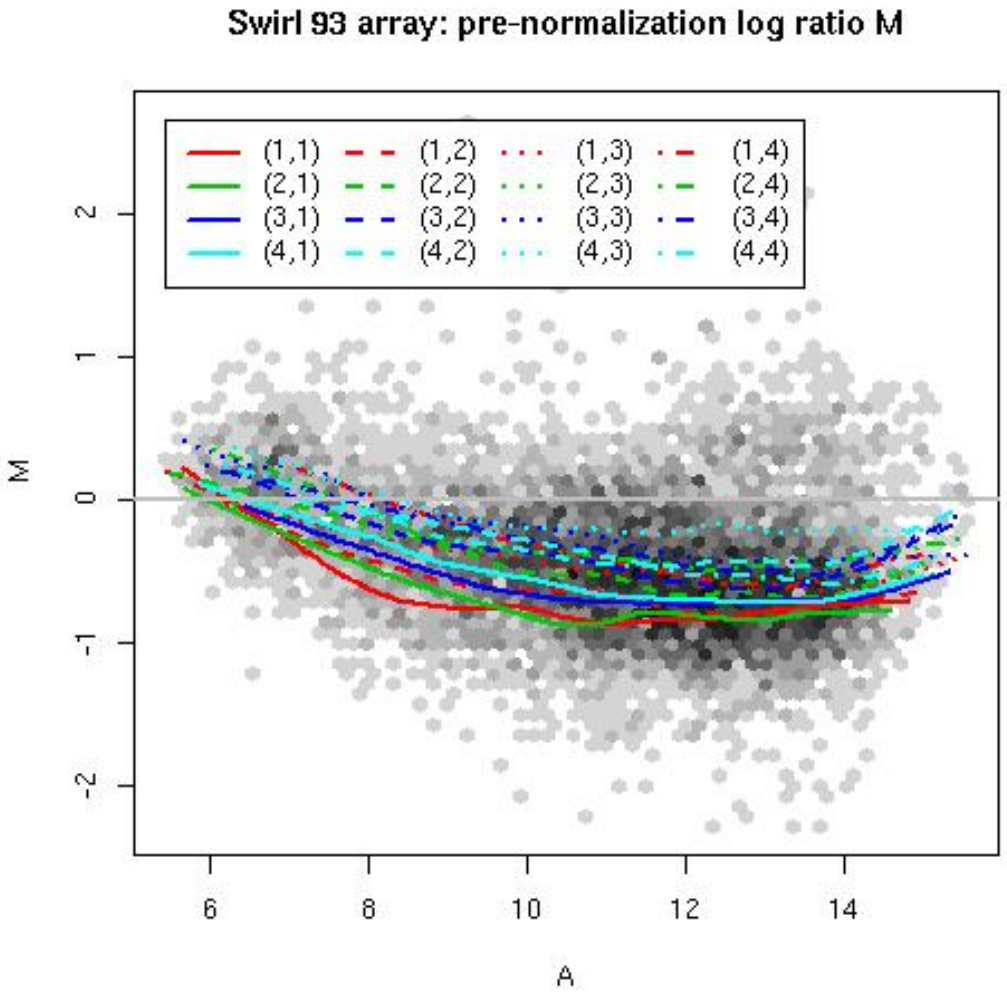
  vs. **$A = (\log_2 R + \log_2 G)/2$**.

- An MA-plot amounts to a 45^o counterclockwise rotation of a

  $\log_2 R$ vs. $\log_2 G$ plot followed by scaling.

MA-plot by print-tip-group

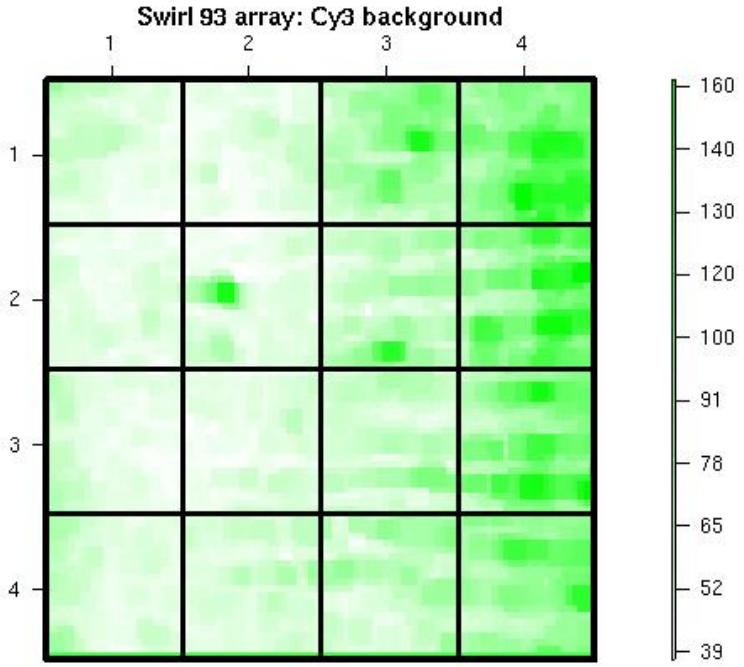
Intensity log-ratio, M



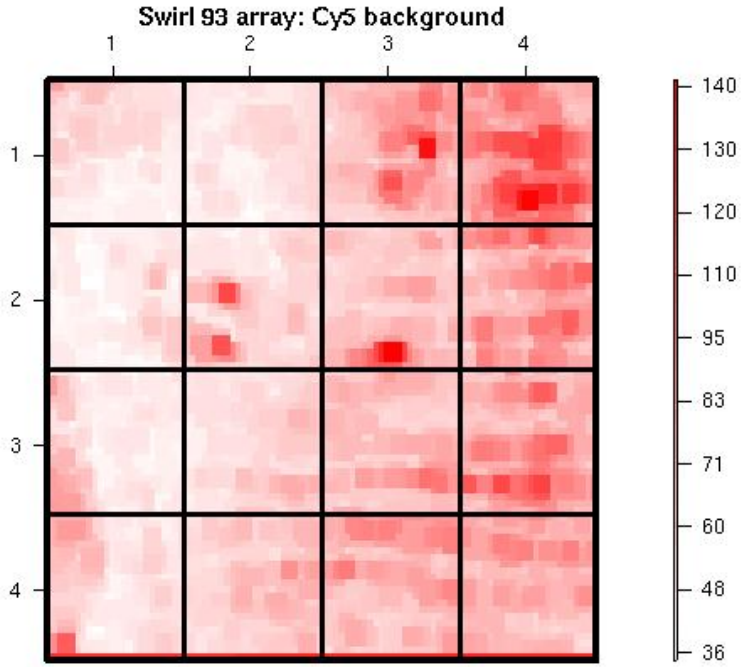
$M = \log_2 R - \log_2 G$,
$A = (\log_2 R + \log_2 G)/2$

Average log-intensity, A

2D spatial images



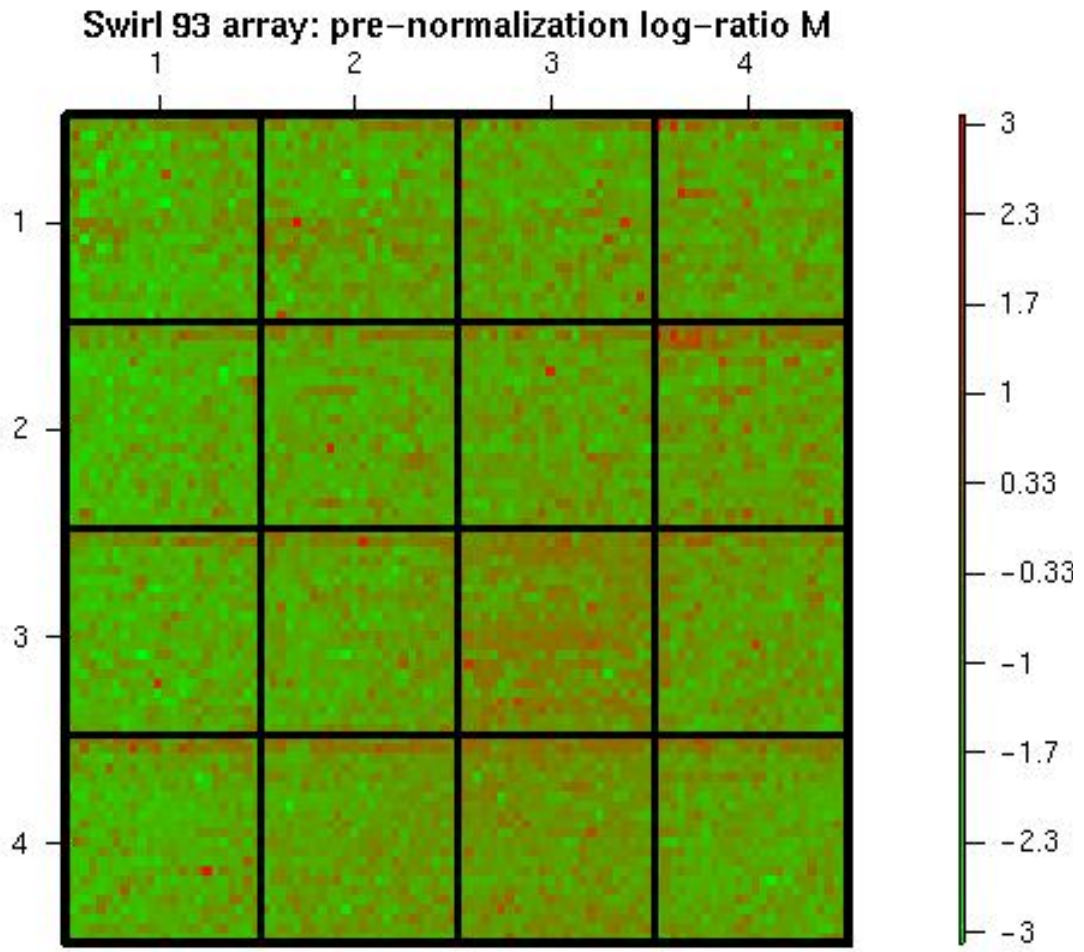
Cy3 background intensity



Cy5 background intensity

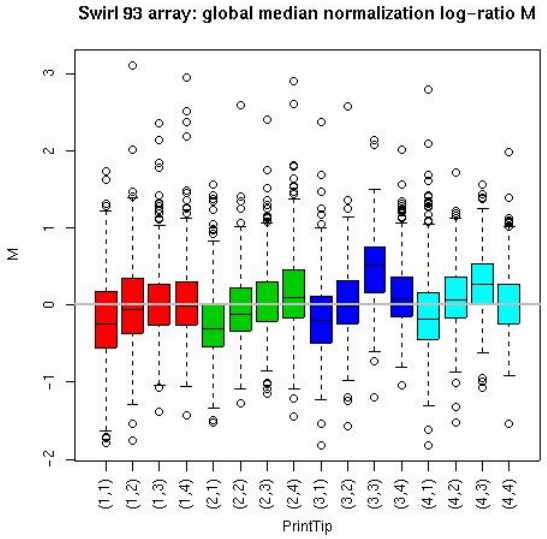
2D spatial images

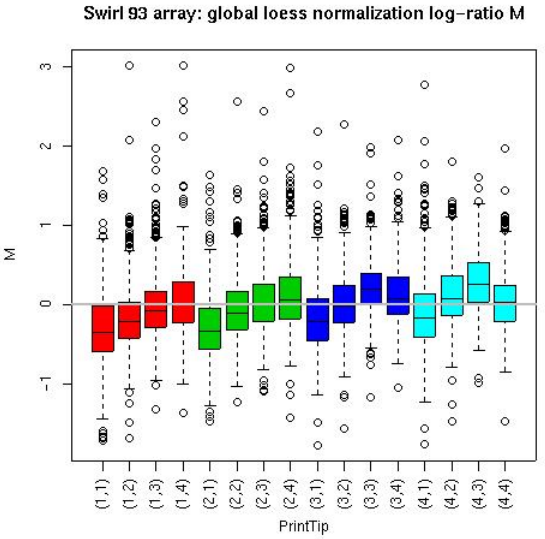
Intensity log-ratio, M



Boxplots of normalized M-L

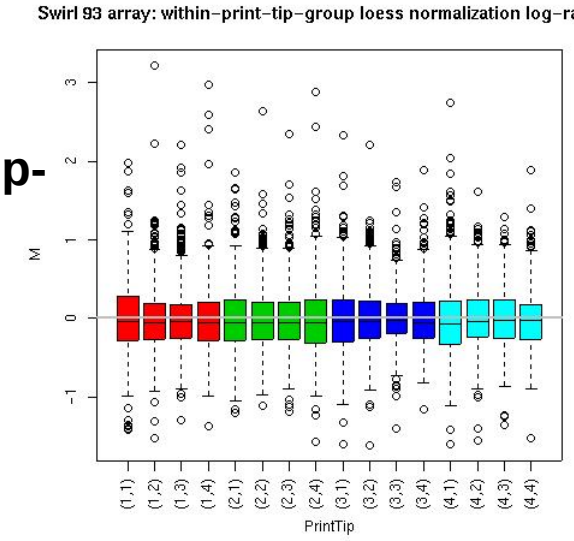
Global median normalization

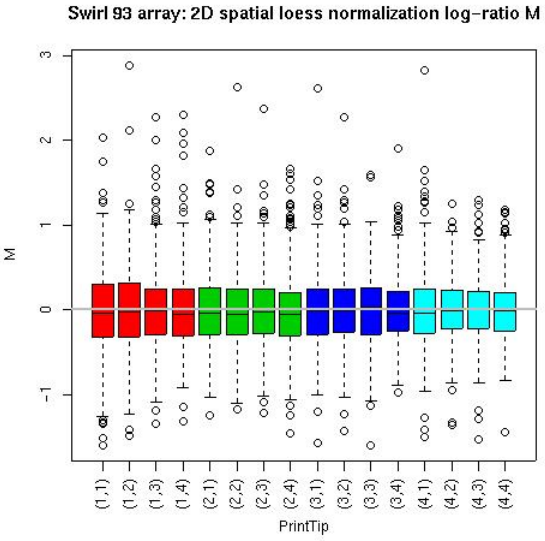




Global loess normalization

Within-print-tip-group loess normalization

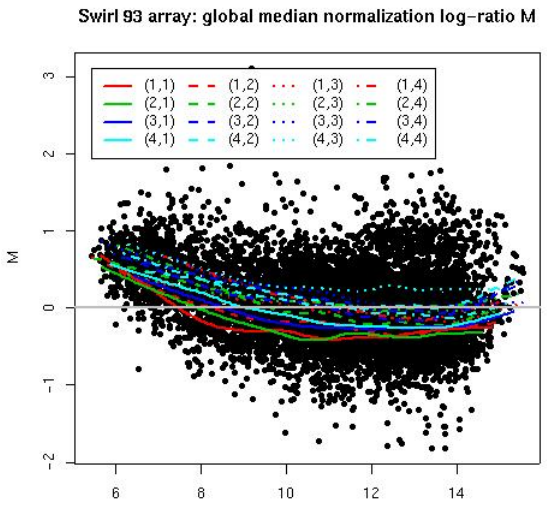


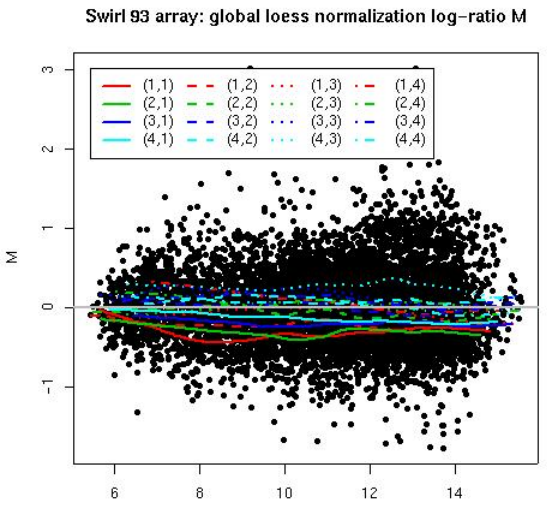


2D spatial normalization

MA-plots of normalized M-L

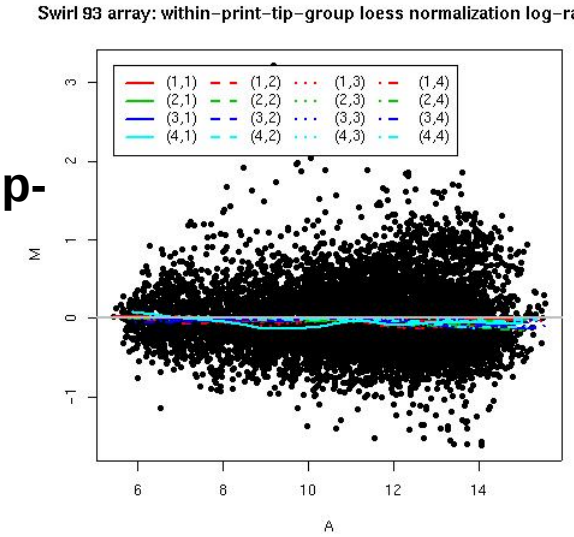
Global median normalization

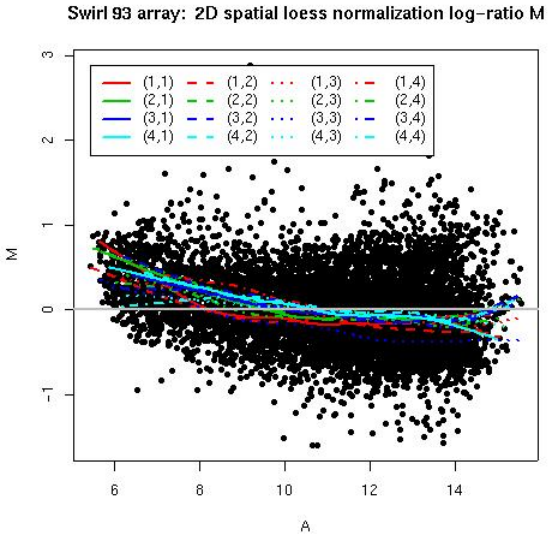




Global loess normalization

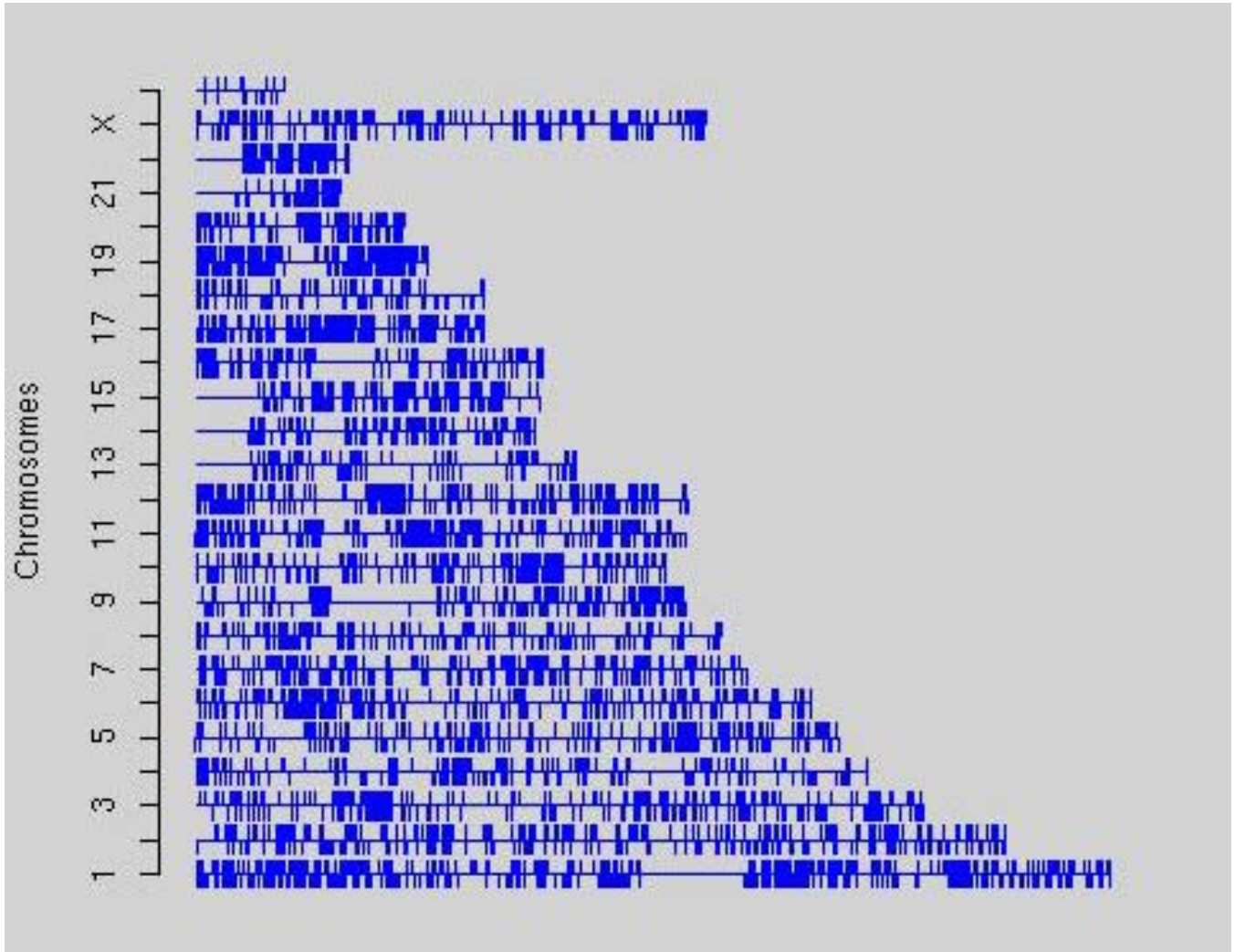
Within-print-tip-group loess normalization



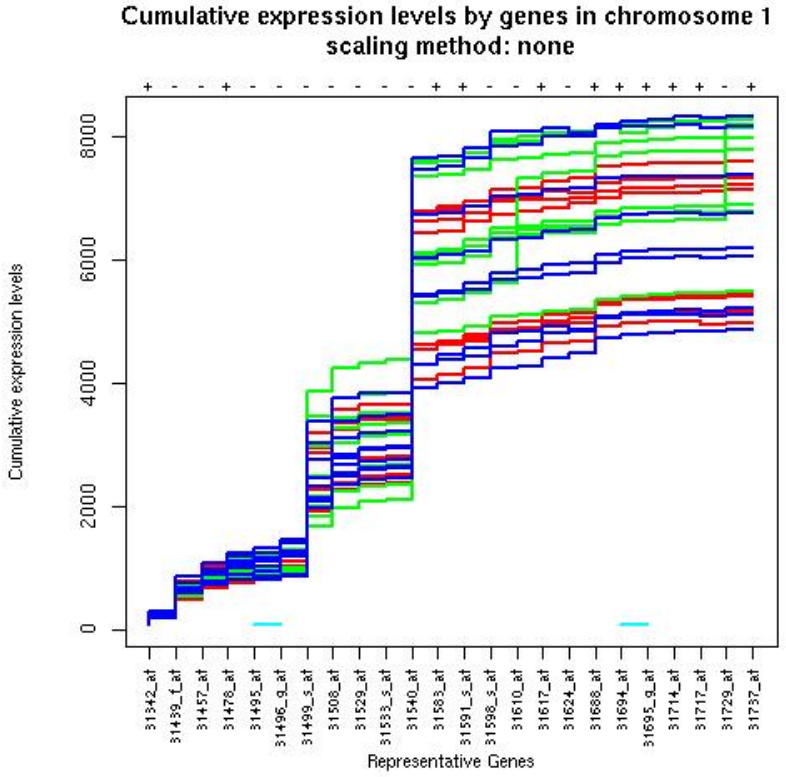


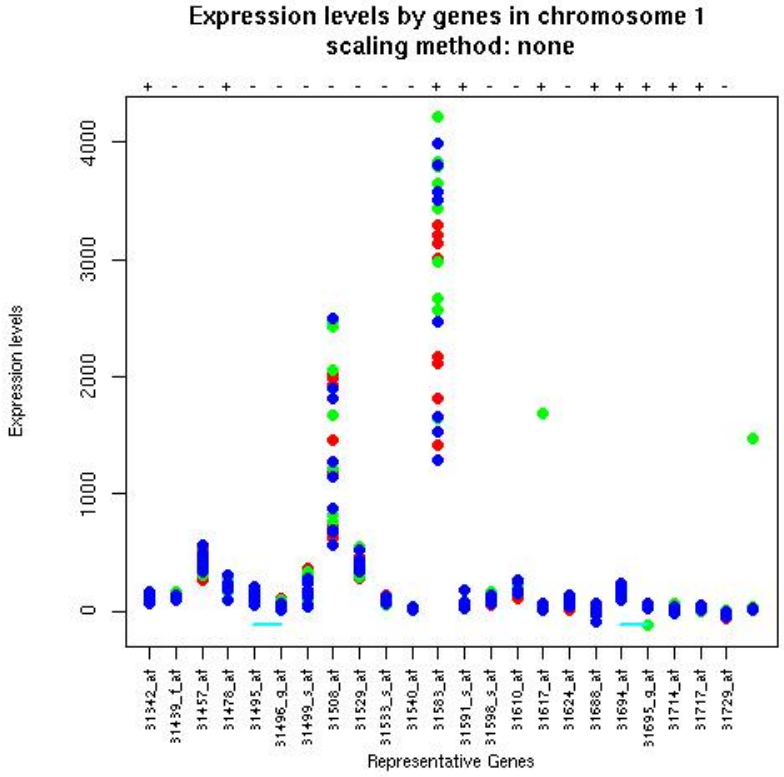
2D spatial normalization

geneplotter: cPlot

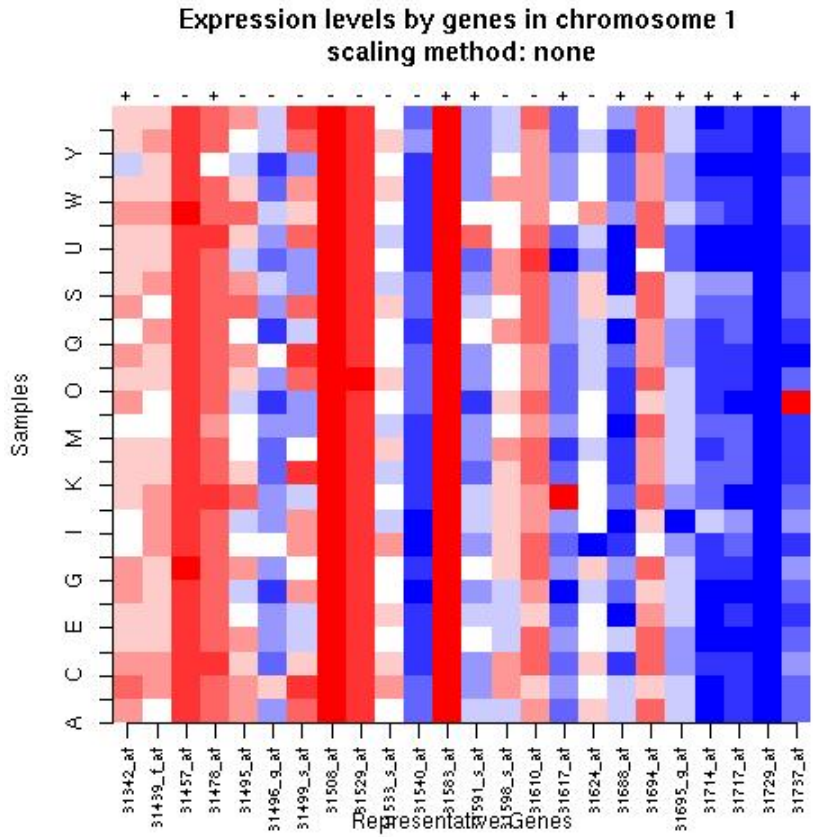


geneplotter: alongChrom

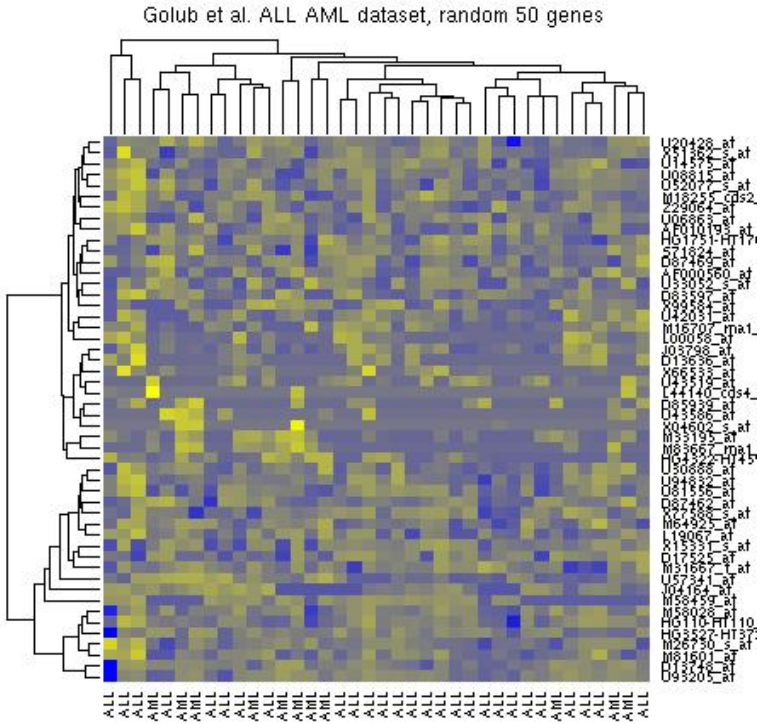


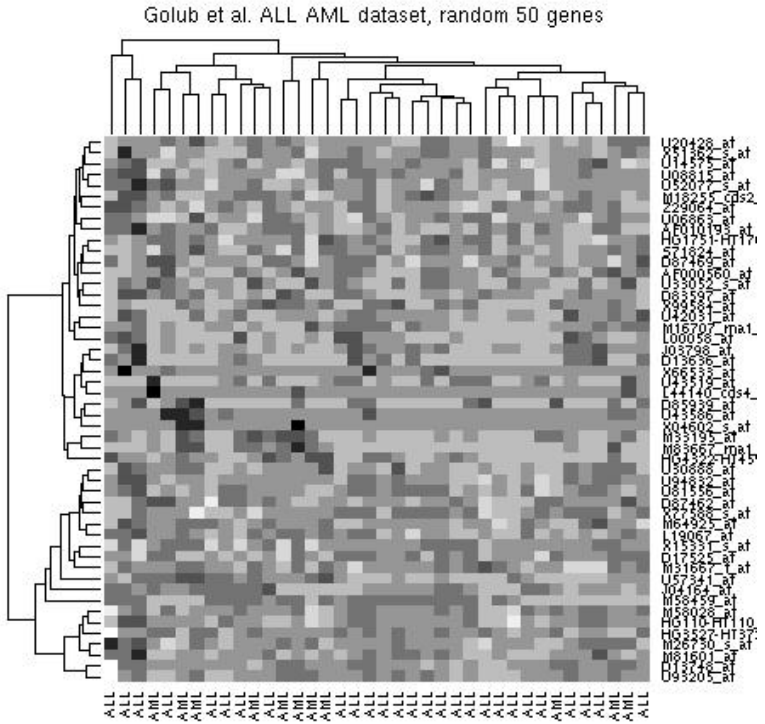


geneplotter: alongChrom

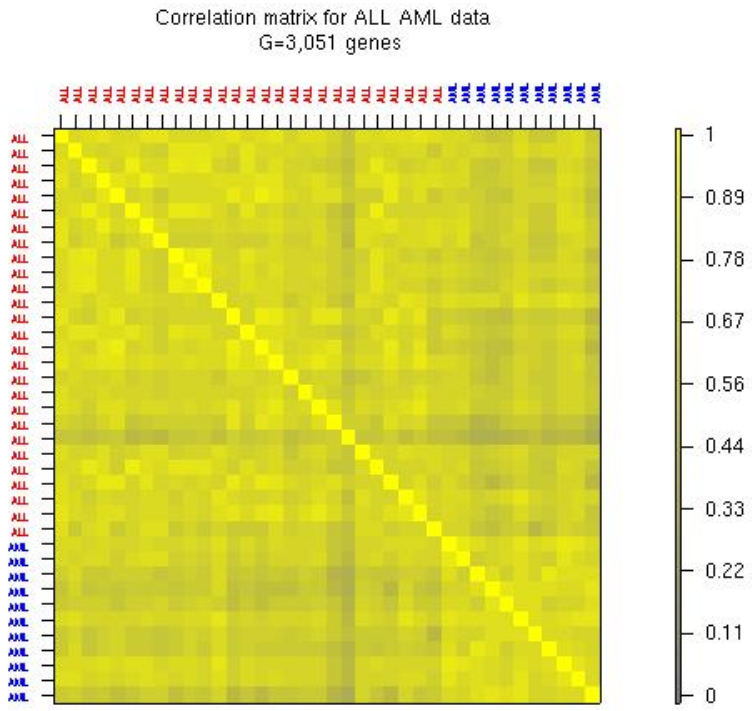


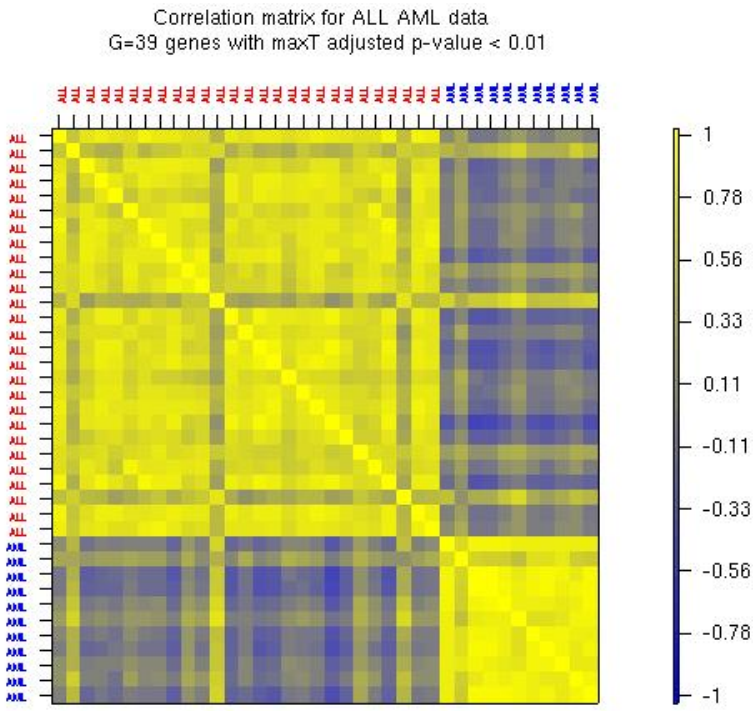
mva: heatmap


Golub et al. ALL AML dataset, random 50 genes


Golub et al. ALL AML dataset, random 50 genes

Correlation matrices





`plot.cor` function from **`sma`** package

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`**).

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`**),
  - HOPACH (coming soon!).
- **`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.

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.

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`**).