Microarray data analysis

Ståle Nygård

Bioinformatics Core Facility, OUS - RR
Web: core.rr-research.no/bioinformatics

and

Institute for Experimental Medical Research, OUS - Ullevål
Web: iemr.no

Sep 09, 2009

Outline

- Experimental design
- Analysis package Bioconductor
- Low level analysis ("Preprocessing" or "cleaning" of the data)
- High level analysis
  - Gene focused analysis
    - Find list of interesting genes.
    - Characterize the list of genes: Functional enrichment, network construction
  - Patient focused analysis
    - Classification of the patients into disease subtypes
    - Predicting outcome (e.g time to death/relapse) of patients

Experimental design

Issues to consider

- How many replications
- Pooling vs non-pooling
- Strategies for pairing hybridization targets on two-channel (cDNA) arrays

Some consensus points:

- Biological replication is essential. You should have at least 5 biological cases. NB! Power analysis methods for microarray data exist (e.g. Pawitan et al, 2005).
- Pooling biological samples can be useful. Technical variation often seen to be smaller than biological variation.
- Avoid confounding by extraneous factors
- Groups to be compared should be hybridized to the same arrays (two-channel cDNA arrays)

Bioconductor

- An open source and open development software project for the analysis and comprehension of genomic data.
- Started in 2001.
- Is based on the R programming language.
- In addition a huge number of analysis methods, Bioconductor contains a large number of meta-data packages.

Working with Bioconductor

Make a pheno data file (a tab-delimited text-file), e.g. "pd-hf.txt".

Name	FileName	Group
JBM942	JBM 942.CEL	AB
JBM812	JBM 812.CEL	AB
⋮	⋮	⋮
JBM439	JBM 439.CEL	AB
JBM494	JBM 494.CEL	AB-HF
JBM496	JBM 496.CEL	AB-HF
⋮	⋮	⋮
JBM937	JBM 937.CEL	AB-HF

Store the file in a folder called e.g. "C:/microarray", where you also have the microarray data (the cel files).

Working with Bioconductor

Set working directory.

```
>setwd("C:/microarray")
```

Install and load bioconductor packages:

```
>source("http://bioconductor.org/biocLite.R")
>biocLite("affy")
>library(affy)
```

Read the pheno data file

$$> pd <- read.AnnotatedDataFrame("pd - hf.txt", header = TRUE, row.names = 2)$$

Read the microarray data (the cel files)

$$> data <- ReadAffy(filenames = rownames(pData(pd)), phenoData = pd, verbose = TRUE)$$

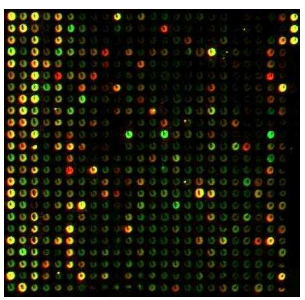
Quality control

Some quality measures:

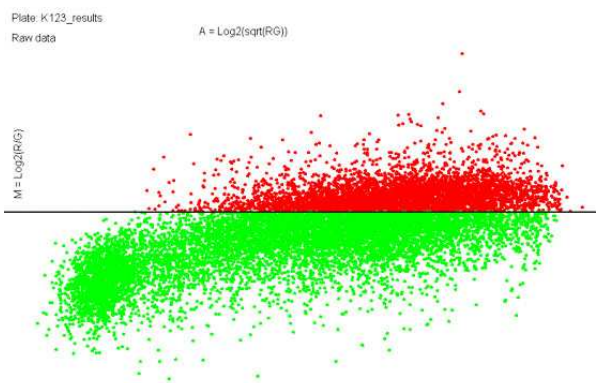
- Correlation between observed and expected values for spike-in probes (with known quantity)
- Variation of replicated control or spike-in probes
- Variation in background signal

Visual inspection:

- Look for spatial effects
- For two-channel (cDNA) arrays look for (unwanted) dependence between differential and overall expression using Ratio Intensity (RI) (also called MA) plots.



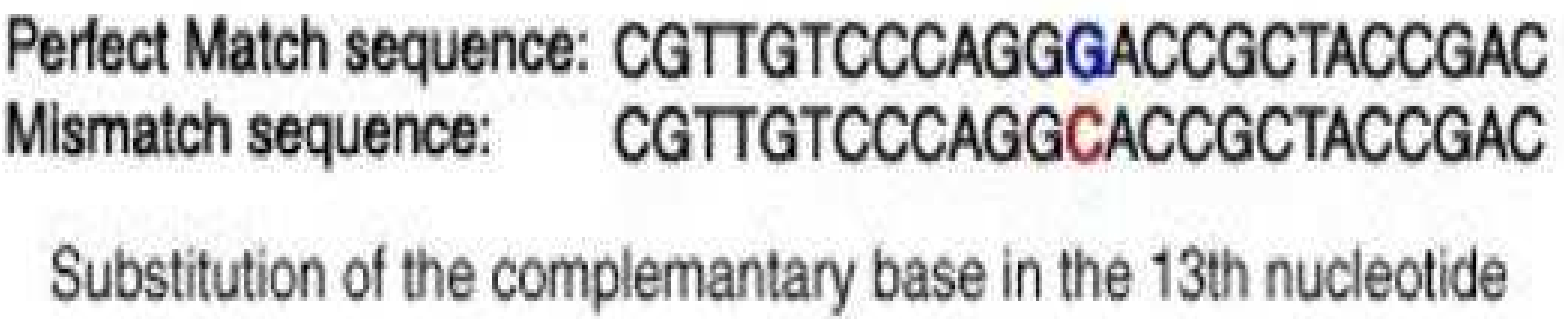
Microarray pseudo image for investigating spatial effects.



Ratio Intensity plot.

Normalization of Affymetrix data

- Early approach: PM-MM
- Problem: MM detects also signal from the PM, i.e. is not only measuring non-specific binding



GCRMA (Irizarry et al, 2004)

- GeneChip Robust Multiarray average
- Uses a stochastic model for background adjustment that uses probe sequence information

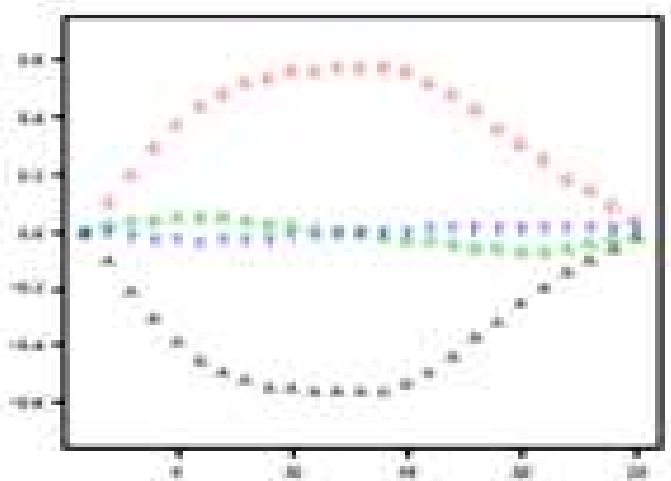


Figure: The effect of the bases A,T, C and G plotted against probe position.

$$PM = O_{PM} + N_{PM} + S$$

$$MM = O_{MM} + N_{MM} + \phi S$$

Here O represents optical noise, N represent non-specific binding and S is a quantity proportional to RNA expression. The parameter $0 < \phi < 1$ accounts for the fact that for some probe-pairs the MM detects signal.

Comparison of some common Affymetrix preprocessing methods

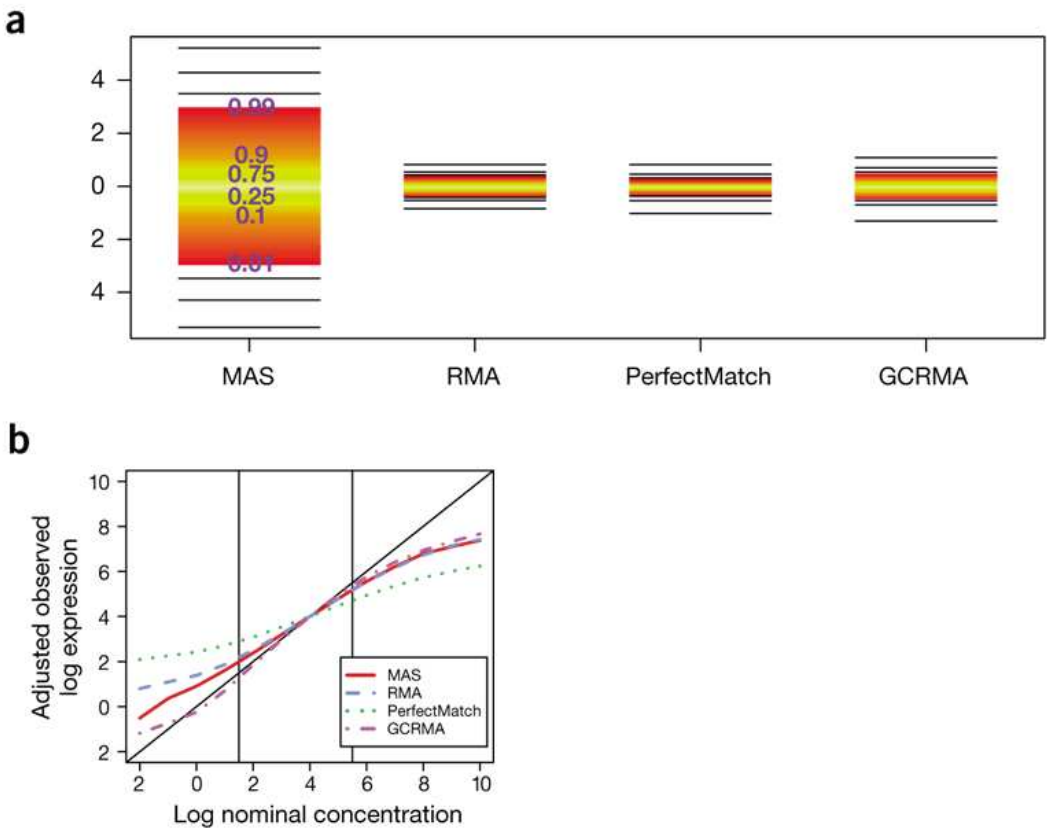


Figure: Figure a shows the boxplots of log fold changes for unaltered non-spiked in probes. The boxplots should be narrow. Figure b shows expected versus observed values for spiked-in probes. The values should be on a straight line.

GCRMA in Bioconductor

```
> biocLite("gcrma")
> library("gcrma")
> eset <- gcrma(data)
```

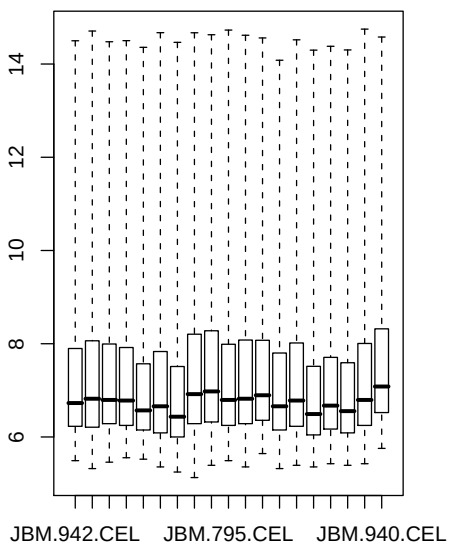
Some check on how the normalization worked.

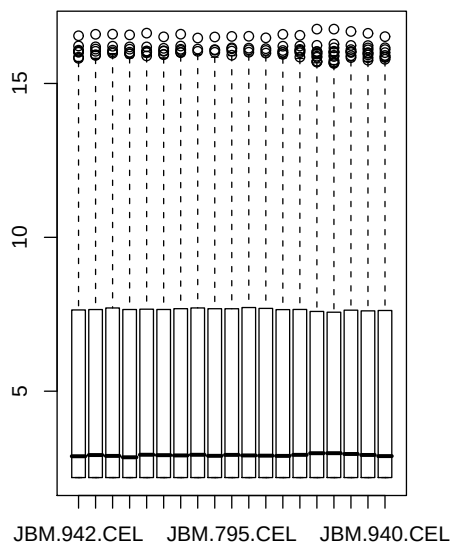
Boxplot of arrays before normalization

```
> boxplot(data)
```

And after normalization

```
> boxplot(data.frame(exprs(eset)))
```





Normalizing cDNA arrays

- There are many normalization methods for cDNA arrays
- They differ mainly in the way spot segmentation (distinguishing foreground from background intensities) is carried out.
- Background adjustment can increase the variability of the processed expression data, and should therefore maybe be avoided (Qin and Kerr, 2004).
- Many methods use lowess smoothing of RI plots (straightening the RI plots)
- Model based normalization using ANOVA has also been proposed (Kerr et al, 2000).

Detecting differential expression

We want to test differential expression between two groups for $i = 1, \ldots, p$ genes (p of order 10000). This can be done using ordinary two sample *t* statistic

$$t_i = \frac{\bar{x}_i - \bar{y}_i}{\widehat{\sigma}_i},$$

where $\widehat{\sigma}_i$ is the (estimated) standard deviation for the difference $\bar{x}_i - \bar{y}_i$.

Variance estimates can be improved by "borrowing strength" across genes in a technique called *variance shrinkage*:

$$z_i = \frac{\bar{x}_i - \bar{y}_i}{\sqrt{B\widehat{\sigma}^2_{\text{all}} + (1-B)\widehat{\sigma}^2_i}}.$$

Bootstrap estimated test statistic

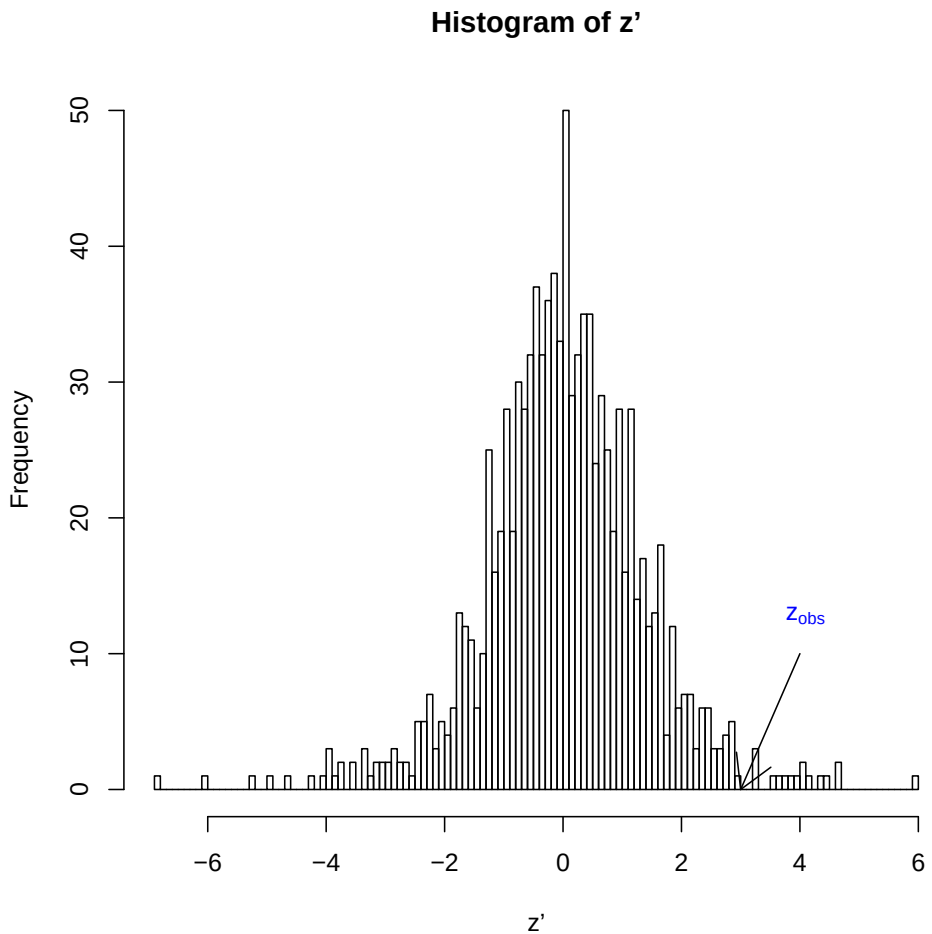
Variance shrinkage is often accompanied by bootstrap estimation of test statistic under H_0
For B bootstrap samples:

- Draw random observations x' and y' from the set observations including both groups $\{x_1, \ldots, x_n, y_1, \ldots, y_n\}$:

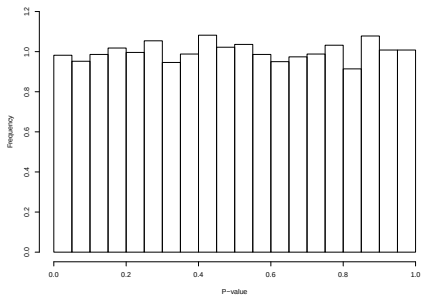
  $$\Downarrow \text{(draw)}$$

  $\{x'_1, \ldots, x'_n\}, \{y'_1, \ldots, y'_n\}$
- Calculate the "null" statistic z' from the $x's$ and the $y's$.

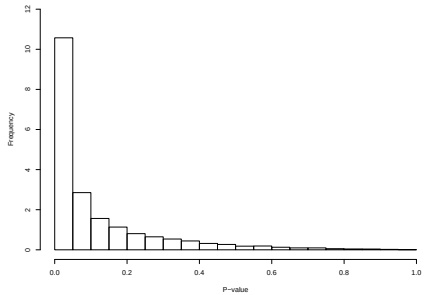
Compare observed test statistic z_{obs} with the B z'-values.



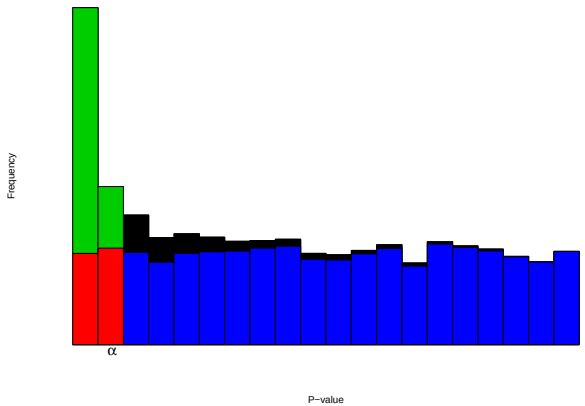
Microarrays and false discovery rate



p values for null genes



p values for non-null genes.



True positives
False positives
True negatives
False negatives

p values for all genes on the microarray

From: Mette Langaas, *Multiple hypothesis testing: theory and applications to genomics. Lecture notes*

SAM (Tusher et al, 2001)

- SAM=Significance Analysis of Microarrays
- Test statistic for gene i:

$$z = \frac{\bar{x}_i - \bar{y}_i}{s_i + s_0},$$

  where s_i is the standard deviation for gene i, and s_0 is common to all genes, chosen to minimize coefficient of variation.
- Is accompanied by bootstrapping of labels to calculate p-values.
- The SAM software also calculates the false discovery rate (FDR) associated with each gene, also called local false discovery rate or q-value.

SAM in Bioconductor

$> biocLite("siggenes")$
$> library(siggenes)$
We want to compare the 5 AB mice with heart failure (AB-HF) to the 5 AB mice without heart failure (AB).
$> sam.HF < -sam(exprs(eset), cl = c(rep(0, 5), rep(1, 5)))$
How many genes are regulated if we don't correct for multiple testing:
$> length(which(sam.HF@p.value < 0.05))$
How many genes are regulated if we do correct for multiple testing:
$> length(which(sam.HF@q.value < 0.05))$

Investigating the list of regulated genes

Install and load the required annotation packages

$> biocLite(c("mouse4302.db"))$

$> library(mouse4302.db)$

Find the gene symbols of the list of the 10 most significant genes

$> unlist(mget(featureNames(eset)[order(sam.HF@q.value)[1:45]], mouse4302SYMBOL))$

Look the genes up in entrez gene:

http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene

To manually examine all significant genes is a tedious work. May be better to look for altered functional categories.

Functional enrichment analysis

- The Gene Ontology annotates genes to *biological processes*, *molecular functions and cellular components*.
- There are many softwares that test categories for enrichment of regulated genes.
- The Bioconductor package TopGO (Alexa et al 2006) is one of the most advanced ones, taking dependencies between the GO categories into account.

Clustering

- Genes are grouped together by degree of correlation.
- Problem: Many thousand genes and many are correlated just by chance.
- Main utility of cluster diagrams is to get an overview of up and down-regulation in the array data.

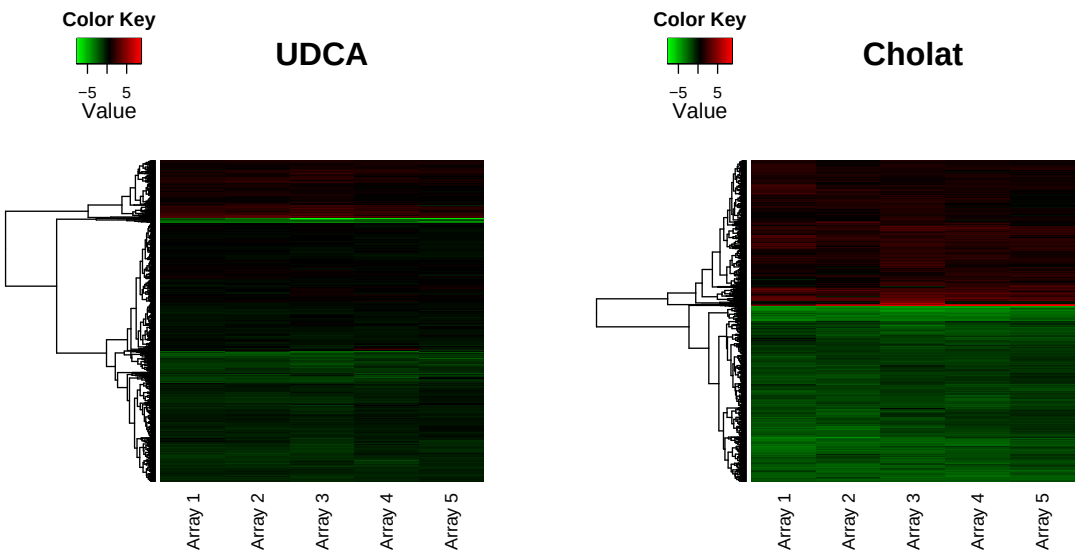
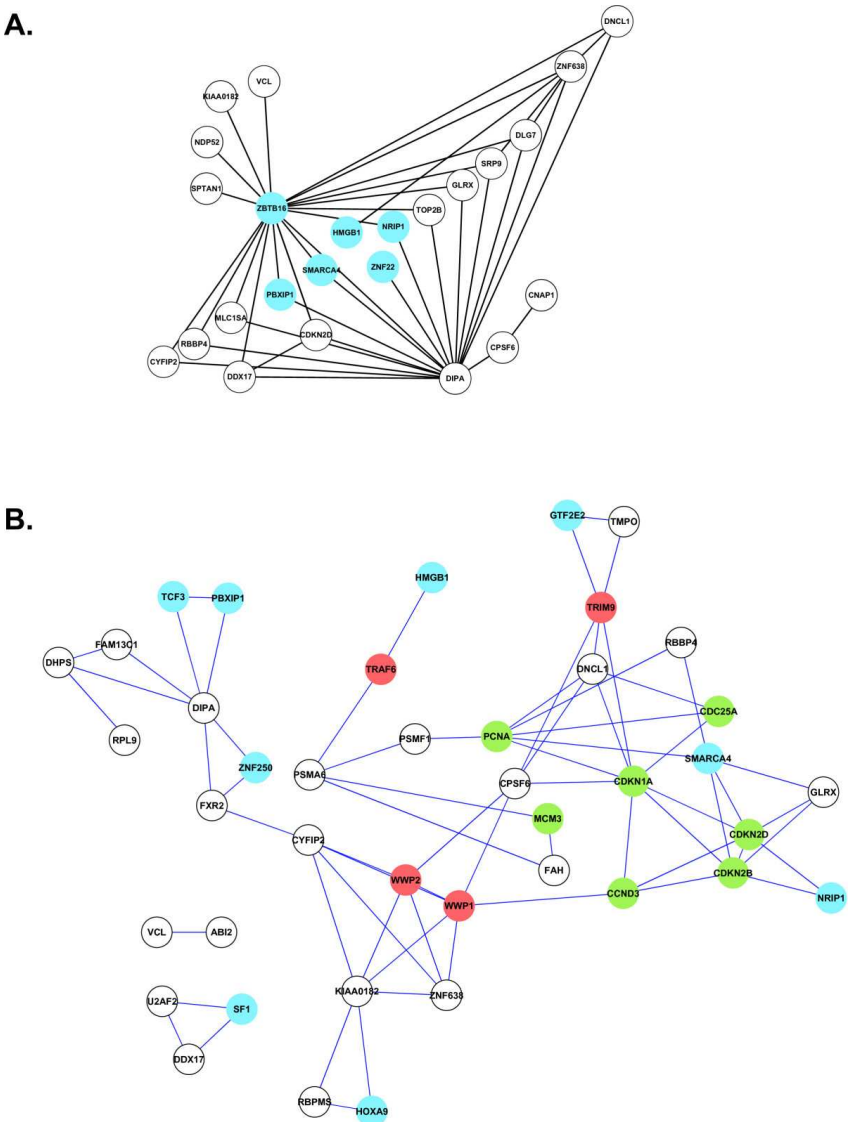


Figure: Cluster diagrams of mice with mild degree (UDCA) and severe degree (Cholate) of schlerosing cholangitis (Nakken et al, 2008).

Network construction of regulated genes

- Network construction from genomic data is difficult.
- Many possible combinations of interactions.
- Network construction could be guided by including external information about interactions.
- Seeded Bayesian Networks (Djebbari and Quackenbush, 2008) guide the network construction by including interactions reported in literature and protein-protein interaction databases.

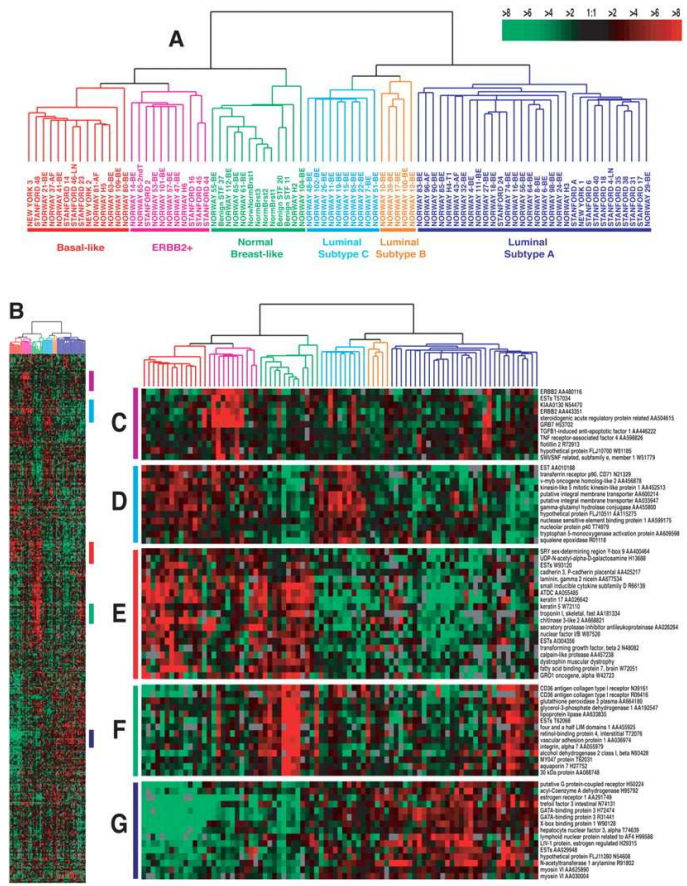
Example of network constructed from the seeded BN method



Networks arising from a Bayesian Network analysis of gene expression data of Golub et al. and rendered in Cytoscape using (A) no prior information and (B) prior network seeds deduced from a combination of the literature and the protein-protein interaction data of Rual et al.

Patient focus analysis: Classification/prediction

- There is a difference between patient and gene focused analysis.
- The list of genes most correlated with phenotype (gene focused analysis) is not necessarily best for classifaction/prediction (patient focused analysis). Reason: Genes are correlated, and groups of genes contain same information.
- Best classification/prediction rule may be obtain using a few genes "representing" the whole group.



Classification/prediction approach

- Instead of looking at each gene's correlation to the phenotype one by one (gene focused analysis), the optimal classification/prediction rule looks at the effect of all genes simultaneously. We then answer the question: what is the effect of gene i when we account for the effect of all other genes.
- Best prediction rule picks out genes with orthogonal (independent) information on the phenotype.
- Methodological problem: How to fit a model with a much larger number (p) of explanatory variables (the genes) than the number of individuals (n). This is called the $p > n$ (p larger than n) problem.
- The solution is to reduce the number of dimensions

Variance-bias trade-off

- All these methods are in fact biased, i.e. underestimating the effect of each gene.
- But they have reduced variance, leading to smaller prediction error.
- Prediction error=bias^2+variance

Variance-bias trade-off

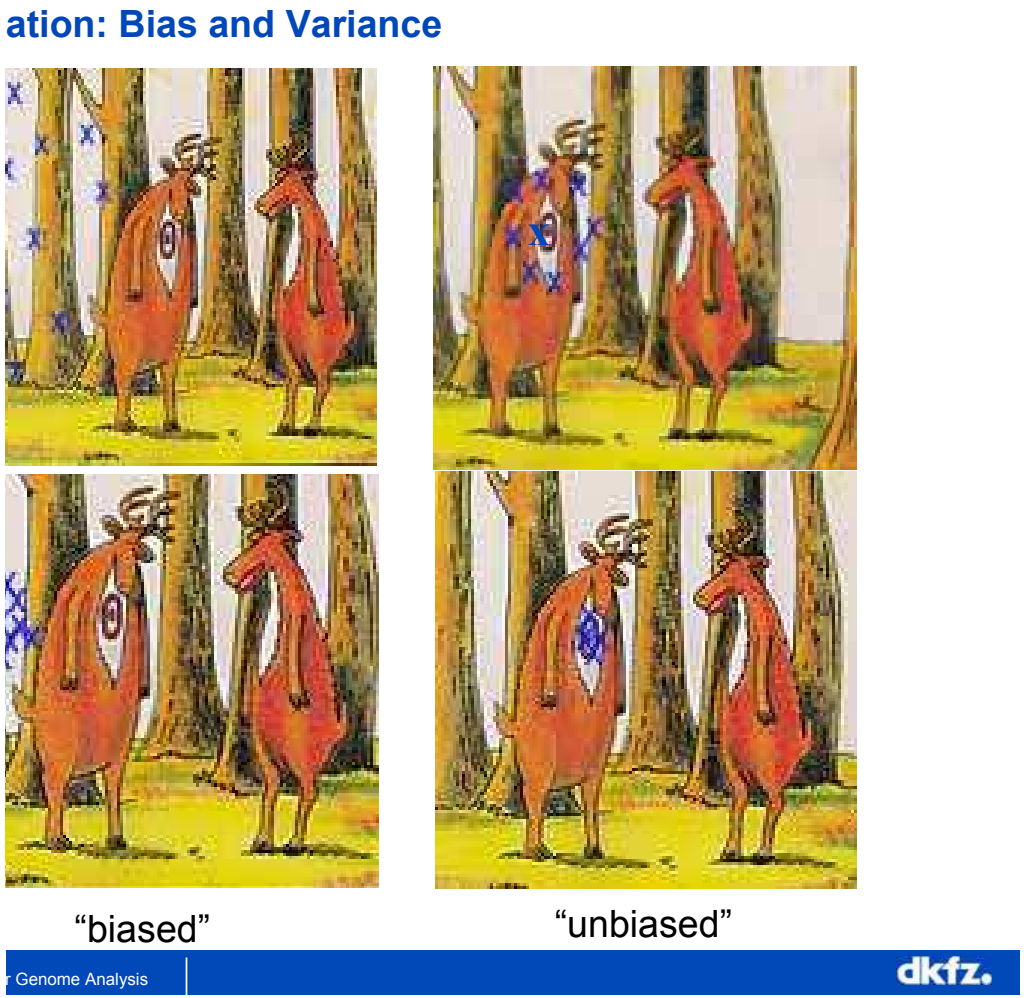


Figure: From Tim Beissbarth, University of Gottingen, Bioconductor course, lecture notes.

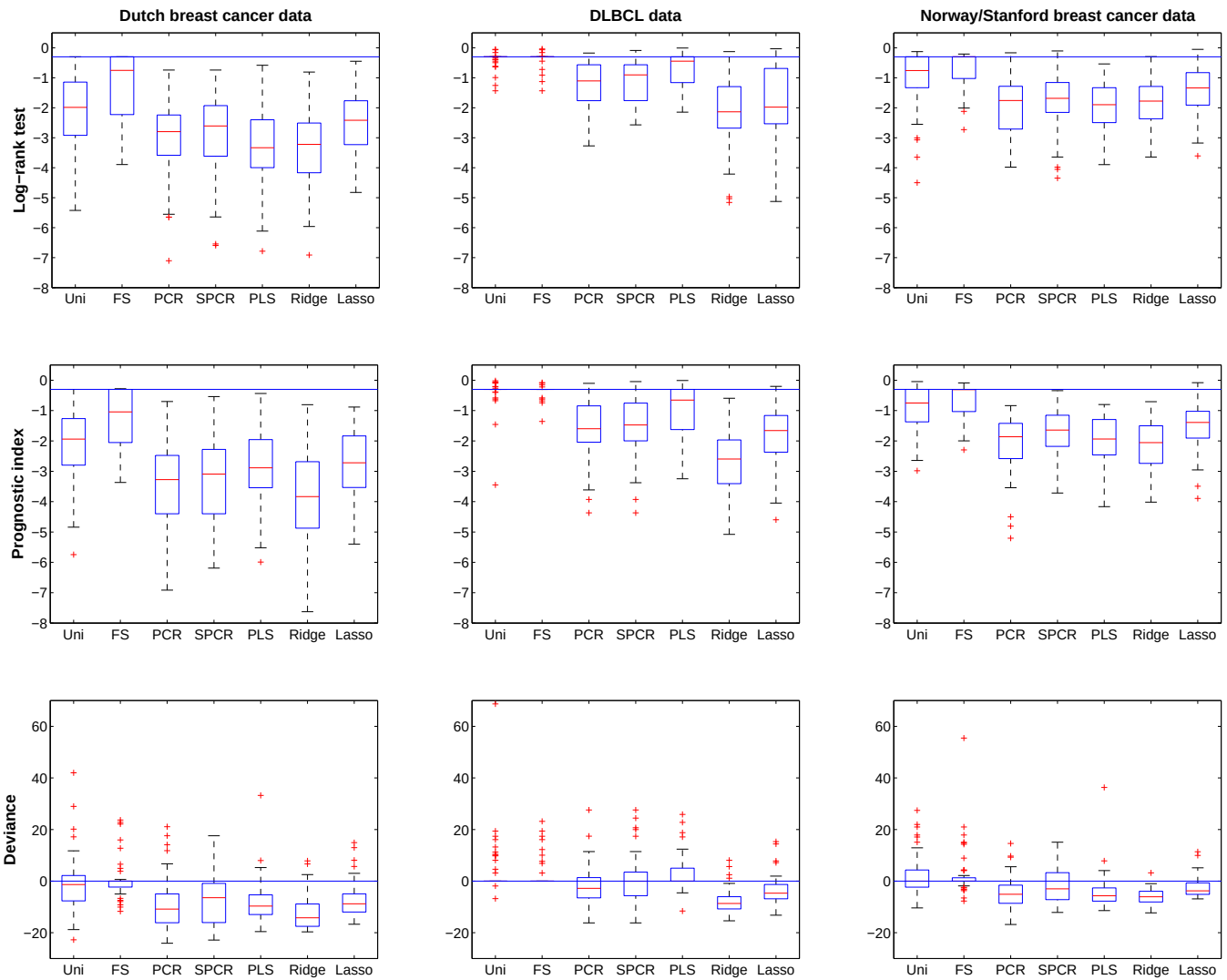
Dealing with survival data

- In addition to the $p > n$ problem, survival or time to event data have the problem of censoring. Event (e.g. death) does not always occur before end of study.
- The Cox model is the most common model dealing with censoring.
- In the Cox model the *hazard rate*, i.e. the instantaneous risk of failure at time t, is modeled by

$$h(t \mid \mathbf{x}) = h_0(t) \exp(\mathbf{x}^T \boldsymbol{\beta}).$$

- Bøvelstad et al (2007) performed a thorough comparison of the seven most common dimension reduction method on three well-known microarray/survival data sets.
- To evaluate prediction performance, models were trained on one part of the data (training set) and evaluated on another part of the data (test set). If this not is done, too large models are favored. "Too large" means that it includes variables randomly correlated to the phenotype in the training data (false positives).

Results of the comparison study of Bøvelstad et al (2007)



References, further reading and software packages

Miscellaneous

Allison et al (2006). Microarray data analysis: from disarray to consolidation and consensus. *Nature review genetics 7*, 55-61. **A nice review on microarray data analysis in general.**

Experimental design

Pawitan et al (2005). False discovery rate, sensitivity and sample size for microarray studies. *Bioinformatics 21*, 3017-3024. **Method for power analysis and sample size calculation for microarray data.** Bioconductor package: OCplus.

Preprocessing

Quin and Kerr (2004). Empirical evaluation of data transformation and ranking statistics for microarray analysis. *Nucleic Acids Research 32*, 5471-5479. **Comparative study of different preprocessing techniques for cDNA data.**

Quackenbush (2002). Microarray data normalization and transformation. *Nature genetics 32*, 496-501. **Review on normalization methods for microarray data.**

References, further reading and Bioconductor packages

Wu et al (2004). A Model-Based Background Adjustment for Oligonucleotide Expression Arrays. *JASA, 99*, 909-917. **Describes the preprocessing method gcrma for Affymetrix data.** Bioconductor package: gcrma.

Kerr et al (2000). Analysis of variance for gene expression microarray data *Journal of Computational Biology 8*, 819-837. **ANOVA for microarray data. Performs preprocessing and testing for differential at the same time.** Bioconductor package: maanova.

Detecting differential expression

Tusher et al (2009). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS 98*, 5116-5121. **Describes the famous SAM method for testing differential expression and calculation of false discovery rate.** Bioconductor package: siggenes.

Smyth (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology 3*, No.1, Article 3. **Describes a nice package for testing differential expression, which also controls for multiple testing.** Bioconductor package: limma.

References, further reading and Bioconductor packages

False Discovery Rate

Efron (2007). Size, power and false discovery rates. *Annals of Statistics, 35*, 1351-1357. **Describes a good method for calculating FDR.** Bioconductor package: locfdr.

Dudoit et al (2003). Multiple Hypothesis Testing in Microarray Experiments. *Statistical Science, 18*, 71-103. **A review article on multiple testing procedures for microarray data.** Bioconductor package: multtest.

Functional enrichment analysis

Werner (2008). Bioinformatics applications for pathway analysis of microarray data. *Current Opinion in Biotechnology, 19*, 50-54. **A review article on bioinformatics tools for identification of altered biological processes and pathways.**

Alexa et al (2006). Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics 22*, 1600-1607. **Describes a functional enrichment analysis method which takes dependencies between GO categories into account.** Bioconductor package: topGO.

References, further reading and Bioconductor packages

Network construction

Djebbari and Quackenbush (2008). Seeded Bayesian Networks: Constructing genetic networks from microarray data. *BMC Systems Biology 2*, Article number 57. **Method for network construction using both co-regulations in microarray data, literature reported interactions and databases of protein-protein interactions.** Software download: http://www.tm4.org/mev.html.

Classification/prediction

Bøvelstad et al (2007). Predicting survival from microarray data – a comparative study. *Bioinformatics 27*, 2080-2087. **A thorough comparison of the seven most common methods for survival prediction from microarray data. Finds that ridge regression has the best performance.** Software download: http://www.med.uio.no/imb/stat/bmms/software/microsurv

Goeman (2009). **An R package for performing lasso and ridge regression using high-dimensional data as covariates. Convenient for classification/prediction purposes.** Software download: http://cran.r-project.org/web/packages/penalized.