



# 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](http://www.stat.berkeley.edu/~sandrine)

EMBO Practical Course on Analysis and Informatics of Microarray Data

Wellcome Trust Genome Campus, Hinxton, Cambridge, UK

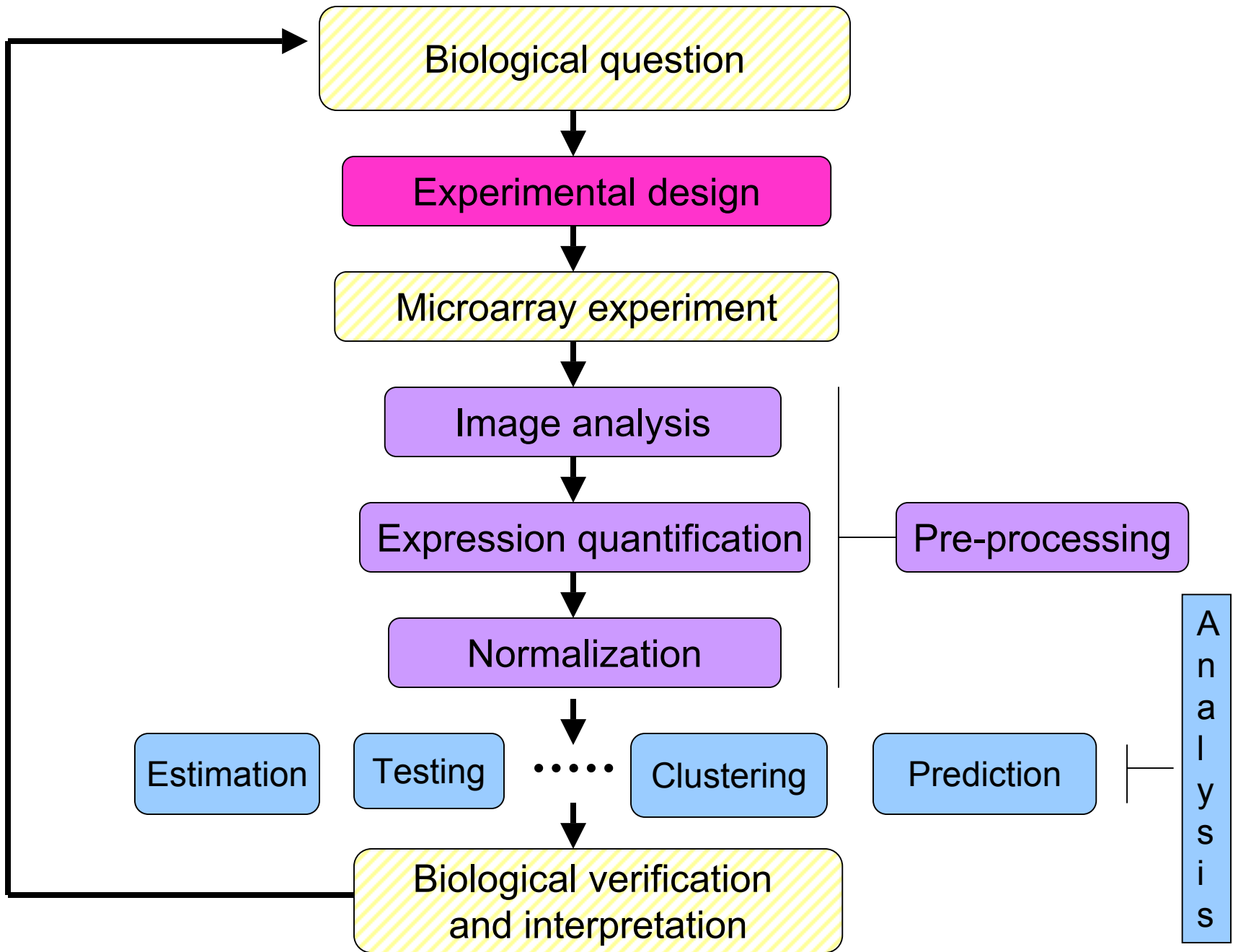
March 18, 2003

© Copyright 2003, all rights reserved

Materials from Bioconductor short  
courses developed with

Robert Gentleman,  
Rafael Irizarry.

Expanded version of this course: Fred  
Hutchinson Cancer Research Center,  
December 2002



# Statistical computing

## Everywhere ...

- Statistical design and analysis:
  - image analysis, normalization, estimation, testing, clustering, prediction, etc.
- Integration of experimental data with biological metadata from WWW-resources
  - gene annotation (GenBank, LocusLink);
  - literature (PubMed);
  - graphical (pathways, chromosome maps).

# Outline

- Overview of the Bioconductor project
- Annotation
- Visualization
- Pre-processing: spotted and Affy arrays
- Differential gene expression
- Clustering and classification

# Acknowledgments

- **Bioconductor core team**
- **Vince Carey**, Biostatistics, Harvard
- **Yongchao Ge**, Statistics, UC Berkeley
- **Robert Gentleman**, Biostatistics, Harvard
- **Jeff Gentry**, Dana-Farber Cancer Institute
- **Rafael Irizarry**, Biostatistics, Johns Hopkins
- **Yee Hwa (Jean) Yang**, Biostatistics, UCSF
- **Jianhua (John) Zhang**, Dana-Farber Cancer Institute

# References

- **Personal webpage**

[www.stat.berkeley.edu/~sandrine](http://www.stat.berkeley.edu/~sandrine)

articles and talks on: image analysis; normalization; identification of differentially expressed genes; cluster analysis; classification.

- **Bioconductor** [www.bioconductor.org](http://www.bioconductor.org)

- software, data, and documentation (vignettes);
- training materials from short courses;
- mailing list.

- **R** [www.r-project.org](http://www.r-project.org)

- software; documentation; RNews.

# Bioconductor

- Bioconductor is an open source and open development software project for the analysis and comprehension of biomedical and genomic data.
- Software, data, and documentation are available from [www.bioconductor.org](http://www.bioconductor.org).



# Bioconductor

- The project was started in the Fall of 2001 by Robert Gentleman, at the Biostatistics Unit of the Dana Farber Cancer Institute.
- There are currently 21 core developers, at various institutions in the US and Europe.
- R and the R package system are used to design and distribute software ([www.r-project.org](http://www.r-project.org)).
- First release (v 1.0): May 2<sup>nd</sup>, 2002, 15 packages.
- Second release (v 1.1): November 18<sup>th</sup>, 2002, 5 new packages.

# Bioconductor

There are two main classes of packages

- **End-user packages:**
  - aimed at users unfamiliar with R or computer programming;
  - polished and easy to use interfaces to a wide variety of computational and statistical methods for the analysis of genomic data.
- **Developer packages:** aimed at software developers, in the sense that they provide ``software to write software".

# Bioconductor packages

Release 1.1, November 18<sup>th</sup>, 2002

- General infrastructure:  
`Biobase`, `reposTools`, `rhdf5`, `tkWidgets`.
- Annotation:  
`annotate`, `AnnBuilder` → data packages.
- Graphics:  
`geneplotter`, `hexbin`.
- Pre-processing for Affymetrix oligonucleotide chip data:  
`affy`, `vsn`, CDF packages.
- Pre-processing for spotted DNA microarray data:  
`marrayClasses`, `marrayInput`, `marrayNorm`, `marrayPlots`,  
`marrayTools`, `vsn`.
- Differential gene expression:  
`edd`, `genefilter`, `multtest`, `ROC`.
- Graphs:  
`graph`.

# Ongoing efforts

- Variable (feature) selection;
- Prediction;
- Cluster analysis;
- Cross-validation;
- Multiple testing;
- Quality measures for microarray data;
- Interactions with MAGE-ML;
- Biological sequence analysis;
- Etc.

# Computing needs

- 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 **WWW** 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

- Interactive tools for linking experimental data in real time, to [biological metadata from WWW resources](#).  
E.g. PubMed, GenBank, LocusLink.
- Scenario. Normalize spotted array data with [marrayNorm](#), obtain list of differentially expressed genes from [multtest](#) 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.

# Bioconductor

- **Widgets.** Small-scale graphical user interfaces (GUI), providing point & click access for specific tasks (**tkWidgets**).
- E.g. File browsing and selection for data input, basic analyses.
- **Object-oriented class/method design.** Allows efficient representation and manipulation of large and complex biological datasets of multiple types (cf. MIAME standards).

# Object-oriented programming

- The Bioconductor project has adopted the **object-oriented programming – OOP –** paradigm presented in J. M. Chambers (1998). *Programming with Data*.
- Tools for programming using the class/method mechanism are provided in the **R methods** package.
- Tutorial: [www.omegahat.org/RSMETHODS/index.html](http://www.omegahat.org/RSMETHODS/index.html)



# OOP

- A **class** provides a software abstraction of a real world object. It reflects how we think of certain objects and what information these objects should contain.
- Classes are defined in terms of **slots** which contain the relevant data.
- An object is an **instance** of a class.
- A class defines the structure, inheritance, and initialization of objects.

# OOP

- A **method** is a function that performs an action on data (objects).
- Methods define how a particular function should behave depending on the class of its arguments.
- Methods allow computations to be adapted to particular data types, i.e., classes.
- A **generic function** is a dispatcher, it examines its arguments and determines the appropriate method to invoke.
- Examples of generic functions include `plot`, `summary`, `print`.

# 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.

# 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.

# marrayRaw class

## Pre-normalization intensity data for a batch of arrays

|           |      |   |
|-----------|------|---|
| maRf      | maGf | Matrix of red and green foreground intensities                |
| maRb      | maGb | Matrix of red and green background intensities                |
| maW       |      | Matrix of spot quality weights                                |
| maLayout  |      | Array layout parameters - <b>marrayLayout</b>                 |
| maGnames  |      | Description of spotted probe sequences<br>- <b>marrayInfo</b> |
| maTargets |      | Description of target samples - <b>marrayInfo</b>             |
| maNotes   |      | Any notes   |

# AffyBatch class

Probe-level intensity data for a batch of arrays (same CDF)

|                          |   |  |
|--------------------------|---|--|
| <code>cdfName</code>     | Name of CDF file for arrays in the batch                          |  |
| <code>nrow</code>        | <code>ncol</code>   | Dimensions of the array  |
| <code>exprs</code>       | <code>se.exprs</code>   | Matrices of probe-level intensities and SEs<br>rows → probe cells, columns → arrays. |
| <code>phenoData</code>   | Sample level covariates, instance of class <code>phenoData</code> |  |
| <code>annotation</code>  | Name of annotation data   |  |
| <code>description</code> | MIAME information   |  |
| <code>notes</code>       | Any notes   |  |

# exprSet class

**exprs**

Matrix of expression measures, genes x samples

**se.exprs**

Matrix of SEs for expression measures, genes x samples

**phenoData**

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

**annotation**

Name of annotation data

**description**

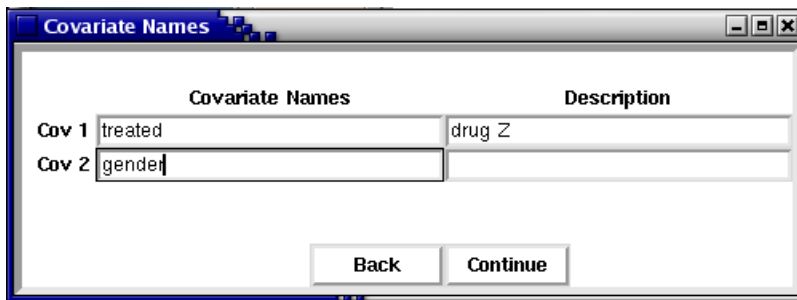
MIAME information

**notes**

Any notes



# Reading in phenoData

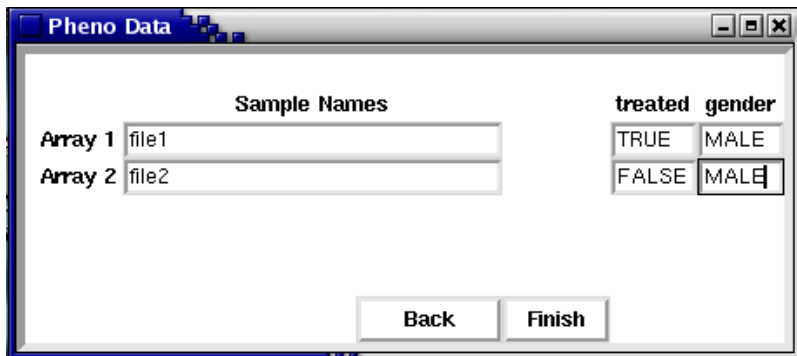


A dialog box titled "Covariate Names" with a table for entering covariate information. The table has two columns: "Covariate Names" and "Description".

|       | Covariate Names | Description |
|-------|-----------------|-------------|
| Cov 1 | treated         | drug Z      |
| Cov 2 | gender          |             |

Buttons: Back, Continue

tkSampleNames

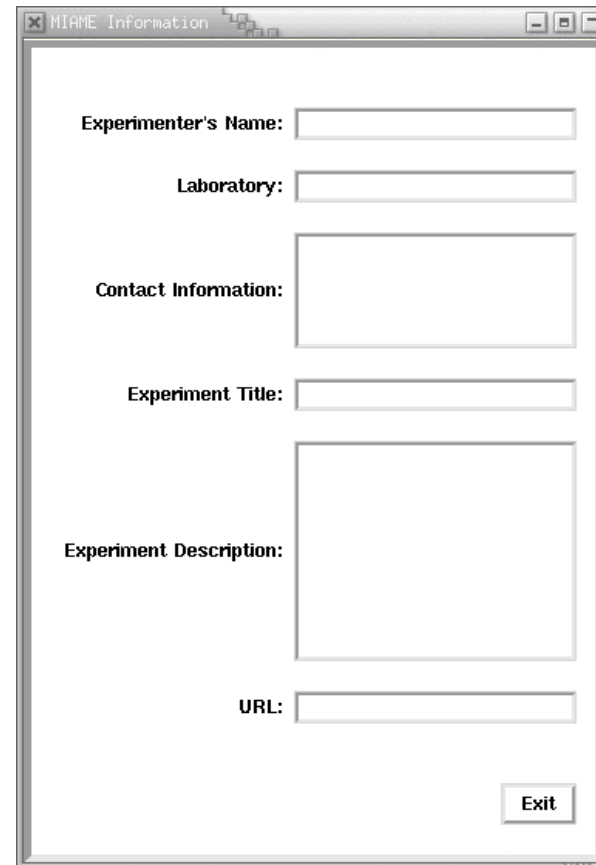


A dialog box titled "Pheno Data" with a table for entering sample information. The table has three columns: "Sample Names", "treated", and "gender".

|         | Sample Names | treated | gender |
|---------|--------------|---------|--------|
| Array 1 | file1        | TRUE    | MALE   |
| Array 2 | file2        | FALSE   | MALE   |

Buttons: Back, Finish

tkphenoData



A dialog box titled "MIAME Information" with several input fields for experiment details.

Fields:

- Experimenter's Name:
- Laboratory:
- Contact Information:
- Experiment Title:
- Experiment Description:
- URL:

Buttons: Exit

tkMIAME

# Pedagogy

Extensive documentation and training resources for R and Bioconductor are available on the WWW.

- **R manuals and tutorials** are available from the R website.
- **R help system**
  - detailed on-line documentation, available in text, HTML, PDF, and LaTeX formats;
  - e.g. `help(genefilter)` , `?pubmed`.
- **R demo system**
  - user-friendly interface for running demonstrations of R scripts;
  - e.g. `demo(marrayPlots)` , `demo(affy)` .
- **Bioconductor short courses**
  - modular training segments on software and statistical methodology;
  - lectures and computer labs available on WWW for self-instruction.

# Vignettes

- Bioconductor has adopted a new documentation paradigm, the vignette.
- A **vignette** is an **executable document** consisting of a collection of documentation text and code chunks.
- Vignettes form **dynamic, integrated, and reproducible statistical documents** that can be automatically updated if either data or analyses are changed.
- Vignettes can be generated using the **Sweave** function from the R **tools** package.

# Vignettes

- Each Bioconductor package contains at least one vignette, located in the `doc` subdirectory of an installed package and accessible from the help browser.
- Vignettes provide task-oriented descriptions of the package's functionality and can be used interactively.
- Vignettes are available separately from the Bioconductor website or as part of the packages.

# Vignettes

- Tools are being developed for managing and using this repository of step-by-step tutorials
  - **Biobase**: `openVignette` – Menu of available vignettes and interface for viewing vignettes (PDF).
  - **tkWidgets**: `vExplorer` – Interactive use of vignettes.
  - **reposTools**.

# Sweave

- The **Sweave** system allows the generation of integrated statistical documents intermixing text, code, and code output (textual and graphical).
- Functions are available in the R **tools** package.
- See ? **Sweave** and manual  
[www.ci.tuwien.ac.at/~leisch/Sweave/](http://www.ci.tuwien.ac.at/~leisch/Sweave/)

# Sweave input

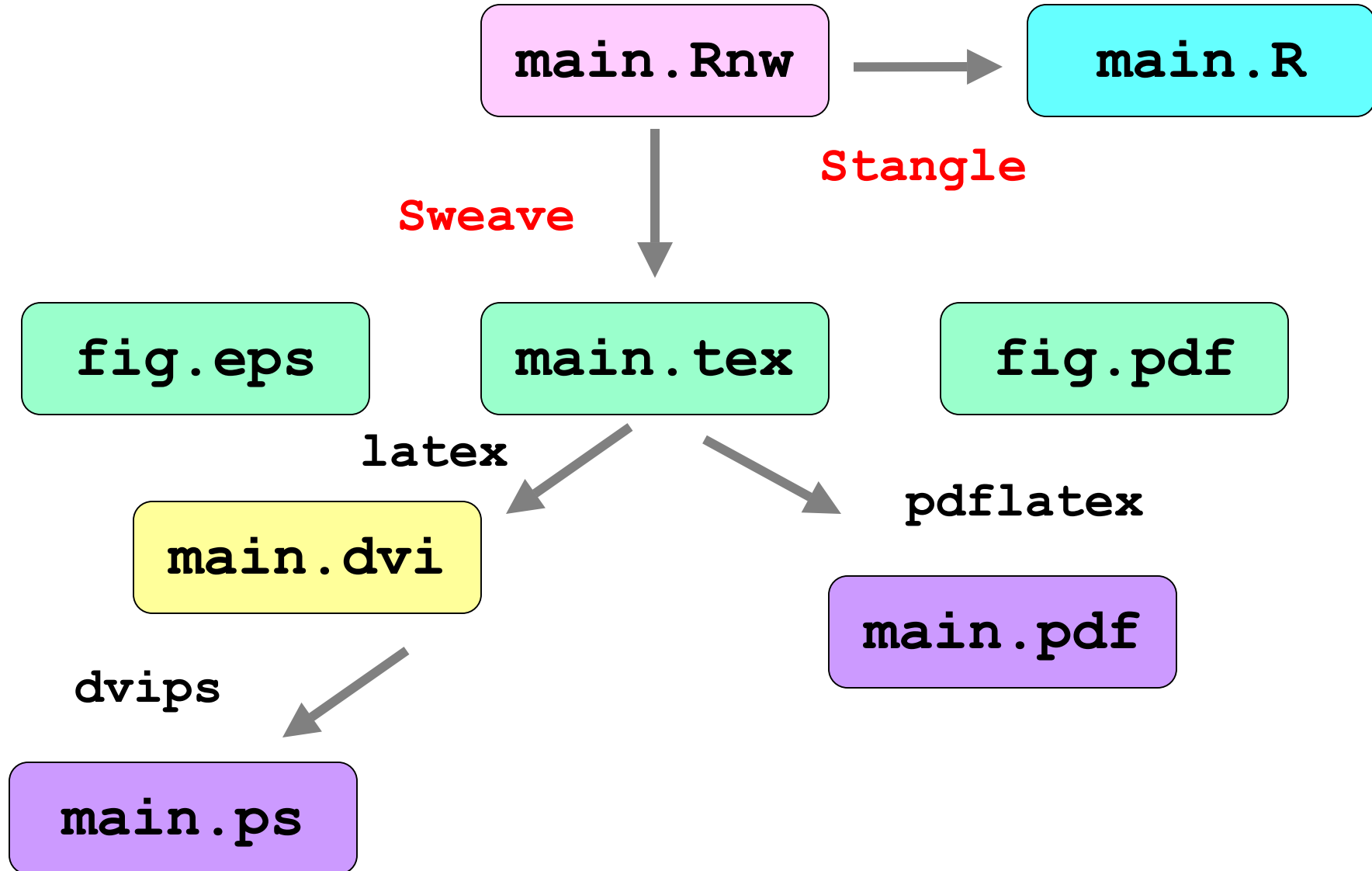
- Input: a text file which consists of a sequence of code and documentation **chunks**, or segments (noweb file).
  - **Documentation chunks**
    - start with @
    - can be text in a markup language like LaTeX.
  - **Code chunks**
    - start with `<<name>>=`
    - can be R or S-Plus code.
  - File extension: `.rnw`, `.Rnw`, `.snw`, `.Snw`.

# Sweave output

- Output: a single document, e.g., `.tex` file or `.pdf` file containing
  - the documentation text,
  - the R code,
  - the code output: text and graphs.
- The document can be automatically regenerated whenever the data, code, or documentation text change.
- **Stangle** or **tangleToR**: extract only the code.



# Sweave



# Annotation

# annotate package

- 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<br>HGU95A chips | "41046_s_at"                         |
| LocusLink, LocusID                    | "9203"                               |
| GenBank accession #                   | "X95808"                             |
| Gene symbol                           | "ZNF261"                             |
| PubMed, PMID                          | "10486218"<br>"9205841"<br>"8817323" |
| Chromosomal location                  | "X", "Xq13.1"                        |

# Annotation data packages

- The Bioconductor project provides packages that contain only **data**.
- These data packages are built using **AnnBuilder**.
- They can be downloaded from the Bioconductor website and also using **update.packages**.
- Data packages contain many different mappings to interesting data.
  - Mappings between Affy IDs and other probe IDs: **hgu95a** for HGU95A GeneChip series, also, **hgu133a**, **hu6800**, **mgu74a**, **rgu34a**.
  - Affy CDF data packages.
- The packages are updated and expanded regularly as updated and new data become available.

# 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**.
- 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

```
> 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 = hgu95aSUFUNC)
[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: matching IDs**

- Instead of relying on the general R functions for environments, new user-friendly functions have been written for accessing and working with specific identifiers.
- E.g. `getGO`, `getGODesc`, `getLL`, `getPMID`, `getSYMBOL`.

# annotate: matching IDs

```
> getSYMBOL("41046_s_at", data="hgu95a")
41046_s_at
"ZNF261"

> gg<- getGO("41046_s_at", data="hgu95a")
> getGODesc(gg, "MF")
$"c("GO:0003677", "GO:0007275")"
[1] "DNA binding"

> getLL("41046_s_at", data="hgu95a")
41046_s_at
9203

> getPMID("41046_s_at", data="hgu95a")
$"41046_s_at"
[1] 10486218 9205841 8817323
```

# 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

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

- The **XML** package is used to parse query results.

# annotate: querying GenBank

[www.ncbi.nlm.nih.gov/Genbank/index.html](http://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.nlm.nih.gov/entrez/query.fcgi?tool=biocductor&cmd=Search&db=Nucleotide&term=X95808>

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

# annotate: querying LocusLink

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



# annotate: querying PubMed

[www.ncbi.nlm.nih.gov](http://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.

# **annotate: PubMedAbst class**

Class structure for storing and processing PubMed abstracts in R

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

# **annotate: high-level tools for querying PubMed**

- **pm.getabst**: download the specified PubMed abstracts (stored in XML) and create a list of **pubMedAbst** objects.
- **pm.titles**: extract the titles from a list 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](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: PubMed HTML report**

- The new function `pmAbst2html` takes a list of `pubMedAbst` objects and generates an HTML report with the titles of the abstracts and links to their full page on PubMed.

```
pmAbst2html (absts [[1]], filename="pm.html")
```

BioConductor Abstract List - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: file:///C:/Sandrine/Current/Talks/EMBO03/pm.html What's Related

Google Sandrine Dudoit Welcome to Bioc PH 240D - Sprin Group In Biosta Berkeley Progra Home Page, Stat

## BioConductor Abstract List

| Article Title  | Publication Date |
|--|------------------|
| <a href="#">Conditional targeting of the DNA repair enzyme hOGG1 into mitochondria.</a>  | Nov 2002         |
| <a href="#">Inter-individual variation, seasonal variation and close correlation of OGG1 and ERCC1 mRNA levels in full blood from healthy volunteers.</a>                    | Sep 2002         |
| <a href="#">A limited association of OGG1 Ser326Cys polymorphism for adenocarcinoma of the lung.</a>   | May 2002         |
| <a href="#">Protection of human lung cells against hyperoxia using the DNA base excision repair genes hOgg1 and Fpg.</a>   | Jul 2002         |
| <a href="#">The human OGG1 DNA repair enzyme and its association with orolaryngeal cancer risk.</a>  | Jul 2002         |
| <a href="#">Human OGG1 undergoes serine phosphorylation and associates with the nuclear matrix and mitotic chromatin in vivo.</a>  | Jun 2002         |
| <a href="#">hOGG1 Ser(326)Cys polymorphism and modification by environmental factors of stomach cancer risk in Chinese.</a>  | Jun 2002         |
| <a href="#">Association of the hOGG1 Ser326Cys polymorphism with lung cancer risk.</a>   | Apr 2002         |
| <a href="#">Reciprocal "flipping" underlies substrate recognition and catalytic activation by the human 8-oxo-guanine DNA glycosylase.</a>                                   | Mar 2002         |
| <a href="#">Expression of 8-oxoguanine DNA glycosylase is reduced and associated with neurofibrillary tangles in Alzheimer's disease brain.</a>                              | Jan 2002         |
| <a href="#">Structure and chromosome location of human OGG1.</a>   | Month 1999       |
| <a href="#">Expression and differential intracellular localization of two major forms of human 8-oxoguanine DNA glycosylase encoded by alternatively spliced OGG1 mRNAs.</a> | May 1999         |
| <a href="#">Genetic polymorphisms and alternative splicing of the hOGG1 gene, that is involved in the repair of 8-hydroxyguanine in damaged DNA.</a>                         | Jun 1998         |
| <a href="#">Augmented expression of a human gene for 8-oxoguanine DNA glycosylase (MutM) in B lymphocytes of the dark zone in lymph node germinal centers.</a>               | Nov 1997         |
| <a href="#">Opposite base-dependent reactions of a human base excision repair enzyme on DNA containing 7,8-dihydro-8-oxoguanine and abasic sites.</a>                        | Oct 1997         |
| <a href="#">Molecular cloning and functional expression of a human cDNA encoding the antimutator enzyme 8-hydroxyguanine-DNA glycosylase.</a>                                | Jul 1997         |
| <a href="#">Cloning and characterization of hOGG1, a human homolog of the OGG1 gene of Saccharomyces cerevisiae.</a>   | Jul 1997         |

Document: Done

pmAbst2html  
function from  
annotate package

[pm.html](#)

# annotate: analysis reports

- A simple interface, [ll.htmlpage](#), can be used to generate an HTML report of analysis 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 |
|------------------------|------------------|------------|----------|----------|-------------|-------------|-------------|
| <a href="#">7791</a>   | X95735_at        | 7          | -0.295   | 1.59     | -10.6       | 2e-05       | 2e-05       |
| <a href="#">1471</a>   | M27891_at        | 20         | -0.81    | 2.08     | -9.78       | 2e-05       | 2e-05       |
| <a href="#">2184</a>   | M55150_at        | 15         | 0.488    | 1.24     | -8.03       | 2e-05       | 0.00014     |
| <a href="#">4067</a>   | M16038_at        | 8          | -0.284   | 1.1      | -7.98       | 2e-05       | 0.00016     |
| <a href="#">334</a>    | L09209_s_at      | 11         | -0.162   | 1.36     | -7.97       | 2e-05       | 2e-04       |
| <a href="#">6929</a>   | M31523_at        | 19         | 0.855    | -0.391   | 7.55        | 2e-05       | 5e-04       |
| <a href="#">5928</a>   | X74262_at        | 1          | 0.869    | -0.565   | 7.42        | 2e-05       | 0.00078     |
| <a href="#">7155</a>   | Z15115_at        | 3          | 1.94     | 0.945    | 7.35        | 2e-05       | 0.001       |
| <a href="#">26999</a>  | L47738_at        | 5          | 0.734    | -0.779   | 7.31        | 2e-05       | 0.00114     |
| <a href="#">4602</a>   | U22376_cds2_s_at | 6          | 1.86     | 0.294    | 7.28        | 2e-05       | 0.00116     |
| <a href="#">65108</a>  | HG1612-HT1612_at | 1          | 1.91     | 0.888    | 7.11        | 2e-05       | 0.0017      |
| <a href="#">34</a>     | M91432_at        | 1          | 0.431    | -0.771   | 7.08        | 2e-05       | 0.0018      |
| <a href="#">5925</a>   | L41870_at        | 13         | -0.438   | -1.3     | 7.08        | 2e-05       | 0.0018      |
| <a href="#">546</a>    | U72936_s_at      | NA         | -0.097   | -1.07    | 7.07        | 2e-05       | 0.0018      |
| <a href="#">7430</a>   | X51521_at        | 6          | 1.92     | 1.07     | 7.06        | 2e-05       | 0.00186     |
| <a href="#">4056</a>   | U50136_ma1_at    | 5          | 0.71     | 1.51     | -6.97       | 2e-05       | 0.00232     |
| <a href="#">54741</a>  | Y12670_at        | 1          | -0.167   | 0.892    | -6.96       | 2e-05       | 0.00238     |
| <a href="#">7203</a>   | X74801_at        | 1          | 0.611    | -0.183   | 6.95        | 2e-05       | 0.00238     |
| <a href="#">3576</a>   | Y00787_s_at      | 4          | -0.371   | 2.32     | -6.87       | 2e-05       | 0.00288     |
| <a href="#">6709</a>   | J05243_at        | 9          | 0.413    | -0.982   | 6.86        | 2e-05       | 0.00288     |
| <a href="#">1725</a>   | U26266_s_at      | 19         | -0.209   | -1.16    | 6.85        | 4e-05       | 0.00294     |
| <a href="#">3205</a>   | U82759_at        | 7          | -0.64    | 0.504    | -6.82       | 2e-05       | 0.00306     |
| <a href="#">945</a>    | M23197_at        | 19         | -0.881   | 0.354    | -6.79       | 2e-05       | 0.0033      |
| <a href="#">1509</a>   | M63138_at        | 11         | 1.21     | 2.12     | -6.77       | 2e-05       | 0.00344     |
| <a href="#">6955</a>   | M12959_s_at      | 14         | 1.13     | 0.132    | 6.76        | 2e-05       | 0.00352     |
| <a href="#">967</a>    | X62654_ma1_at    | 12         | 0.0513   | 1.33     | -6.76       | 2e-05       | 0.00352     |
| <a href="#">5341</a>   | X07743_at        | 2          | -0.959   | 0.535    | -6.74       | 2e-05       | 0.00378     |
| <a href="#">140465</a> | M31211_s_at      | 12         | 0.108    | -0.953   | 6.71        | 2e-05       | 0.00404     |
| <a href="#">7336</a>   | U62136_at        | 8          | -0.163   | -0.92    | 6.68        | 2e-05       | 0.00428     |
| <a href="#">3660</a>   | X15949_at        | 4          | -0.541   | -1.33    | 6.61        | 2e-05       | 0.00492     |
| <a href="#">9655</a>   | U72936_s_at      | NA         | -0.097   | -1.07    | 7.07        | 2e-05       | 0.0018      |

l1.htmlpage  
function from  
**annotate**  
package

[genelist.html](#)

# **annotate: chromLoc class**

Location information for one gene

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

# **annotate: chromLocation class**

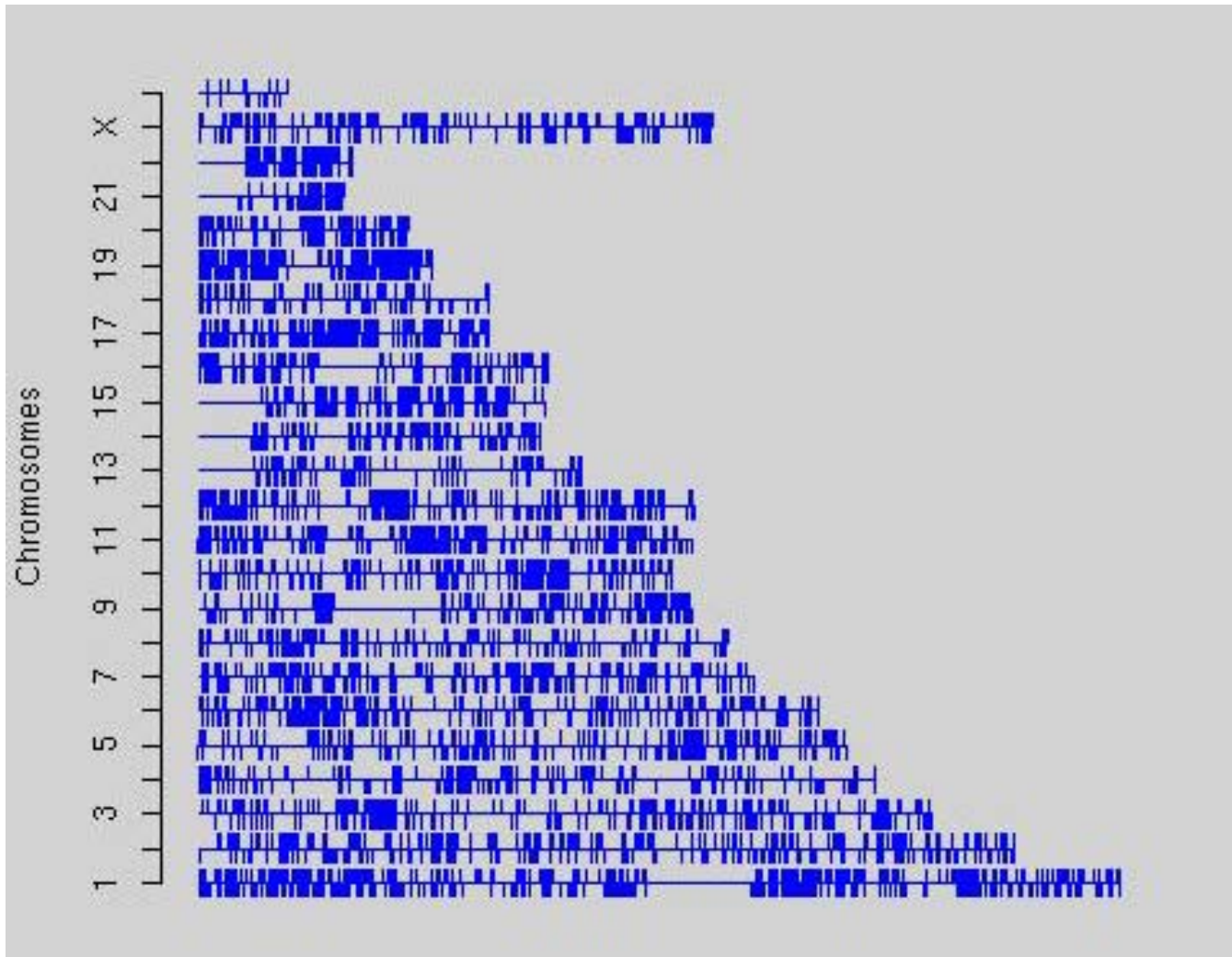
Location information for a set of genes

- **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**.

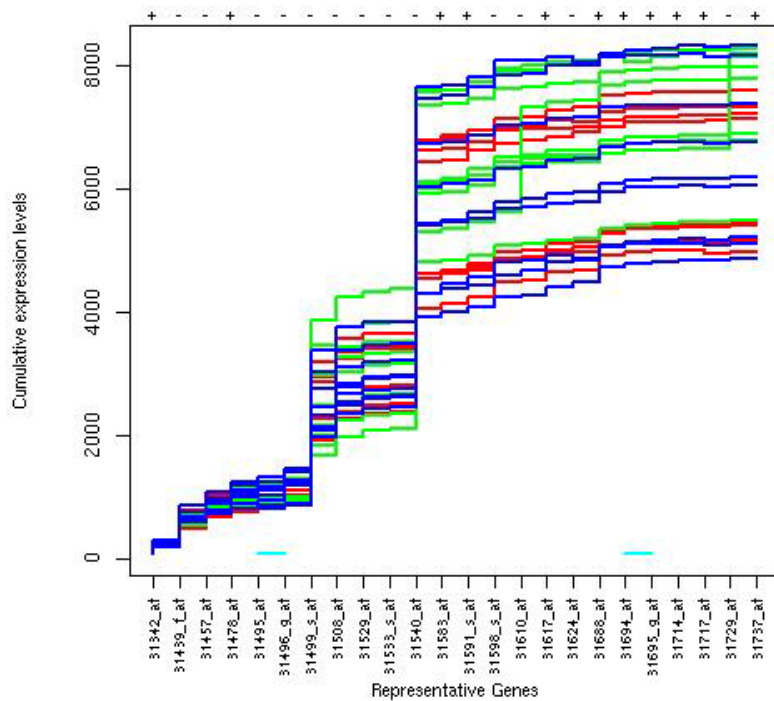
# Visualization

# geneplotter: cPlot

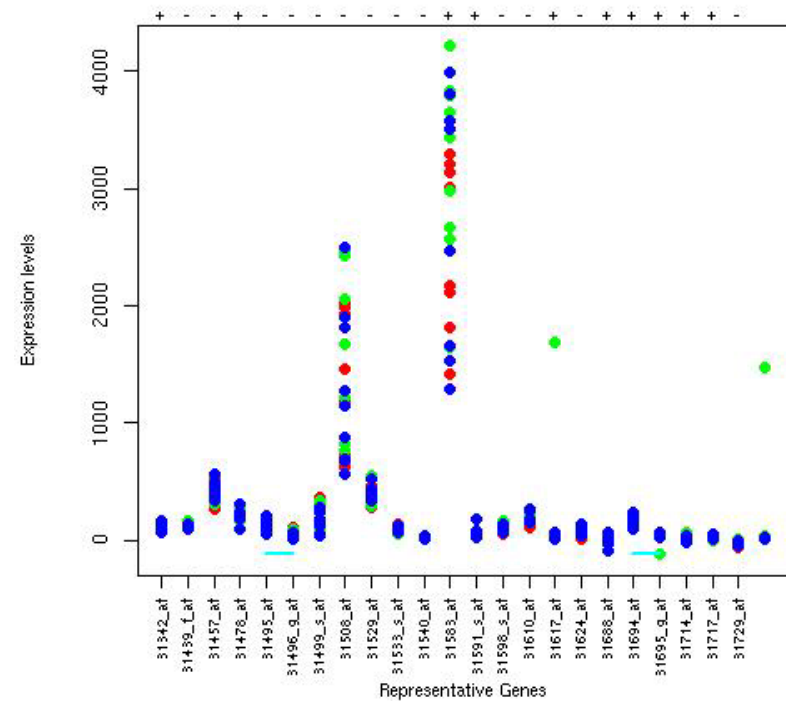


# geneplotter: a longChrom

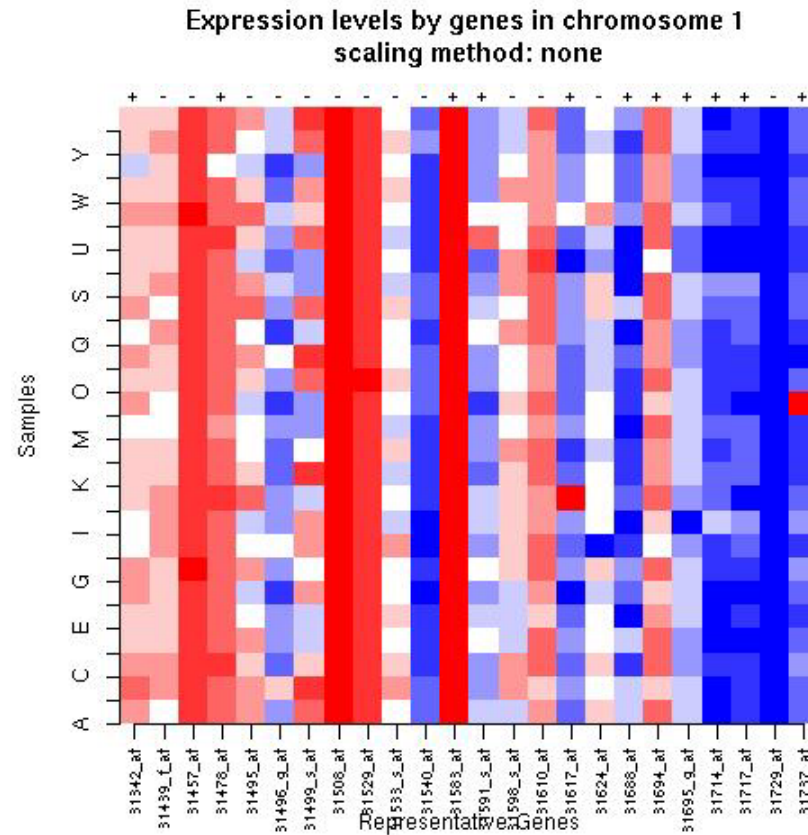
Cumulative expression levels by genes in chromosome 1  
scaling method: none



Expression levels by genes in chromosome 1  
scaling method: none



# genepLOTter: alongChrom



# **Pre-processing DNA microarray data**



# Pre-processing

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

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**.

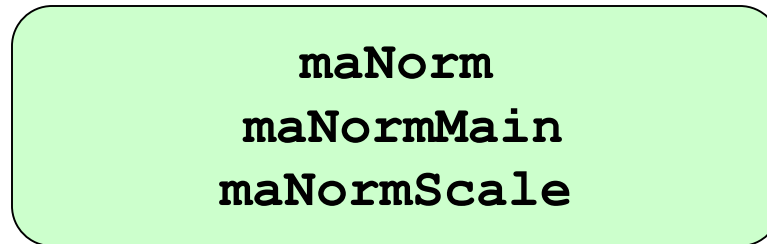
One can then use other Bioconductor packages, e.g., **genefilter**, **geneplotter**.

# marray packages

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



Class `marrayRaw`



Class `marrayNorm`



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

Class `exprSet`

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

# affy package

CEL and CDF  
files



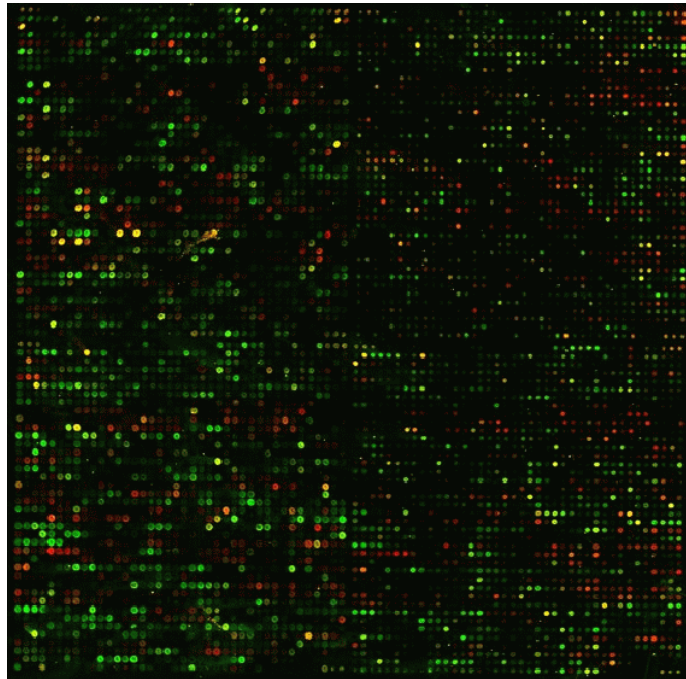
Class `AffyBatch`



Class `exprSet`

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

# Pre-processing: spotted DNA microarrays



# 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, classification, 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;
  - different amounts of Cy3- and Cy5-labeled mRNA;
  - different scanning parameters;
  - print-tip, spatial, or plate effects, etc.

# 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.

# **marray**: Pre-processing spotted DNA microarray data

- **marrayClasses**:
  - class definitions for spotted DNA microarray data (cf. MIAME);
  - basic methods for manipulating microarray objects: printing, plotting, subsetting, class conversions, etc.
- **marrayInput**:
  - reading in intensity data and textual data describing probes and targets;
  - automatic generation of microarray data objects;
  - widgets for point & click interface.
- **marrayPlots**: diagnostic plots.
- **marrayNorm**: robust adaptive location and scale normalization procedures.
- **marrayTools**: miscellaneous tools for functional genomics cores facilities at UCB and UCSF.



# 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 and green foreground intensities                |
| maRb      | maGb | Matrix of red and green background intensities                |
| maW       |      | Matrix of spot quality weights                                |
| maLayout  |      | Array layout parameters - <b>marrayLayout</b>                 |
| maGnames  |      | Description of spotted probe sequences<br>- <b>marrayInfo</b> |
| maTargets |      | Description of target samples - <b>marrayInfo</b>             |
| 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 - <code>marrayLayout</code>                 |   |
| maGnames   | Description of spotted probe sequences<br>- <code>marrayInfo</code> |   |
| maTargets  | Description of target samples - <code>marrayInfo</code>             |   |
| maNormCall | Function call   |   |
| maNotes    | Any notes   |   |

# `marrayInput` package

- `marrayInput` provides functions for reading microarray data into R and creating microarray objects of class `marrayLayout`, `marrayInfo`, and `marrayRaw`.
- Input
  - Image quantitation data, i.e., output files from image analysis software.  
E.g. `.gpr` for **GenePix**, `.spot` for **Spot**.
  - Textual description of probe sequences and target samples.  
E.g. `gal` files, `god` lists.

# marrayInput package

- Widgets for graphical user interface

`widget.marrayLayout`,

`widget.marrayInfo`,

`widget.marrayRaw`.

The screenshot shows a window titled "MarrayRaw builder" with a "Files" button at the top. Below it are several input sections:

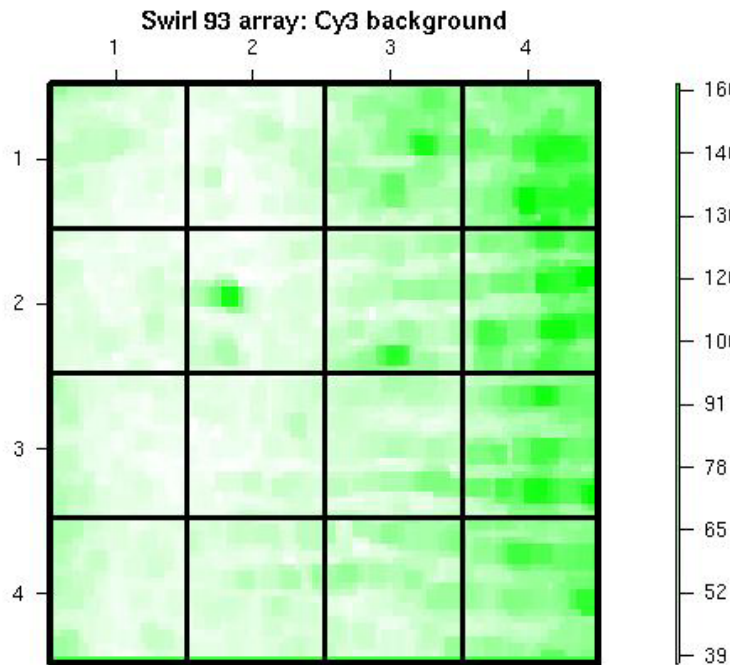
- Name of the marrayRaw object:** A text field containing "swirl".
- Foreground and background intensities:** A section with four input fields: "Green Foreground" (Gmean), "Green Background" (morphG), "Red Foreground" (Rmean), and "Red Background" (morphR). There is also a "Weights" field which is currently empty.
- Layout:** A text field containing "swirl.layout" and a "Browse" button.
- Target Information:** A text field containing "swirl.samples" and a "Browse" button.
- Gene Information:** A text field containing "swirl.gnames" and a "Browse" button.
- Notes:** An empty text area.

At the bottom of the window is a row of five buttons: "Layout", "Target", "Genes", "Build", and "Quit".

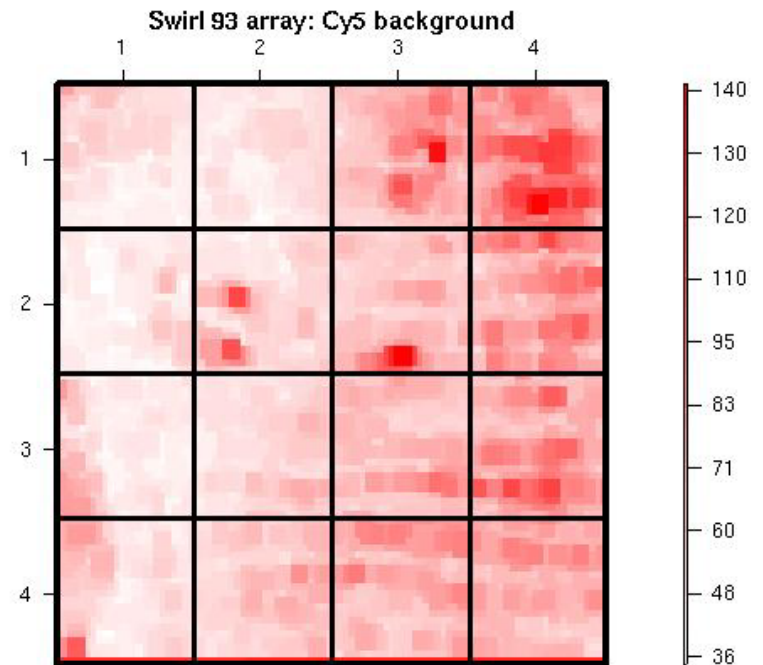
# marrayPlots package

- See demo (`marrayPlots`).
- **Diagnostic plots** of spot statistics.  
E.g. red and green log intensities, intensity log ratios  $M$ , average log intensities  $A$ , spot area.
  - `maImage`: 2D spatial color images.
  - `maBoxplot`: boxplots.
  - `maPlot`: scatter-plots with fitted curves and text highlighted.
- **Stratify** plots according to layout parameters such as `print-tip-group`, `plate`.  
E.g. MA-plots with loess fits by `print-tip-group`.

# 2D spatial images maImage



**Cy3 background intensity**



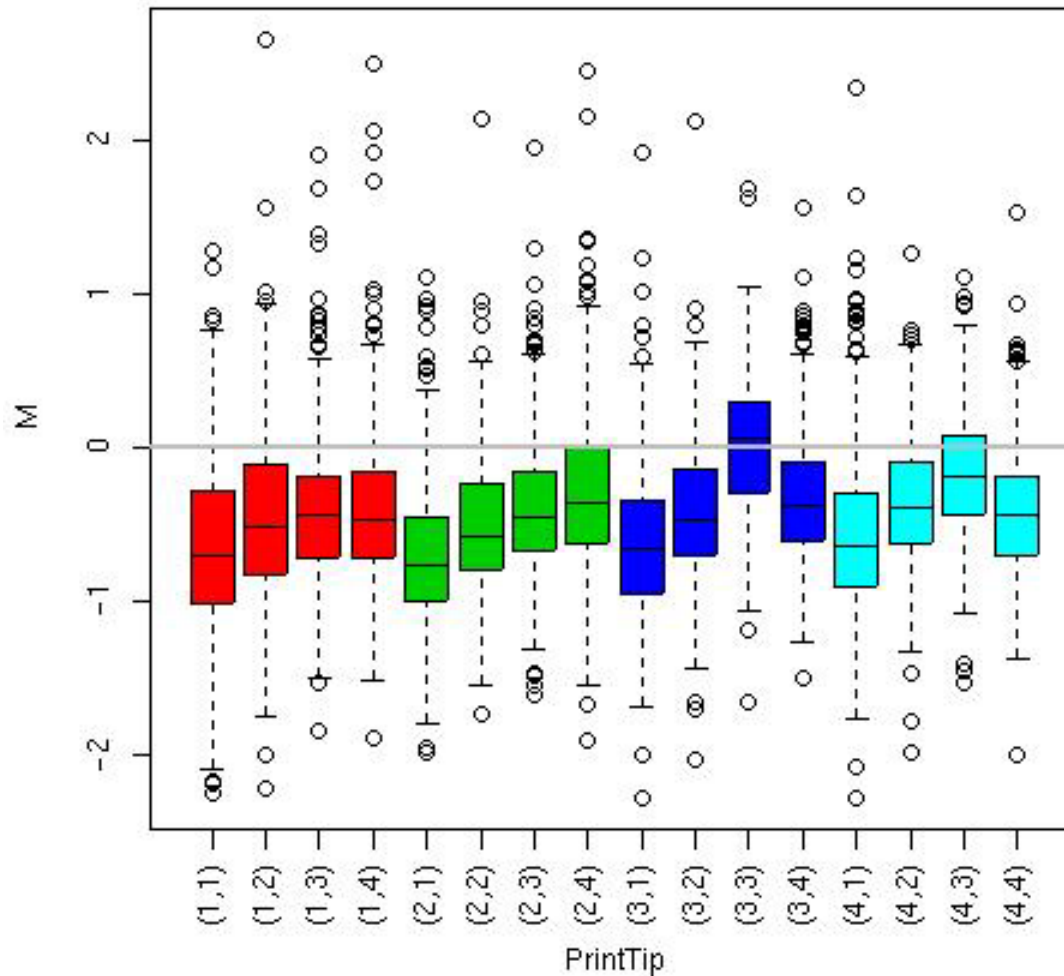
**Cy5 background intensity**

# Boxplots by print-tip-group

## maBoxplot

Swirl 93 array: pre-normalization log-ratio M

Intensity  
log ratio, M



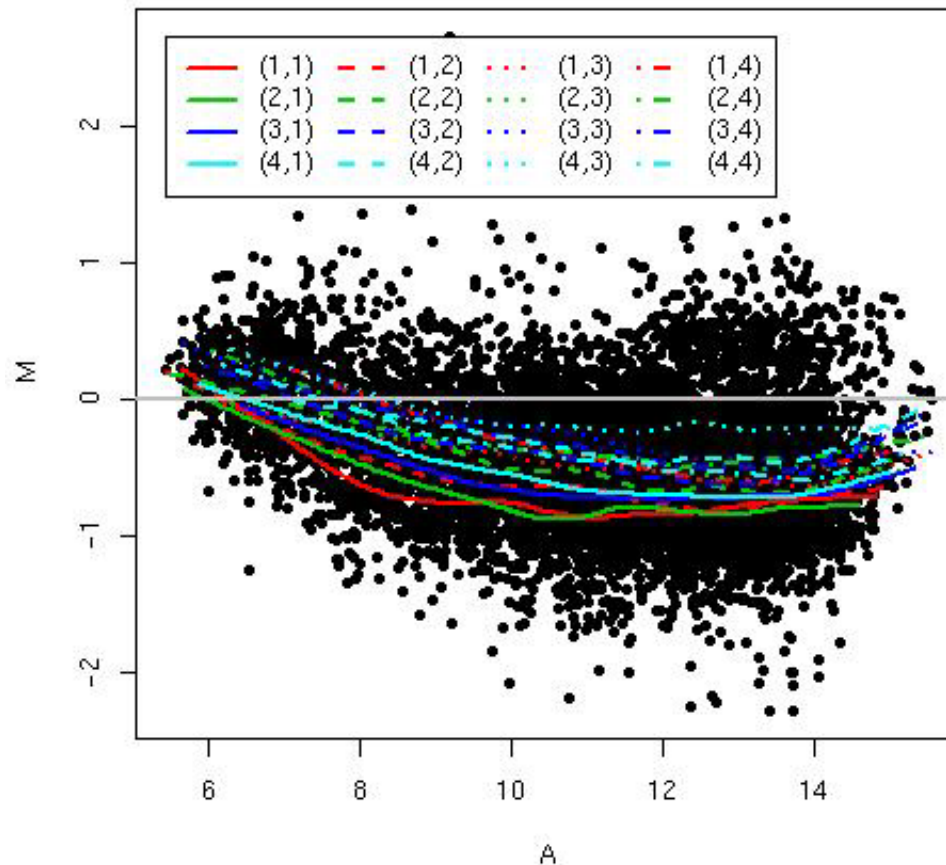


# MA-plot by print-tip-group

## maPlot

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

Swirl 93 array: pre-normalization log-ratio M



Intensity  
log ratio, M

Average  
log intensity, A

# **marrayNorm** package

- **maNormMain**: main normalization function, allows **robust adaptive location and scale normalization** for a batch of arrays
  - intensity or A-dependent location normalization (**maNormLoess**);
  - 2D spatial location normalization (**maNorm2D**);
  - median location normalization (**maNormMed**);
  - scale normalization using MAD (**maNormMAD**);
  - composite normalization;
  - your own normalization function.
- **maNorm**: simple wrapper function.  
**maNormScale**: simple wrapper function for scale normalization.

# **marrayTools** package

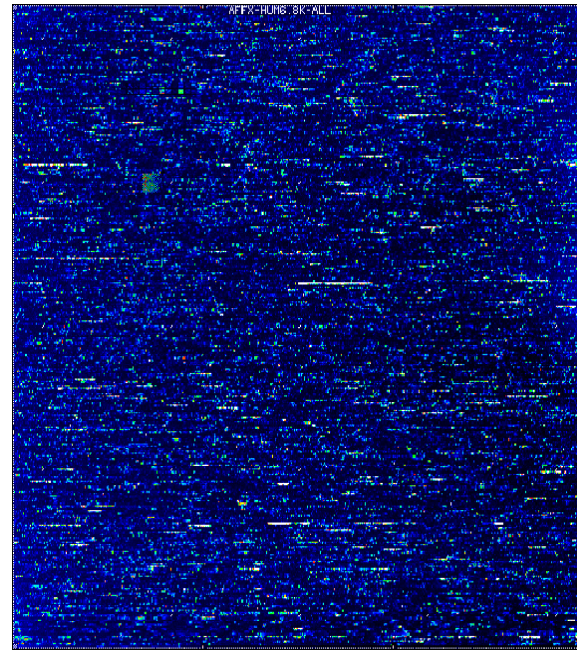
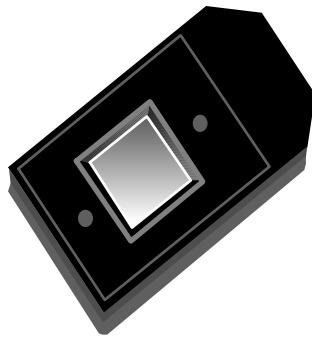
- The **marrayTools** package provides additional functions for handling two-color spotted microarray data (see devel. version).
- The **spotTools** and **gpTools** functions start from Spot and GenePix image analysis output files, respectively, and automatically
  - read in these data into R,
  - perform standard normalization (within print-tip-group loess),
  - create a directory with a standard set of diagnostic plots (jpeg format), excel files of quality measures, and tab delimited files of normalized log ratios  $M$  and average log intensities  $A$ .

# swirl dataset

- Microarrays:
  - 8,448 probes (768 controls);
  - 4 x 4 grid matrix;
  - 22 x 24 spot matrices.
- 4 hybridizations: swirl mutant and wild type mRNA.
- Data stored in object of class `marrayRaw`: `data(swirl)`.
- ```
> maInfo(maTargets(swirl))[ ,3:4]
```

|   | experiment Cy3 | experiment Cy5 |
|---|----------------|----------------|
| 1 | swirl          | wild type      |
| 2 | wild type      | swirl          |
| 3 | swirl          | wild type      |
| 4 | wild type      | swirl          |

# Pre-processing: oligonucleotide chips



# Probe-pair set

## GeneChip® Expression Array Design

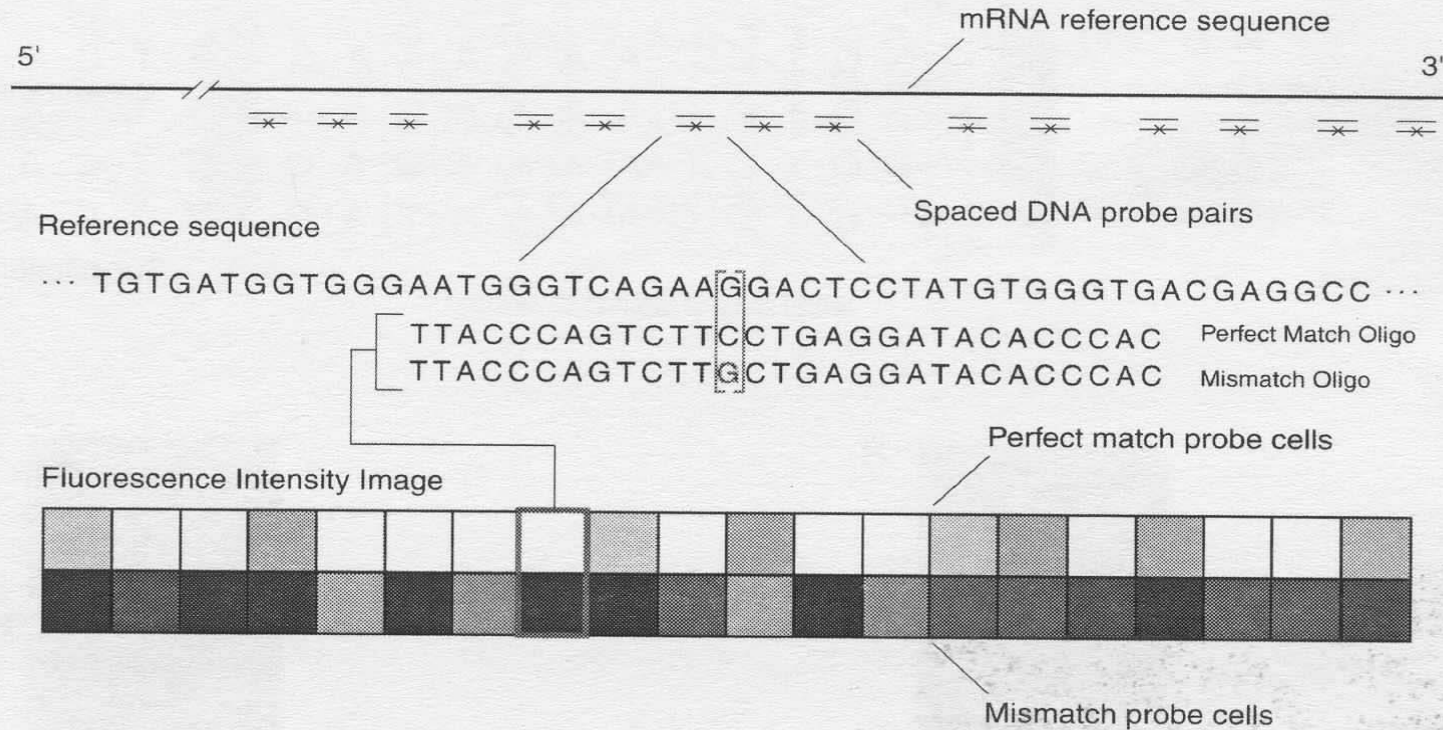


Figure 1-3 Expression tiling strategy

# Terminology

- Each gene or portion of a gene is represented by 16 to 20 oligonucleotides of 25 base-pairs.
- **Probe**: an oligonucleotide of 25 base-pairs, i.e., a 25-mer.
- **Perfect match (PM)**: A 25-mer complementary to a reference sequence of interest (e.g., part of a gene).
- **Mismatch (MM)**: same as PM but with a single homomeric base change for the middle (13<sup>th</sup>) base (transversion purine  $\leftrightarrow$  pyrimidine, G  $\leftrightarrow$  C, A  $\leftrightarrow$  T) .
- **Probe-pair**: a (PM,MM) pair.
- **Probe-pair set**: a collection of probe-pairs (16 to 20) related to a common gene or fraction of a gene.
- **Affy ID**: an identifier for a probe-pair set.
- The purpose of the MM probe design is to measure non-specific binding and background noise.

# Affymetrix files

- Main software from Affymetrix company, *MicroArray Suite - MAS*, now version 5.
- **DAT** file: Image file,  $\sim 10^7$  pixels,  $\sim 50$  MB.
- **CEL** file: Cell intensity file, probe level PM and MM values.
- **CDF** file: Chip Description File. Describes which probes go in which probe sets and the location of probe-pair sets (genes, gene fragments, ESTs).



# **affy**: Pre-processing Affymetrix data

- Class definitions for probe-level data: **AffyBatch**, **ProbSet**, **Cdf**, **Cel**.
- Basic methods for manipulating microarray objects: printing, plotting, subsetting.
- Functions and widgets for data input from **CEL** and **CDF** files, and automatic generation of microarray data objects.
- Diagnostic plots: 2D spatial images, density plots, boxplots, MA-plots, etc.

# **affy**: Pre-processing Affymetrix data

- Background estimation.
- Probe-level normalization: quantile and curve-fitting normalization (Bolstad et al., 2002).
- Expression measures: MAS 4.0 AvDiff, MAS 5.0 Signal, MBEI (Li & Wong, 2001), RMA (Irizarry et al., 2003).
- Main functions: **ReadAffy**, **rma**, **expresso**, **express**.

# affy classes: AffyBatch

Probe-level intensity data for a batch of arrays (same CDF)

|                          |                                                                   |                                                                                      |
|--------------------------|-------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| <code>cdfName</code>     | Name of CDF file for arrays in the batch                          |                                                                                      |
| <code>nrow</code>        | <code>ncol</code>                                                 | Dimensions of the array                                                              |
| <code>exprs</code>       | <code>se.exprs</code>                                             | Matrices of probe-level intensities and SEs<br>rows → probe cells, columns → arrays. |
| <code>phenoData</code>   | Sample level covariates, instance of class <code>phenoData</code> |                                                                                      |
| <code>annotation</code>  | Name of annotation data                                           |                                                                                      |
| <code>description</code> | MIAME information                                                 |                                                                                      |
| <code>notes</code>       | Any notes                                                         |                                                                                      |

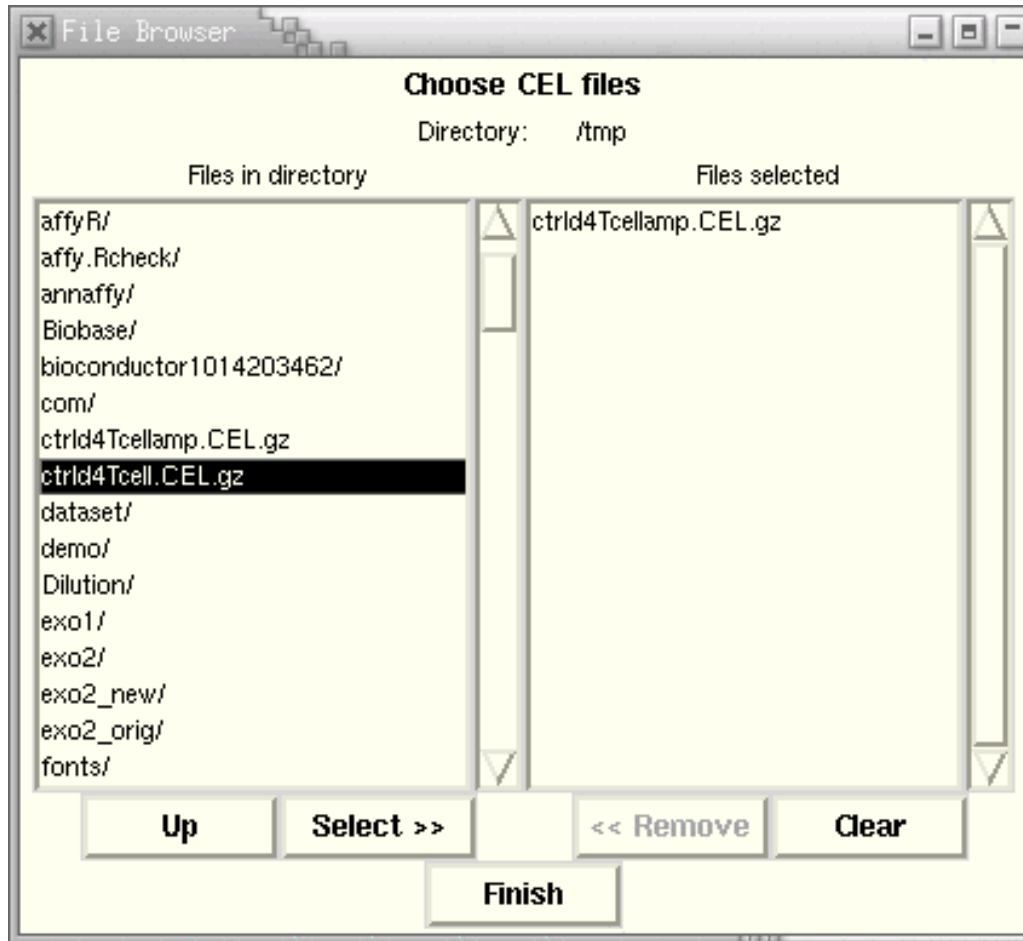
# affy classes

- **ProbeSet**: PM, MM intensities for individual probe sets.
  - **pm**: matrix of PM intensities for one probe set, rows → 16-20 probes, columns → arrays.
  - **mm**: matrix of MM intensities for one probe set, rows → 16-20 probes, columns → arrays.Apply **probeset** to **AffyBatch** object to get a list of **ProbeSet** objects.
- **Cel**: Single array cel intensity data.
- **Cdf**: Information contained in a **CDF** file.

# CDF data packages

- Data packages containing necessary CDF information are available at [www.bioconductor.org](http://www.bioconductor.org).
- Packages contain **environment** objects, which provide mappings between AffyIDs and matrices of probe locations, rows  $\rightarrow$  probe-pairs, columns  $\rightarrow$  PM, MM (e.g., 20X2 matrix for hu6800).
- **cdfName** slot of **AffyBatch**.
- **HGU95Av2** and **HGU133A** provided in **affy** package.

# Reading in data: ReadAffy



Creates object  
of class **AffyBatch**

# Accessing PM and MM data

- **probeNames**: method for accessing AffyIDs corresponding to individual probes.
- **pm**, **mm**: methods for accessing probe-level PM and MM intensities → probes x arrays matrix.
- Can use on **AffyBatch** objects.

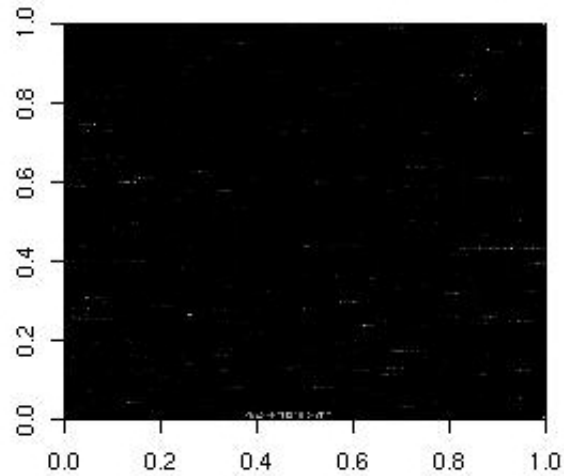
# Diagnostic plots

- See demo (`affy`).
- Diagnostic plots of probe-level intensities, PM and MM.
  - `image`: 2D spatial color images of log intensities (`AffyBatch`, `Cell`).
  - `boxplot`: boxplots of log intensities (`AffyBatch`).
  - `mva.pairs`: scatter-plots with fitted curves (apply `exprs`, `pm`, or `mm` to `AffyBatch` object).
  - `hist`: density plots of log intensities (`AffyBatch`).

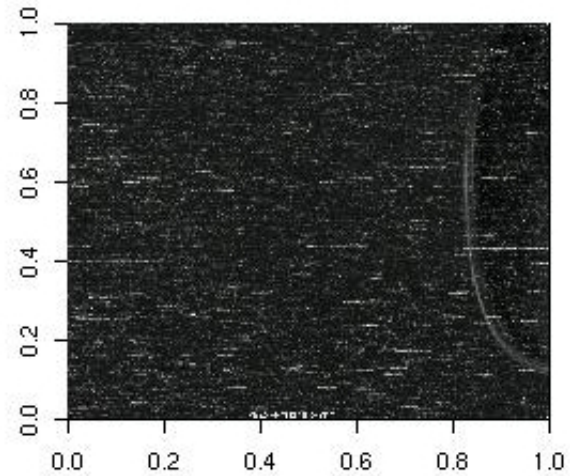


# image

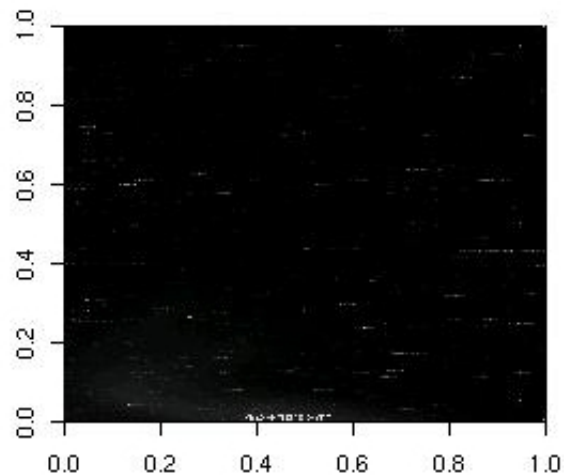
read from file: HIVControl4A.CEL.gz



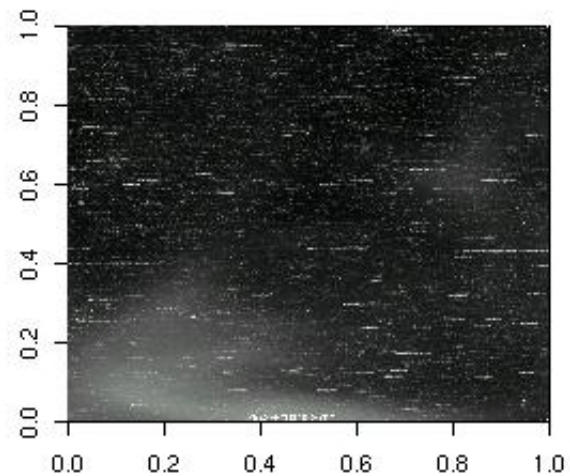
read from file: HIVControl4A.CEL.gz



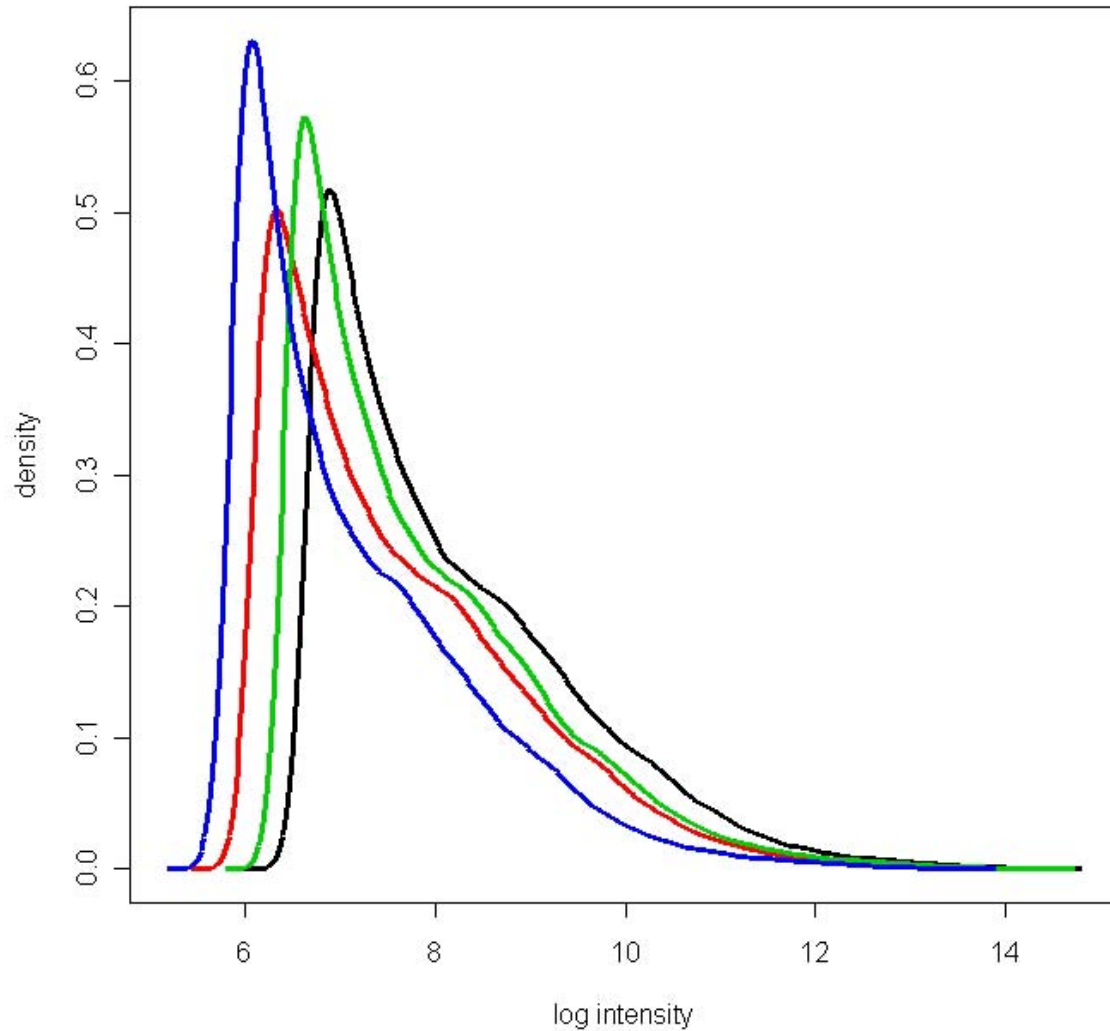
read from file: HIVControl4B.CEL.gz



read from file: HIVControl4B.CEL.gz



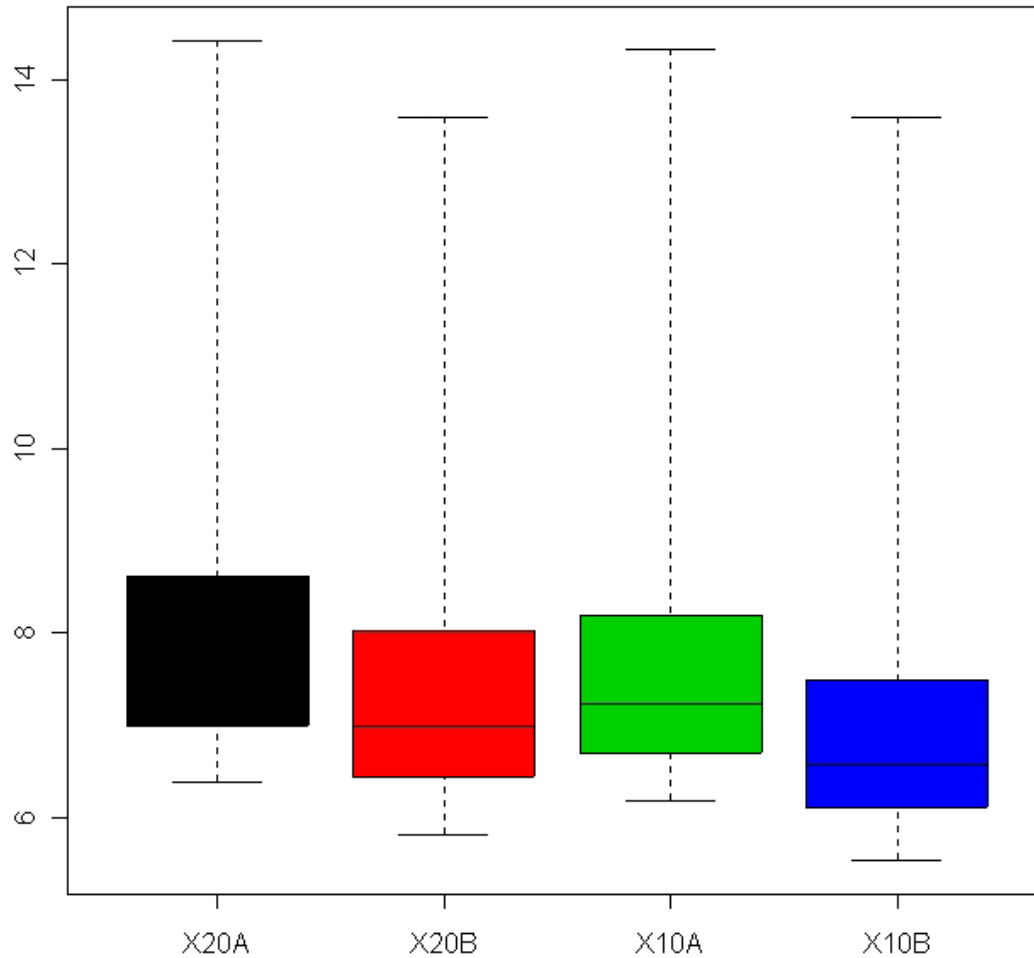
# hist



```
hist(Dilution,col=1:4,type="l",lty=1,lwd=3)
```

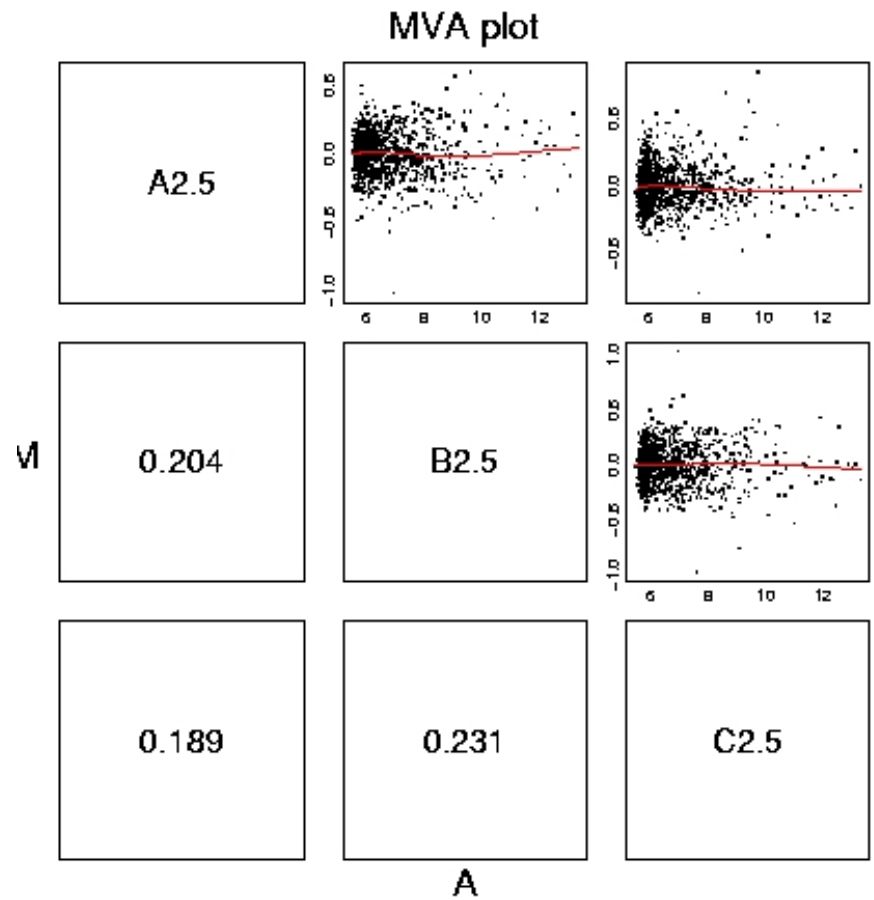
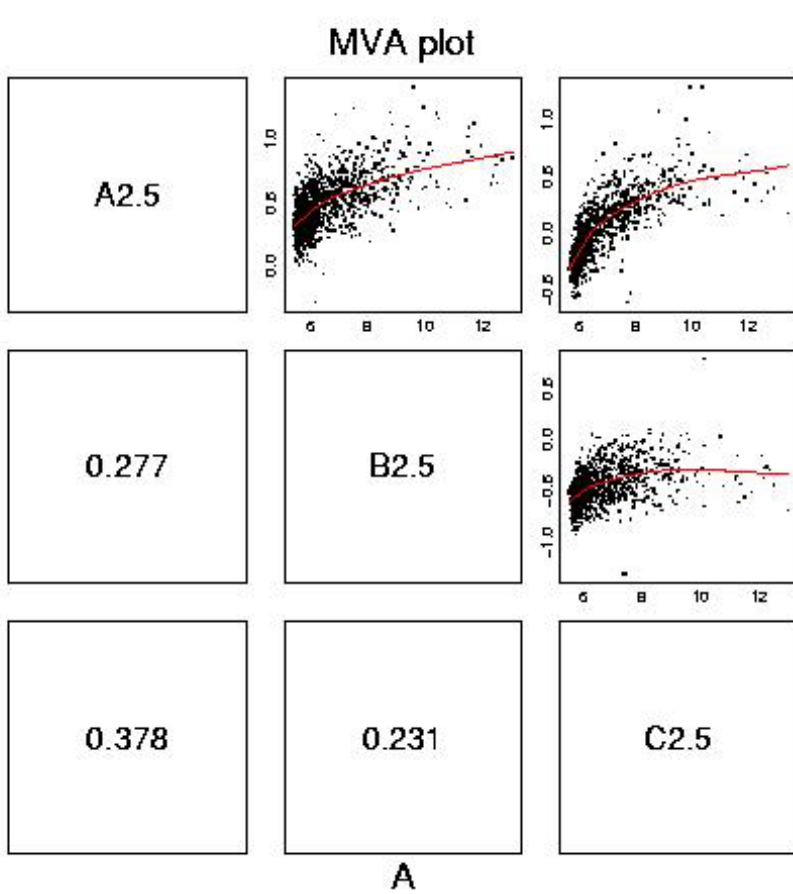
# boxplot

Small part of dilution study



```
boxplot(Dilution, col=1:4)
```

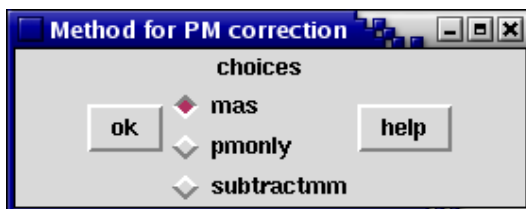
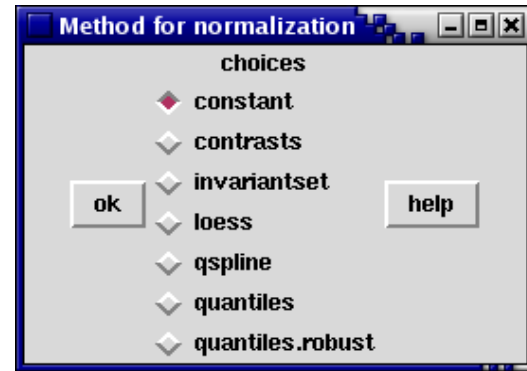
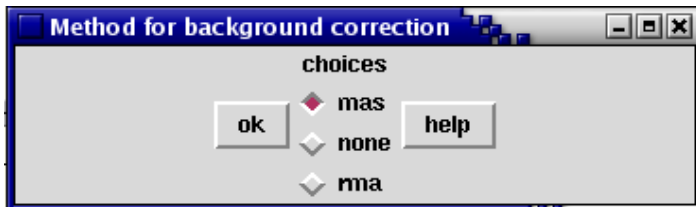
# mva.pairs



# Expression measures

- **expresso**: Choice of common methods for
  - background correction: `bgcorrect.methods`
  - normalization: `normalize.AffyBatch.methods`
  - probe specific corrections: `pmcorrect.methods`
  - expression measures:  
`express.summary.stat.methods`.
- **rma**: Fast implementation of RMA (Irizarry et al., 2003): model-based background correction, quantile normalization, median polish expression measures.
- **express**: Implementing your own method for computing expression measures.
- **normalize**: Normalization procedures in `normalize.AffyBatch.methods` or `normalize.methods(object)`.

# Expression measures: expresso



**expresso (widget=TRUE)**

# Probe sequence analysis

- Examine probe intensity based on location relative to 5' end of RNA sequence of interest.
- Expect probe intensities to be lower at 5' end compared to 3' of mRNA.

- E.g.

```
deg<-AffyRNAdeg (Dilution)
```

```
plotAffyRNAdeg (deg)
```