UNIVERSITÉ DU LUXEMBOURG

CENTRE DE RECHERCHE PUBLIC SANTÉ

PhD Course

Advanced Biostatistics

Lecture 6

Advanced Topics

dr. P. Nazarov

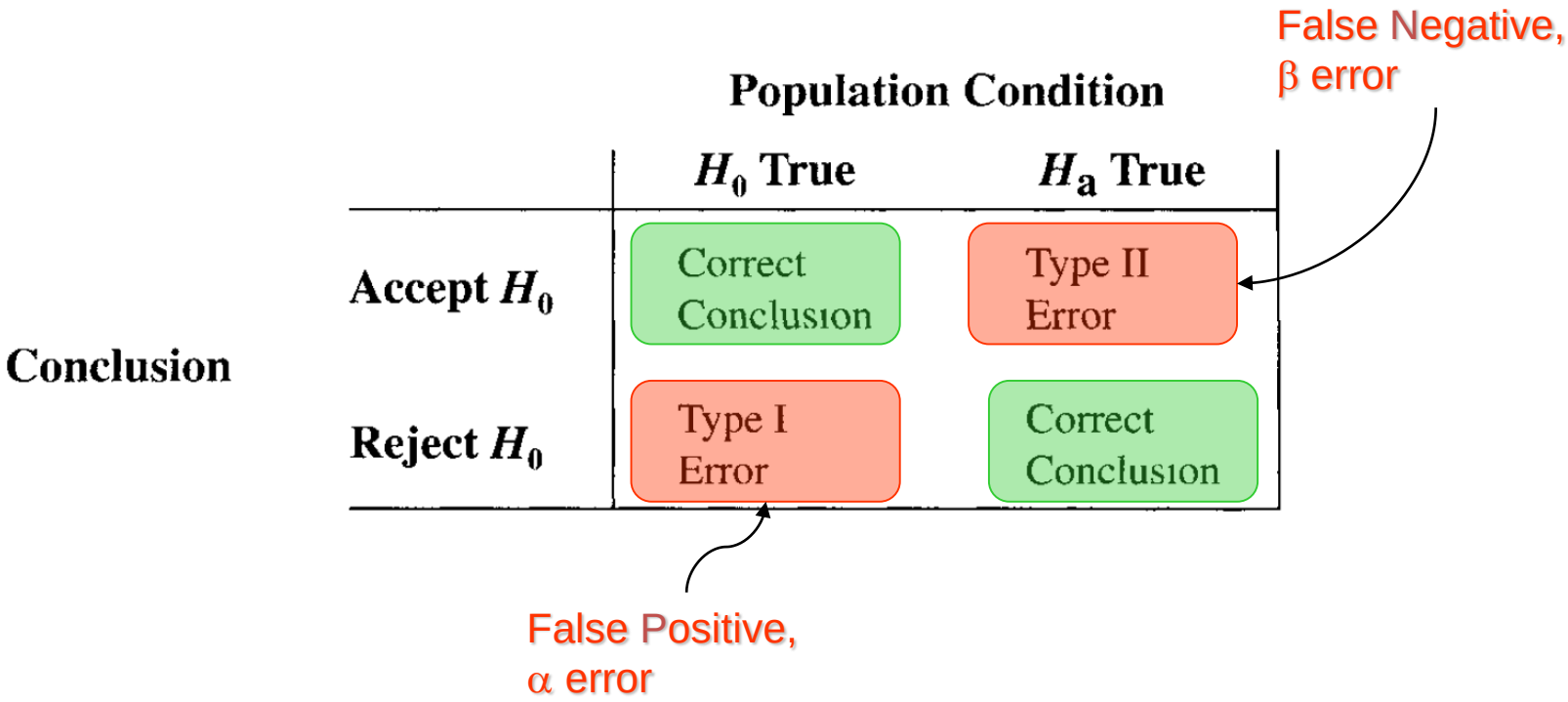
petr.nazarov@crp-sante.lu

17-12-2014

Outline

- Multiple Comparisons (L6.1)
- Survival analysis (L6.2)
- Microarray data analysis (L6.3)
  - Principles
  - Pipeline for data analysis
  - Experiment description
  - APT import
  - QC, differential expression analysis
  - Differential expression analysis
- RNASeq data analysis (L6.4)
- Enrichment analysis (L6.5)

Correct Results and Errors

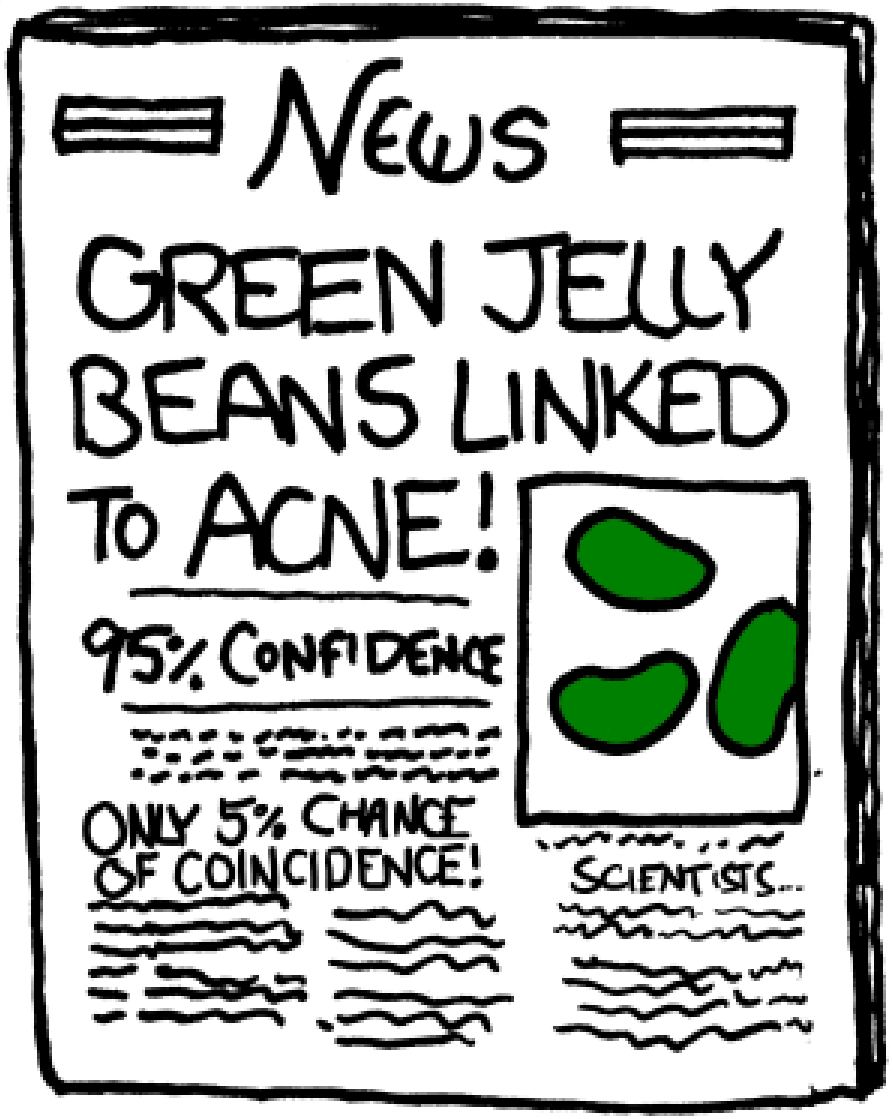


Probability of an error in a multiple test:

$1-(0.95)^{\text{number of comparisons}}$

Differential Expression Analysis

Example



http://www.xkcd.com/882/

False Discovery Rate

False discovery rate (FDR)

FDR control is a statistical method used in multiple hypothesis testing to correct for multiple comparisons. In a list of rejected hypotheses, FDR controls the expected proportion of incorrectly rejected null hypotheses (type I errors).

Conclusion	Population Condition: H_0 is TRUE	H_0 is FALSE	Total
Accept H_0 (non-significant)	U	T	$m - R$
Reject H_0 (significant)	V	S	R
Total	m_0	$m - m_0$	m

$$FDR = E\left(\frac{V}{V+S}\right)$$

False Discovery Rate

Assume we need to perform *m* = 100 comparisons, and select maximum **FDR = α = 0.05**

Independent tests

The Simes procedure ensures that its expected value $\mathrm{E}\left[\frac{V}{V+S}\right]$ is less than a given α (Benjamini and Hochberg 1995). This procedure is valid when the m tests are independent. Let $H_1 \ldots H_m$ be the null hypotheses and $P_1 \ldots P_m$ their corresponding p-values. Order these values in increasing order and denote them by $P_{(1)} \ldots P_{(m)}$. For a given α, find the largest k such that $P_{(k)} \leq \frac{k}{m}\alpha$.

Then reject (i.e. declare positive) all $H_{(i)}$ for $i = 1, \ldots, k$.

Note that the mean α for these m tests is $\frac{\alpha(m+1)}{2m}$ which could be used as a rough FDR, or RFDR, "α adjusted for m indep. tests." The RFDR calculation shown here provides a useful approximation and is not part of the Benjamini and Hochberg method; see AFDR below.

False Discovery Rate: Benjamini & Hochberg

Assume we need to perform *m* = 100 comparisons, and select maximum **FDR = α = 0.05**

$$FDR = E\left(\frac{V}{V+S}\right)$$

Expected value for FDR < α if $P_{(k)} \leq \frac{k}{m}\alpha$

$$\frac{mP_{(k)}}{k} \leq \alpha$$

```
p.adjust(pv, method="fdr")
```

Other Methods

Bonferroni – simple, but too stringent, not recommended

Holm – a more powerful and less stringent version of Bonferroni (ok)

L6.1. Multiple Comparisons

p-value or FDR?

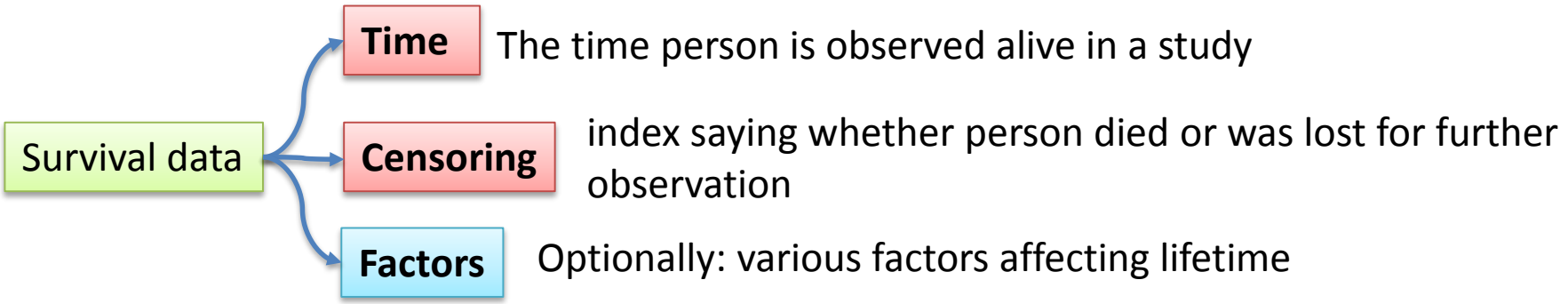
Let's generate a completely ran dom experiment (script L6.1)

Survival Data

Survival analysis
is a branch of statistics which deals with analysis of time to events, such as death in biological organisms and failure in mechanical systems (i.e. **reliability theory** in engineering).

Survival analysis attempts to answer questions such as:

- What is the proportion of a population which will survive past a certain time?
- Of those that survive, at what rate will they die or fail?
- Can multiple causes of death or failure be taken into account?
- How do particular circumstances or characteristics increase or decrease the probability of survival?



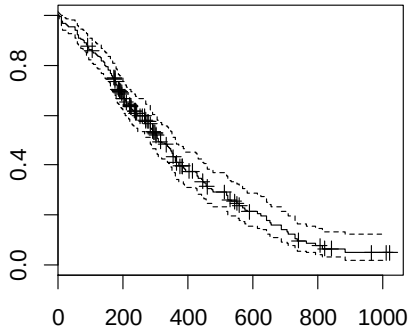
http://www.partek.com/Survival%20Analysis?mkt_tok=3RkMMJWWfF9wsRogv6nMZKXonjHpfsX56%2BwqW6a3lMI%2F0ER3fOvrPUfGjI4CRMNqI%2BSLDwEYGJlv6SgFTrnDMbZlzLgJXRQ%3D

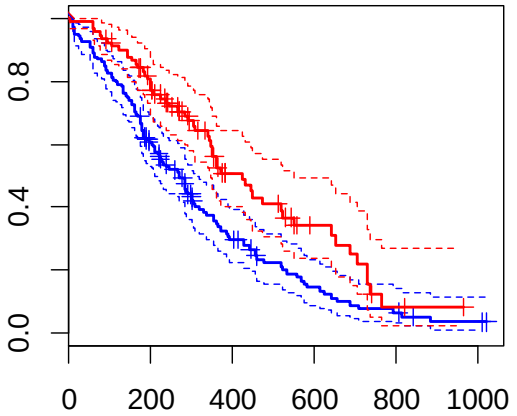
L6.2. Survival Analysis

Example: Lung

```
library(survival)
str(lung)

## create a survival object
## lung$status: 1-censored, 2-dead
sData = Surv(lung$time,event = lung$status == 2)
print(sData)

## Let's visualize it
fit = survfit(sData~1)
plot(fit)

## Let's visualize it for male/female
fit.sex = survfit(sData ~ lung$sex)
plot(fit.sex, col=c("blue","red"), conf.int = TRUE)

## Rank test for survival data
dif.sex = survdiff(sData ~ lung$sex)
dif.sex

## build Cox regression model
mod = coxph(sData ~ sex + age, data=lung)
summary(mod)
```

"event" should be:
0 – for censored
1 – for dead patients



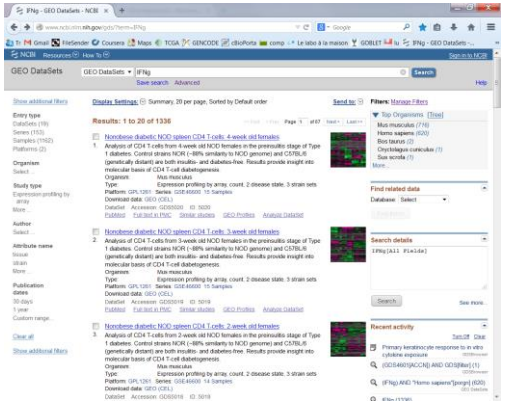


ovarian

L6.3. Microarrays

Public Repositories

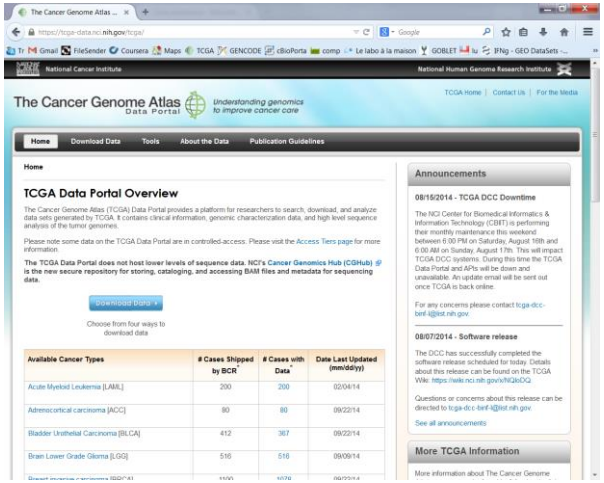
GEO: http://www.ncbi.nlm.nih.gov/gds



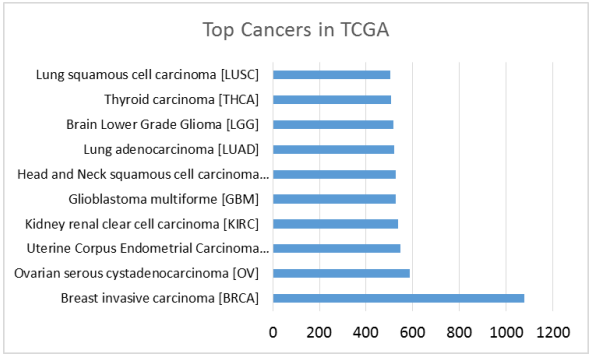
Browse Content

Repository Browser	
DataSets:	3847
Series:	50810
Platforms:	13387
Samples:	1237318

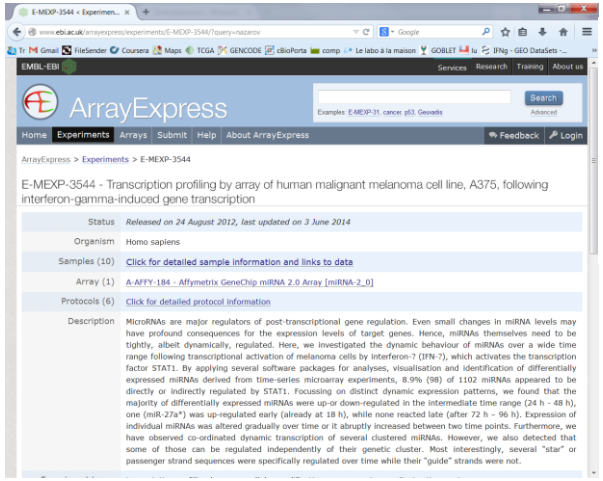
TCGA: https://tcga-data.nci.nih.gov/tcga/



Sep 2014 – more then 10k patients



ArrayExpress: http://www.ebi.ac.uk/arrayexpress/



Data Content

Updated today at 06:00

- 52801 experiments
- 1555904 assays
- 24.99 TB of archived data

Analysis via:
http://www.cbioportal.org/public-portal/

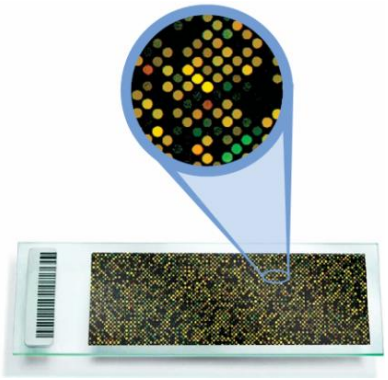
Data for our course: http://edu.sablab.net/transcript

L.6.3. Microarray Data

Types of Microarrays

Two-color Arrays (2C)

- Agilent full genome
- Thematic arrays



Pro

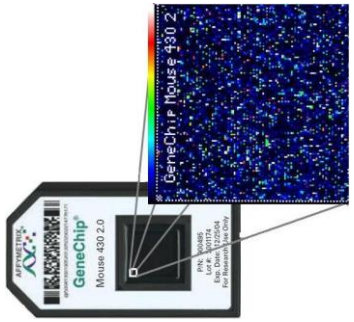
- Direct comparison
- Less sensitive to inaccuracies of spotting

Con

- Dye effects: need for “dye-swaps”
- Non-flexibility in analysis

One-color Arrays (1C)

- Affymetrix GeneChip
- Affymetrix Exon
- Affymetrix mRNA



Pro

- Flexible analysis
- High level of standardization

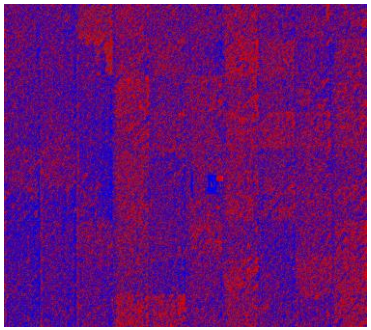
Con

- Price

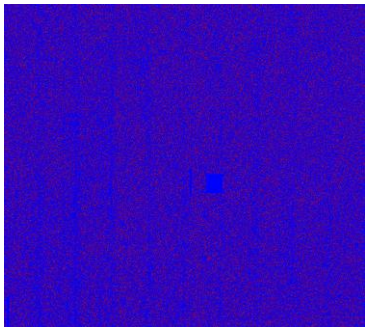
Microarray Data

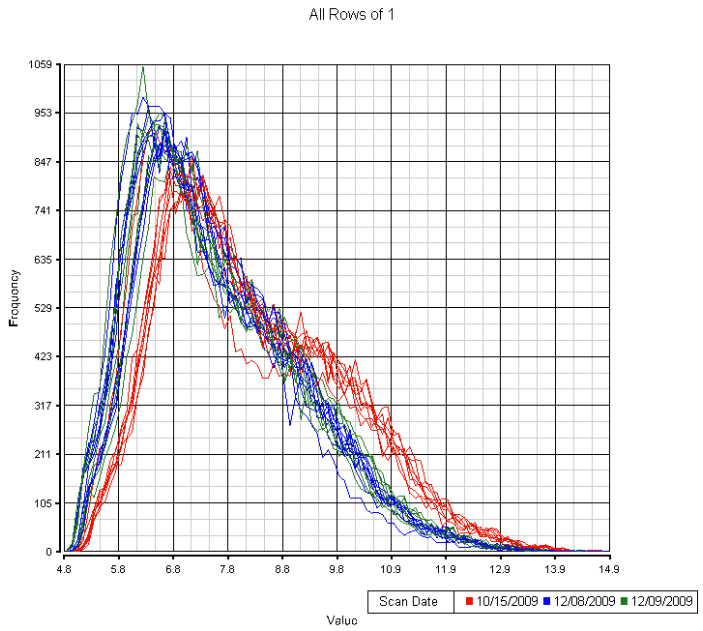
One-color Arrays

Raw

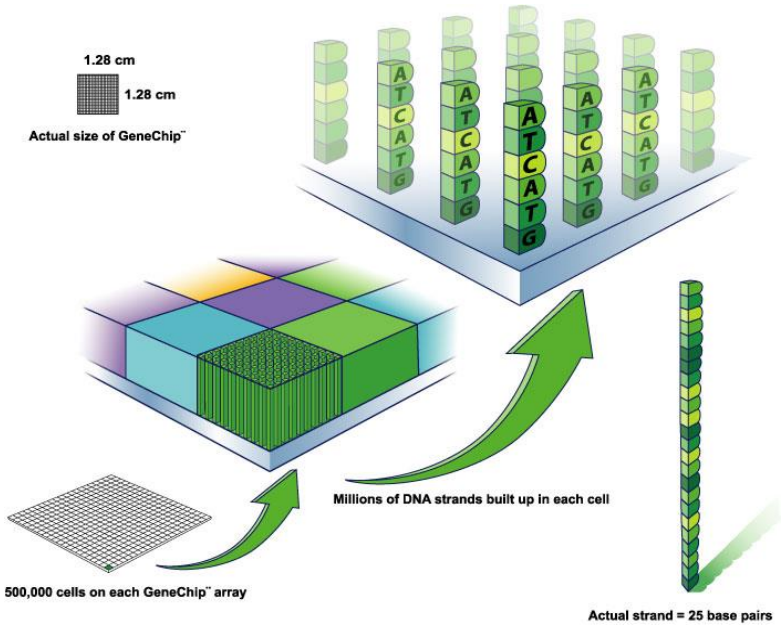


Normalized





High reproducibility and quality of spotting is required.
Affymetrix – "photolithography"-like technique



$$LogIntensity = log_2(I)$$

Background is "removed" during normalization step

Filtering may help removing uninformative features

Affymetrix: Probes, Probesets and Transcript clusters

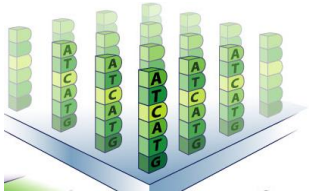
Probes
25-mer sequences targeted on a single region of transcriptome (hopefully)

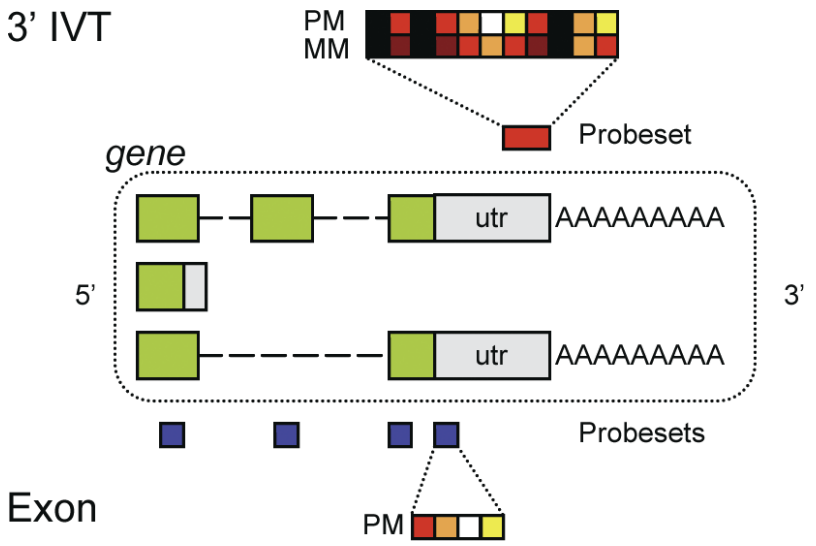
Probesets
groups of closely located or overlapped probes (on average 4 probes)

Exons
HuExon and HTA arrays allow measuring exon expression

Transcript clusters
For majority of features - synonymous to "genes". However, some distinct transcripts of genes are considered as different transcript clusters.

In old versions of Affy arrays (hgu95, hgu133, etc), there were:
PM – perfect match probes
MM – mismatch probes (having replacement in th 13th character)
This was done for background estimation.
But this approach is not used now!!





Okoniewski M, Comprehensive Analysis of Affymetrix Exon Arrays Using BioConductor, PLoS CompBiol, 2008

Normalization of Affymetrix Arrays by RMA

Background correction

↓

Normalization b/w arrays

↓

Estimate expression

Background and signal are strictly positive.
Noise is additive in log scale:

$$PM_{ij} = \underset{\text{exponential}}{Sijn} + \overset{\text{normal}}{Bijn}$$

Quantile **normalization** b/w arrays: makes distribution of probes the same across all arrays

Probeset expression is estimated from a linear model:

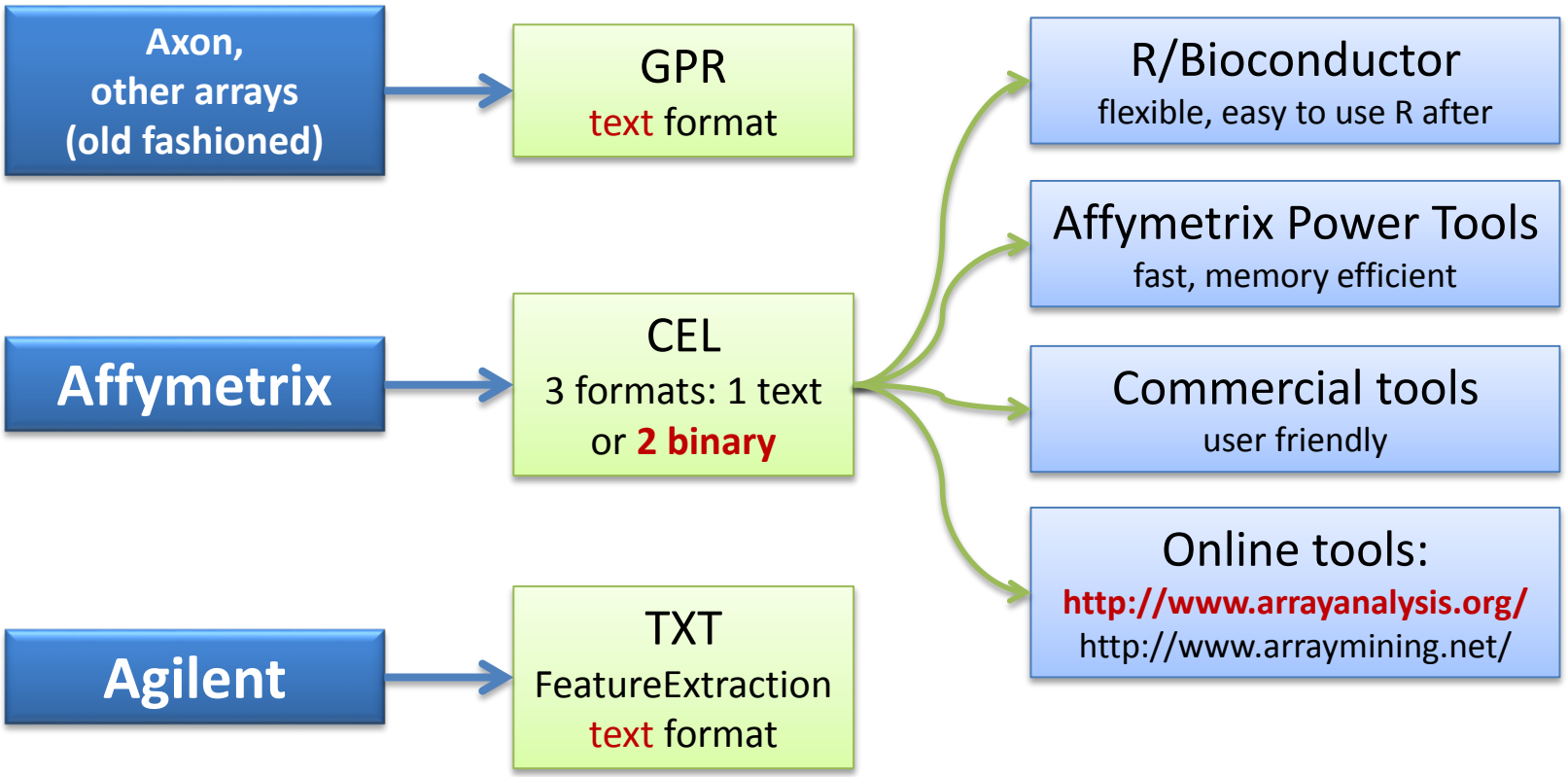
$$\underset{\text{observed}}{Y_{ijn}} = \underset{\text{expression}}{\mu_{in}} + \underset{\text{probe affinity}}{\alpha_{jn}} + \underset{\text{error with 0 mean}}{\varepsilon_{ijn}}$$

i – array
j -- probe
n -- probeset

"Median polish" helps avoid outliers effect

Microarray Data

File Formats



Microarray Data

Analysis Pipeline

Raw CEL files

Text files with $\log_2$ gene (or probeset) expression

Lists of differentially expressed gene

Image Analysis

Pre-processing

Quality control

Statistical Analysis

Affymetrix© software

- Background correction
- Normalization
- Summarization

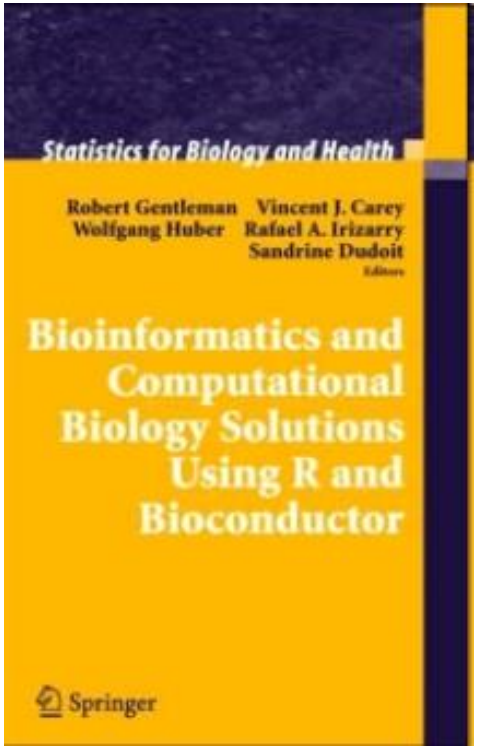
Remove low quality arrays, if necessary

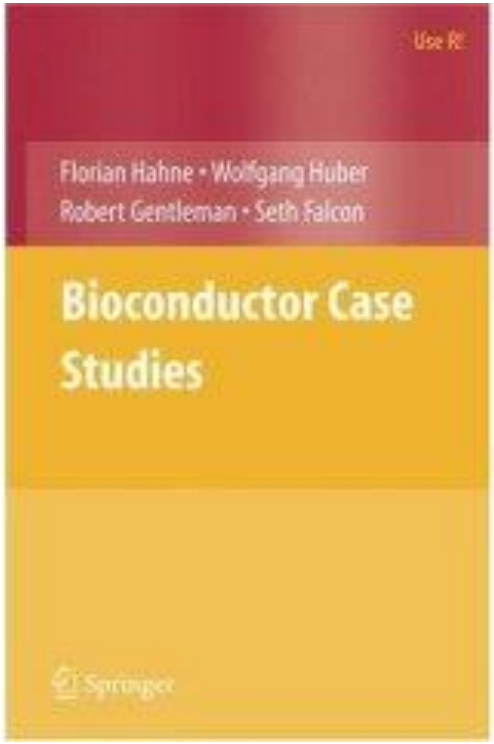
- Significance analysis for differentially expressed genes (DEG)
- Prediction analysis
- Co-regulation analysis
- Etc.

Bioinformatics

L6.3. Microarray Data Analysis

R / Bioconductor



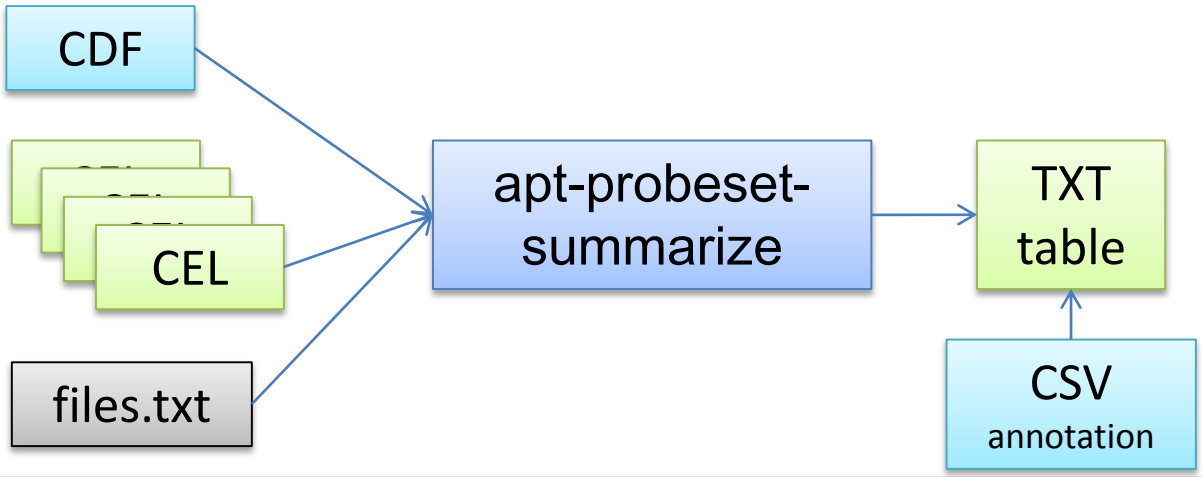


Affymetrix Power Tools

apt-probeset-summarize is a program for doing background subtraction, normalization and summarizing probe sets from Affymetrix expression microarrays. It implements analysis algorithms such as RMA, Plier, and DABG (detected above background).
The main features of **apt-probeset-summarize** not common in other implementations are: Quantile normalization using a subset (sketch) of the data which results in much smaller memory usage.

http://www.affymetrix.com/support/developer/powertools/changelog/apt-probeset-summarize.html

```
apt-probeset-summarize
    -a rma-sketch -d chip.cdf -o output-dir  --cel-files  files.txt
```



http://edu.sablab.net/data/gz/

Differential Expression Analysis

Factorial design

Design Matrix
design = model.matrix(…)

Model
mod = lmFit(x,design)

Contrast Matrix
CM = makeContrast (…, design)

Contrasts Fit
mod2= contrasts.fit(mod,desig)

Empirical Bayesian Method
EB = eBayes (mod2)

Generate List of Genes
topTable (EB,…)

https://sablab.net/scripts

limmaEBS2Class.r

limmaEBSnClass.r

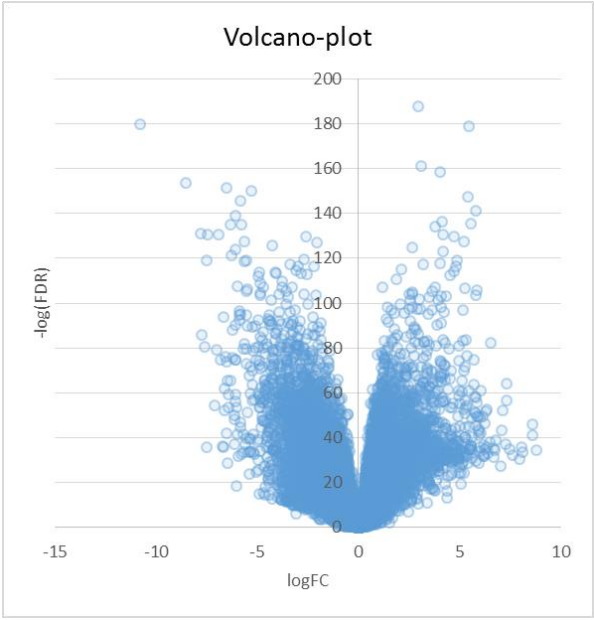
Differential Expression Analysis

http://edu.sablab.net/data/txt/lusc.zip

1. Find genes significantly differentially expressed in SCC vs normal tissue
   - apply ***limma***
   - Keep genes with FDR > 0.001
   - keep only genes with |logFC| > 2

2. Make a “volcano plot”:
   -log10(FDR) vs LogFC

3. Save lists of up and down regulate genes – we shall need them



htts://sablab.net/scripts

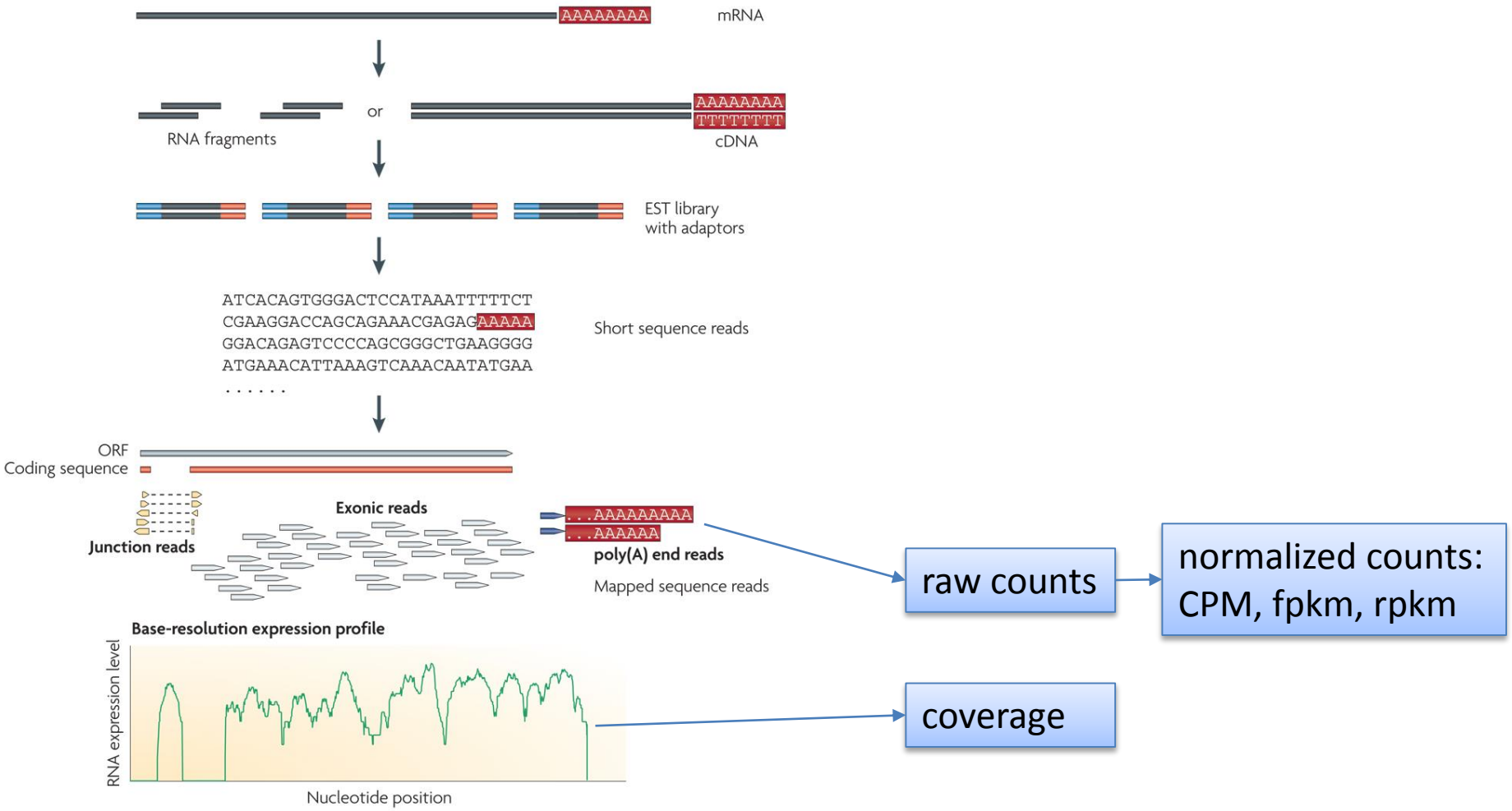
L6.3. Microarray Data Analysis

```
################################################################################
# L6.2. Import and Analysis
################################################################################
## clear memory
rm(list = ls())

##------------------------------------------------------------------------------
## L6.2.1. Loading results after APT and QC
##------------------------------------------------------------------------------

## load data after APT
## alternative: http://edu.sablab.net/data/gz/rma-sketch.summary.txt
Data = read.table("e:/data/kreis/+data+/miR.pub/cel/res/rma-sketch.summary.txt",
                  header=T, sep="\t", as.is=T)

## load sample description
## alternative: http://edu.sablab.net/data/gz/Affymetrix_miRNA2.txt
Meta= read.table("e:/data/kreis/+data+/miR.pub/cel/files.txt",
                  header=T, sep="\t", as.is=T)
str(Data)
Meta

## keep only human miRNA
Data = Data[grep("hsa-",Data$probeset_id),]

# if order of Data columns and Meta rows are the same - simply change columns
if (sum(names(Data[,-1])!= Meta[,1]) == 0) names(Data)[-1] = Meta[,2]

source("http://sablab.net/scripts/plotDataPDF.r")
x11()
plotDataPDF(Data,add.legend=T,col=rainbow(ncol(Data)))
x11()
boxplot(Data[,-1],outline=F,col=rainbow(ncol(Data)),las=2)
```

```
##------------------------------------------------------------------------------
## L6.2.2 Analysis
##------------------------------------------------------------------------------

## let's filter out miR with low expression
## and put the rest into matrix Y
thr = 3
idx.keep = logical(nrow(Data))|T
idx.keep[apply(Data[,-1],1,max)<=thr]=F
sum(idx.keep)

## Y contains now the data
Y = as.matrix(Data[idx.keep,-1])
colnames(Y) = names(Data)[-1]
rownames(Y) = sub("_st","",Data[idx.keep,1])
str(Y)

## plot heatmap of scaled data
heatmap(t(scale(t(Y))))

## plot PCA
PC = prcomp(t(Y))
## plot 3D
library(rgl)
plot3d(PC$x[,1],PC$x[,2],PC$x[,3],
       size = 2,
       col  = rainbow(ncol(Data)),
       type = "s",
       xlab = "PC1",
       ylab = "PC2",
       zlab = "PC3")
text3d(PC$x[,1]+0.5,PC$x[,2]+0.5,PC$x[,3]+0.5,colnames(Y))

## DEA
source("http://sablab.net/scripts/limmaEBS2Class.r")

idx=c(grep("T000",colnames(Y)),
      grep("T48",colnames(Y)))
res=limmaEBS2Class(Y[,idx],rownames(Y),classes=c("T00","T00","T48","T48"),
                   plotTop=20)
```

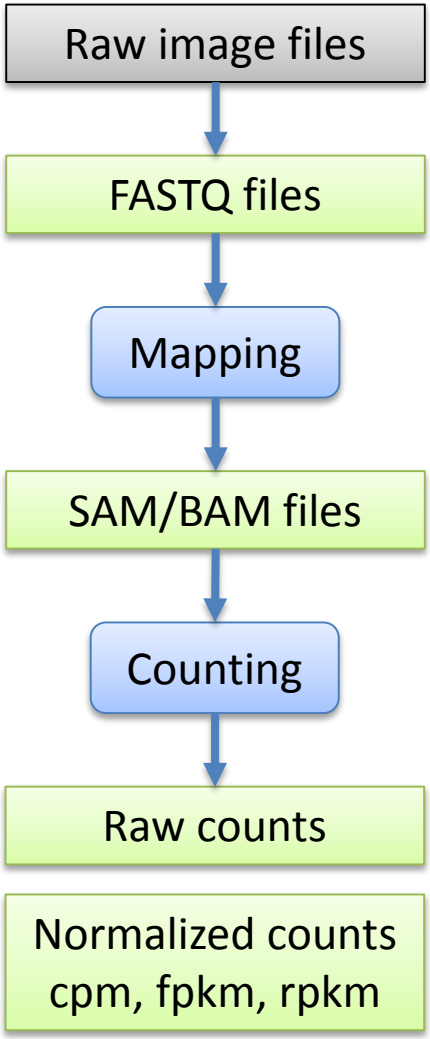
RNASeq

Next Generation Sequencing: RNA-Seq



Wang Z et al. RNA-Seq: a revolutionary tool for transcriptomics. **Nat Rev Genet. 2009**

File Types




```
@HWI-ST508:152:D06G9ACXX:2:1101:1160:2042 1:Y:0:ATCACG
NAAGACCGAATTCTCCAAGCTATGGTAAACATTGCACTGGCCTTTCATCTG
+
#11??+2<<<CCB4AC?32@+1@AB1**1?AB<4=4>=BB<9=>?######
```

Read – a short sequence identified in RNA-Seq experiment
Library – set (10^5 – 10^8) of reads from a single sample

```
@HD         VN:1.0 SO:coordinate
@SQ         SN:seq1 LN:5000
@SQ         SN:seq2     LN:5000
@CO         Example of SAM/BAM file format.

B7_591:4:96:693:509 73   seq1       1          99          36M         *
            0            0          CACTAGTGGCTCATTGTAAATGTGTGGTTTAACTCG
                                    <<<<<<<<<<<<<<;<<<<<<<<<5<<<<<;:<;7
            MF:i:18      Aq:i:73    NM:i:0      UQ:i:0      H0:i:1

H1:i:0EAS54_65:7:152:368:113        73          seq1        3           99
            35M          *          0           0
            CTAGTGGCTCATTGTAAATGTGTGGTTTAACTCGT
            <<<<<<<<<<0<<<<655<<7<<<:9<<3/:<6): MF:i:18     Aq:i:66
            NM:i:0       UQ:i:0     H0:i:1      H1:i:0
```

For the list of tools see:
http://en.wikipedia.org/wiki/List_of_RNA-Seq_bioinformatics_tools

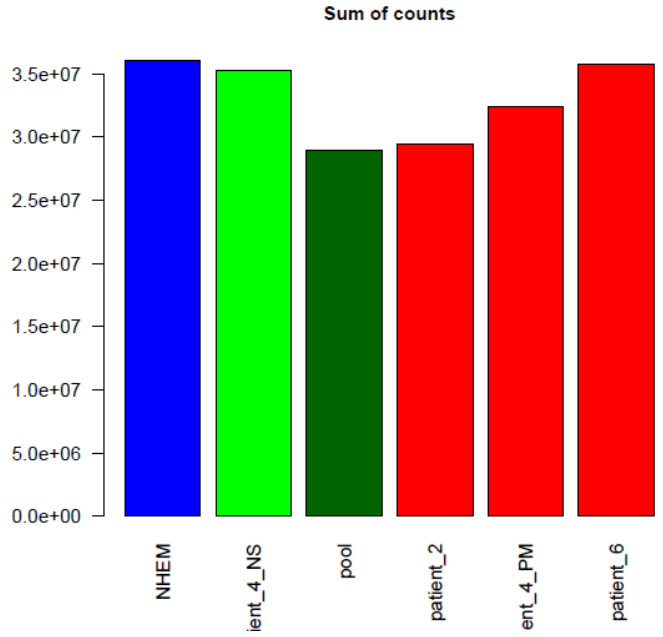
Normalization

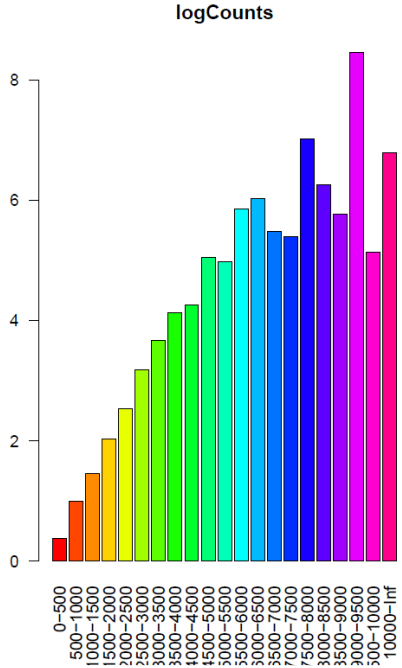
Problems:

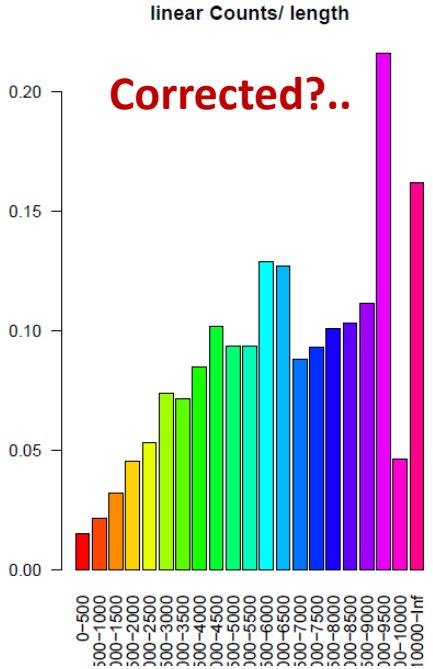
- Libraries has different size (different number of reads from samples)
- Long transcripts produce more reads

Solutions (?) :

- Accounting for library size during analysis (standard) or direct correction for it
- Correction for transcript size (but which transcript is expressed?)

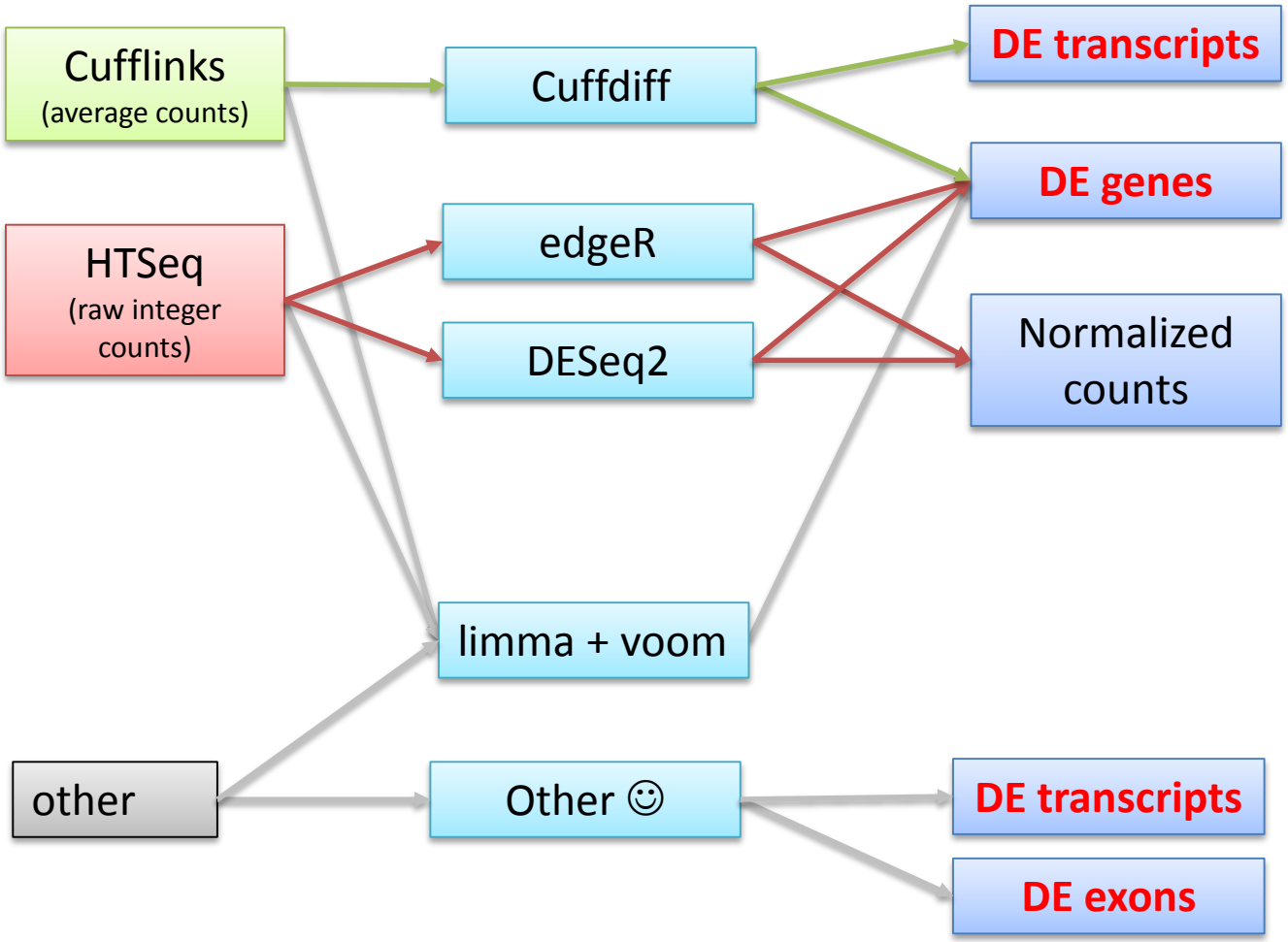






6.4. RNA-Seq Data

Differential Expression Analysis



L6.4. RNASeq Data

Differential Expression Analysis (edgeR)

htts://sablab.net/scripts LibDEA.r

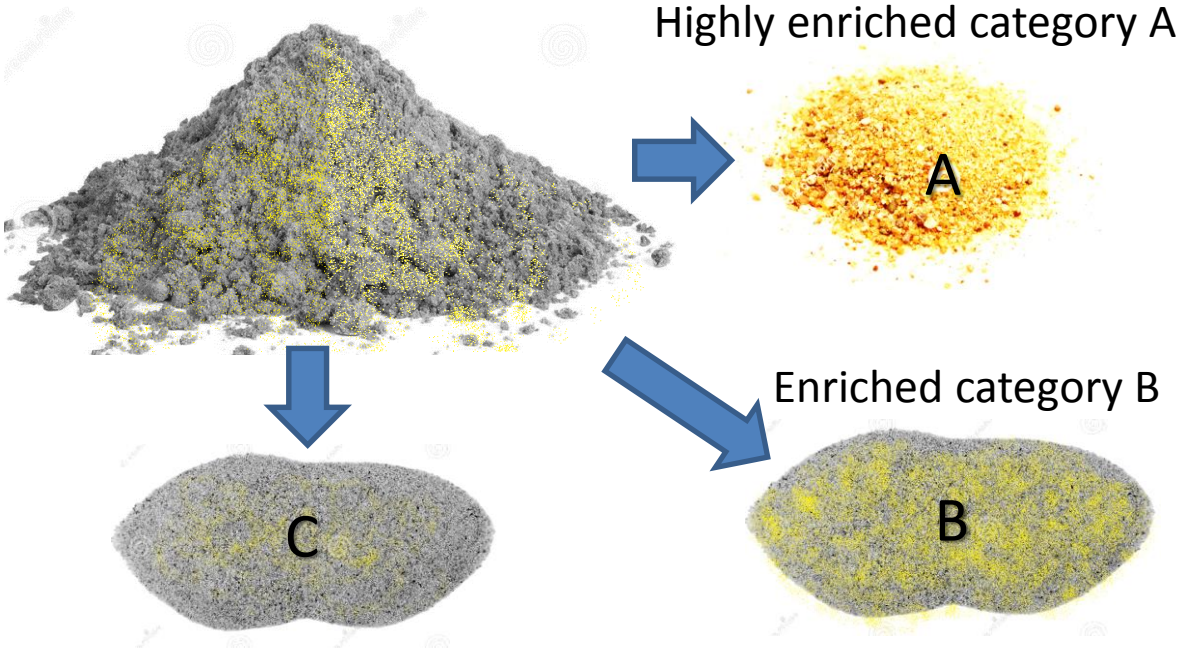
Differential Expression Analysis (DESeq2)

htts://sablab.net/scripts LibDEA.r

Enrichment Analysis

1. Category Enrichment Analysis

Are interesting genes overrepresented in a subset corresponding to some biological process?



Method of the analysis:
Fisher's exact test

Someone grabs "randomly" 20 balls from a box with 100x ● and 100x ●

How surprised will you be if he grabbed

●●●●●●●●●●●●●●●●●●●●

(17 red , 3 green)

sand belongs to: http://www.dreamstime.com/photos-images/pile-sand.html ;)))

1. Category Enrichment Analysis

Fisher's exact test: based on hypergeometrical distributions

Hypergeometrical: distribution of objects taken from a "box", without putting them back

$$P = 1 - \sum_{i=0}^{k-1} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}}$$

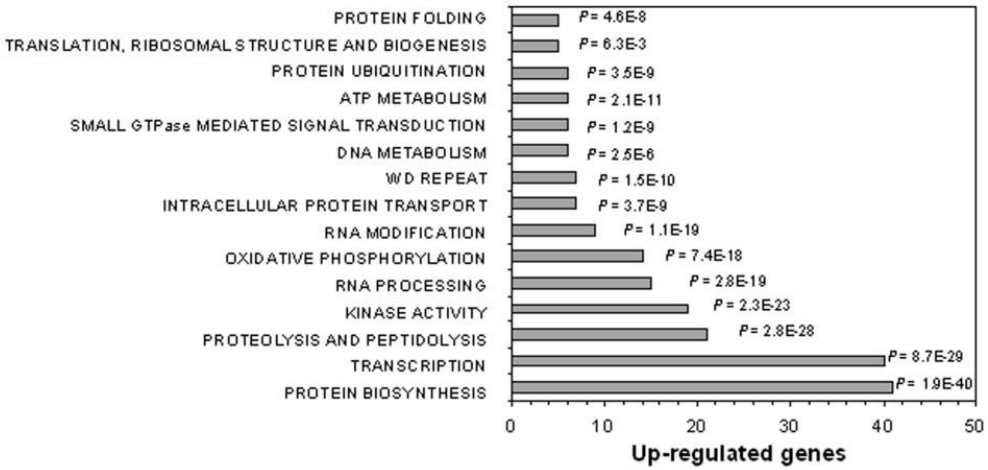
N: total number of genes

M: total number of genes annotated with this term

n: number of genes in the list

k: number of genes in the list annotated with this term

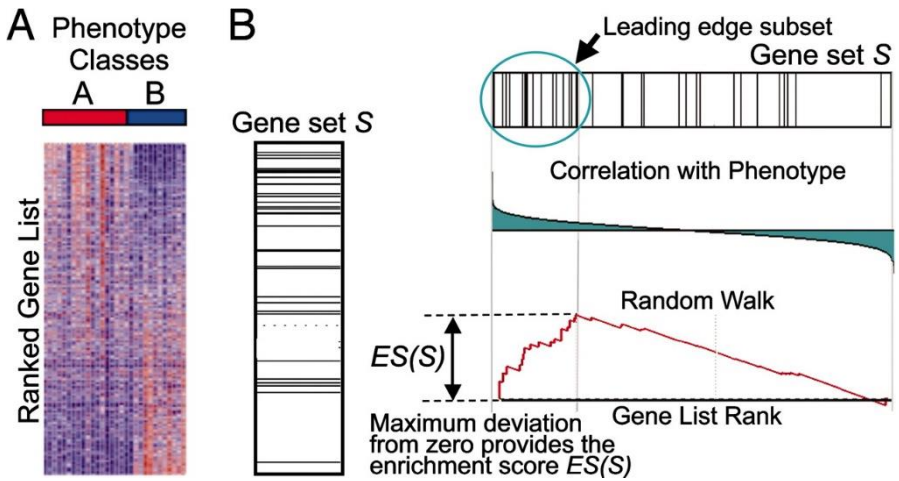
$$C_k^n = C_n^k = \binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

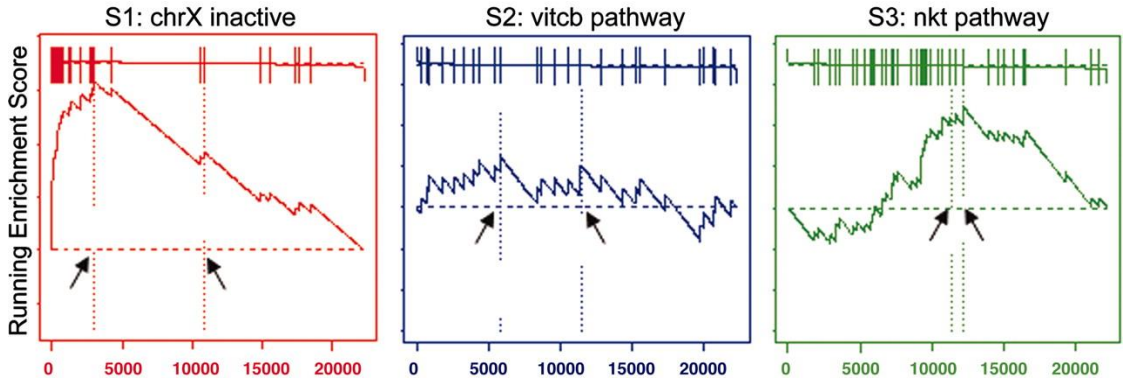


Okamoto et al. Cancer Cell International 2007 7:11 doi:10.1186/1475-2867-7-11

2. Gene Set Enrichment Analysis (GSEA)

Is direction of genes in a category random?





A. Subramanian et al. PNAS **2005**,102,43

L6.5. Enrichment Analysis

Example: GO enrichment

http://edu.sablab.net/transcript

Strategy 1:

Take all DEG and use them in enrichment.

- Safe
- No additional assumptions
- Cannot distinguish ↑ and ↓ functions

Strategy 2:

Separate DEG to down- and up- regulated genes. Then perform independent enrichment by these 2 groups

- Can be biased (gene can be ↑↓)
- Assume ↑gene => ↑function
- Can distinguish ↑ and ↓ functions

Enrichr
http://amp.pharm.mssm.edu/Enrichr/enrich

BioCompendium
http://biocompendium.embl.de/

L6.5. Enrichment Analysis

LUSC Example

http://edu.sablab.net/data/txt/lusc.zip

http://amp.pharm.mssm.edu/Enrichr/

http://biocompendium.embl.de/

http://string-db.org

0. Prepare lists of DE genes…

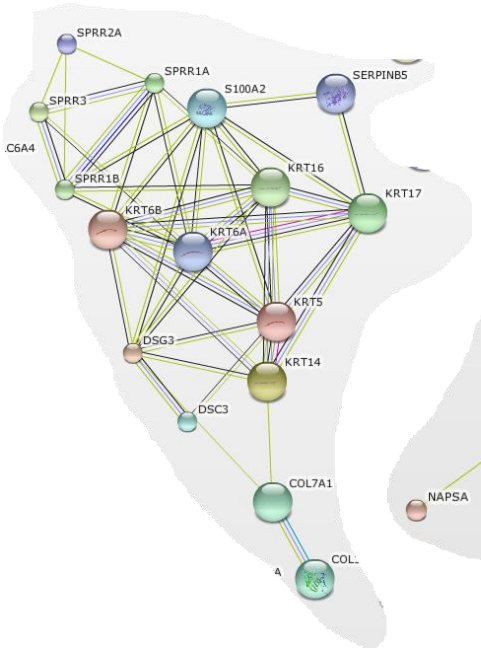
1. Put up-regulated into **enrich**

3. Check: Down CMAP, Disease Signatures from GEO up,

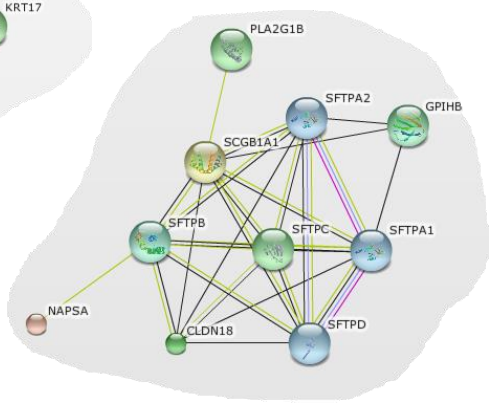
4. Try **biocompendium**

5. Put top 100 genes into String to see PP-interactions

Up regulated



Down regulated



L6.5. Enrichment Analysis

In R

```
################################################################################
## enrichGOens - warpup for topGO package: enrichment analysis of GO-terms
## based on Ensembl IDs
##------------------------------------------------------------------------------
## genes   - vector with list of ENSEBML IDs (character)
## fdr     - vector of FDR for each gene (numeric)
## fc      - vector of logFC for each gene (numeric)
## thr.fdr - significance threshold for FDR (numeric)
## thr.fc  - significance threshold for absolute logFC (numeric)
## db      - name of GO database: "BP","MF","CC" (character)
## genome  - R-package for genome annotation used. For human - 'org.Hs.eg.db' (character)
## do.sort - if TRUE - resulted functions sorted by p-value (logical)
## randomFraction - for testing only, the fraction of the genes to be randomized (numeric)
##------------------------------------------------------------------------------
## (c)GNU GPL P.Nazarov 2014. petr.nazarov[at]crp-sante.lu
################################################################################
```

https://sablab.net/scripts

enrichGOens.r

```
enrichGOens =
function(genes,fdr,fc,thr.fdr=0.05,thr.fc=0,db="BP",genome="org.Hs.eg.db",do.sort=TRUE,
         randomFraction=0){
  ## load libraries
  if (!require(genome,character.only=TRUE)){
    cat("MESSAGE enrichGO: '",genome,"' package is not found. Installing...\n",sep="")
    source("http://bioconductor.org/biocLite.R")
    biocLite(genome)
    library(genome,character.only=TRUE)
  }
  if (!require("topGO")){
    cat("MESSAGE enrichGO: ' topGO ' package is not found. Installing...\n")
    source("http://bioconductor.org/biocLite.R")
    biocLite("topGO")
    library("topGO")
  }
  if (!exists("sortDataFrame")) source("http://sablab.net/scripts/sortDataFrame.r")
  ## prepare gene categories and score
  myGO2genes <- annFUN.org(db, mapping = "org.Hs.eg.db", ID = "ensembl")
  score = (-log10(fdr)*abs(fc))
  names(score)=genes
  score[fdr>=thr.fdr | abs(fc)<=thr.fc]=0

  ## add randomness if required, to test stability
  if (randomFraction>0){
    ## define remove probability: low score have more chances
    prob1 = 1/(1+score)
    prob1[is.na(prob1)]=0
    prob1[score == 0] = 0
    ## define add probability: high score has more chances
    prob2 = -log10(fdr)*abs(fc)
    prob2[is.na(prob2)]=0
    prob2[score > 0] = 0
    ## add and remove
    n=round(sum(score>0)*randomFraction)
    score[sample(1:length(genes),n,prob=prob2)]=1+rexp(n,1/mean(score[score>0]))
    score[sample(1:length(genes),n,prob=prob1)]=0
  }
  ## create topGOdata object
  SelectScore = function(sc){return(sc>0)} ## simple function for significance
  GOdata = new("topGOdata",  ##constructor
               ontology = db,
               allGenes = score,
               geneSelectionFun = SelectScore,
               annot = annFUN.GO2genes,
               GO2genes = myGO2genes)
  ## run testing
  resFisher = runTest(GOdata, algorithm = "classic", statistic = "fisher")
  ## transform results into a table
  enrichRes = GenTable(GOdata, classicFisher = resFisher,
               ranksOf = "classicFisher",topNodes = length(resFisher@score))
  enrichRes$classicFisher[grep("<",enrichRes$classicFisher)] = "1e-31"
  enrichRes$classicFisher = as.double(enrichRes$classicFisher)
  enrichRes$FDR = p.adjust(enrichRes$classicFisher,"fdr")
  enrichRes$Score = -log10(enrichRes$FDR)
  ## by default sorted by p-value. If needed - sort by ID
  if (!do.sort) enrichRes = sortDataFrame(enrichRes,"GO.ID") ## remove sorting
  return(enrichRes)
}
```

Questions ?

Thank you for your attention !