

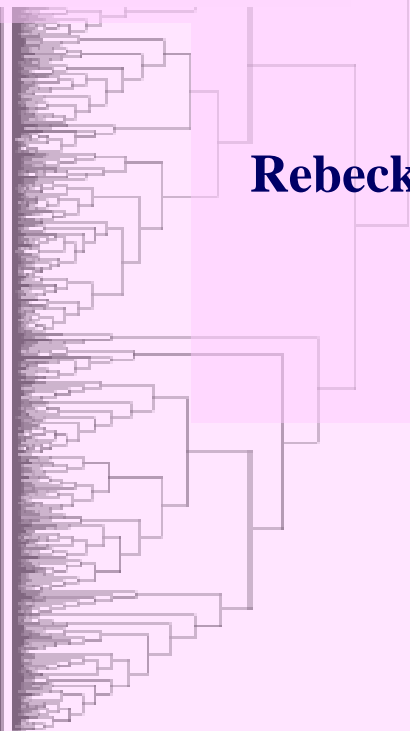
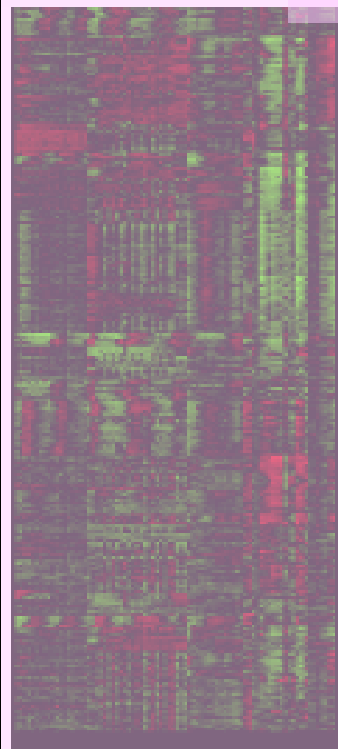
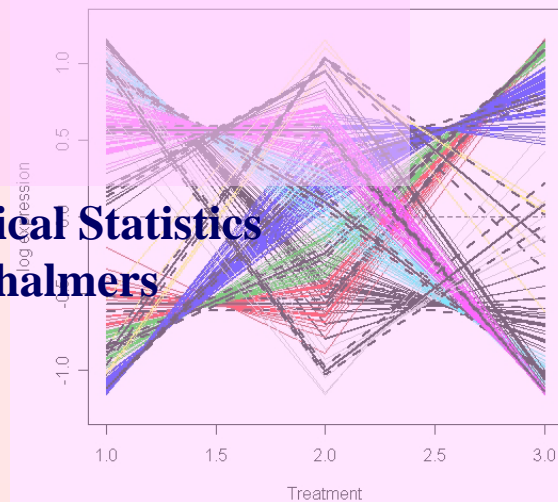
Microarray Data Analysis

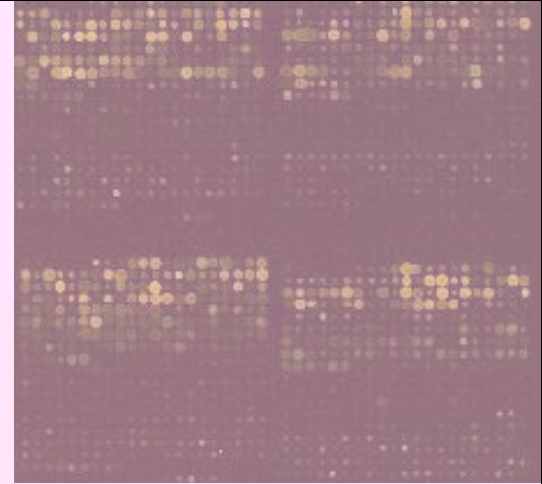
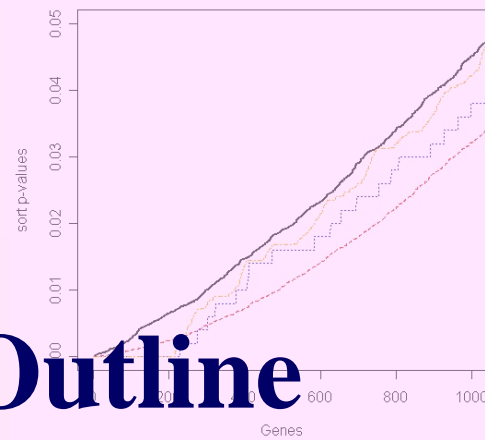
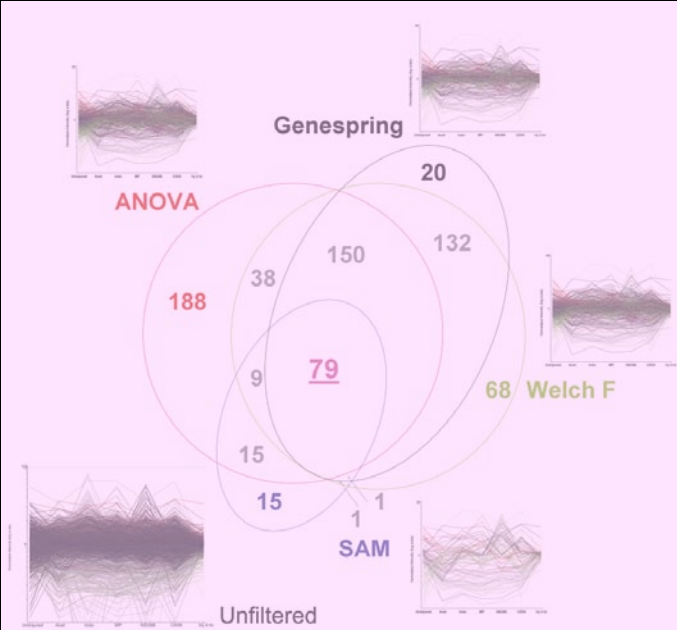
Honolulu, July 14, 2009

Rebecka Jornsten

Department of Statistics
Rutgers University

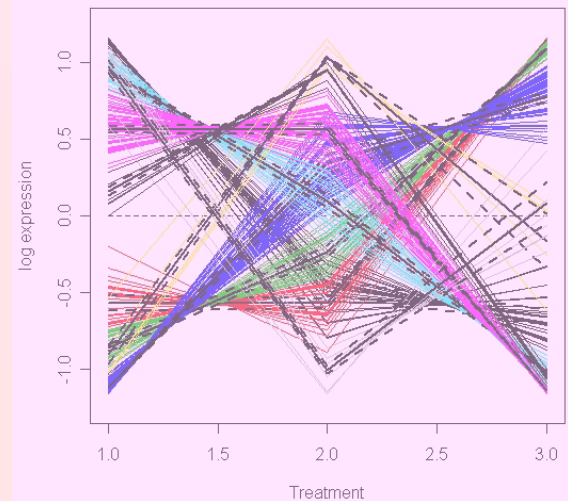
Department of Mathematical Statistics
Gothenburg University/Chalmers





Outline

- Detecting Differentially Expressed Genes.
- Clustering



Gene Expression Data

After normalization

Gene expression data on p genes for n samples

mRNA samples

Genes		sample1	sample2	sample3	sample4	sample5	...
	1	0.46	0.30	0.80	1.51	0.90	...
	2	-0.10	0.49	0.24	0.06	0.46	...
	3	0.15	0.74	0.04	0.10	0.20	...
	4	-0.45	-1.03	-0.79	-0.56	-0.32	...
	5	-0.06	1.06	1.35	1.09	-1.09	...

(log)Gene expression level of gene i in mRNA sample j

Which genes are “interesting”?

Question: what kind of experimental setup do I have?

- **Categorical outcome** (e.g. cancer type).
 - T-test, ANOVA
- **Continuous outcome** (e.g. survival).
 - Regression models, survival models.
- **Time-course**
 - Functional data analysis.

Question: do I also have covariates? (e.g. patient’s age, etc)?

The detection of interesting genes is based on a “**test statistic**” - **T**

- How do we know if an observed value of T is significant?
- We are performing thousands of tests - have to adjust the critical values for *multiple testing*.

Let's say we have R_1 samples from sample type 1, and R_2 samples from sample type 2.

A natural estimate of differential expression of gene g is to take the difference of the two sample means:

$$\hat{\mu}_{2g} - \hat{\mu}_{1g}.$$

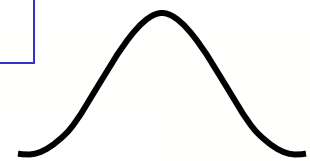
Since errors add up, the associated level of uncertainty with this difference is the standard error

$SE - diff = \sqrt{\frac{s_{1g}^2}{R_1} + \frac{s_{2g}^2}{R_2}}$, where s_{jg}^2 is the sample variance of gene g in sample type j .

Our test-statistic is

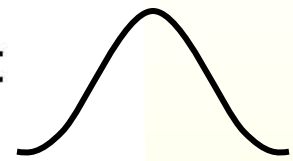
$$T_g = \frac{\hat{\mu}_2 - \hat{\mu}_1}{SE - diff}$$

How large can T_g get just by chance if in fact $\mu_{1g} = \mu_{2g}$?



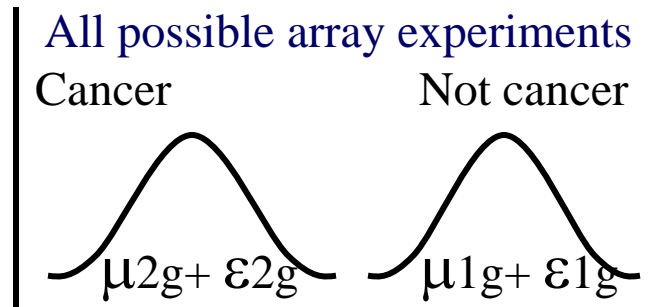
$$\mu_{1g} + \epsilon_{1g}$$

Cancer



$$\mu_{2g} + \epsilon_{2g}$$

Not cancer



We formulate a *null hypothesis*

$$H_0 : \mu_{1g} = \mu_{2g}$$

Under the null, the t-statistic T_g is t-distributed with degrees of freedom (size of tails) depending on R_1, R_2 as well as the individual variances of gene g 's expression in each sample type.

We reject the null if $|T_g|$ exceeds a critical cutoff
(based on the tails of the t-distribution).

Note: if the sample sizes R_1, R_2 are small - lots of parameters to estimate here (μ_1, μ_2 and the variances). Cutting costs?

If we are willing to assume that the variances of expression are equal in both sample types

$\sigma_{1g} = \sigma_{2g}$, we can *pool* the estimates.

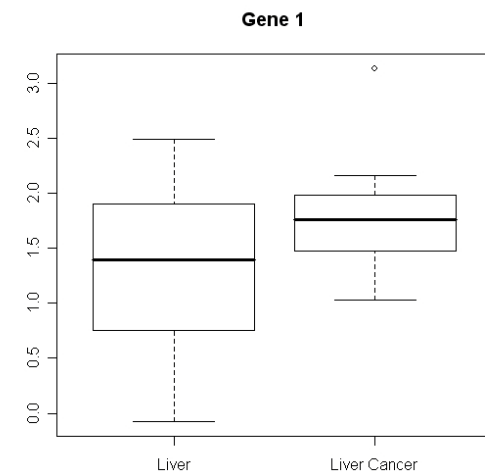
We get a pooled estimate of the sample variance as

$$s_{pg}^2 = \frac{(R_1 - 1)s_{1g}^2 + (R_2 - 1)s_{2g}^2}{R_1 + R_2 - 2}.$$

We use this pooled estimate in the calculation of T_g .

Under the null, the t-statistic T_g is t-distributed with degrees of freedom $R_1 + R_2 - 2$.

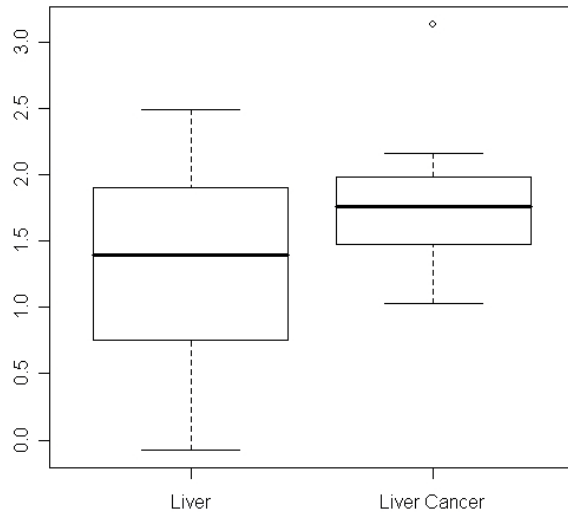
NOTE: if the assumption is incorrect, this test is invalid!



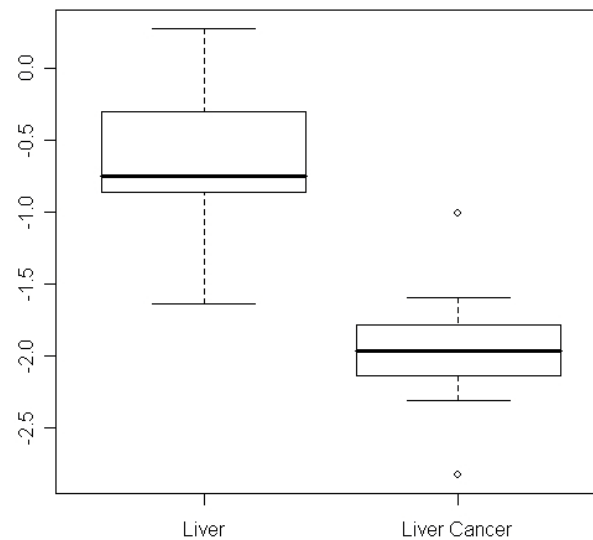
Example: 20 liver, 20 liver cancer samples

To pool or not to pool?

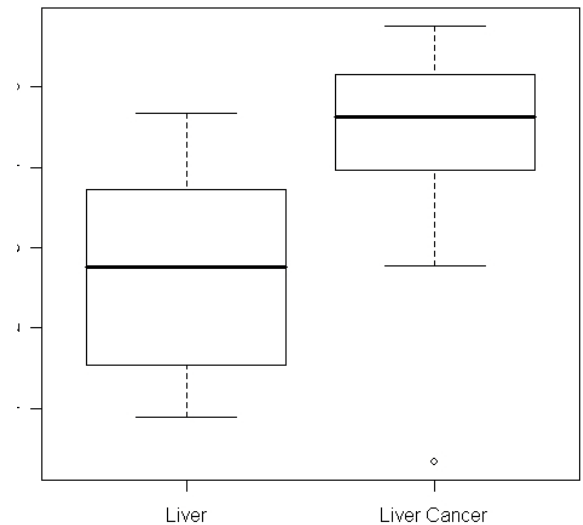
Gene 1



Gene 10



Gene 7

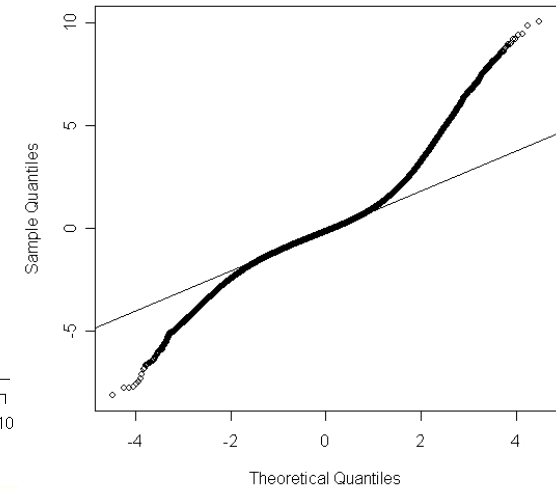
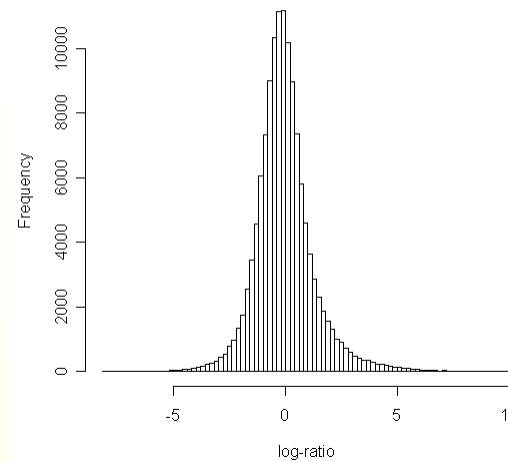


Significance analysis

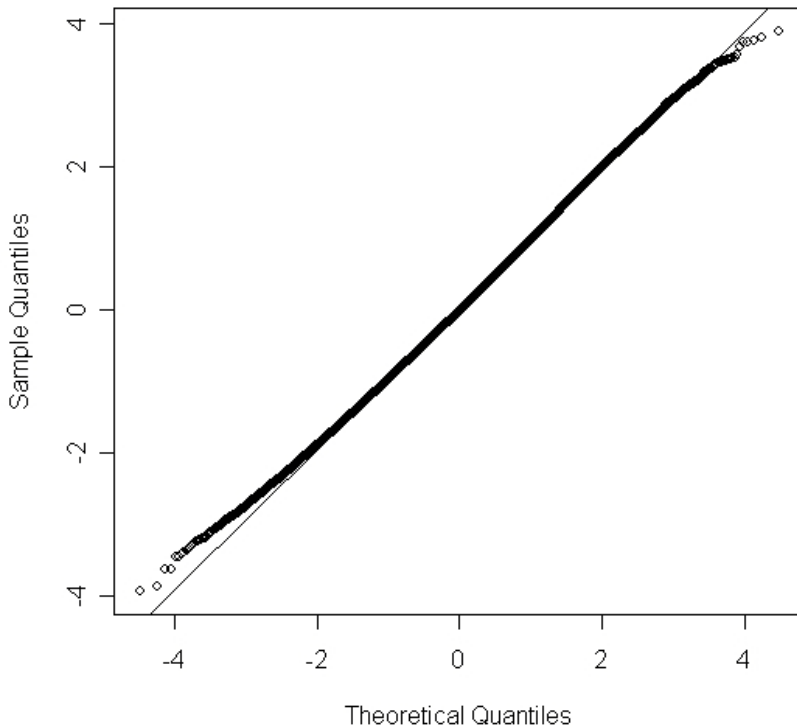
Assumptions of the t-test?

- Independent sampling
- Normal errors

Histogram of liver samples and QQ-plot



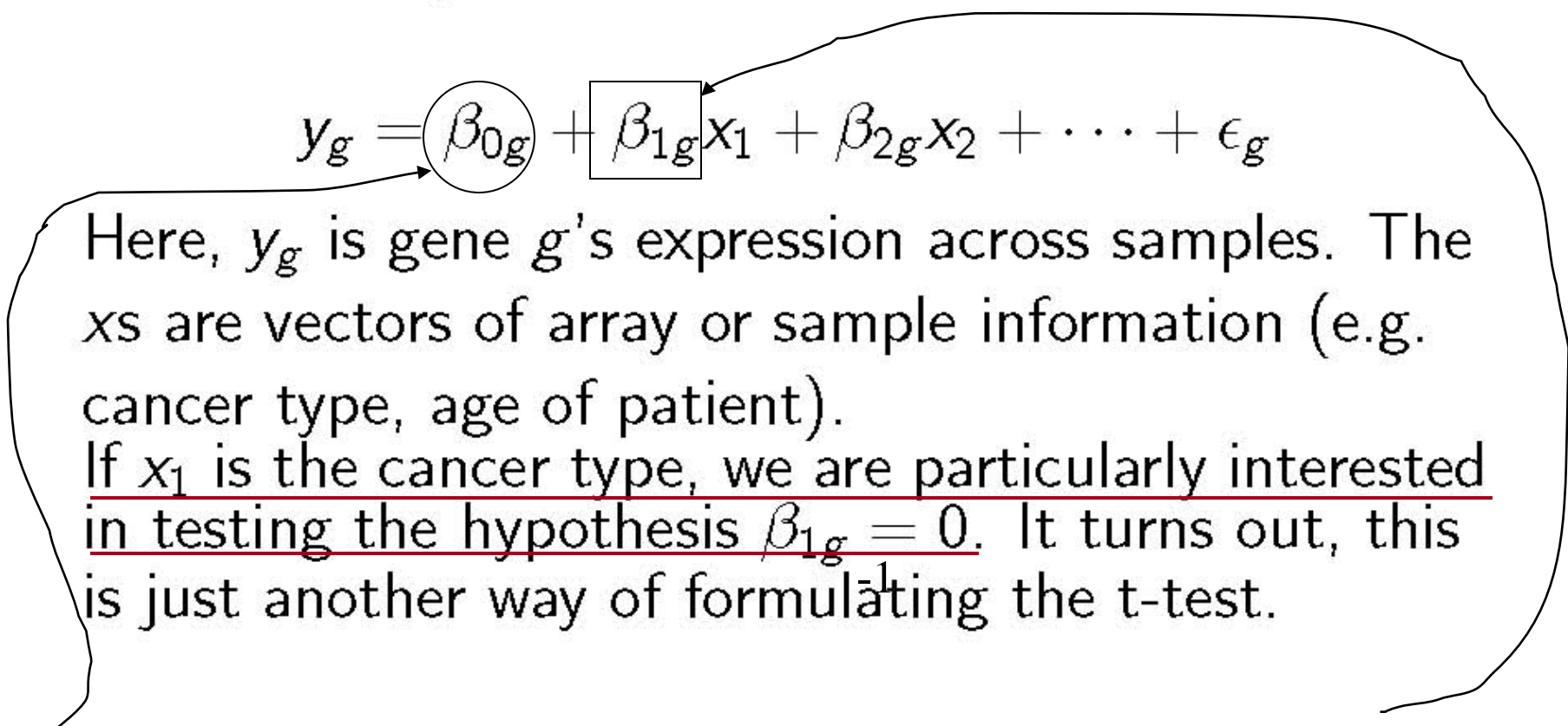
Normal Q-Q Plot



QQ-plot after standardization of the liver data: subtract the mean, divide by the standard deviation, for each gene

Linear Models (ANOVA) –another view of the t-test

We can often write our gene expression data in terms of a *regression model*.

$$y_g = \beta_{0g} + \beta_{1g}x_1 + \beta_{2g}x_2 + \cdots + \epsilon_g$$


Here, y_g is gene g 's expression across samples. The x s are vectors of array or sample information (e.g. cancer type, age of patient).

If x_1 is the cancer type, we are particularly interested in testing the hypothesis $\beta_{1g} = 0$. It turns out, this is just another way of formulating the t-test.

baseline expression

Impact on expression due
to covariate x_1

Linear Models – another view of the t-test

Design matrix $\tilde{X} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \end{pmatrix}$, $\tilde{X}'\tilde{X} = \begin{pmatrix} n & n_1 \\ n_1 & n_2 \end{pmatrix}$

cancer {

not cancer {

Cancer type covariate x1

$$y_g = \beta_{0g} + \beta_{1g}x_1 + \epsilon_g = \tilde{X}\beta_g + \tilde{\epsilon}$$

Then, $\hat{\beta}_g = (X'X)^{-1}X'y_g$ is the solution to the least-squares criterion $\sum_{samples} (y_{gi} - \beta_{0g} - \beta_{1g}x_i)^2$.

With the above X , one can show that $\hat{\beta}_{0g} = \hat{\mu}_{1g}$ and $\hat{\beta}_{1g} = \hat{\mu}_{2g} - \hat{\mu}_{1g}$.

That is, a t-test is the same thing as testing that the second coefficient in this linear regression model is 0!

Why did we bother rephrasing the t-test as a regression problem?

Well, we know a lot about estimation in regression.

If $y_g = X\beta_g + \epsilon_g$, where $\epsilon_g \sim N(0, W_g\sigma_g^2)$, then $\hat{\beta}_g = (X'X)^{-1}X'y_g$ is an **unbiased** estimate of the true β_g .

Furthermore, the variance of this estimate can be expressed as

$$V(\hat{\beta}_g) = \sigma_g^2(X'X)^{-1}(X'W_gX)(X'X)^{-1} = \sigma_g^2 V_g$$

To test the a null hypothesis $\beta_{gk} = 0$ (kth coefficient) we use the test-statistic

$$T_g = \frac{\hat{\beta}_{gk}}{\sqrt{\hat{\sigma}_g^2 v_{gk}}}$$

Here, v_{gk} is the kth diagonal element of matrix V_g . Under the null, T_g is t-distributed with degrees of freedom $n-K$, where K is the number of parameters in the regression model.

(Note, we can choose to pool or not to pool by incorporating different weights W_g in the error distribution $\epsilon_g \sim N(0, W_g \sigma_g^2)$.)

Why did we bother reformulating the testing in terms of linear regression?

A) Easier to extend to more complicated models.

B) Known fixes of certain 'problems' in linear regression models.

Example of A) Multiple-factor experiments involving different tissues, different cell-lines, time, etc.

Example of B) When the sample size is small, s_g^2 can be a poor estimate of σ_g^2 and this can have an adverse effect on testing.

More on A) later. What about B)?

Let's be Bayesian about it... Let's try to *regularize* the estimate of σ_g^2 .

How can we obtain better estimates of σ_g^2 ?
Well, what if we pool strength across genes g , assuming they have similar variance?

We can think of this as putting a *prior* on σ_g^2 : example: $\frac{1}{\sigma_g^2} \sim \frac{1}{d_0 s_0^2} \chi_{d_0}^2$

The posterior estimate $\tilde{s}_g^2 = \frac{d_0 s_0^2 + d_g s_g^2}{d_0 + d_g}$ is a regularized estimate of σ_g^2 .

The bigger d_0 is, the more we 'shrink' s_g^2 toward the common variance estimate s_0^2 .

We obtain a new t-statistic

$$\tilde{t}_{gj} = \frac{\hat{\beta}_{gj}}{\sqrt{\tilde{s}_g^2 v_{gj}}} \sim t_{d_0 + d_g}$$

Examples of regularized t-tests are
Baldi and Long (where we regularize toward a common variance defined by the the k nearest genes to gene g), * Speed and Lonnstedt (the B-statistic), * Smyth (LIMMA package in R), * Tibshirani et al (SAM).

Common to all these methods: we use the data as a whole to regularize our individual gene estimates. This idea of letting the prior be guided by data is called *empirical Bayes*

Significance analysis	Permutation tests					
	Group	1	1	1	2	2
	Data	0.46	0.30	0.80	1.51	0.90
	...					
	Permute the labels!					
	1	2	2	1	1	...
	2	1	2	1	2	...
	2	1	1	2	1	...
	1	1	2	2	2	...

What do we do if we can't assume the errors are normally distributed?

By permuting the sample labels we make the null is true! – since labels are randomly assigned the means of groups 1 and 2 have to be equal.

If we compute the t-statistics with the permuted labels we obtain the sampling distribution of the t under the null, but we don't have to assume normality of errors!

P-value=proportion permutations with $|t(\text{permuted})| > |t(\text{observed})|$

Reject the null if this P-value is less than some cut-off, say 5%

Multiple testing

For each gene we obtain test-statistic T_g . We compare each T_g to the critical cut-off t^* , where t^* corresponds to the $1-\alpha/2$ quantile of the appropriate t-distribution.

Each test has a probability $1-\alpha$ of leading to a false rejection (T_g exceeds t^* even when the null is true).

We pick α small (t^* large) to keep the likelihood of a false rejection under control.

We're performing many tests (10000s of genes), and each test has a small probability α of a false rejection.... What will this mean for the data set as a whole?

If α is 0.05, **the probability that we make at least one false rejection** is ~ 0.05 for 1 test, ~ 0.40 for 10 tests
 ~ 0.994 for 100 tests, and ~ 1 for >1000 tests. Hm?

Significance analysis

Multiple testing

	Not rejected	Rejected	
True Nulls	U	V	M0
True Alternatives	T	S	M1
	M-R	R	M

Family-wise error rate $\text{FWER} = \text{Prob}(V \geq 1)$, at least one false rejection

False discovery rate $\text{FDR} = E(V/R \mid R > 0)$, **proportion** of rejections that are false.

Family-wise error rate $\text{FWER} = \text{Prob}(V \geq 1)$, at least one false rejection

False discovery rate $\text{FDR} = E(V/R \mid R > 0)$, proportion of rejections that are false.

Do we care more about FWER or FDR?

- Well, if any false rejections are unacceptable we go with FWER
- If we don't care about a few false rejection, provided that they make up a small proportion of the total number of rejections, we go with FDR.

To control FWER we can use the classical Bonferroni correction:

Adjust the p-value for gene g by the number of tests performed:

$$P\text{-adjusted}(g) = P(g) * M$$

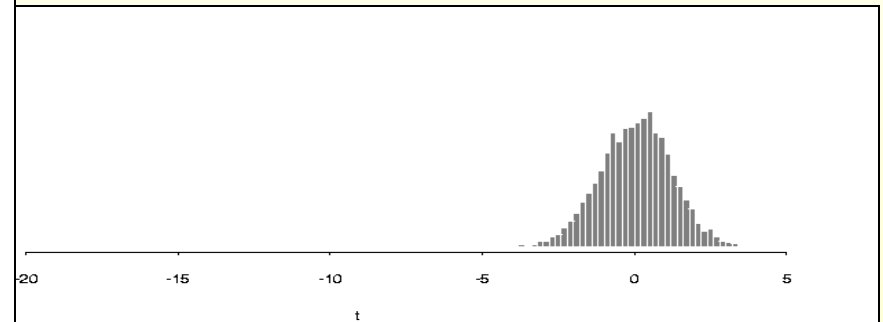
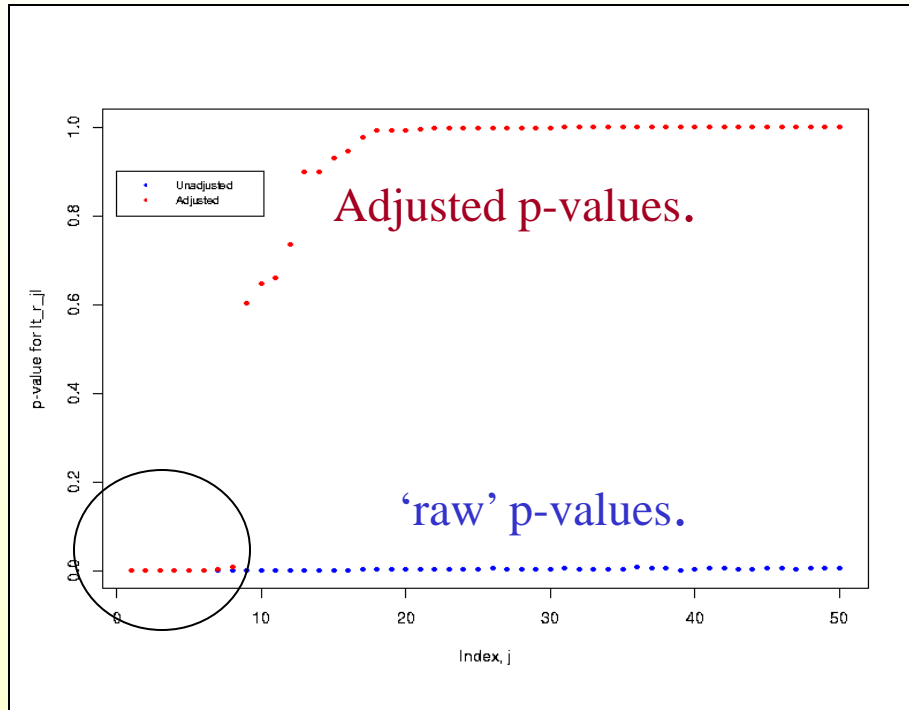
To control FDR we can use the Benjamini-Hochberg correction:

Adjust the p-value for gene g by a factor that depends on its rank-order. If gene g has the k-th smallest p-value:

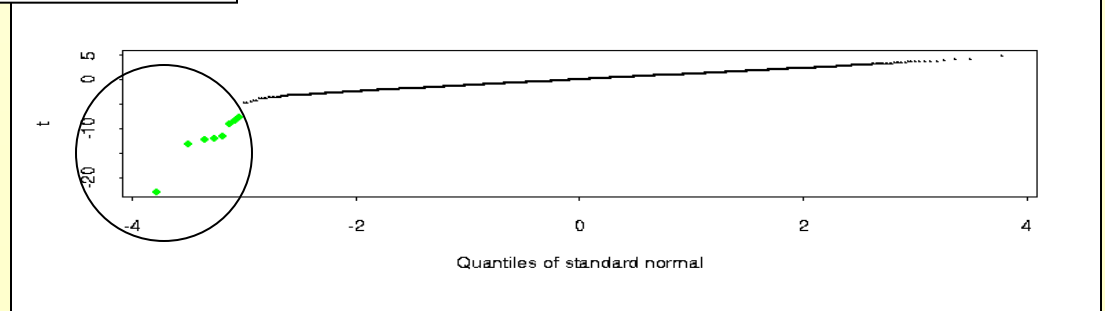
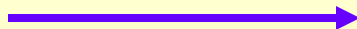
$$P\text{-adjusted}(g) = P(g) * M/k$$

(where we make sure adjusted p-values retain the same rank-order and don't exceed 1)

Significance analysis Example: 8×2 replicate experiment.
(courtesy: M. Callows, LBNL, technical report UCB statistics)
8 control samples, 8 treatment samples. Compute the t-statistic.
Use permutations of the control/treatment labels to get adjusted p-values. p-value $< .01$ significant - found 8 genes.



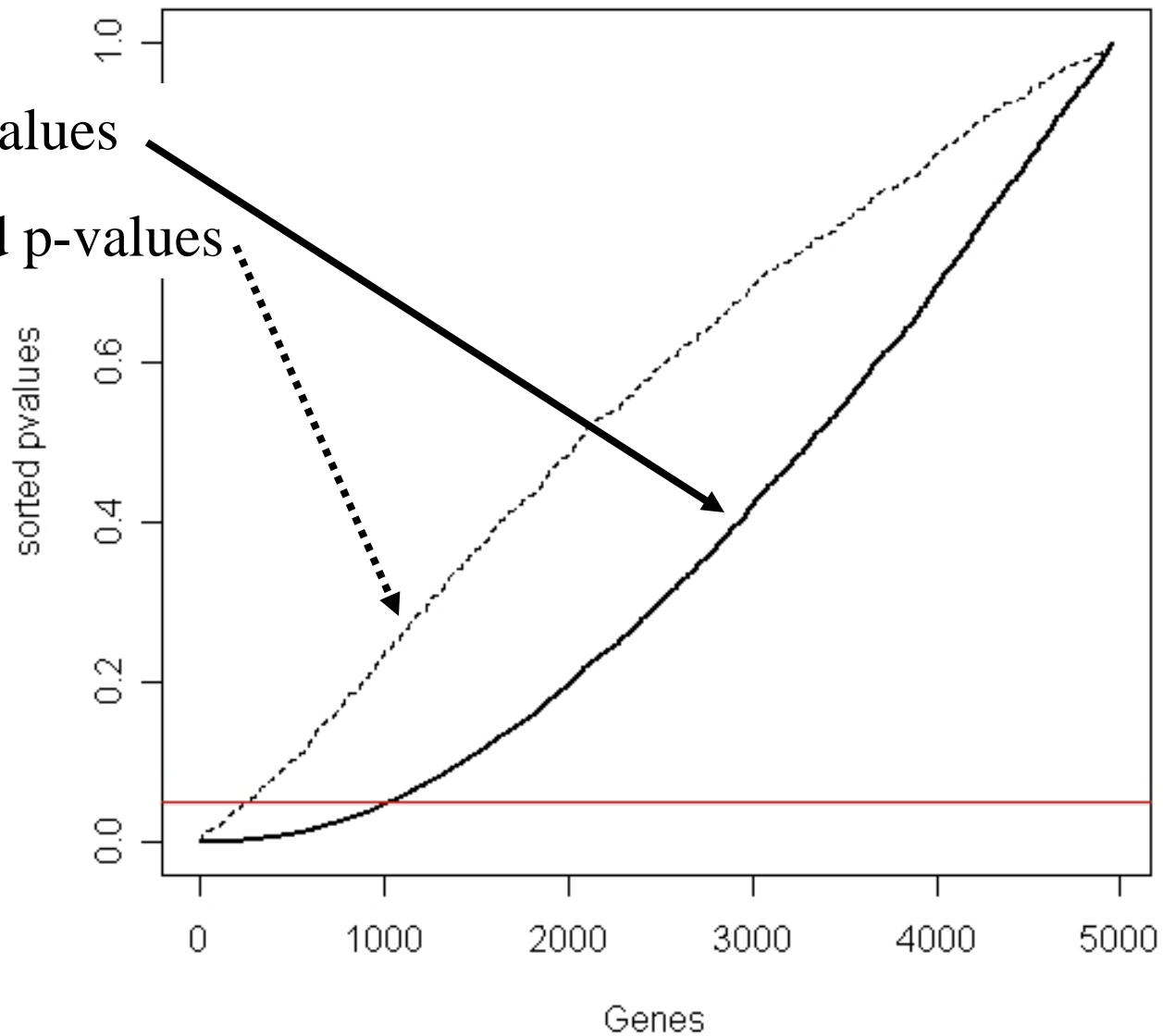
Histogram, QQ-plot of t-statistics.



Another example from the liver cancer data

Raw p-values

Adjusted p-values



Significance analysis

Examples from liver cancer data: number of genes declared significant at the 5% level.

Test		t-test	ebayes	permutation
Sample size 20	Raw	2638	2647	2656
	BF	420	462	1001
	BH	2164	2206	2198
		2124	2131	2103
Sample size 3	Raw	298	488	375
	BF	0	0	216
	BH	0	66	228

What if you have more than one experimental factor of interest?

Example: two-factor experiments with time-course and cell-line.

Model for the data:

$$y_g = \alpha_{cell-line} + \beta_{time} + \gamma_{cell/time} + \epsilon_g.$$

Note, models like these can also be reformulated as a linear regression model with a design matrix X .

Now you can test for cell-line effects (α), time-effects (β), and cell-line/time interactions (γ).

More complex experiments and methods.

For more complex experiments (many covariates, time course experiments) there are many new statistical methods out there.

Most are some kind of variant of regression, with a twist on regularization.

See e.g. J. Storey (UWash) for time course modeling, M. Yuan et al (G.Tech) for detection of differential expression patterns/classes, M. West (Duke) for Bayesian methods for gene selection, Hongzhe Li (UPenn) for incorporating pathway information into testing, and many many more...

New methods vs standard ones....

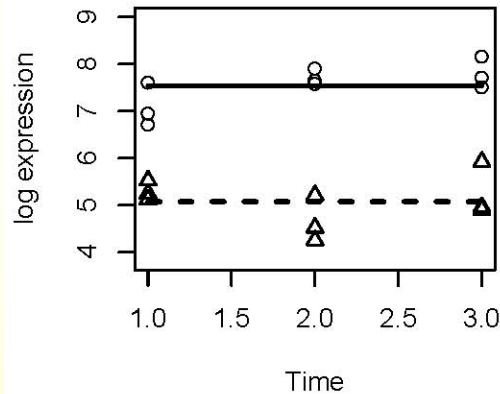
PRO: More specific answers (differential expression patterns).

CON: Often not as easy to use (R-code or R-packages).

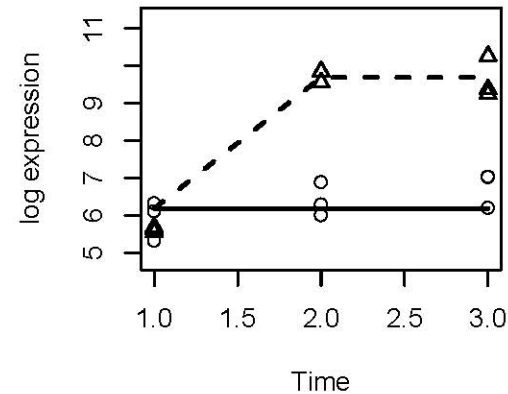
Significance analysis

More advanced methods - not just declaring a gene differentially expressed, but say in what sense... (Muir et al, West et al, Yuan et al, Jornsten,...)

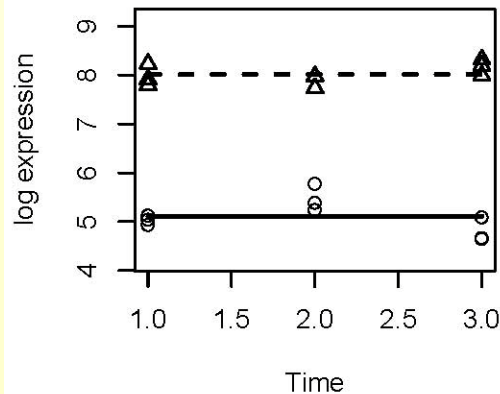
Class 2



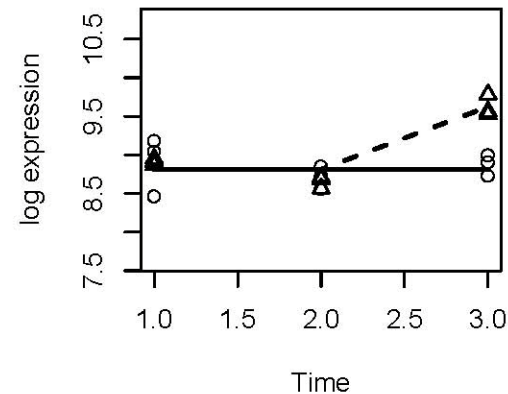
Class 10



Class 3



Class 5



Biological knowledge = prior or validation?

Recent methods research – incorporating biological knowledge into the analysis.

- 2 paths: (1) using biological knowledge for validation
- (2) using biological knowledge as prior information

Example of (1): looking at the over(under)-representation of certain GO terms in the list of differentially expressed genes.

	<i>Example 1</i>		<i>Example 2</i>	
	non-DE	DE	non-DE	DE
Non-GO1	8750	2225	8750	1025
GO1	1250	325	1250	1300

Biological knowledge = prior or validation?

	<i>Example 1</i>		<i>Example 2</i>	
	non-DE	DE	non-DE	DE
Non-GO1	8750	2225	8750	1025
GO1	1250	325	1250	1200

Assessing significance of a table entry like this one: Fisher’s test or similar tests of independence. Complication: GO terms are linked and genes are correlated!

Fix: resampling based methods – array permutation. BUT – the nested structure of GO terms etc makes this a complicated problem.

Alternative: Resample gene lists at random.

What is the null? More GO-terms than the average gene list, or independence?

Biological knowledge = prior or validation?

(2) using biological knowledge as prior information

Lots of work in this area.

Examples:

- *Rocke – combine test statistics from (pre-defined) groups of genes (e.g. pathways). Assess significance of a combined test statistic via permutation tests or random sampling of gene sets.
- *Hongzhe Li – let gene-specific regression models depend on a “state” variable (DE or nonDE), and the state-variable depend on the state of “neighboring” genes (genes in the same pathway being DE or nonDE etc).

A research area that is very active! Other prior info: literature, PPI, meta-analysis...

Classification – feature selection.

An alternative analysis view from significance analysis.

Instead of looking at genes one at a time, view the problem as a diagnosis or prediction problem... Which genes are “markers” of a disease?

This is a non-standard model selection problem in statistics because there are many more genes than patients or samples ($p > n$).

Variable selection: wrappers, filters, ... Problem: genes are correlated.

Recent methods research focus in statistics – regularized regression with penalties on grouped variables (e.g. group-lasso).

Methods allow for the discovery of groups of genes (e.g. pathways) that are predictive of disease.

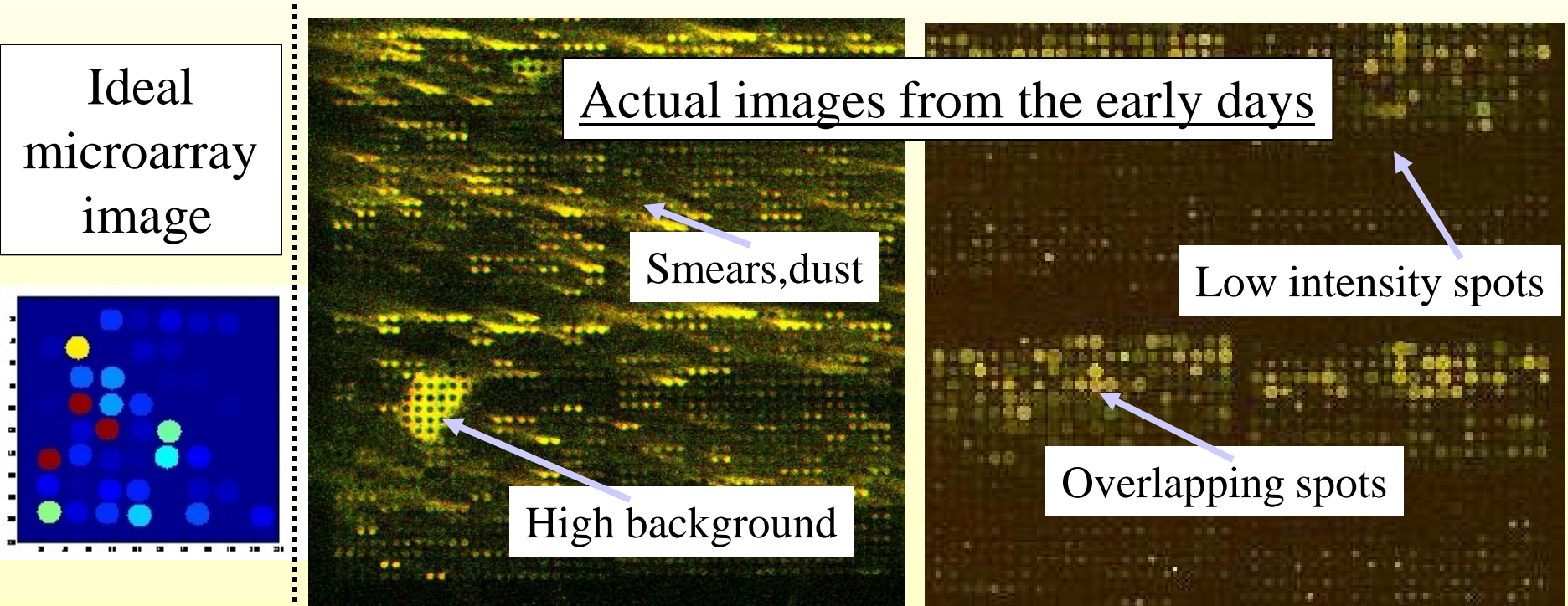
(deMol et al, West et al)

Imputation

Should we ignore the missing values? Impute them?

What is the effect on the subsequent analysis?

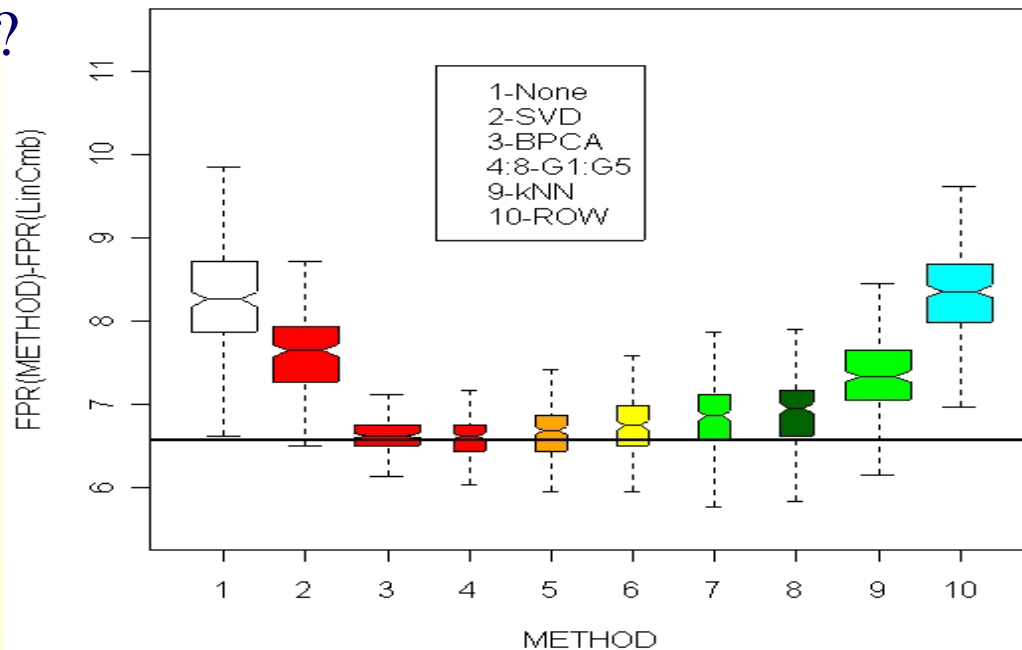
First ask yourself ‘why do I have missing values?’ Are they ‘missing’ or do they carry information, e.g. saturation, below detection? If the latter, you should NOT impute but model the missingness directly (or adjust calibration of scanner).



Imputation

“ROWimpute”, K-nearest-neighbors (kNN), Transform based methods (SVD, BPCA), and many, many more

So....how do we decide **which method to use**? Is there a ‘best method’ for imputation?



With few values are missing (~1% missing values), forgoing imputation is not a bad strategy. If more values are missing, imputing can really help, but only if you do it right...

Some methods are clearly always a bad choice: kNN and ROWimpute.

Clustering = exploratory analysis

We wish to group data units (genes or samples) that are similar, or partition the data set into dissimilar groups.

- *decide on what you mean by similarity (e.g. correlated, close in an average sense)

- *choose an algorithm that uses this similarity metric to group the data.

These choices are subjective.

We cannot easily say that one clustering outcome is “better” than another – *different clustering methods focus on different aspects of the data*

Using the metric to generate clusters.

To generate clusters based on a metric we have to use a **rule for assigning units to the same/different groups**.

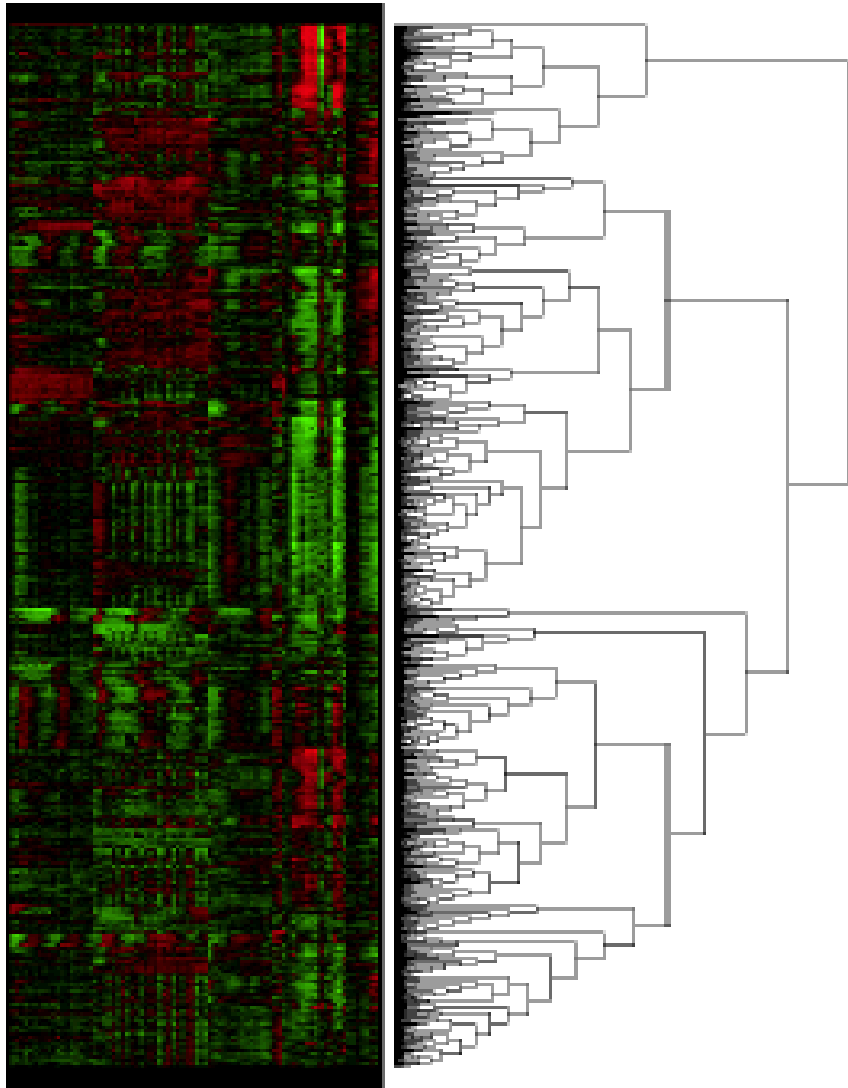
A very popular method is hierarchical clustering.

Start with all N units as individual clusters

- Join the two units, or groups of units, that are the most similar
- Here, the similarity of groups of units is determined via the “linkage” function, i.e. how to combine the *dissimilarities between the group members* into one *dissimilarity between the groups* (e.g. *nearest or farthest neighbor, or average neighbor distance*).
- Repeat until only one cluster remains

Clustering

Example: two-way clustering (both samples and genes)



Dendograms: the length of the branches depicted are proportional to the dissimilarity between the daughter-branches. Long branches indicate good separation.

Caution: Hierarchical clustering is highly **non-robust**: small changes to the data can alter the look of the dendrogram substantially.

Both the choices of linkage and the dissimilarity metric play a role

Partitioning methods

Divide the data into K groups such that an objective function is optimized. Examples:

- **kmeans** – partition in order to minimize the distance from each unit to the closest cluster representative=mean of the cluster.
- **Kmedian, PAM** – same as above, but using the median (multivariate) as the cluster representative
- And many more....

Depending on the objective function these methods are more or less robust. These three methods tend to produce clusters of equal size and shape.

- **kmeans** –simple, intuitive and fast. Non-robust because the mean is used as the cluster representative
- **kmedian,PAM** – more robust than kmeans, also fast

Model-based clustering methods.

- If we are willing to assume that gene expression is approximately normally distributed, and expression patterns come from a number of more or less distinct shapes....

Clustering becomes a regular statistical modeling problem!

- We assume that there are **K** types of expression patterns (or clusters) in the data.
- Within **each cluster** we assume that the gene have expression pattern μ_k and the cluster has shape Σ_k .
- We assume that a **proportion** π_k of the genes belong to cluster **k**.
- We need to estimate the parameters (μ_k, Σ_k, π_k)

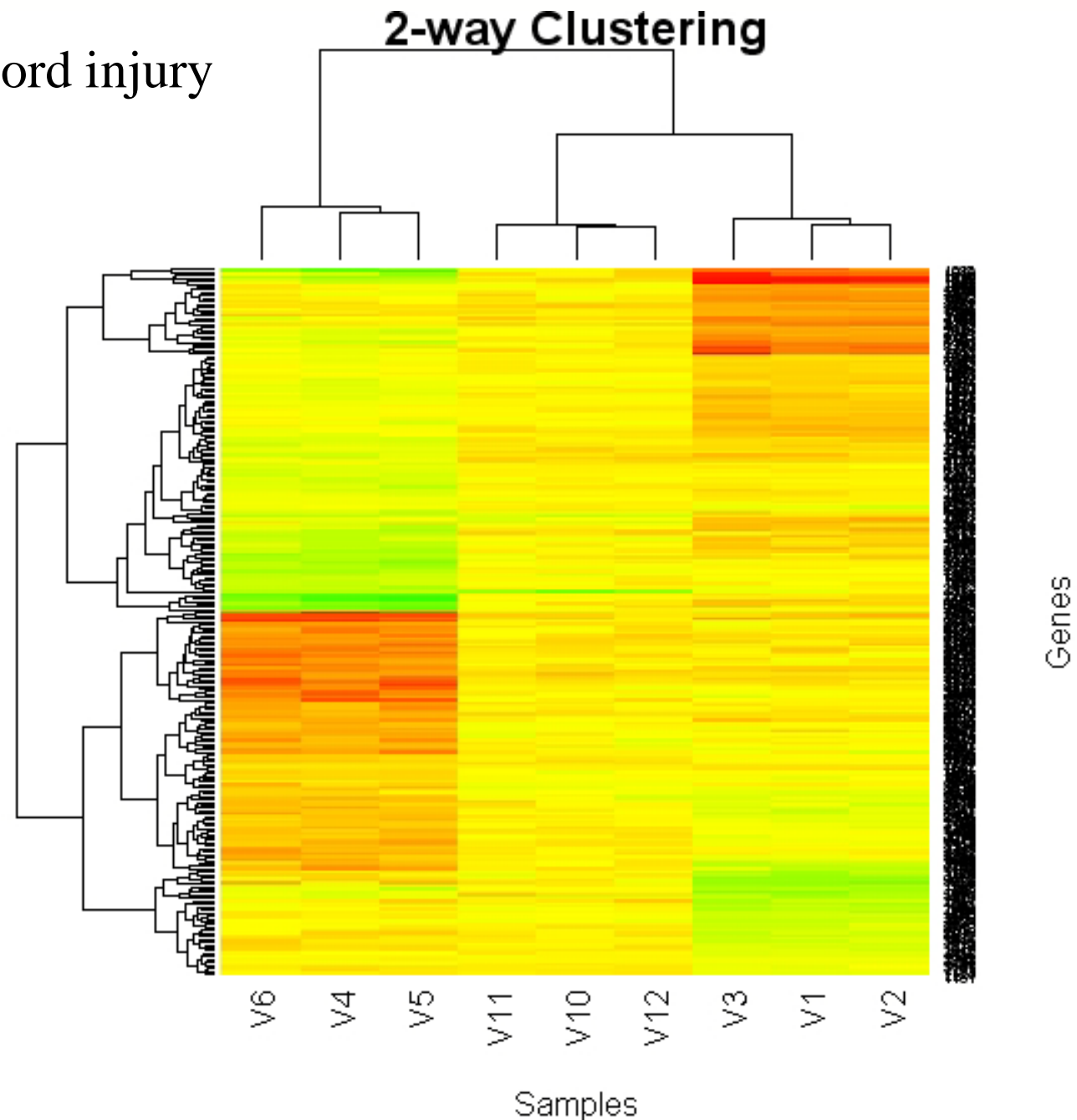
Model-based clustering methods.

- Pro: we're using a specific parametric model to describe the gene expression data, and we can check the fit of this model
- Pro: **We can use standard statistical model selection techniques to select the number of clusters K , and the parameters that defines the pattern of each cluster**
- Con: can converge to a local optimum
- Con: we make assumptions on the distribution of the data.

Clustering

An example: Spinal cord injury

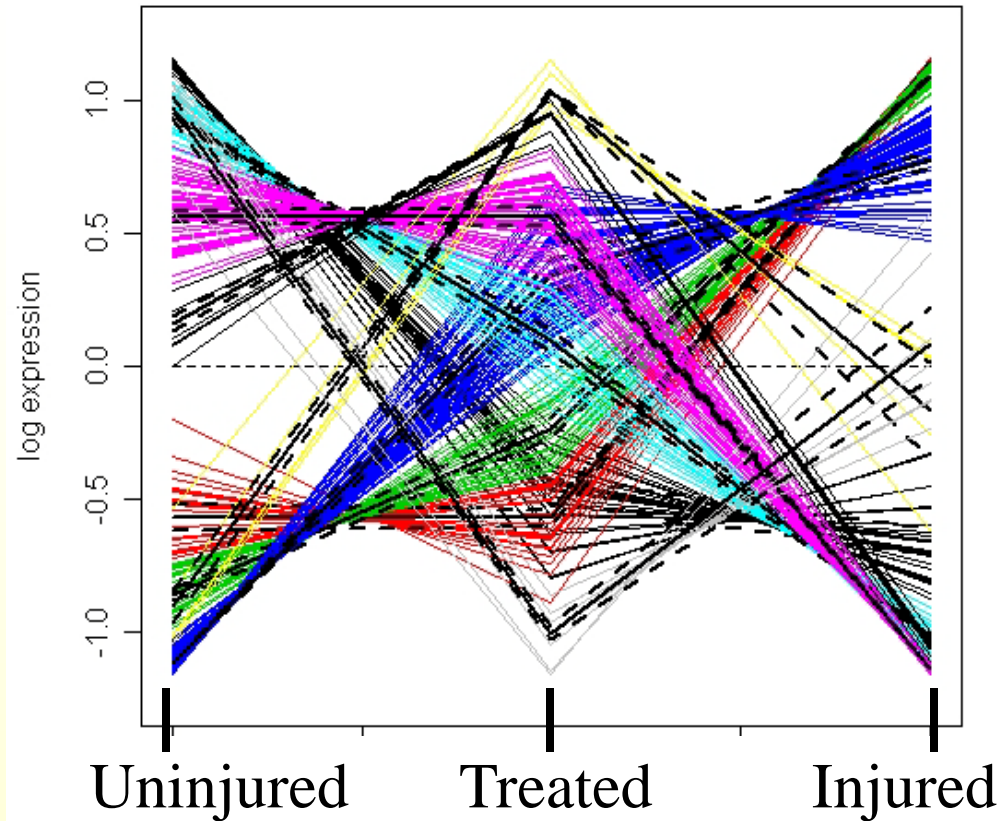
- There are 3 types of samples; uninjured tissue, injured tissue treated with an anti-inflammatory drug, and untreated injured tissue.
- Here is a simple two-way hierarchical clustering.
- We see that the sample types cluster together



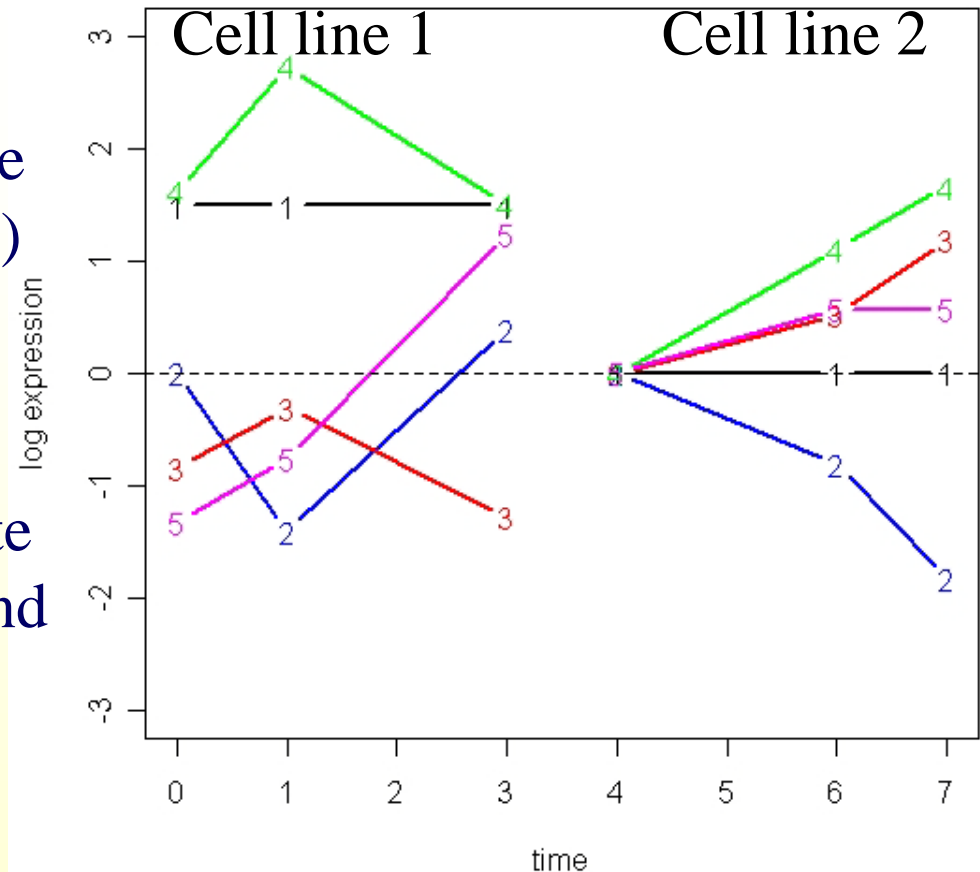
Clustering

An example: Spinal cord injury

- Here are the selected gene clusters (9 of them)
- As you can see, a few **cluster models indicate that the uninjured and injured-treated have similar expression** while the injured-untreated differs from the others (magenta, red, green)
- - this would suggest that the drug treatment suppresses the injury effect!



- Here is a more complex example with two-factors in the cluster model (time & cell-line)
- Clustering detects 5 distinct patterns.
- After clustering we can allocate genes to specific time-course and cell-line patterns.



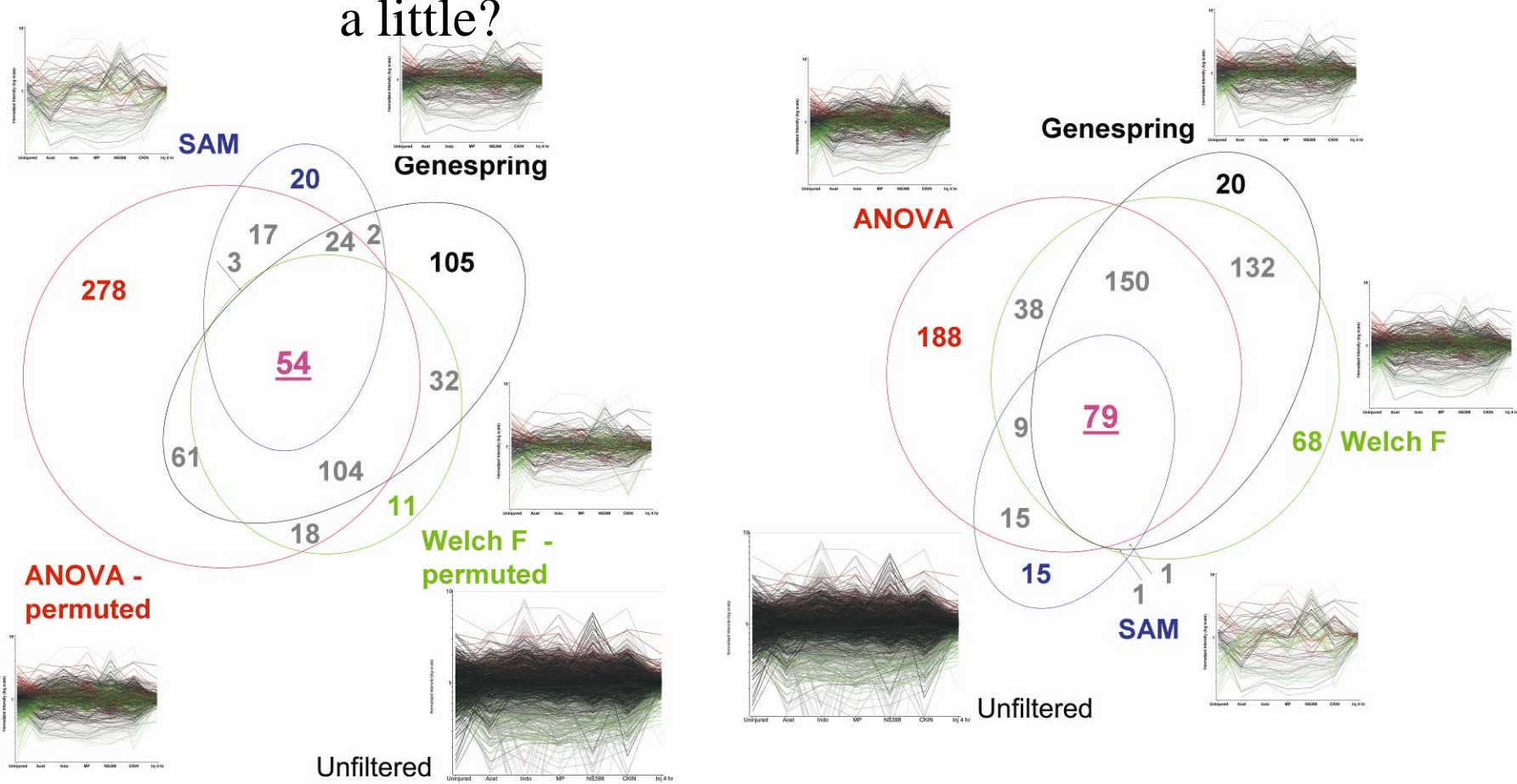
- Lots of current work in this area (Raftery et al. (UWash), Hongzhe Li (UPenn), W. Pan (UMinn) and many more....) The goal of these statistically oriented clustering methods is to make clustering less subjective.

New methods in clustering: variable selection and prior information.

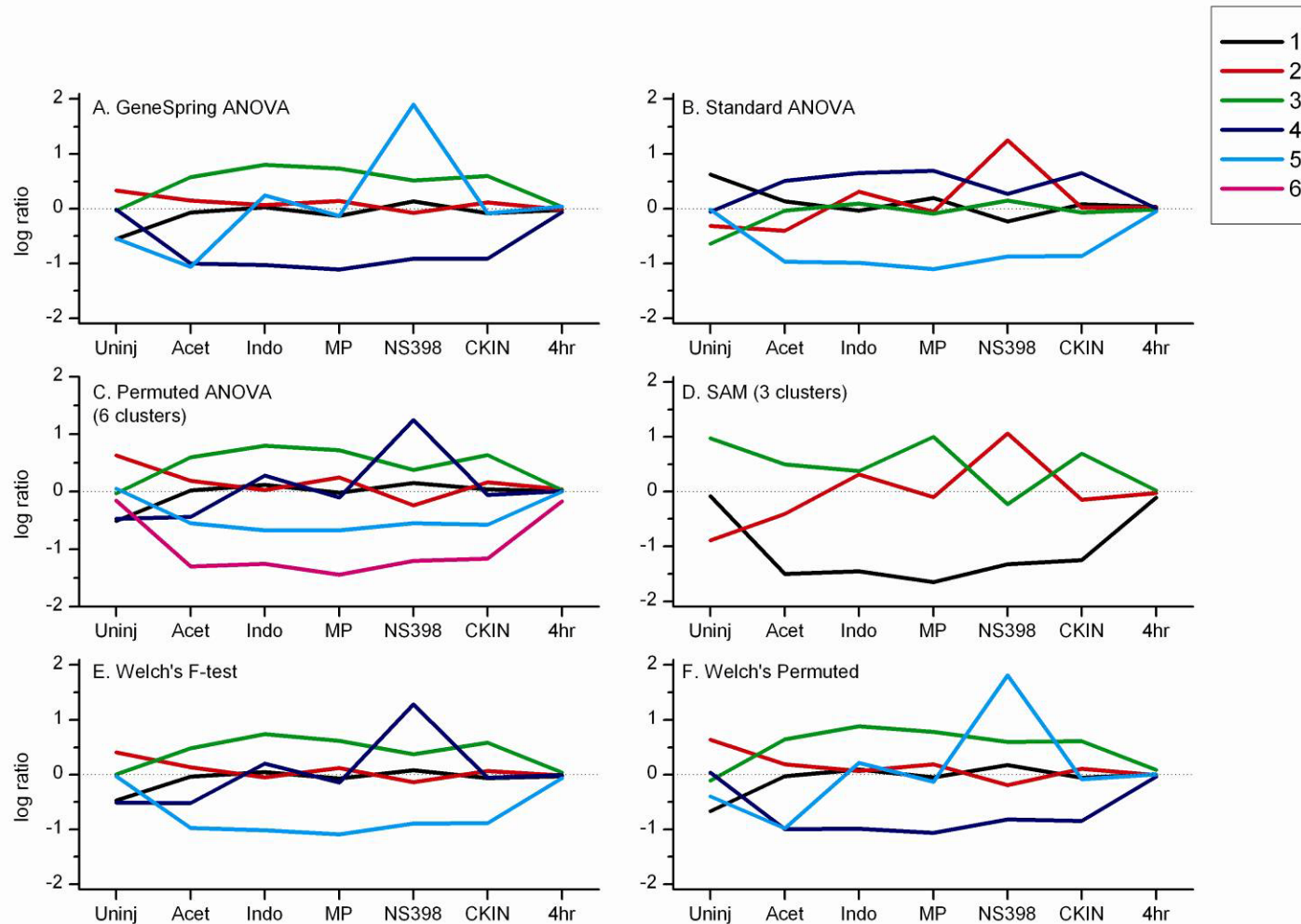
- Variable selection in clustering – experiments are “in or out” when it comes to finding gene clusters (e.g. Raftery et al). Genes “in or out” to find sample clusters (Zhu et al). Cluster specific models (Jornsten et al, Yuan et al).
- Prior knowledge and clustering: “must-link”, adjusted distances, letting cluster parameters depend on group terms (e.g. Foudas et al).

Clustering

Stability analysis: how much will analysis and clustering results differ if I alter the data a little?



Stability analysis: how much will results differ if I alter the data a little?



Stability analysis: how much will results differ if I alter the data a little?

Sometimes resampling based methods (stability analysis) can be used to generate new clustering methods. E.g. Tseng and Wong – resampling and look for *consensus* clusters. Are genes “always” together?

Resample experiments, genes, replicates, add noise,....

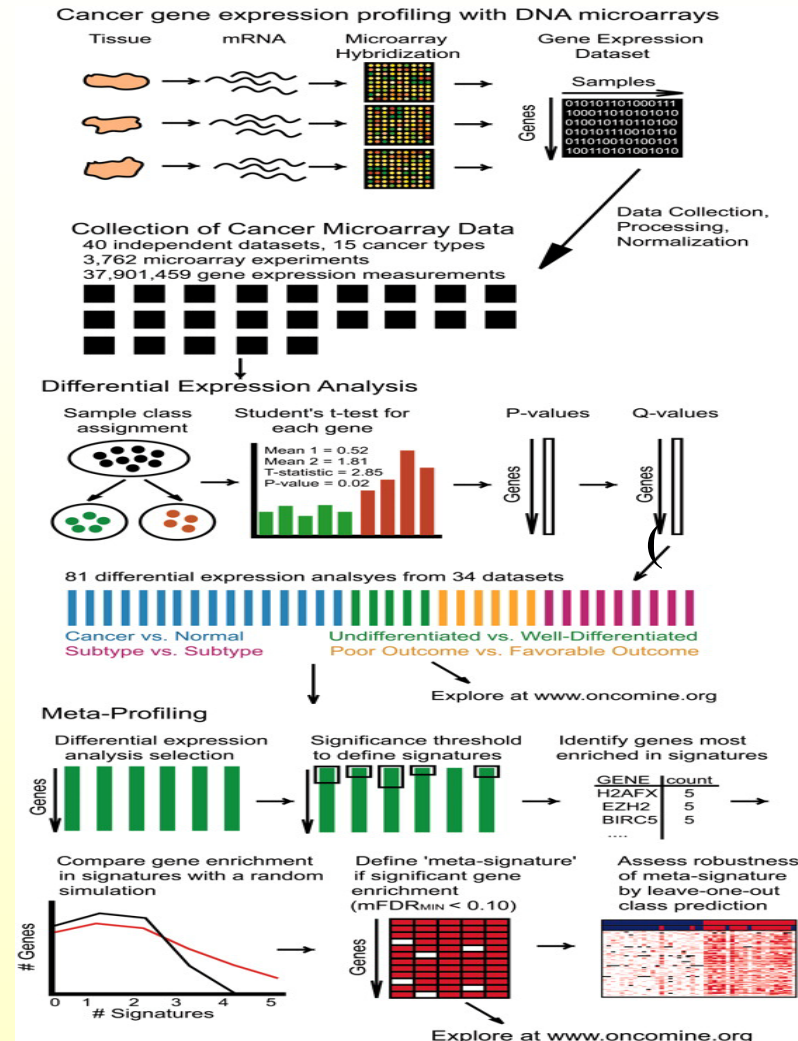
Clustering

.Currently, much focus on combining multiple data sets and background knowledge into analysis of gene expression data.

.The base models are often very similar to what we have discussed (regression, model-based clustering).

.BUT the extra knowledge we incorporate tries to make results somewhat consistent with prior belief

Challenge: how to weigh the different types of experimental evidence together.



Rhodes, Daniel R. et al. (2004) Proc. Natl. Acad. Sci. USA 101, 9309-9314

Take-home Message

1. Significance Analysis

- Don't forget to check the assumptions of the test you use: normality of error? equal variance? independent sampling?
- Choose an appropriate test. Which parameter are you interested in? Can you formulate a model that gives you a direct estimate of this parameter? Do you need to regularize your test-statistic? Did you take covariates (confounders) into account?
- Imputation affects downstream analysis so it's worthwhile doing it properly
- Are values missing or do you have a calibration problem?
- Ask a statistician about the design issues before you spend \$\$\$.

2. Clustering

- is exploratory, but can be useful for data reduction, increased understanding of the data structure
- The choices of distance metric and clustering algorithm drive the results – different approaches focus on different aspects of the data. You get what you ask for – no more, no less!
- Stability analysis is a great way to check how much your assumptions drive the analysis (run clustering after making small changes to the data, e.g. # genes, one sample in/out).