Vall d'Hebron
Institut de Recerca

EXPRESSION DATA ANALYSIS WITH MICROARRAYS

Ricardo Gonzalo Sanz and Alex Sanchez-Pla
{ricardo.Gonzalo,alex.sanchez}@vhir.org
Statistical and Bioinformatics Unit (www.ueb.vhir.org).
Vall d'Hebron Institut de Recerca

Vall d'Hebron Hospital · Generalitat de Catalunya Departament de Salut · Institut Català de la Salut · UAB Universitat Autònoma de Barcelona · Generalitat de Catalunya Departament d'Economia i Coneixement · Institució CERCA Centres de Recerca de Catalunya · Instituto de Salud Carlos III

Vall d'Hebron
Institut de Recerca

1. Introduction to R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to microarray technology
4. An example of microarray data analysis

Vall d'Hebron
Institut de Recerca

1. Introduction to R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to microarray technology
4. An example of microarray data analysis

1. Introduction to R and Bioconductor

Vall d'Hebron
Institut de Recerca

[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Reporting Bugs
Development Site
Conferences
Search

R Foundation

Foundation
Board
Members
Donors
Donate

Help With R

Getting Help

Documentation

Manuals
FAQs
The R Journal
Books
Certification
Other

Links

Bioconductor
Related Projects

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

- The R Foundation welcomes five new ordinary members: Jennifer Bryan, Dianne Cook, Julie Josse, Tomas Kalibera, and Balasubramanian Narasimhan.
- **R version 3.3.2 (Sincere Pumpkin Patch)** has been released on Monday 2016-10-31.
- **The R Journal Volume 8/1** is available.
- The **useR! 2017** conference will take place in Brussels, July 4 - 7, 2017, and details will be appear here in due course.
- **R version 3.3.1 (Bug in Your Hair)** has been released on Tuesday 2016-06-21.
- **R version 3.2.5 (Very, Very Secure Dishes)** has been released on 2016-04-14. This is a rebadging of the quick-fix release 3.2.4-revised.
- **Notice XQuartz users (Mac OS X)** A security issue has been detected with the Sparkle update mechanism used by XQuartz. Avoid updating over insecure channels.
- The **R Logo** is available for download in high-resolution PNG or SVG formats.
- **useR! 2016**, hase taken place at Stanford University, CA, USA, June 27 - June 30, 2016.
- **The R Journal Volume 7/2** is available.
- **R version 3.2.3 (Wooden Christmas-Tree)** has been released on 2015-12-10.
- **R version 3.1.3 (Smooth Sidewalk)** has been released on 2015-03-09.

https://cran.r-project.org/

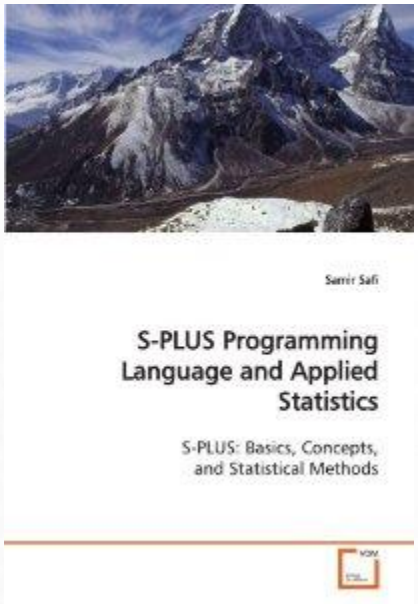
Vall d'Hebron
Institut de Recerca

What is R

- The S language was developed in 1976 at Bell Laboratories by John Chambers to ...
  - facilitate interactive exploration and visualization of data of varying complexity.
  - allow them to perform on all types of statistical analyzes.

- S language was (and is) commercial.

- R ("GNU" S) is born as a free alternative to S



Vall d'Hebron
Institut de Recerca

Why R?

- Free
- High quality methods implemented
- Platform independent
  - Linux, Mac, Other
- Constantly evolving
  - New version /6 months
- Programming language
  - Powerful & Flexible
  - Open source
  - Great for repetitive tasks

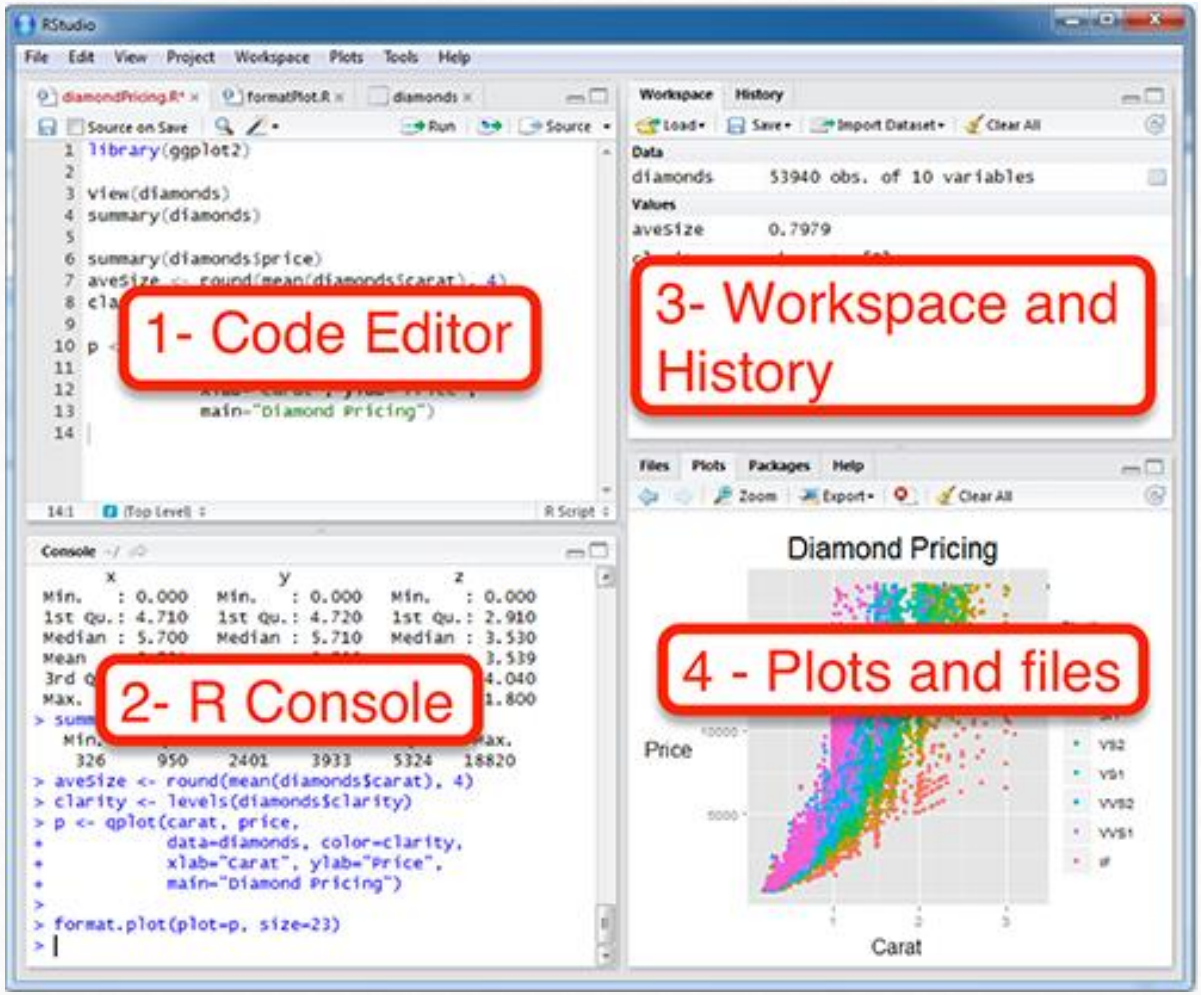
- Statistical tool
  - Modern
  - Most existing methods
    - (new method in R)
  - Great graphics.

Vall d'Hebron
Institut de Recerca

Why not R?

- Console-based interface
  - But GUI projects available
  - R-commander, DeduceR
- Community-based quality control
  - No company behind (no money back)
  - But thousands of users for most packages
- Constantly evolving
  - One new version every 6 months

R interfaces - Rstudio



https://www.rstudio.com/

Vall d'Hebron
Institut de Recerca

BIOCONDUCTOR

- An open source and open development software project for the analysis and comprehension of genomic data.
- Started in 2001. The core team is based primarily at the *Fred Hutchinson Cancer Research Center.*
- Primarily based on the R programming language.
- There are two releases of Bioconductor every year.
  - Started with 15packages
  - Now there are more than 1000

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Vall d'Hebron
Institut de Recerca

BIOCONDUCTOR

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Search:

Home | Install | Help | Developers | About

About *Bioconductor*

Bioconductor provides tools for the analysis and comprehension of high-throughput genomic data. Bioconductor uses the R statistical programming language, and is open source and open development. It has two releases each year, 1296 software packages, and an active user community. Bioconductor is also available as an AMI (Amazon Machine Image) and a series of Docker images.

News

- Bioconductor 3.4 is available.
- Bioconductor F1000 Research Channel launched.
- Orchestrating high-throughput genomic analysis with *Bioconductor* (abstract) and other recent literature.
- Read our latest newsletter and course material.
- Use the support site to get help installing, learning and using Bioconductor.

Install »

Get started with *Bioconductor*

- Install *Bioconductor*
- Explore packages
- Get support
- Latest newsletter
- Follow us on twitter
- Install R

Learn »

Master *Bioconductor* tools

- Courses
- Support site
- Package vignettes
- Literature citations
- Common work flows
- FAQ
- Community resources
- Videos

Use »

Create bioinformatic solutions with *Bioconductor*

- Software, Annotation, and Experiment packages
- Amazon Machine Image
- Latest release annoucement
- Support site

Develop »

Contribute to *Bioconductor*

- Developer resources
- Use Bioc 'devel'
- 'Devel' Software, Annotation and Experiment packages
- Package guidelines
- New package submission
- Build reports

http://bioconductor.org/

Vall d'Hebron
Institut de Recerca

BIOCONDUCTOR

- Essentially Bioconductor is a set of R packages
- A bioconductor package
  - Implements a different, new functionality
    - To manipulate or make tests on omics data
    - To use annotations
    - …
  - It can also be an Annotations database
  - Or even an experimental dataset
- BioConductor also provides training materials

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Vall d'Hebron
Institut de Recerca

1. Introduction to R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to microarray technology
4. An example of microarray data analysis

2. Installation of R and Bioconductor

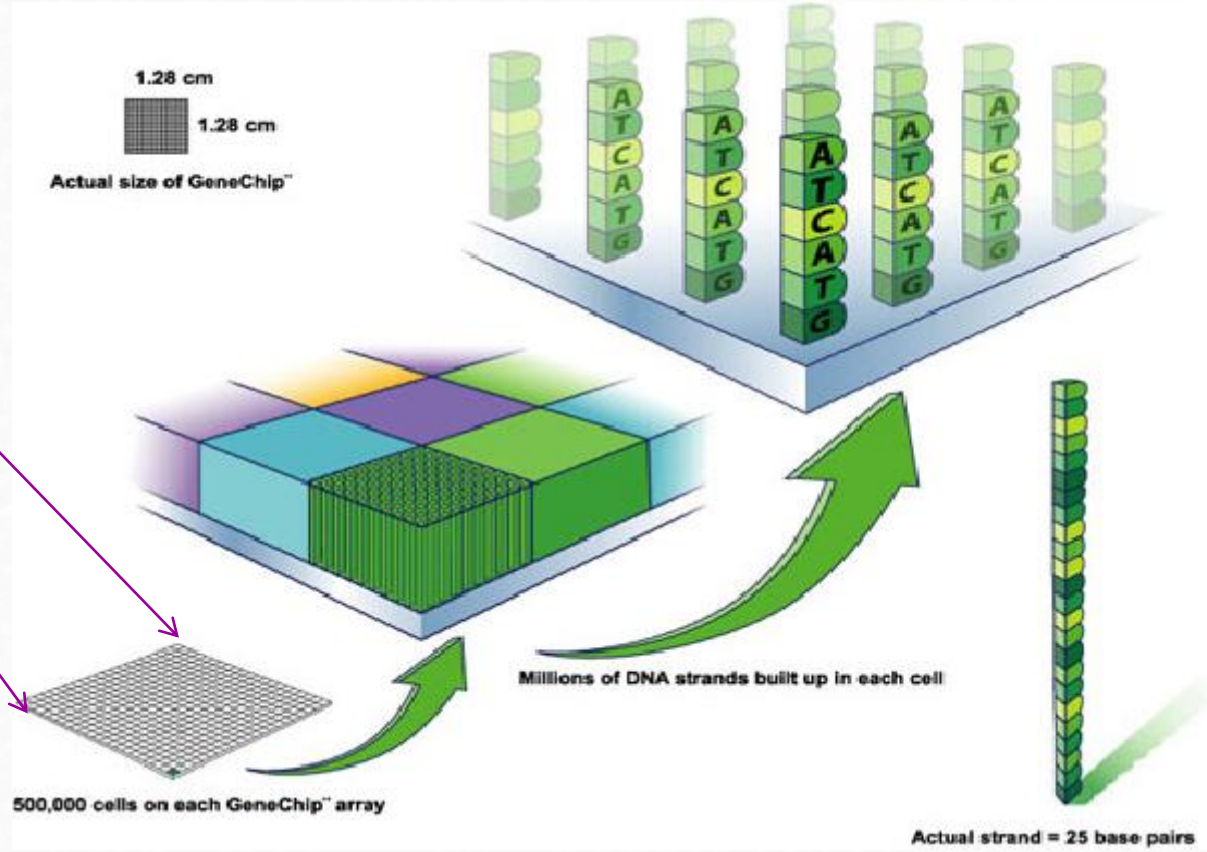
Vall d'Hebron
Institut de Recerca

Please, follow instructions of “Basic Introduction to R and Bioconductor.pdf” file, from to web page course

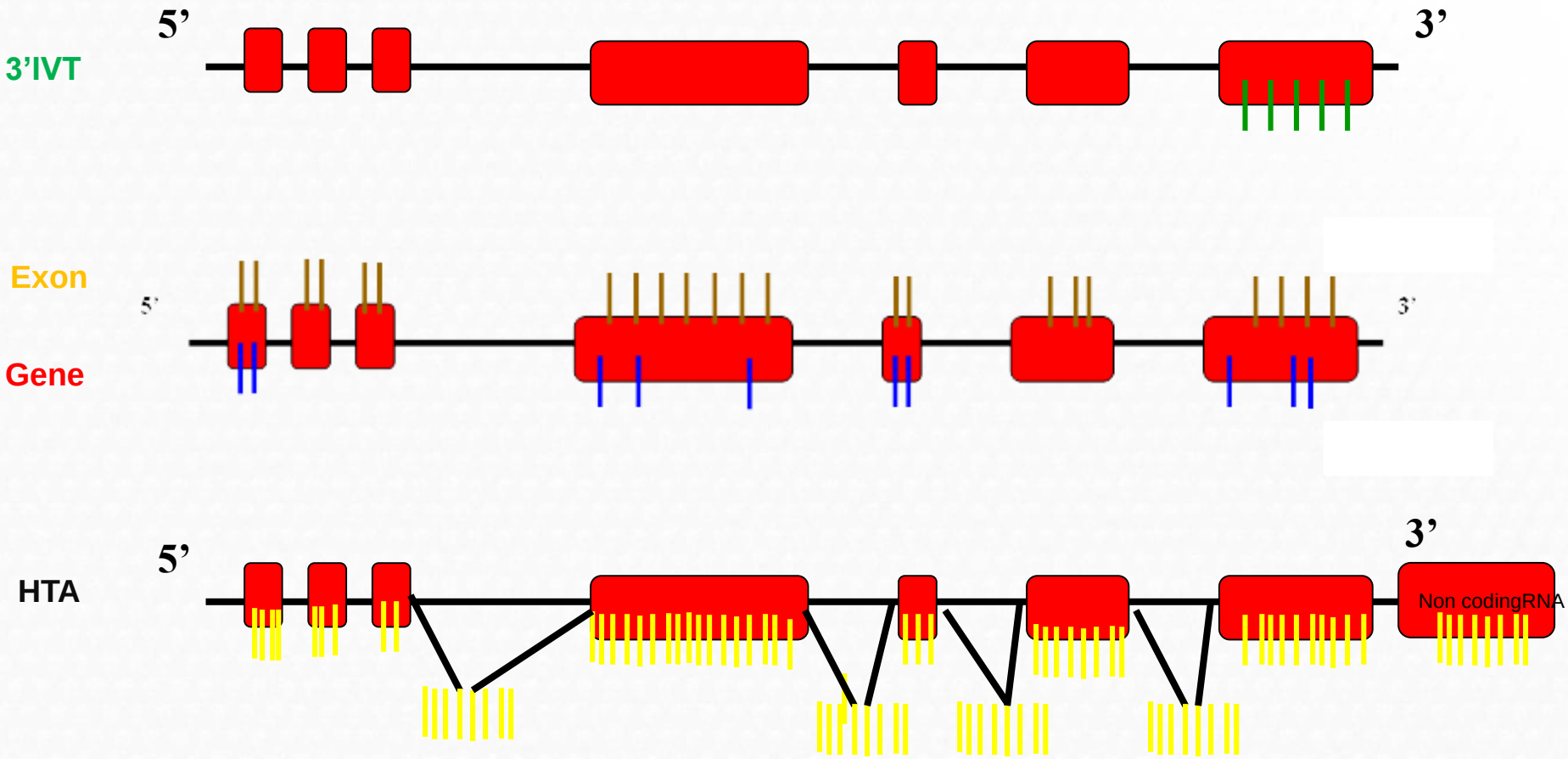
Vall d'Hebron
Institut de Recerca

1. Introduction to R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to microarray technology
4. An example of microarray data analysis

3. Introduction to microarray technology

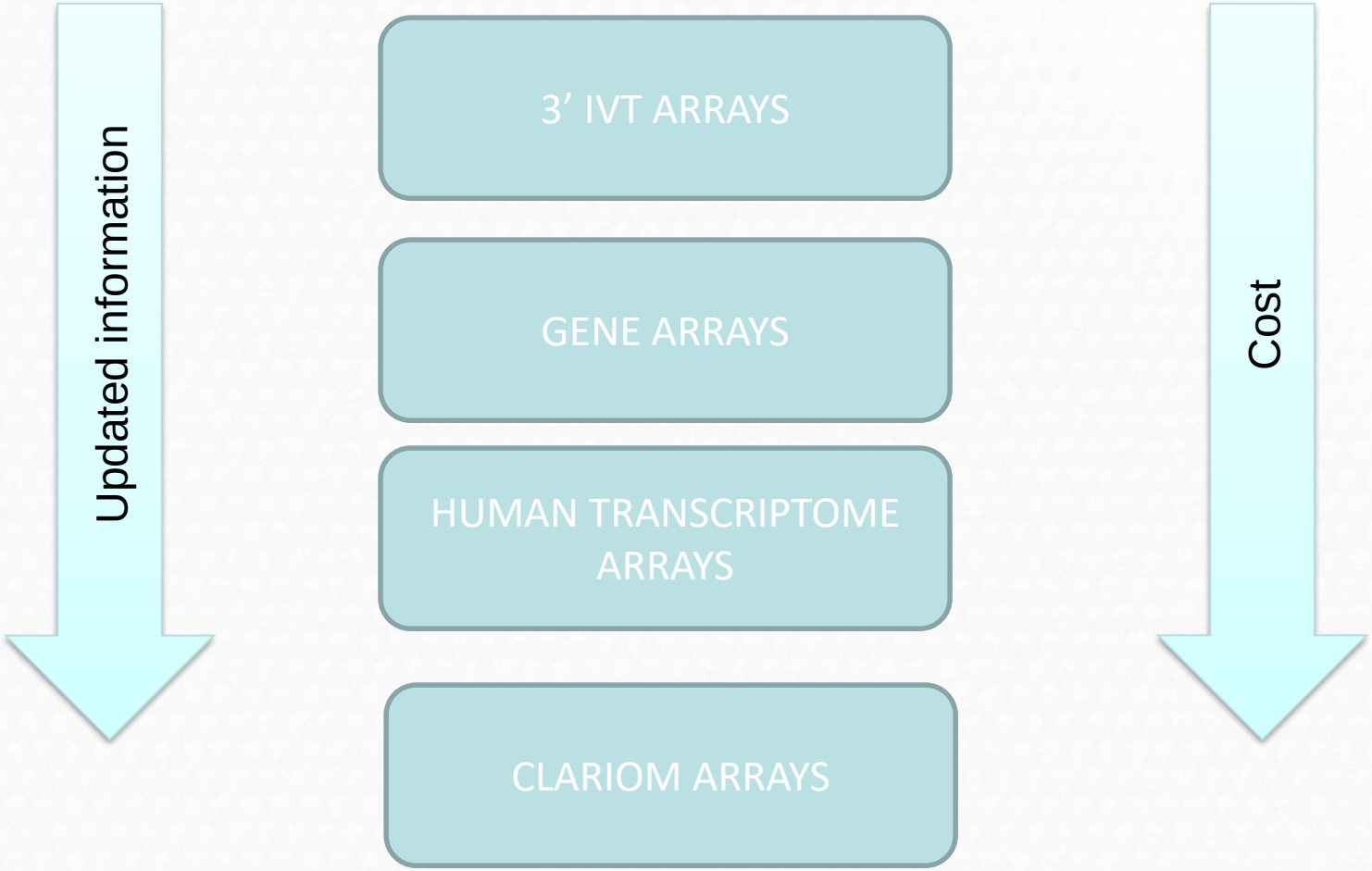


3. Introduction to microarray technology

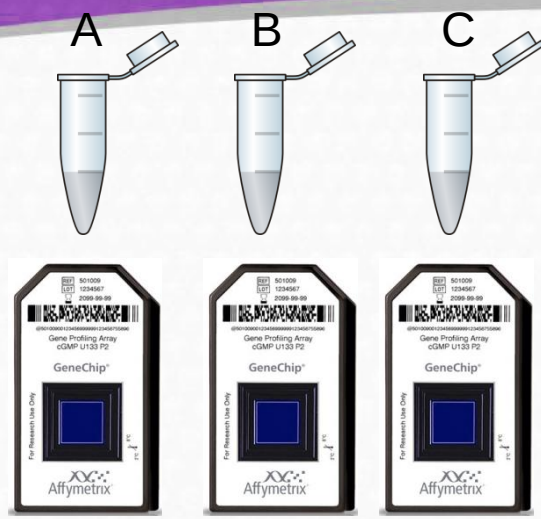


3. Introduction to microarray technology

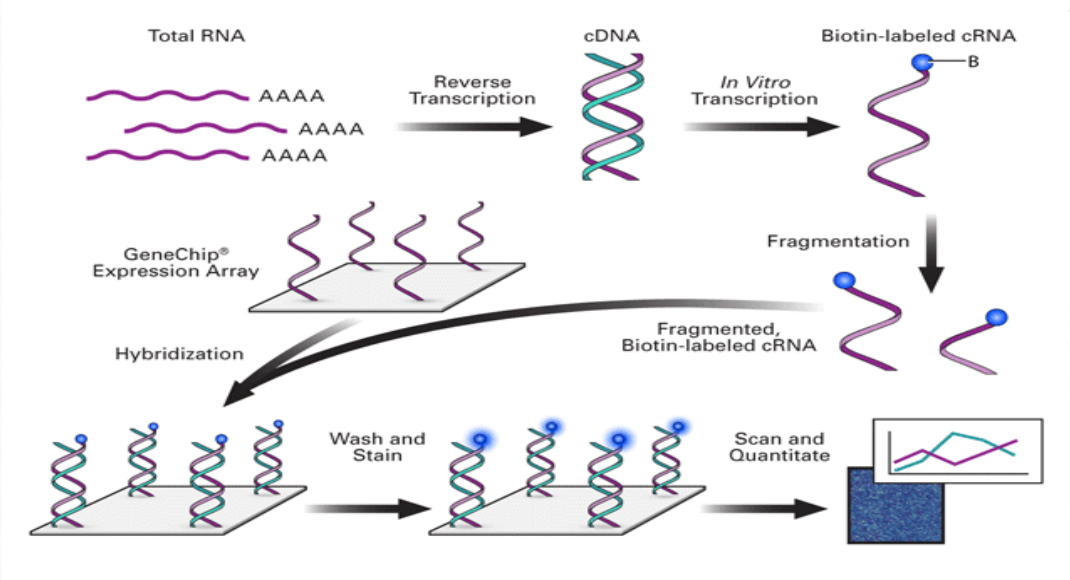
Vall d'Hebron
Institut de Recerca

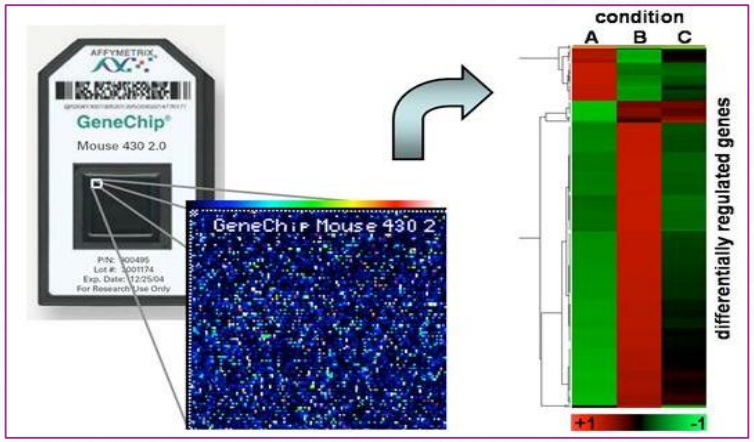


3. Introduction to microarray technology



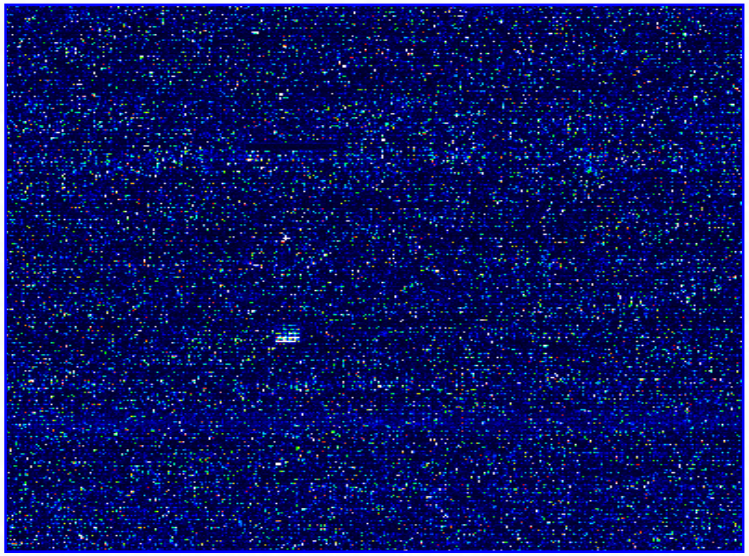
One sample hybridized per array





The sample is stained with **one dye (absolute fluorescence measure)**

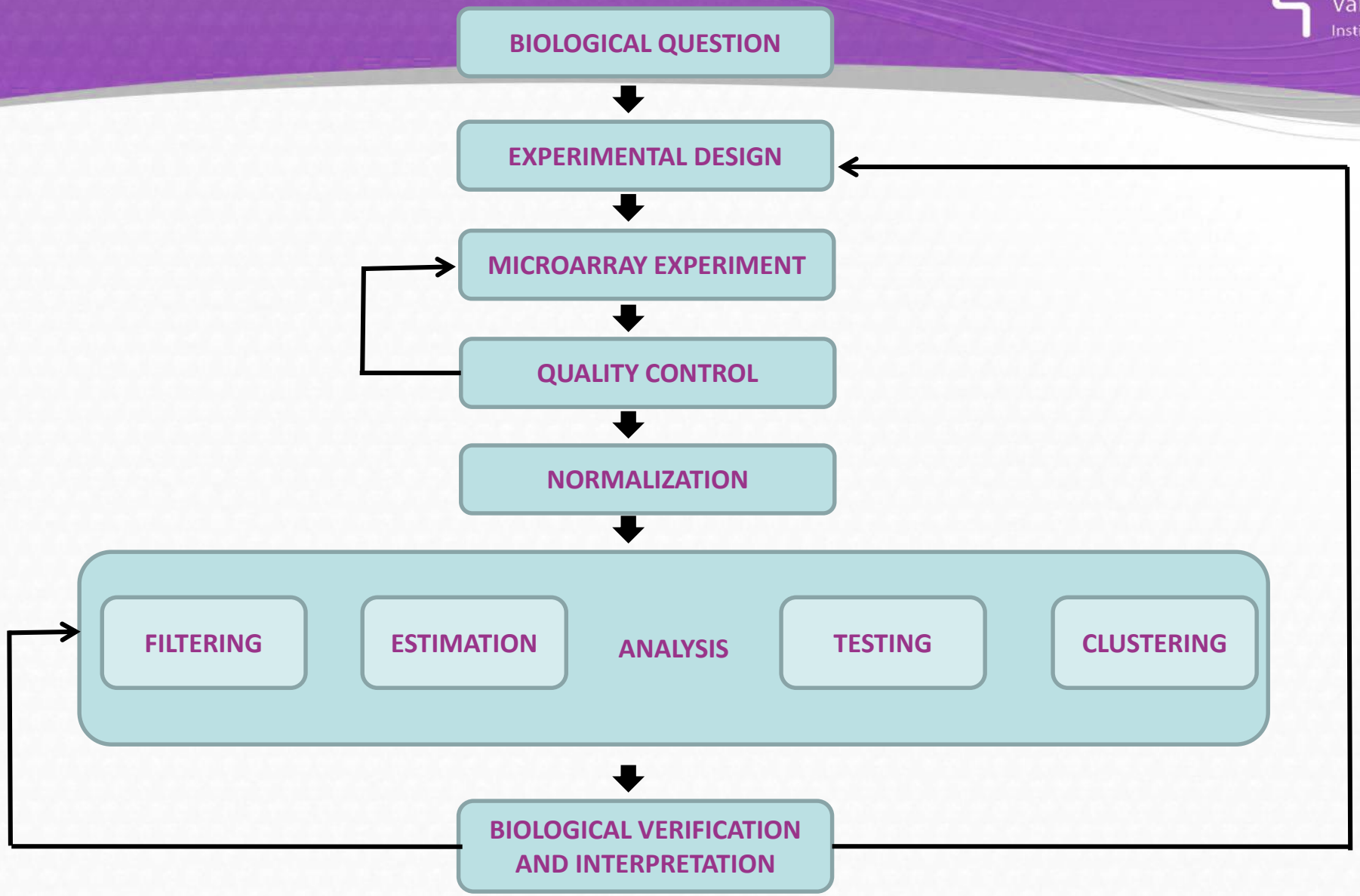
3. Introduction to microarray technology

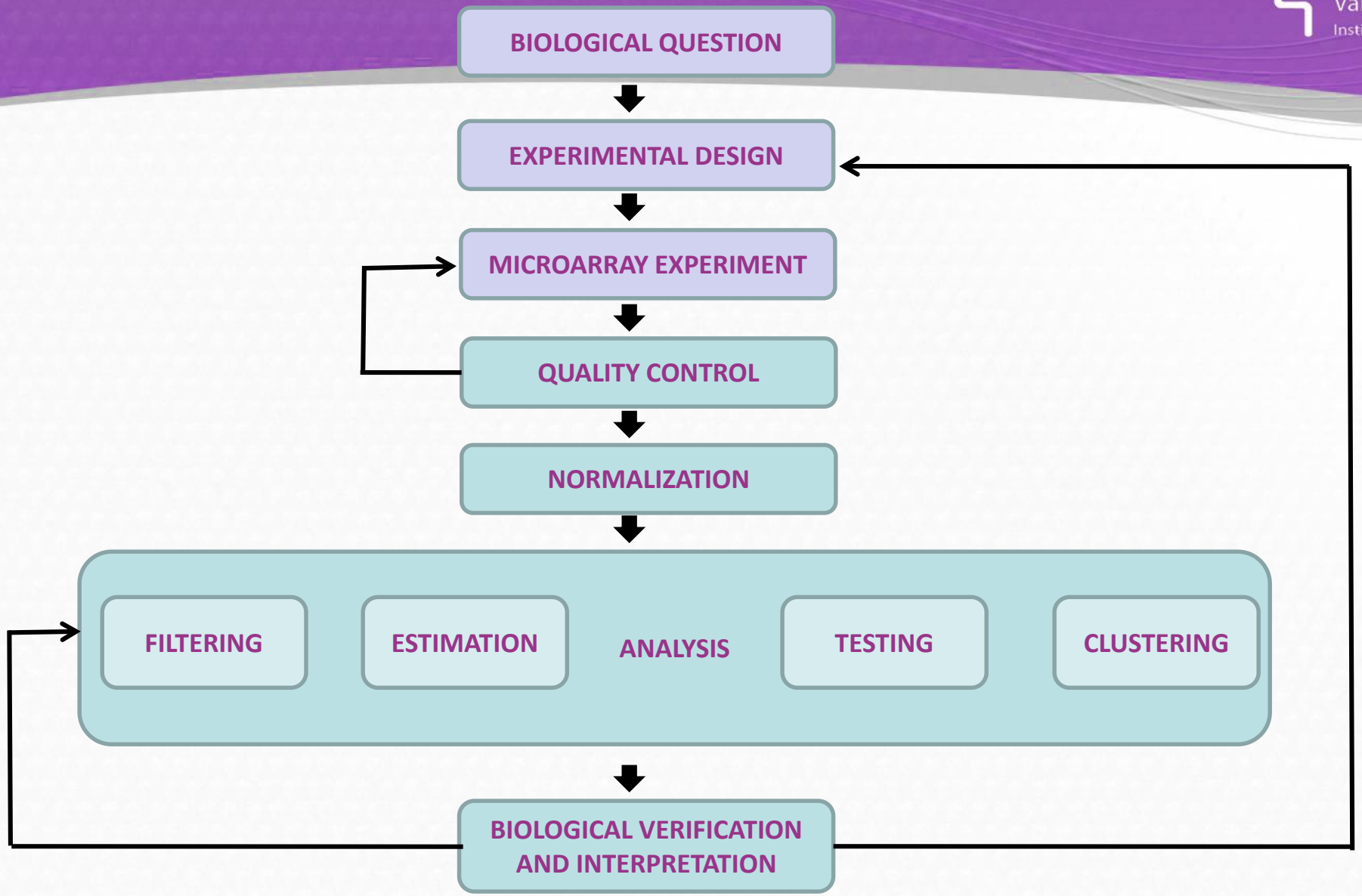


"CEL" FILES

Vall d'Hebron
Institut de Recerca

1. Introduction to R and Bioconductor
2. Installation of R and Bioconductor
3. Introduction to microarray technology
4. An example of microarray data analysis


BIOLOGICAL QUESTION
EXPERIMENTAL DESIGN
MICROARRAY EXPERIMENT
QUALITY CONTROL
NORMALIZATION
FILTERING
ESTIMATION
ANALYSIS
TESTING
CLUSTERING
BIOLOGICAL VERIFICATION AND INTERPRETATION


BIOLOGICAL QUESTION
EXPERIMENTAL DESIGN
MICROARRAY EXPERIMENT
QUALITY CONTROL
NORMALIZATION
FILTERING
ESTIMATION
ANALYSIS
TESTING
CLUSTERING
BIOLOGICAL VERIFICATION AND INTERPRETATION

4. Example of a microarray analysis with R. LOAD THE DATA

Vall d'Hebron
Institut de Recerca

GENE EXPRESSION ONMIBUS DATABASE

- Public functional genomic repository from NCBI
- Array and sequence-based data are accepted
- It is mandatory to upload your microarrays CEL files before publishing any article about them

http://www.ncbi.nlm.nih.gov/geo/

4. Example of a microarray analysis with R. LOAD THE DATA

Vall d'Hebron
Institut de Recerca

GENE EXPRESSION ONMIBUS DATABASE

The data for the example: GDS4155

Keyword or GEO Accession | Search

Search for GDS4155[ACCN] | Search | Clear | Show All | Advanced Search

DataSet Record GDS4155: Expression Profiles Data Analysis Tools Sample Subsets			
Title:	Dopaminergic transcription factors Ascl1, Lmx1a, Nurr1 combined effect on embryonic fibroblasts		
Summary:	Analysis of induced dopaminergic (iDA) neurons generated from E14.5 mouse embryonic fibroblasts (MEFs) reprogrammed by infection with lentiviruses expressing dopaminergic transcription factors Ascl1, Lmx1a and Nurr1. Results provide insight into the molecular basis of MEF to iDA reprogramming.		
Organism:	*Mus musculus*		
Platform:	GPL6246: [MoGene-1_0-st] Affymetrix Mouse Gene 1.0 ST Array [transcript (gene) version]		
Citation:	Caiazzo M, Dell'Anno MT, Dvoretskova E, Lazarevic D et al. Direct generation of functional dopaminergic neurons from mouse and human fibroblasts. *Nature* 2011 Jul 3;476(7359):224-7. PMID: 21725324		
Reference Series:	GSE27174	**Sample count:**	8
Value type:	transformed count	**Series published:**	2011/07/04

4. Example of a microarray analysis with R. LOAD THE DATA

Sample	Title
GSM671653	Fibroblasts dopaminergic induced rep1
GSM671654	Fibroblasts dopaminergic induced rep2
GSM671655	Fibroblasts dopaminergic induced rep3
GSM671656	Fibroblasts dopaminergic induced rep4
GSM671657	Fibroblasts not induced rep1
GSM671658	Fibroblasts not induced rep2
GSM671659	Fibroblasts not induced rep3
GSM671660	Fibroblasts not induced rep4

INDUCED

WT

4. Example of a microarray analysis with R. LOAD THE DATA

Two types of files are necessary to begin the data analysis:

1. CEL files
2. Targets file

fileName	grupos	ShortName	Colors
GSM671653.CEL	Induced	53_Ind	red
GSM671654.CEL	Induced	54_Ind	red
GSM671655.CEL	Induced	55_Ind	red
GSM671656.CEL	Induced	56_Ind	red
GSM671657.CEL	WT	57_WT	blue
GSM671658.CEL	WT	58_WT	blue
GSM671659.CEL	WT	59_WT	blue
GSM671660.CEL	WT	60_WT	blue

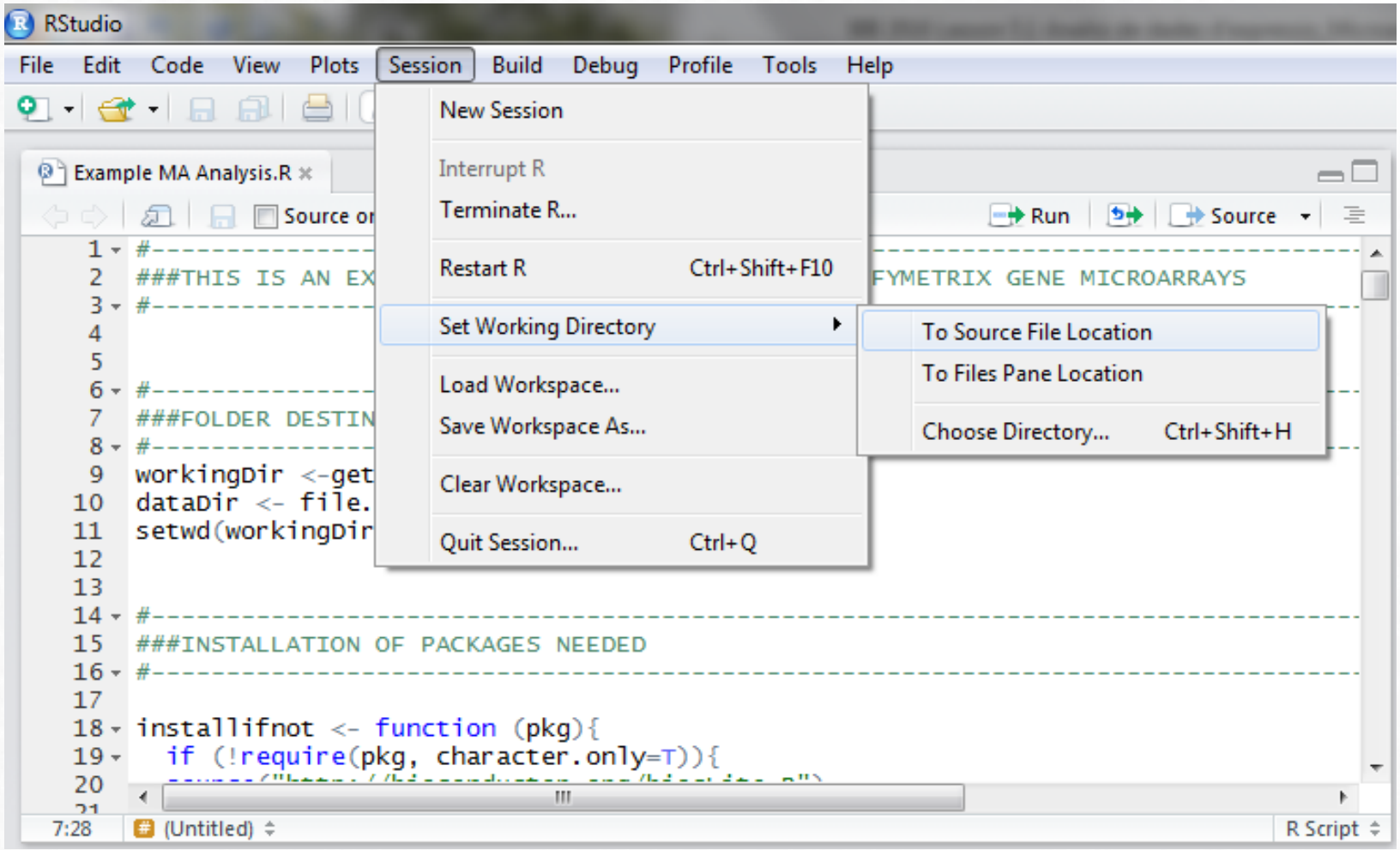
4. Example of a microarray analysis with R. LOAD THE DATA

Vall d'Hebron
Institut de Recerca

We have **to define the folders** before begin to analyze

- make a folder name it for exemple “microarrays”
- inside this folder create two more:
  - name the second “dades”
  - name the second “results”
- save the CEL and target files in the “dades” folder
- open “Example MA Analysis.R” with RStudio

4. Example of a microarray analysis with R. LOAD THE DATA



4. Example of a microarray analysis with R. LOAD THE DATA

Vall d'Hebron Institut de Recerca

We have to define the working folders:

```
workingDir <-getwd()
dataDir <- file.path(workingDir, “dades")
resultsDir <- file.path(workingDir, "/results")
setwd(resultsDir)
```

And to install and load necessary packages…

```
installifnot("pd.mogene.1.0.st.v1")
installifnot("mogene10sttranscriptcluster.db")
installifnot("oligo")
installifnot("limma")
installifnot("Biobase")
installifnot("arrayQualityMetrics")
installifnot("genefilter")
installifnot("multtest")
installifnot("annotate")
installifnot("xtable")
installifnot("GOstats")
installifnot("gplots")
installifnot("scatterplot3d")
```

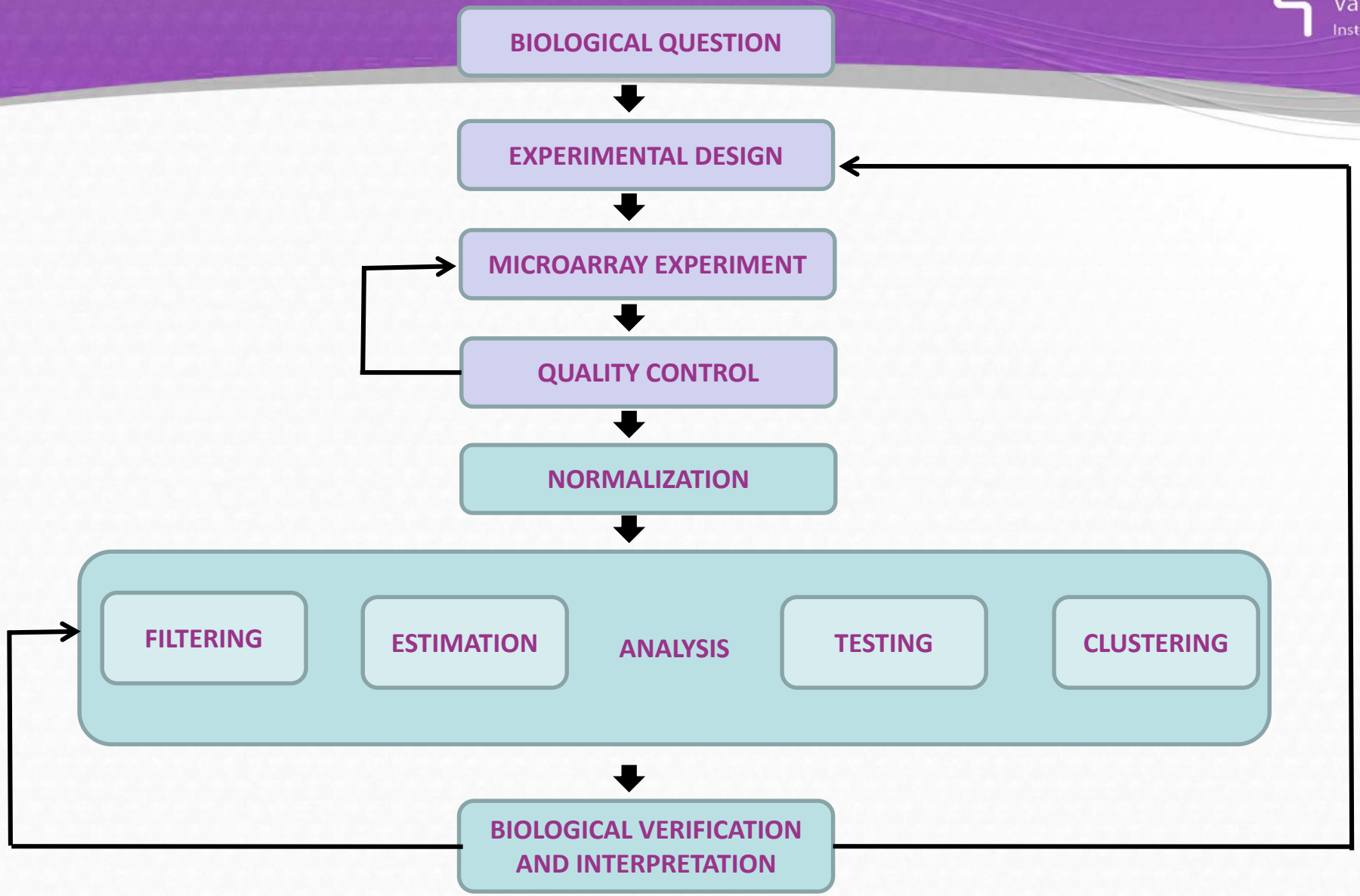
4. Example of a microarray analysis with R. LOAD THE DATA

We load the data:

```
#TARGETS
targets <-read.csv(file=file.path(dataDir,"targets.csv"), header = TRUE,
sep=";")
dades

#CELFILES
CELfiles<-list.celfiles(file.path(dataDir))
CELfiles
rawData<-read.celfiles(file.path(dataDir,CELfiles))

#DEFINE SOME USEFUL VARIABLES FOR THE GRAPHICS
sampleNames <- as.character(targets$ShortName)
sampleColor<- as.character(targets$Colors)
```


BIOLOGICAL QUESTION
EXPERIMENTAL DESIGN
MICROARRAY EXPERIMENT
QUALITY CONTROL
NORMALIZATION
FILTERING
ESTIMATION
ANALYSIS
TESTING
CLUSTERING
BIOLOGICAL VERIFICATION AND INTERPRETATION

4. Example of a microarray analysis with R. QUALITY CONTROL OF THE DATA

Vall d'Hebron Institut de Recerca

- First of all we have to decide if the data are good to work with.
- Microarray experiments generate huge quantities of data

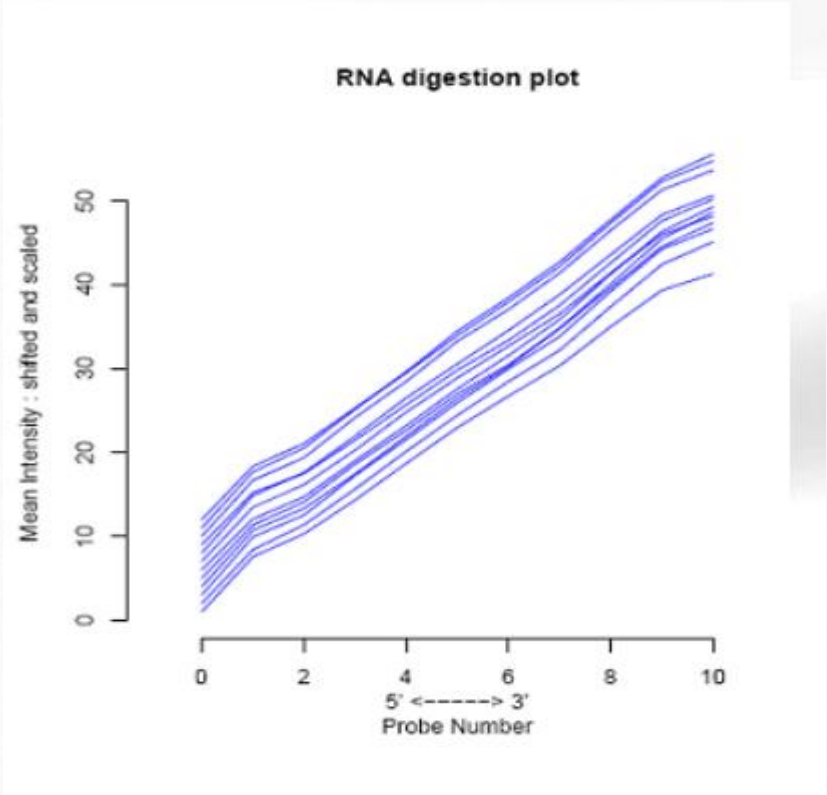
It is hard to decide if things "seem to be all right" just by looking at the numbers.

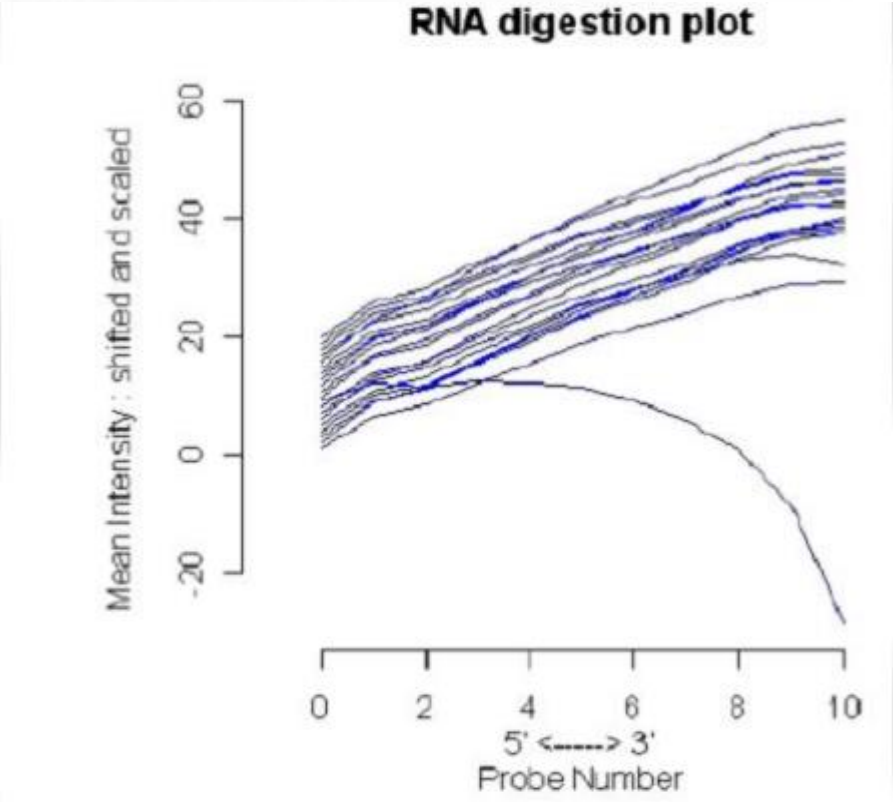
- Standard statistical approach **use plots** to check the quality
  - ✓ show all data together
  - ✓ highlight structures
  - ✓ may help to detect problems ("unusual patterns")

4. Example of a microarray analysis with R. QUALITY CONTROL OF THE DATA

Vall d'Hebron
Institut de Recerca

RNA digestion plot. Only for 3'Arrays.

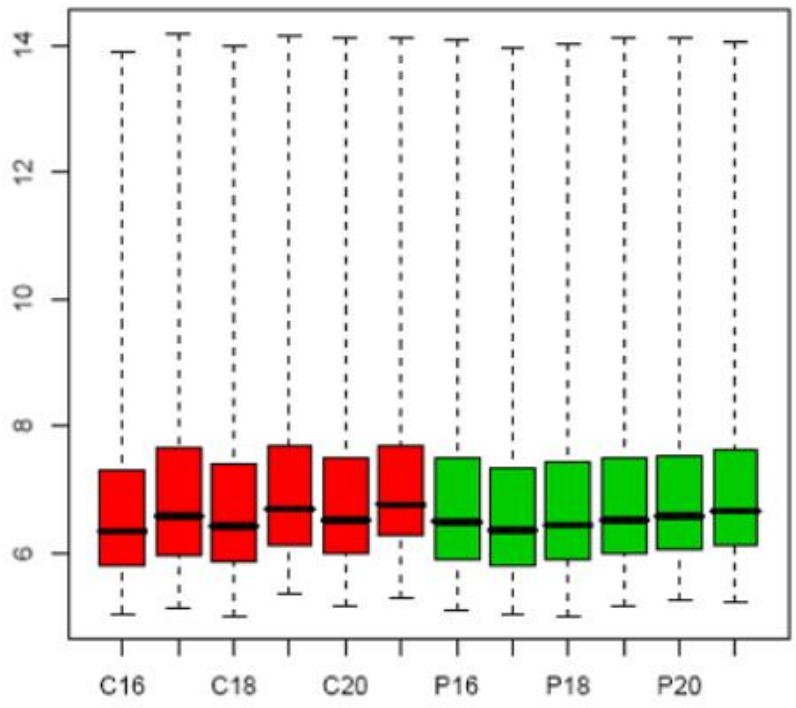


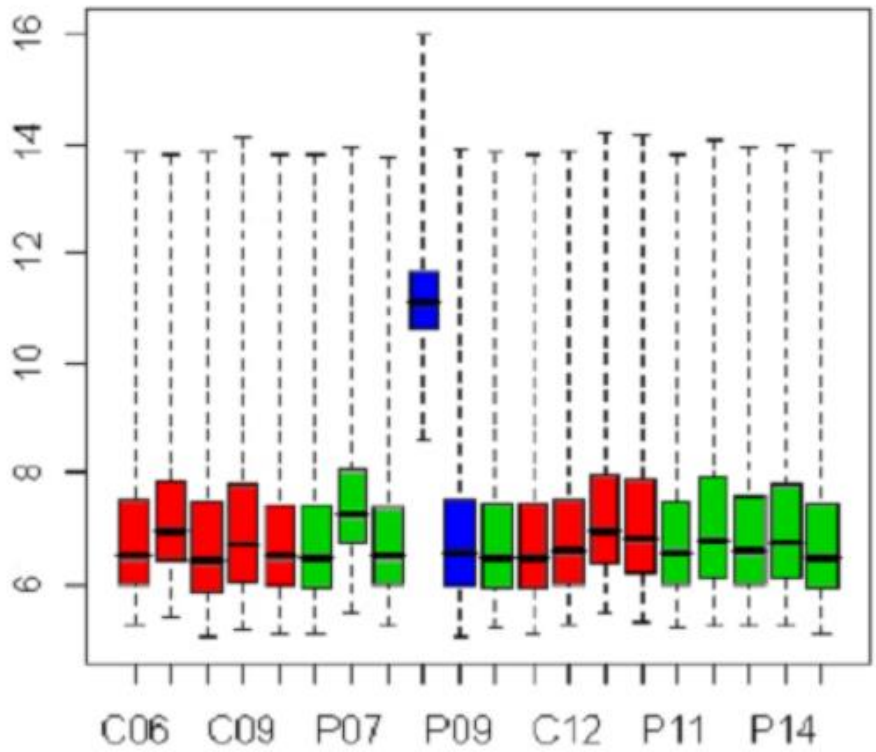


4. Example of a microarray analysis with R. QUALITY CONTROL OF THE DATA

Vall d'Hebron
Institut de Recerca

Boxplot intensities. Raw data/Normalized data

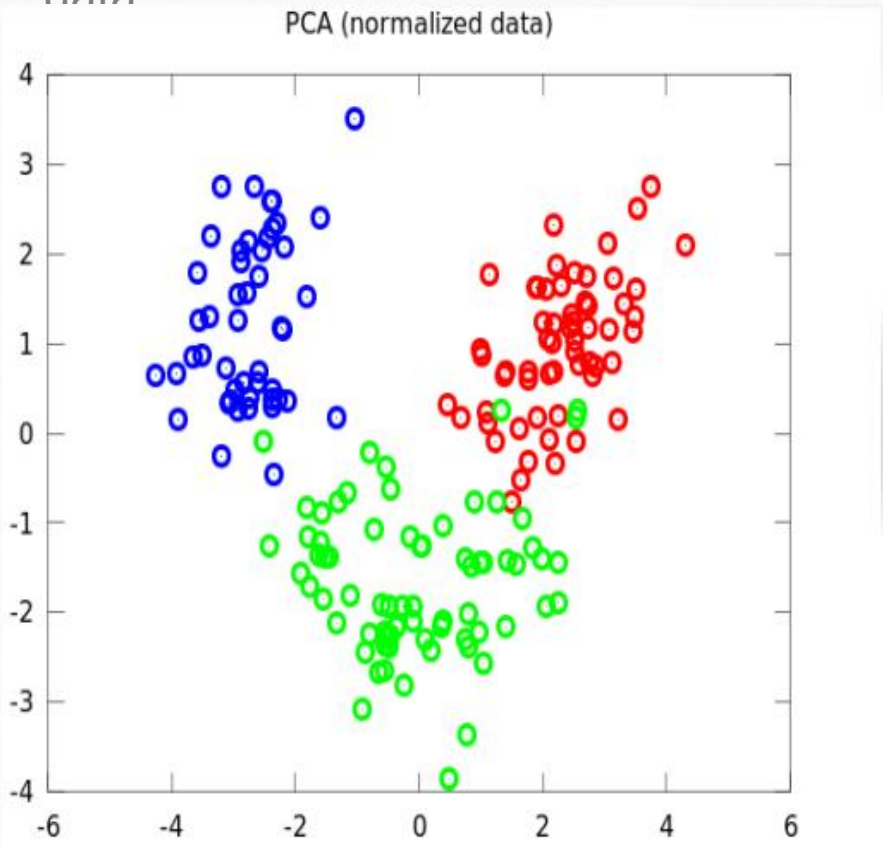


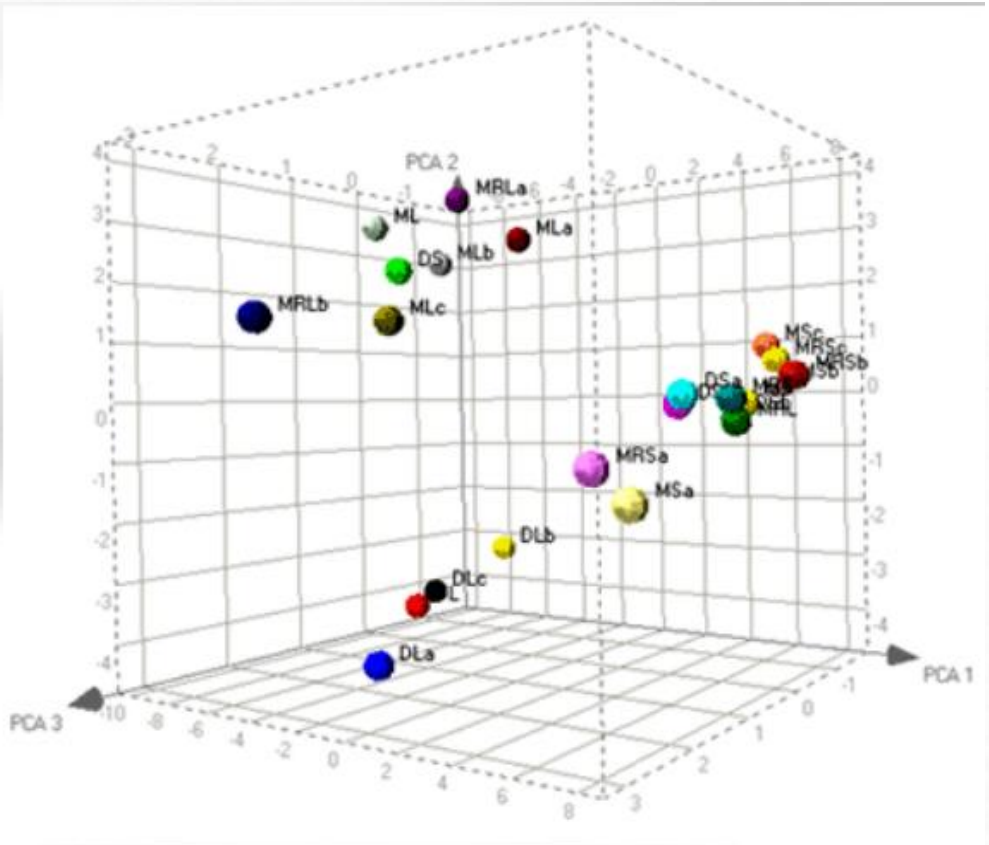


4. Example of a microarray analysis with R. QUALITY CONTROL OF THE DATA

Vall d'Hebron
Institut de Recerca

Principal Component Analysis. Raw data/Normalized data





4. Example of a microarray analysis with R. QUALITY CONTROL OF THE DATA

Vall d'Hebron Institut de Recerca

Let's do with our data:

```
#BOXPLOT
boxplot(rawData, which="all",las=2, main="Intensity distribution of RAW data",
        cex.axis=0.6, col=sampleColor, names=sampleNames)

#HIERARQUICAL CLUSTERING
clust.euclid.average <- hclust(dist(t(exprs(rawData))),method="average")
plot(clust.euclid.average, labels=sampleNames, main="Hierarchical clustering
of RawData", cex=0.7,  hang=-1)
```

4. Example of a microarray analysis with R. QUALITY CONTROL OF THE DATA

Vall d'Hebron Institut de Recerca

Let's do with our data:

```
#PRINCIPAL COMPONENT ANALYSIS
plotPCA <- function ( X, labels=NULL, colors=NULL, dataDesc="", scale=FALSE,
formapunts=NULL, myCex=0.8,...)
{
  pcX<-prcomp(t(X), scale=scale) # o prcomp(t(X))
  loads<- round(pcX$sdev^2/sum(pcX$sdev^2)*100,1)
  xlab<-c(paste("PC1",loads[1],"%"))
  ylab<-c(paste("PC2",loads[2],"%"))
  if (is.null(colors)) colors=1
  plot(pcX$x[,1:2],xlab=xlab,ylab=ylab, col=colors, pch=formapunts,
       xlim=c(min(pcX$x[,1])-100000,
max(pcX$x[,1])+100000),ylim=c(min(pcX$x[,2])-100000, max(pcX$x[,2])+100000))
  text(pcX$x[,1],pcX$x[,2], labels, pos=3, cex=myCex)
  title(paste("Plot of first 2 PCs for expressions in", dataDesc, sep=" "),
cex=0.8)
}

plotPCA(exprs(rawData), labels=sampleNames, dataDesc="raw data",
colors=sampleColor,formapunts=c(rep(16,4),rep(17,4)), myCex=0.6)
```

4. Example of a microarray analysis with R. QUALITY CONTROL OF THE DATA

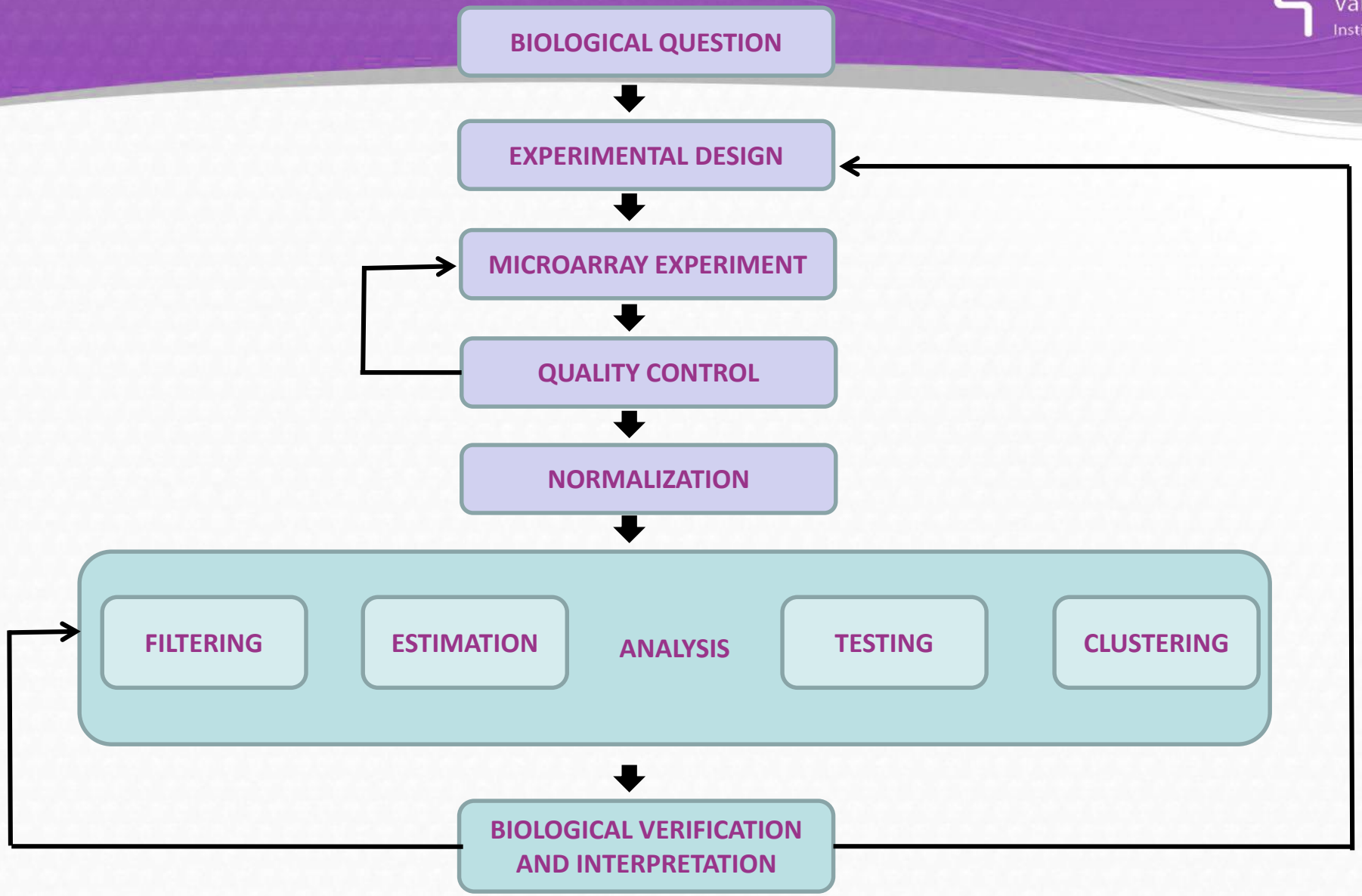
Vall d'Hebron Institut de Recerca

Let's do with our data:

```
#SAVE TO A FILE
pdf("QCPlots_Raw.pdf")
boxplot(rawData, which="all",las=2, main="Intensity distribution of RAW data",
        cex.axis=0.6, col=sampleColor, names=sampleNames)

plot(clust.euclid.average, labels=sampleNames, main="Hierarchical clustering
        of RawData", cex=0.7,  hang=-1)

plotPCA(exprs(rawData), labels=sampleNames, dataDesc="raw data",
        colors=sampleColor, formapunts=c(rep(16,4),rep(17,4)), myCex=0.6)
dev.off()
```


BIOLOGICAL QUESTION
EXPERIMENTAL DESIGN
MICROARRAY EXPERIMENT
QUALITY CONTROL
NORMALIZATION
FILTERING
ESTIMATION
ANALYSIS
TESTING
CLUSTERING
BIOLOGICAL VERIFICATION AND INTERPRETATION

4. Example of a microarray analysis with R. NORMALIZATION

Vall d'Hebron Institut de Recerca

It is very important (essential) to normalize your data.

Why normalization?

1. To remove systematic biases:
   - Sample preparation
   - Variability in hybridization
   - Scanner settings
   - Experimenter bias
2. To achieve a measured scale such that:
   - Has the same origin for all spots
   - Use the same unit for all arrays
   - Linear relationship with mRNA quantity
3. To cure poor data

4. Example of a microarray analysis with R. NORMALIZATION

Vall d'Hebron Institut de Recerca

Exists different methods:

- **RMA (Robust Multiarray Average):** Performs background correction, normalization, and summarization in a modular way. RMA does not take in account unspecific probe hybridization in probe set background calculation (Irizarry et al., 2003)

- **GCRMA:** is a version of RMA with a background correction component that makes use of a probe sequence information (Wu et al., 2004)

- **PLIER (Probe logarithmic error intensity estimate)**: this method produces an improved signal by accounting for experimentally observed patterns in probe behavior and handling error at the appropriately low and high signal values

4. Example of a microarray analysis with R. NORMALIZATION

Vall d'Hebron Institut de Recerca

Nevertheless the steps they perform are common.

General steps:

1. Background correction: correction of the scale origin for spots
2. Normalization: standardizing the scale unit. Rescaling
3. Probe level intensity calculation
4. Summary of information of several spots into a single measure for each gene

4. Example of a microarray analysis with R. NORMALIZATION

Let's do with our data:

```
eset<-rma(rawData)

#SAVE TO A FILE
write.exprs(eset,"NormData.txt")
```

4. Example of a microarray analysis with R. NORMALIZATION

It could be interesting to perform again the quality control plots:

```
#BOXPLOT
boxplot(eset, las=2, main="Intensity distribution of Normalized data",
cex.axis=0.6, col=sampleColor, names=sampleNames)
```

4. Example of a microarray analysis with R. NORMALIZATION

Vall d'Hebron Institut de Recerca

It could be interesting to perform again the quality control plots:

```
#PRINCIPAL COMPONENT ANALYSIS
plotPCA <- function ( X, labels=NULL, colors=NULL, dataDesc="", scale=FALSE,
formapunts=NULL, myCex=0.8,...)
{
  pcX<-prcomp(t(X), scale=scale) # o prcomp(t(X))
  loads<- round(pcX$sdev^2/sum(pcX$sdev^2)*100,1)
  xlab<-c(paste("PC1",loads[1],"%"))
  ylab<-c(paste("PC2",loads[2],"%"))
  if (is.null(colors)) colors=1
  plot(pcX$x[,1:2],xlab=xlab,ylab=ylab, col=colors, pch=formapunts,
       xlim=c(min(pcX$x[,1])-10, max(pcX$x[,1])+10),ylim=c(min(pcX$x[,2])-10,
max(pcX$x[,2])+10))
  text(pcX$x[,1],pcX$x[,2], labels, pos=3, cex=myCex)
  title(paste("Plot of first 2 PCs for expressions in", dataDesc, sep=" "),
cex=0.8)
}

plotPCA(exprs(eset), labels=sampleNames, dataDesc="NormData",
colors=sampleColor,formapunts=c(rep(16,4),rep(17,4)), myCex=0.6)
```

4. Example of a microarray analysis with R. NORMALIZATION

Vall d'Hebron Institut de Recerca

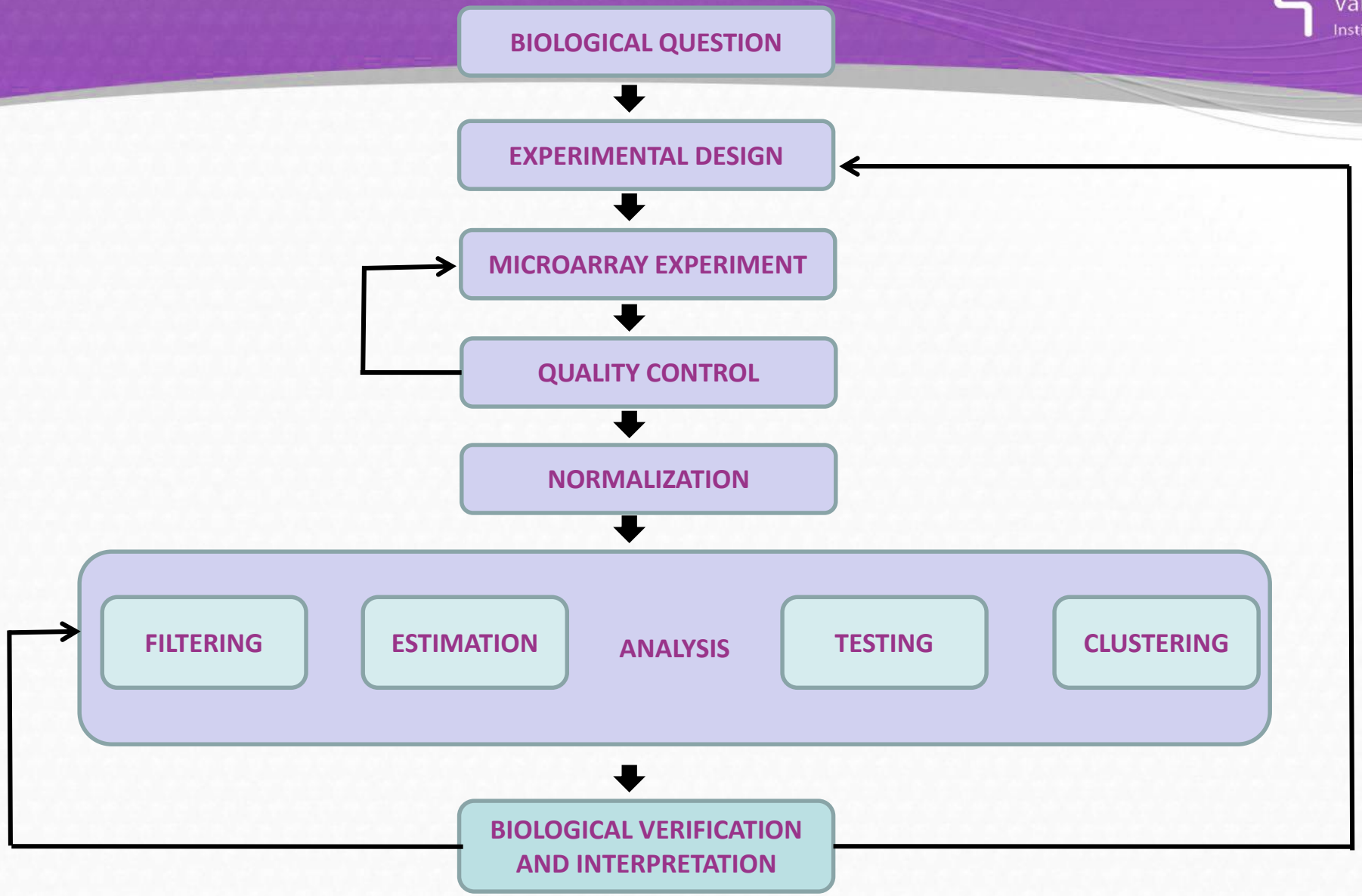
It could be interesting to perform again the quality control plots:

```
#SAVE TO A FILE
pdf("QCPlots_Norm.pdf")
boxplot(rawData, las=2, main="Intensity distribution of Normalized data",
cex.axis=0.6, col=sampleColor, names=sampleNames)

plotPCA(exprs(eset), labels=sampleNames, dataDesc="selected samples",
colors=sampleColor,formapunts=c(rep(16,4),rep(17,4)), myCex=0.6)
dev.off()



#ARRAY QUALITY METRICS
arrayQualityMetrics(eset,  reporttitle="QualityControl", force=TRUE)
```

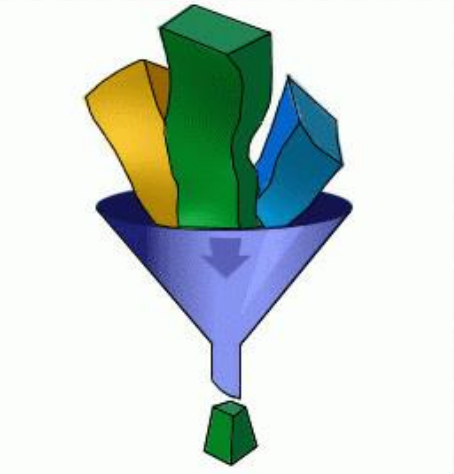

BIOLOGICAL QUESTION
EXPERIMENTAL DESIGN
MICROARRAY EXPERIMENT
QUALITY CONTROL
NORMALIZATION
FILTERING
ESTIMATION
ANALYSIS
TESTING
CLUSTERING
BIOLOGICAL VERIFICATION AND INTERPRETATION

4. Example of a microarray analysis with R. DATA FILTERING

- In a microarray experiment only a few hundreds/thousand of genes change their expression due to the different conditions

- Researcher is interested in keeping the number of tests/genes as low as possible while keeping the interesting genes in the selected subset.

Genes that do not change introduce noise, therefore is better not to be present when the statistical analysis is done



4. Example of a microarray analysis with R. DATA FILTERING

Vall d'Hebron
Institut de Recerca

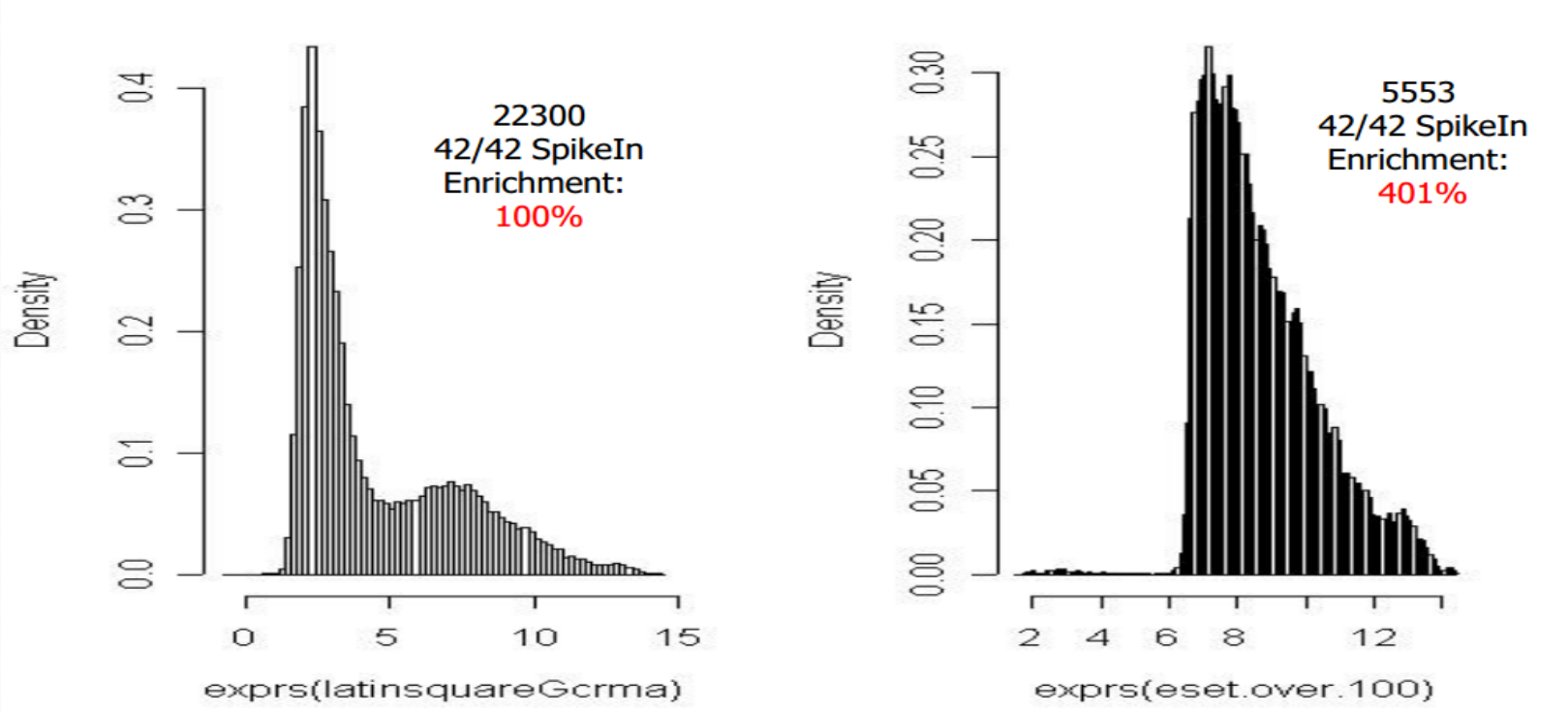
Exists different types of filtering:

- Annotation features (specific):
  - Specific gene features (i.e. GO term, presence of transcriptional regulative elements in promoters, etc.)

- Signal features (non specific)
  - % intensities greater of a user defined value
  - Interquantile range (IQR) greater of a defined value

4. Example of a microarray analysis with R. DATA FILTERING

Signal filtering: This technique has as its premise the removal of genes that are deemed to be not expressed or unchanged according to some specific criterion that is under the control of the user.



4. Example of a microarray analysis with R. DATA FILTERING

Let's do with our data:

```
annotation(eset) <- "org.Mm.eg.db"
eset_filtered <- nsFilter(eset, var.func=IQR,
         var.cutoff=0.75, var.filter=TRUE,
         filterByQuantile=TRUE)

#NUMBER OF GENES OUT
print(eset_filtered$filter.log$numLowVar)

#NUMBER OF GENES IN
print(eset_filtered$eset)
```

Vall d'Hebron
Institut de Recerca

Statistical inference of differential expression

Class comparison problem:

- Identify genes whose expression is significantly associated with different conditions:
  - ✓Treatment, cell type …
  - ✓Dose, time,….
- Estimate effects/differences between groups.

Vall d'Hebron Institut de Recerca

Which situations could we found (here the easiest)?

- Indirect comparisons: 2 groups unpaired
  - E.g. 10 individuals: 5 suffer diabetes, 5 healthy
  - One sample from each individual
  - Test: Two sample t-test

- Direct comparisons: 2 groups paired
  - E.g. 10 individuals with brain stroke
  - Two samples from each patient: one from healthy region1 and one from affected region
  - Test: Paired t-test

Vall d'Hebron
Institut de Recerca

Some issues in gene selection

- Some related with small sample sizes
  - Variance instability (very low variances produces a high t statistic value)
  - Non-normality of the data
- Related to the big number of variables (test to perform)
  - Multiple testing problem

Standard t test is not strictly correct to be used here, better to use a "modified version": *moderated t test*

4. Example of a microarray analysis with R. COMPARISONS

Vall d'Hebron
Institut de Recerca

- Multiple testing problem: It is needed to control for the type I error (false positives). FALSE DISCOVERY RATE

- Finally we will be assigning a p-value for each test/gene. If the p-value is lower than an established threshold….

4. Example of a microarray analysis with R. COMPARISONS

Let's do with our data:

```
#CONTRAST MATRIX.lINEAR MODEL
treat<- targets$grupos
lev<-factor(treat, levels=unique(treat))
design <-model.matrix(~0+lev)
colnames(design)<-levels(lev)
rownames(design) <-sampleNames
print(design)

#COMPARISON
cont.matrix1 <- makeContrasts(Induced.vs.WT=Induced-WT,
        levels=design)
comparison1 <- "Effect of Induction"

#MODEL FIT
fit1<-lmFit(eset_filtered$eset, design)
fit.main1<-contrasts.fit(fit1, cont.matrix1)
fit.main1<-eBayes(fit.main1)
```

4. Example of a microarray analysis with R. RESULTS PRESENTATION

	logFC	AveExpr	t	P.Value	adj.P.Val	B
10470175	5.71	7.36	33.67	0.00	0.00	14.45
10351443	5.78	9.60	32.83	0.00	0.00	14.29
10403796	5.65	8.32	30.96	0.00	0.00	13.91
10522388	5.53	8.43	29.77	0.00	0.00	13.65
10469358	-5.37	7.75	-29.50	0.00	0.00	13.59
10531869	5.75	8.79	28.50	0.00	0.00	13.35
10400926	5.28	9.13	28.25	0.00	0.00	13.29
10499189	-4.97	7.92	-27.86	0.00	0.00	13.19
10474524	-4.38	6.24	-27.44	0.00	0.00	13.08
10455942	5.01	7.94	26.70	0.00	0.00	12.89
10482772	4.71	10.33	26.67	0.00	0.00	12.88
10464370	5.24	8.40	26.60	0.00	0.00	12.86
10382341	4.76	8.90	25.26	0.00	0.00	12.48
10362372	4.52	7.65	25.18	0.00	0.00	12.46
10345791	-4.25	8.08	-24.95	0.00	0.00	12.39
10497713	4.67	8.44	24.89	0.00	0.00	12.37
10517513	-4.25	7.90	-24.46	0.00	0.00	12.25
10466200	-5.23	7.84	-23.94	0.00	0.00	12.08
10469816	-4.19	8.74	-23.64	0.00	0.00	11.99
10540401	5.93	8.76	23.28	0.00	0.00	11.87
10477986	4.37	10.42	23.28	0.00	0.00	11.87
10386211	4.19	7.72	23.05	0.00	0.00	11.80
10560919	5.08	9.02	22.63	0.00	0.00	11.65
10585484	4.16	7.36	22.58	0.00	0.00	11.64
10363082	-3.96	7.76	-21.65	0.00	0.00	11.31
10569370	5.92	8.96	21.30	0.00	0.00	11.18
10563597	3.45	11.27	21.17	0.00	0.00	11.13

1. Gene identifiers
2. Log2 Fold Change
3. Average expression
4. t statistics
5. p-values
6. Log-odd statistics

4. Example of a microarray analysis with R. RESULTS PRESENTATION

Vall d'Hebron
Institut de Recerca

Let's do with our data:

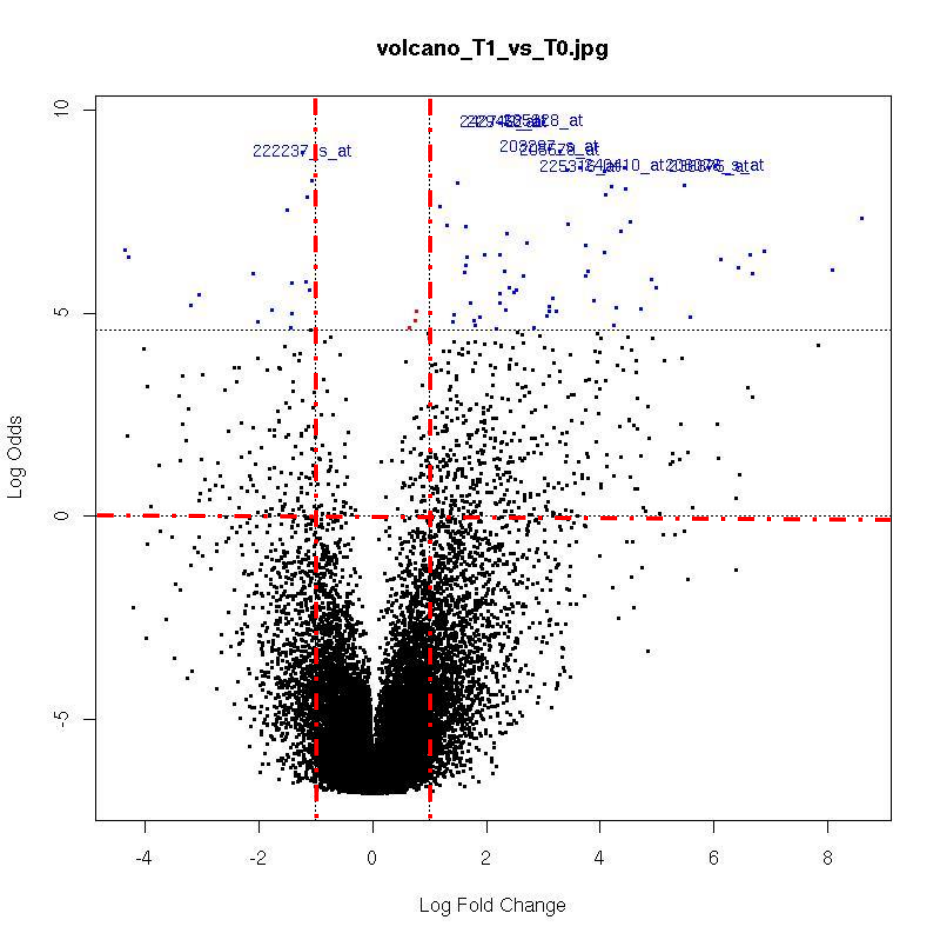
```
#FILTER BY FALSE DISCOVERY RATE AND FOLD CHANGE
topTab <-  topTable (fit.main1, number=nrow(fit.main1), coef="Induced.vs.WT",
adjust="fdr",lfc=abs(3))

#EXPORTED TO HTML
print(xtable(topTab,align="lllllll"),type="html",html.table.attributes="",
      file=paste("Selected.Genes.in.comparison.",comparison1,".html", sep=""))
```

4. Example of a microarray analysis with R. RESULTS PRESENTATION

Vall d'Hebron
Institut de Recerca

Statistics and biological significance representation



4. Example of a microarray analysis with R. RESULTS PRESENTATION

Let's do with our data:

```
volcanoplot(fit.main1, highlight=10, names=fit.main1$ID,
            main=paste("Differentially expressed genes",colnames(cont.matrix1),
sep="\n"))
abline(v=c(-1,1),col="red")


pdf("Volcano.pdf")
volcanoplot(fit.main1, highlight=10, names=fit.main1$ID,
            main=paste("Differentially expressed genes",colnames(cont.matrix1),
sep="\n"))
abline(v=c(-1,1),col="red")
dev.off()
```

4. Example of a microarray analysis with R. RESULTS PRESENTATION

Vall d'Hebron Institut de Recerca

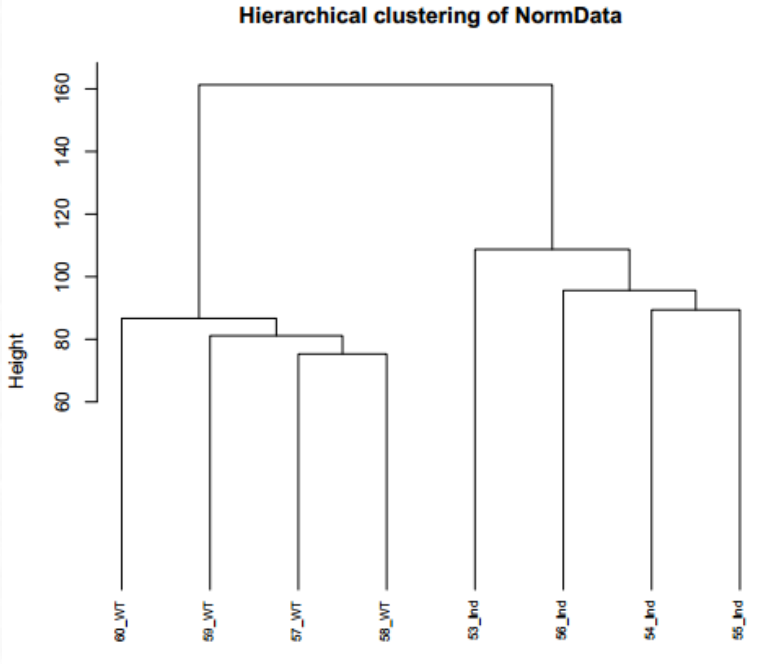
Types:

- **Supervised clustering** try to find the best partition for data that belong to a know set o classes
- **Unsupervised clustering** try to define the number and the size of the classes in which the transcription profiles can be fitted in.
- **Distances** between genes/samples are used to classify them (Euclidian distance, Manhattan distance, Mahalanovis distance….)

4. Example of a microarray analysis with R. RESULTS PRESENTATION

Hierarchical Clustering (HCL)

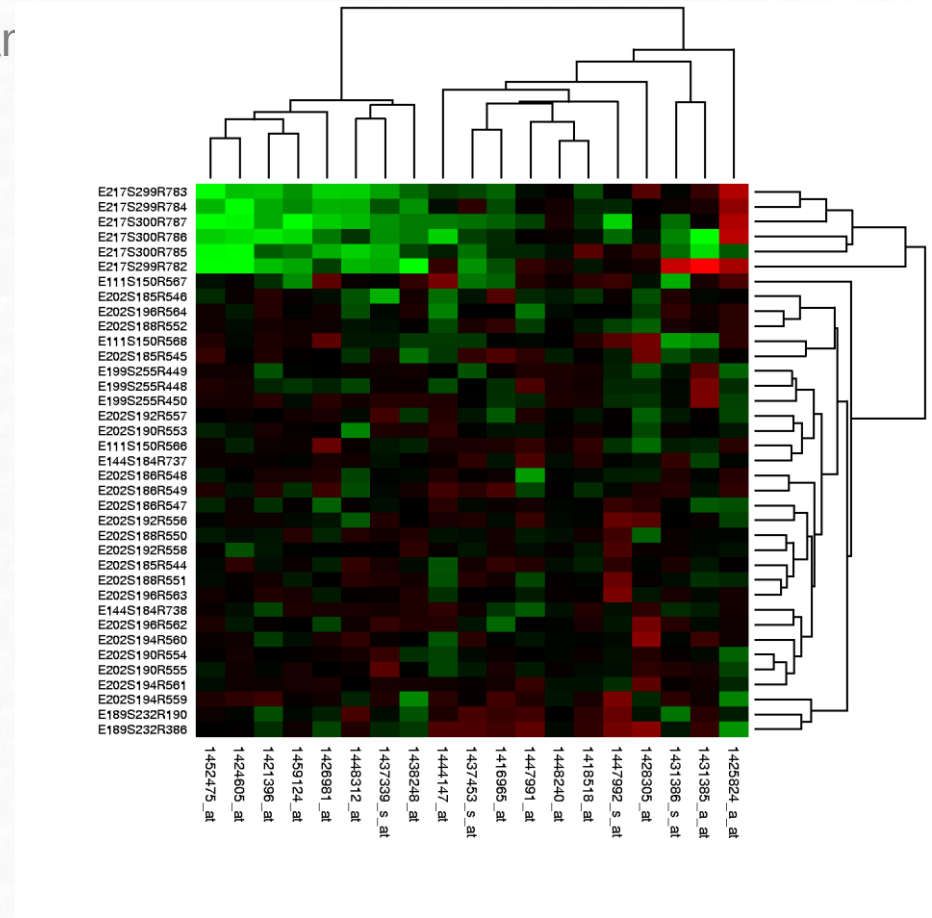
- HCL is an agglomerative /divise clustering method.
- The iterative process continues until all groups are connected in a hierarchical tree.
- Samples more similar between them are closed.



4. Example of a microarray analysis with R. RESULTS PRESENTATION

Heatmaps

- Allow a quick visualization of the possible expression patterns that could exists among sar



4. Example of a microarray analysis with R. RESULTS PRESENTATION

Let's do with our data:

```
#PREPARE THE DATA
my_frame<-data.frame(exprs(eset))
head(my_frame)
HMdata<-merge(my_frame,topTab, by.x=0,by.y=0)
rownames(HMdata)<-HMdata$Row.names
HMdata<-HMdata[,-c(1,10:15)]
head(HMdata)
HMdata2<-data.matrix(HMdata,rownames.force=TRUE)
head(HMdata2)
write.csv2(HMdata2, file="DatatoHM.csv")

#HEATMAP PLOT
my_palette <- colorRampPalette(c("blue", "red"))(n = 299)
heatmap.2(HMdata2,
          Rowv=TRUE,
          Colv=TRUE,
          main="HeatMap Induced.vs.WT FC>=3",
          scale="row",
          col=my_palette,
          sepcolor="white",
          sepwidth=c(0.05,0.05),
          cexRow=0.5,
          cexCol=0.9,
          key=TRUE,
          keysize=1.5,
          density.info="histogram",
          ColSideColors=c(rep("red",4),rep("blue",4)),
          tracecol=NULL,
          srtCol=30)
```

4. Example of a microarray analysis with R. RESULTS PRESENTATION

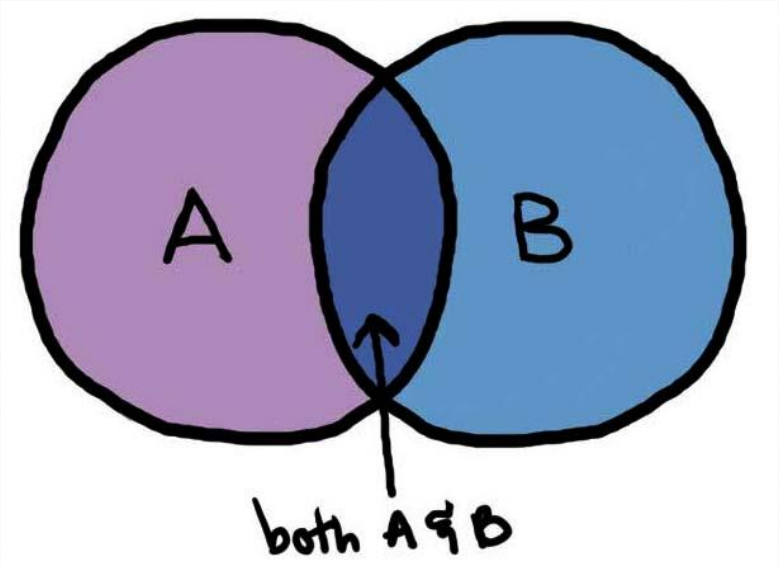
Vall d'Hebron Institut de Recerca

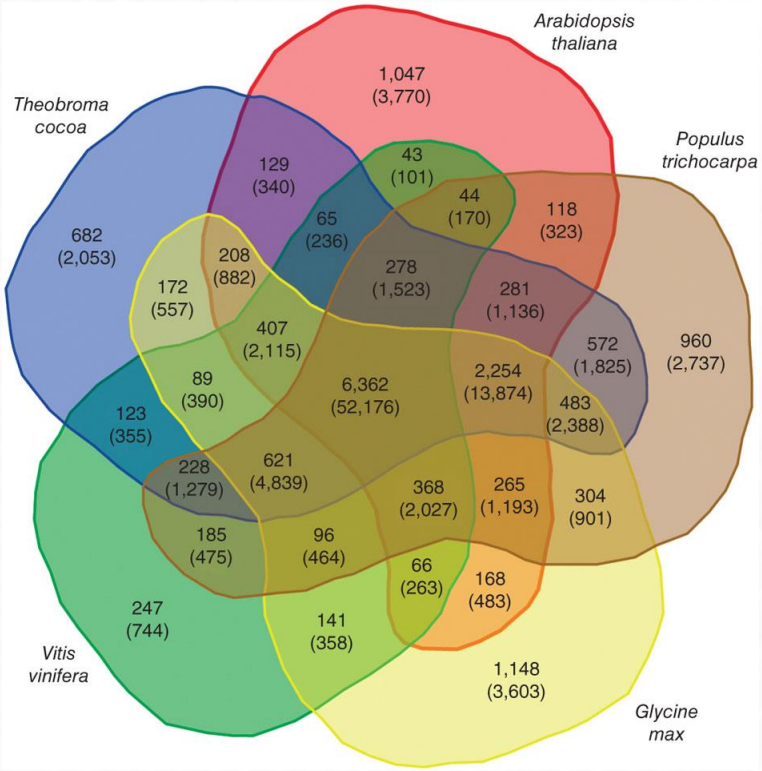
Let's do with our data:

```
#EXPORT TO PDF FILE
pdf("HeatMap InducedvsWT.pdf")
heatmap.2(HMdata2,
          Rowv=TRUE,
          Colv=TRUE,
          main="HeatMap Induced.vs.WT FC>=3",
          scale="row",
          col=my_palette,
          sepcolor="white",
          sepwidth=c(0.05,0.05),
          cexRow=0.5,
          cexCol=0.9,
          key=TRUE,
          keysize=1.5,
          density.info="histogram",
          ColSideColors=c(rep("red",4),rep("blue",4)),
          tracecol=NULL,
          srtCol=30)
dev.off()
```

4. Example of a microarray analysis with R. RESULTS PRESENTATION

- If the study have more than one comparisons it could be interesting to look for common genes in the gene lists (multiple comparisons)





Annotation

- Relation between probes sets and genes.
- An important issue in microarray data analysis is the specific association of probe identifiers with genome annotated transcripts.
- Not of the probes have a “genome annotated transcript”.
- Different database used (Entrez, Gene Symbol, Ensembl,…) generates different results.

4. Example of a microarray analysis with R. RESULTS PRESENTATION

Vall d'Hebron Institut de Recerca

Let's do with our data:

```
all_anota<-data.frame(exprs(eset))
Annot <- data.frame(SYMBOL=sapply(contents(mogene10sttranscriptclusterSYMBOL), paste,
collapse=", "),
                    DESC=sapply(contents(mogene10sttranscriptclusterGENENAME), paste,
collapse=", "))
Annot<-Annot[!Annot$SYMBOL=="NA",]
Annot<-Annot[!Annot$DESC=="NA",]
head(Annot)

anotaGenes <- merge(Annot,all_anota, by.x=0,by.y=0)
head(anotaGenes)
write.table(anotaGenes,file="data.ann.txt",sep="\t")

rownames(anotaGenes)<-anotaGenes[,1]
anotaGenes<-anotaGenes[,-1]
anotaGenes.end <- merge(anotaGenes,topTab, by.x=0,by.y=0)
topTab.end<-anotaGenes.end[,c(1:3,12:17,4:11)]
topTab.end<- topTab.end[order(-topTab.end$B),]

rownames(topTab.end)<-topTab.end[,1]
write.csv(topTab.end,file="TopTable.end.csv")
```