University of Zurich[UZH]

Institute of Molecular Life Sciences

# limma, Affymetrix, RMA, Independent filtering

- Journal club + Projects
- Review of last week: moderation
- design matrices + contrast matrix
- limma mathematical theory
- Affymetrix arrays + RMA

Mark D. Robinson, Statistical Genomics, IMLS

**Journal club signups → by 18.00 / 31.10.2016**

**(Submit a PR to materials/ README.md)**

**Project proposal: for your team, write 2-3 sentences with the plan. Target: mid/ late-November**

| | | | | |
|---|---|---|---|---|
| 24.10.2016 | Mark | limma 1 | | |
| 31.10.2016 | Mark | limma 2 | Topological Data Analysis Generates High-Resolution, Genome-wide Maps of Human Recombination {CS} | x |
| 07.11.2016 | Hubert | RNA-seq quantification | Reliable detection of subclonal single-nucleotide variants in tumour cell populations {CB,L-WY} | A network-based method to evaluate quality of reproducibility of differential expression in cancer genomics studies {TS, SS} |
| 14.11.2016 | Mark | edgeR+friends 1 | Impact of statistical models on the prediction of type 2 diabetes using non-targeted metabolomics profiling {FB,SM,CP} | x |
| 21.11.2016 | Mark | edgeR+friends 2 | Adjusting batch effects in microarray expression data using empirical Bayes methods {KH, SS} | A statistical approach for identifying differential distributions in single-cell RNA-seq experiments {VS, FH} |
| 28.11.2016 | Hubert | classification | A Method for Checking Genomic Integrity in Cultured Cell Lines from SNP Genotyping Data {EP, PCC} | Shrinkage estimation of dispersion in Negative Binomial models for RNA-seq experiments with small sample size {AD, KL, XL} |
| 5.12.2016 | Mark | epigenomics, DNA methylation | Empirical Bayes Analysis of a Microarray Experiment {GA, IA} | x |
| 12.12.2016 | Mark | gene set analysis | x | x |
| 19.12.2016 | Mark | single-cell | The statistical properties of gene-set analysis {AS, FE, MB} | x |

# Project ideas: Consulting/Research

1. **Differential expression of long non-coding RNAs.** 48 samples paired-end RNA-seq data. This collaboration, involving preprocessing and primary DE analysis of the data.
2. **Differential methylation**. Preprocessing and discovery of DMRs for BS-seq data.
3. **Identify somatic mutations in RNA-seq data**. Identify gene mutations in serrated lesions vs conventional adenomas using RNA sequencing data.
4. Standard options:
   - perform a methods comparison
   - recreate some analyses from a published article

# Differential expression, small sample inference

- Table of data (e.g., microarray gene expression data with replicates of each of condition A, condition B)
  - *rows* = features (e.g., genes), *columns* = experimental units (samples)
- Most common problem in statistical bioinformatics: want to infer whether there is a **change in the response** → a statistical test for each row of the table.

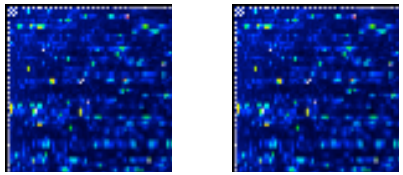
What test might you use? Why is this hard? What issues arise? How much statistical power is there [1] ?

```
> head(y)
        group0     group0      group0     group1     group1      group1
gene1 -0.1874854  0.2584037 -0.05550717 -0.4617966 -0.3563024 -0.03271432
gene2 -3.5418798 -2.4540999  0.11750996 -4.3270442 -5.3462622 -5.54049106
gene3 -0.1226303  0.9354707 -1.10537767 -0.1037990  0.5221678 -1.72360854
gene4 -2.3394536 -0.3495697 -3.47742610 -3.2287093  6.1376670 -2.23871974
gene5 -3.7978820  1.4545702 -7.14796503 -4.0500796  4.7235714 10.00033769
gene6  1.4627078 -0.3096070 -0.26230124 -0.7903434  0.8398769 -0.96822312
```

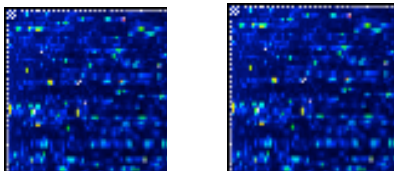
[1] http://www.stat.ubc.ca/~rollin/stats/ssize/n2.html

# A very common experiment

Mutant x 2



WT x 2



Gene X

Which genes are differentially expressed?

$n_1 = n_2 = 2$ microarrays

~30,000 features (e.g., genes) measured

# Ordinary t-tests (1-colour)

$$t_g = \frac{\overline{y}_{\mathrm{mu}} - \overline{y}_{\mathrm{wt}}}{s_g\, c}$$

gives very high false discovery rates

$$c = \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

Residual df = 2

## t-tests with common variance

$$t_{g,\text{pooled}} = \frac{\overline{y}_{\text{mu}} - \overline{y}_{\text{wt}}}{s_0\, c}$$

with residual standard deviation $s_0$ pooled across genes

More stable, but ignores gene-specific variability

$$c = \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

## Posterior Statistics

Posterior variance estimators

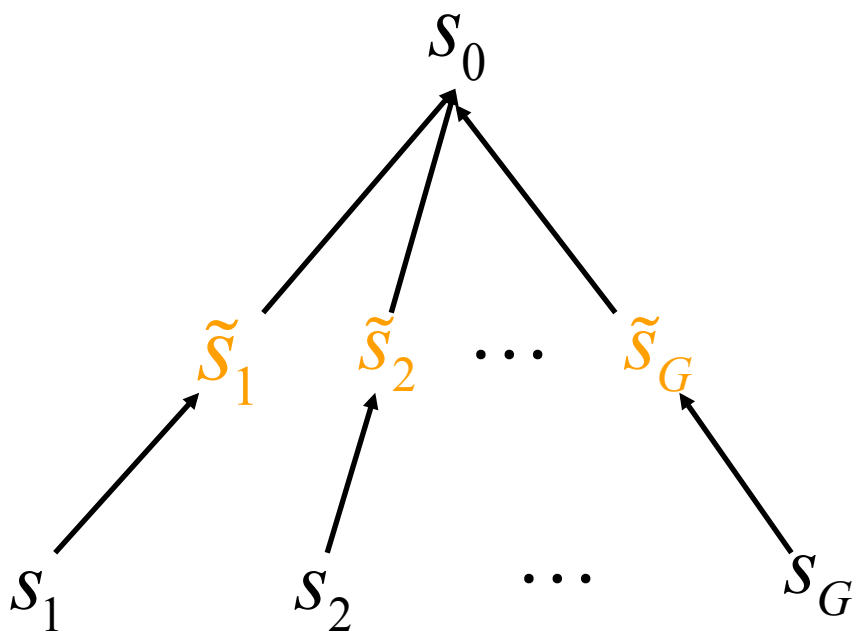
$$\tilde{s}_g^2 = \frac{s_0^2 d_0 + s_g^2 d_g}{d_0 + d_g}$$

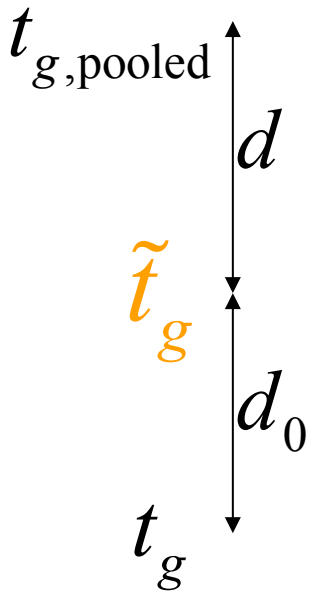
Moderated t-statistics

$$\tilde{t}_g = \frac{\overline{y}_{\mathrm{mu}} - \overline{y}_{\mathrm{wt}}}{\tilde{s}_g \, u}$$

Baldi & Long 2001, Wright & Simon 2003, Smyth 2004

# Shrinkage of standard deviations




The **data decides** whether $\tilde{t}_g$ should be closer to $t_{g,\text{pooled}}$ or to $t_g$

# What layers to add today

- Where does the moderated variance come from?
- Why the degrees of freedom add: $d_0 + d$
- empirical Bayes: how to estimate the hyperparameters ($d_0$ and $s_0$)
- Design matrices + contrast matrices in practice

## Exercise:
## where does the t-distribution come from?

## 10-15 minutes: discuss with your neighbour, use the resources provided and/or search the web to explain .. where does the t-test/t-distribution originate from.

# Unexpected mathematics: Why do degrees of freedom add?

**The construction of the classical t-statistic:**

$$Z = \left(\overline{X}_n - \mu\right)\frac{\sqrt{n}}{\sigma}$$

$$V = (n-1)\frac{S_n^2}{\sigma^2}$$

$$T \equiv \frac{Z}{\sqrt{V/\nu}} = \left(\overline{X}_n - \mu\right)\frac{\sqrt{n}}{S_n},$$

**Stated another way → Exercise (optional): what are a, b above?**

If $T$ is distributed as $(a/b)^{1/2}Z/U$ where $Z \sim N(0,1)$ and $U \sim \chi_\nu$, then $T$ has density function

$$p(t) = \frac{a^{\nu/2}b^{1/2}}{B(1/2,\nu/2)(a+bt^2)^{1/2+\nu/2}}$$

http://en.wikipedia.org/wiki/Bayesian_inference

# Exercise: Derive the posterior

Data $$s_g^2 \sim \sigma_g^2 \frac{\chi^2_{d_g}}{d_g}$$

Prior $$\frac{1}{\sigma_g^2} \sim s_0^2 \frac{\chi^2_{d_0}}{d_0}$$

Posterior $$E\left(\frac{1}{\sigma_g^2} \mid s_g^2\right) = \frac{d_0 + d_g}{s_0^2 d_0 + s_g^2 d_g}$$

$$p(\theta|x) = \frac{f(x|\theta)p(\theta)}{\int f(x|\theta)p(\theta)d\theta}$$

**Optional exercise**

Sketch: i) Let $x=s^2$, $\theta=\sigma^{-2}$; ii) Using the functional form of chi-squared distribution, calculate only the numerator (since denominator does not contain θ); iii) collect terms and see if you can identify the distribution and the parameters of it; iv) What is the mean of this distribution?

## Linear Models

- In general, need to specify:
  - Dependent variable
  - Explanatory variables (experimental design, covariates, etc.)
- More generally:

$$y = X\alpha + \epsilon$$

vector of observed data

design matrix

Vector of parameters to estimate

Obtain a linear model for each gene *g*

$$E(\underset{\sim}{y}_g) = X\underset{\sim}{\alpha}_g$$

$$\mathrm{var}(\underset{\sim}{y}_g) = W_g^{-1}\sigma_g^2$$

# Contrasts -- `contrasts.fit()`

A *contrast* is any linear combination of the coefficients $\alpha_j$ which we want to test equal to zero.

Define contrasts

$$\underset{\sim}{\beta}_g = C^T \underset{\sim}{\alpha}_g$$

were C is the *contrast matrix*.

Want to test

$$H_0 : \beta_{gj} = 0$$

vs

$$H_a : \beta_{gj} \neq 0$$

# Unexpected mathematics: Why do degrees of freedom add?

$$p(\hat{\beta}, s^2 \,|\, \beta = 0) = \int p(\hat{\beta} \,|\, \sigma^{-2}, \beta = 0)\, p(s^2 \,|\, \sigma^{-2})\, p(\sigma^{-2})\, d(\sigma^{-2})$$

The integrand is

$$\begin{aligned}
& \frac{1}{(2\pi v\sigma^2)^{1/2}} \exp\left(-\frac{\hat{\beta}^2}{2v\sigma^2}\right) \\
\times\ & \left(\frac{d}{2\sigma^2}\right)^{d/2} \frac{s^{2(d/2-1)}}{\Gamma(d/2)} \exp\left(-\frac{ds^2}{2\sigma^2}\right) \\
\times\ & \left(\frac{d_0 s_0^2}{2}\right)^{d_0/2} \frac{\sigma^{-2(d_0/2-1)}}{\Gamma(d_0/2)} \exp\left(-\sigma^{-2}\frac{d_0 s_0^2}{2}\right) \\
=\ & \frac{(d_0 s_0^2/2)^{d_0/2}(d/2)^{d/2} s^{2(d/2-1)}}{(2\pi v)^{1/2}\Gamma(d_0/2)\Gamma(d/2)} \\
& \sigma^{-2(1/2+d_0/2+d/2-1)} \exp\left\{-\sigma^{-2}\left(\frac{\hat{\beta}^2}{2v} + \frac{ds^2}{2} + \frac{d_0 s_0^2}{2}\right)\right\}
\end{aligned}$$

# Unexpected mathematics: Why do degrees of freedom add?

$$p(\hat{\beta}, s^2 \mid \beta = 0) = \int p(\hat{\beta} \mid \sigma^{-2}, \beta = 0) p(s^2 \mid \sigma^{-2}) p(\sigma^{-2}) d(\sigma^{-2})$$

$$= \frac{(d_0 s_0^2/2)^{d_0/2} (d/2)^{d/2} s^{2(d/2-1)}}{(2\pi v)^{1/2} \Gamma(d_0/2) \Gamma(d/2)}$$

$$\sigma^{-2(1/2+d_0/2+d/2-1)} \exp\left\{-\sigma^{-2}\left(\frac{\hat{\beta}^2}{2v} + \frac{d s^2}{2} + \frac{d_0 s_0^2}{2}\right)\right\}$$

$\sigma^{-2}$ is chi-squared (or gamma)

$$f(x; k) = \begin{cases} \frac{x^{(k/2)-1} e^{-x/2}}{2^{k/2}\Gamma\left(\frac{k}{2}\right)}, & x \geq 0; \\ 0, & \text{otherwise.} \end{cases}$$

http://en.wikipedia.org/wiki/Chi-squared_distribution

# Unexpected mathematics: Why do degrees of freedom add?

$$p(\hat{\beta}, s^2 \mid \beta = 0) = \int p(\hat{\beta} \mid \sigma^{-2}, \beta = 0) p(s^2 \mid \sigma^{-2}) p(\sigma^{-2}) d(\sigma^{-2})$$

$$p(\hat{\beta}, s^2 \mid \beta = 0)$$
$$= \frac{(1/2v)^{1/2} (d_0 s_0^2/2)^{d_0/2} (d/2)^{d/2} s^{2(d/2-1)}}{D(1/2, d_0/2, d/2)} \left( \frac{\hat{\beta}^2/v + d_0 s_0^2 + d s^2}{2} \right)^{-(1+d_0+d)/2}$$

# Unexpected mathematics: Why do degrees of freedom add?

$$p(\hat{\beta}, s^2 \mid \beta = 0)$$
$$= \frac{(1/2v)^{1/2}(d_0 s_0^2/2)^{d_0/2}(d/2)^{d/2} s^{2(d/2-1)}}{D(1/2, d_0/2, d/2)} \left( \frac{\hat{\beta}^2/v + d_0 s_0^2 + d s^2}{2} \right)^{-(1+d_0+d)/2}$$

The null joint distribution of $\tilde{t}$ and $s^2$ is

$$p(\tilde{t}, s^2 \mid \beta = 0) = \tilde{s} v^{1/2} p(\hat{\beta}, s^2 \mid \beta = 0)$$

http://en.wikipedia.org/wiki/Random_variable#Distribution_functions_of_random_variables

$$f_Y(y) = f_X(g^{-1}(y)) \left| \frac{dg^{-1}(y)}{dy} \right|$$

# Unexpected mathematics: Why do degrees of freedom add?

If $T$ is distributed as $(a/b)^{1/2}Z/U$ where $Z \sim N(0,1)$ and $U \sim \chi_\nu$, then $T$ has density function

$$p(t) = \frac{a^{\nu/2}b^{1/2}}{B(1/2,\nu/2)(a+bt^2)^{1/2+\nu/2}}$$

$$p(\tilde{t}, s^2 \,|\, \beta = 0) = \frac{(d_0 s_0^2)^{d_0/2} d^{d/2} s^{2(d/2-1)}}{B(d/2, d_0/2)(d_0 s_0^2 + d s^2)^{d_0/2+d/2}} \times \frac{(d_0+d)^{-1/2}}{B(1/2, d_0/2+d/2)} \left(1 + \frac{\tilde{t}^2}{d_0+d}\right)^{-(1+d_0+d)/2}$$

This shows that $\tilde{t}$ and $s^2$ are independent with

$$s^2 \sim s_0^2 F_{d,d_0}$$

and

$$\tilde{t} \,|\, \beta = 0 \sim t_{d_0+d}.$$

## Linear Models

- In general, need to specify:
  - Dependent variable
  - Explanatory variables (experimental design, covariates, etc.)
- More generally:

$$y = X\alpha + \epsilon$$

vector of observed data

design matrix

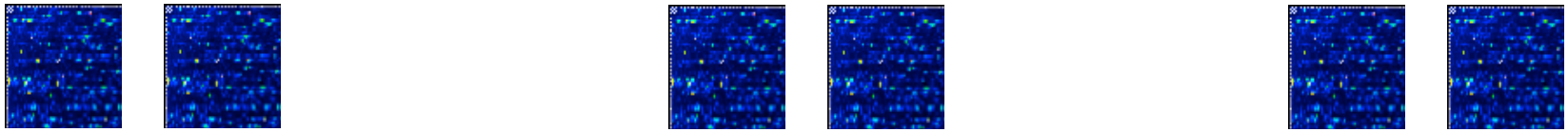
Vector of parameters to estimate

# Linear Models for microarrays

- Combined estimation of precision (moderated variance)
- Extensible to arbitrarily complicated experiments (multiple groups, factorial designs, time courses, paired designs, etc.)
  - NB: only special cases of mixed models are covered
- **Design matrix**: specifies experimental condition of each sample
- **Contrast matrix**: specifies which comparisons are of interest

# Analysis of Variance → Linear model

WT x 2 | Cond A x 2 | Cond B x 2



$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix}$$

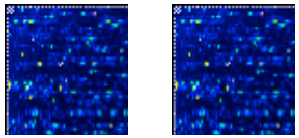
$\alpha_1$ = wt log-expression

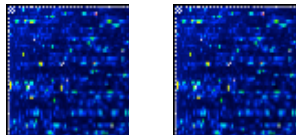
$\alpha_2$ = Cond A - wt

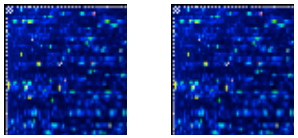
$\alpha_3$ = Cond B - wt

$E[y_1]=E[y_2]=\alpha_1$ $E[y_3]=E[y_4]=\alpha_1+\alpha_2$ $E[y_5]=E[y_6]=\alpha_1+\alpha_3$

# Analysis of Variance → Linear model, alternative parameterization

WT x 2 

Cond A x 2 

Cond B x 2 

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix}$$

$\alpha_1$ = wt log-expression

$\alpha_2$ = Cond A log-expression

$\alpha_3$ = Cond B log-expression

$E[y_1]=E[y_2]=\alpha_1$ $E[y_3]=E[y_4]=\alpha_2$ $E[y_5]=E[y_6]=\alpha_3$

# Linear Model Estimates – `lmFit()`

Obtain a linear model for each gene *g*

$$E(\underset{\sim}{y}_g) = X\underset{\sim}{\alpha}_g$$

$$\text{var}(\underset{\sim}{y}_g) = W_g^{-1}\sigma_g^2$$

Estimate*:*

coefficients $\hat{\alpha}_{gj}$

standard deviations $s_g$

standard errors $\text{se}(\hat{\beta}_{gj})^2 = c_{gj}s_g^2$

# An example use of design and contrast matrices

design matrix

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix}$$

$E[y_1]=E[y_2]=\alpha_1$
$E[y_3]=E[y_4]=\alpha_2$
$E[y_5]=E[y_6]=\alpha_3$

contrast matrix

$$\beta = C\alpha = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_2 - \alpha_1 \\ \alpha_3 - \alpha_2 \end{bmatrix}$$

# Contrasts -- `contrasts.fit()`

A *contrast* is any linear combination of the coefficients $\alpha_j$ which we want to test equal to zero.

Define contrasts

$$\underset{\sim}{\beta}_g = C^T \underset{\sim}{\alpha}_g$$

were C is the *contrast matrix*.

Want to test

$$H_0 : \beta_{gj} = 0$$

vs

$$H_a : \beta_{gj} \neq 0$$

# Limma / Analysis of Variance

$$F = \frac{\text{variance between treatments}}{\text{variance within treatments}}$$

$$F = \frac{MS_{\text{Treatments}}}{MS_{\text{Error}}} = \frac{SS_{\text{Treatments}}/(I-1)}{SS_{\text{Error}}/(n_T - I)}$$

The moderated $t$-statistics also lead naturally to moderated $F$-statistics which can be used to test hypotheses about any set of contrasts simultaneously. Appropriate quadratic forms of moderated $t$-statistics follow $F$-distributions just as do quadratic forms of ordinary $t$-statistics. Suppose that we wish to test all contrasts for a given gene equal to zero, i.e., $H_0 : \boldsymbol{\beta}_g = 0$. The correlation matrix of $\hat{\boldsymbol{\beta}}_g$ is $R_g = U_g^{-1} C^T V_g C U_g^{-1}$ where $U_g$ is the diagonal matrix with unscaled standard deviations $(v_{gj})^{1/2}$ on the diagonal. Let $r$ be the column rank of $C$. Let $Q_g$ be such that $Q_g^T R_g Q_g = I_r$ and let $\mathbf{q}_g = Q_g^T \mathbf{t}_g$. Then

$$F_g = \mathbf{q}_g^T \mathbf{q}_g / r = \mathbf{t}_g^T Q_g Q_g^T \mathbf{t}_g / r \sim F_{r, d_0 + d_g}$$

## Aside: Marginal Distributions to calculate

Fun fact: Under usual likelihood model, $s_g$ is independent of the estimated coefficients.

Under the hierarchical model, $s_g$ is independent of the moderated t-statistics instead

$$s_g^2 \sim s_0^2 F_{d,d_0}$$

Thus, the set of $s_g$ can be used to estimated $d_0$ and $s_0$

University of Zurich[UZH]

Institute of Molecular Life Sciences

# Affymetrix + RMA + IRLS

# Affymetrix probe design

Early platforms (11 or 20 probes in a set), 25bp probes, 3' biased

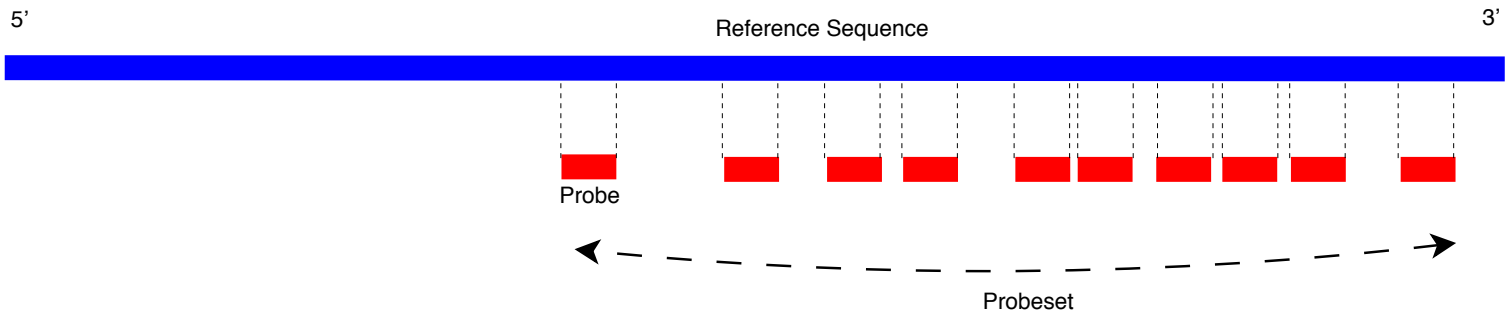



Figure 1.1: Multiple probes interrogating the sequence for a particular gene make up probesets.


Figure 1.2: Pefect Match and Mismatch Probes.

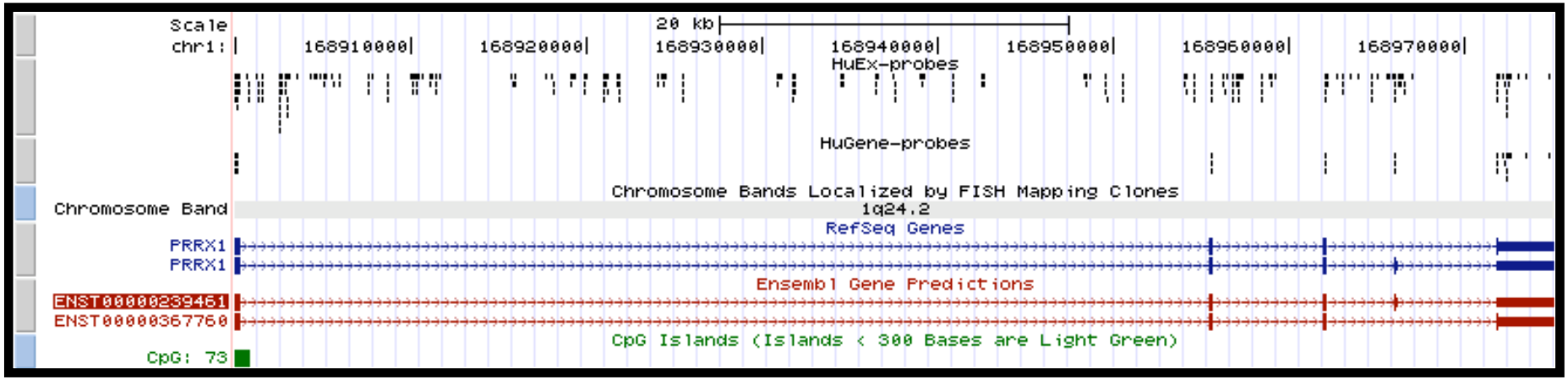
# Latest Affymetrix design: "whole transcript" arrays

Still 25 base pair probes, multiple probes per transcript ("probesets")

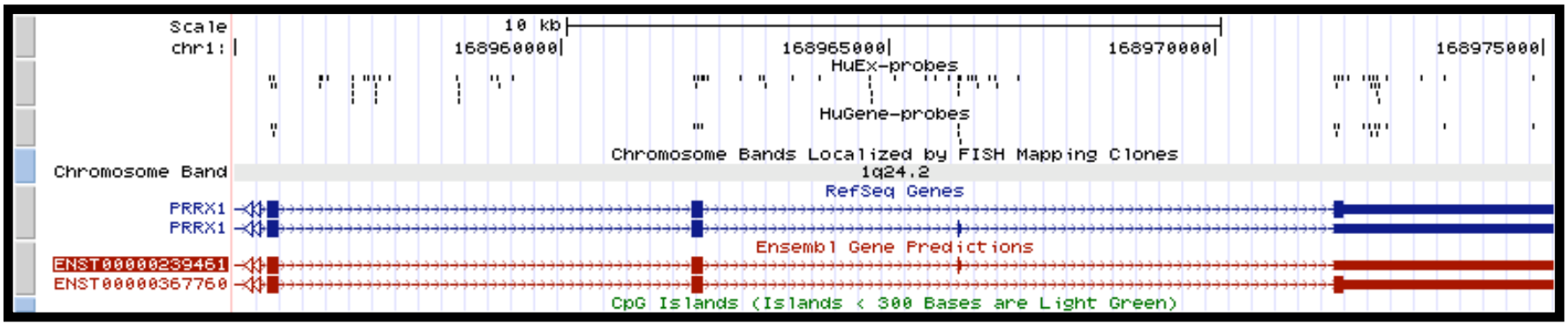
No more mismatch probes.

Reference Sequence

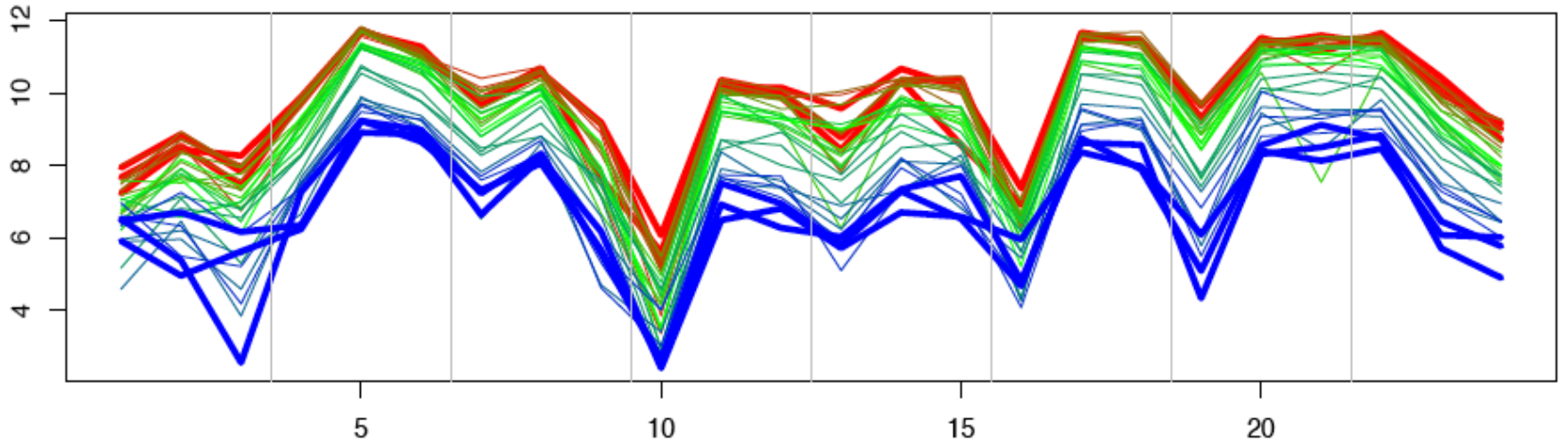
- HuExon: *Human **Exon** 1.0 ST* (~40 probes per gene, 4 probes per "exon", annotated and predicted transcripts)
- HuGene: *Human **Gene** 1.0 ST* (~25 probes per gene, annotated genes only)
- NEW in 2013: HTA (Human Transcriptome Array): updated content + junction probes



# The nature of Affymetrix Probe Level Data




HuGene data [red–heart,blue–brain,mixtures] 10 ENSG00000116132




- Data for one gene that is differentially expressed between heart (red is 100% heart) and brain (blue is 100% brain).
- 11 mixtures x 3 replicates = 33 samples (33 lines)
- Note the parallelism: probes have different **affinities**

# "Summarization": Going from probesets to summarized expression level

MAS 4.0

$$AvDiff = \frac{1}{|A|} \sum_{j \in A} (PM_j - MM_j)$$

MAS 5.0

$$CT_j = \begin{cases} MM_j, & \text{if } MM_j < PM_j \\ \text{less than } PM_j, & \text{if } MM_j \geq PM_j \end{cases}$$

$$signal = TukeyBiweight\{log(PM_j - CT_j)\}$$

dChip (MBEI)

$$PM_{ij} - MM_{ij} = \theta_i \cdot \phi_j + \varepsilon_{ij}, \qquad \varepsilon_{ij} \sim N(0, \sigma^2)$$

$\theta_i$ expression index
$\phi_j$ probe-specific affinity
$\varepsilon_{ij}$ noise component

RMA, GCRMA

# Robust multichip analysis (RMA)

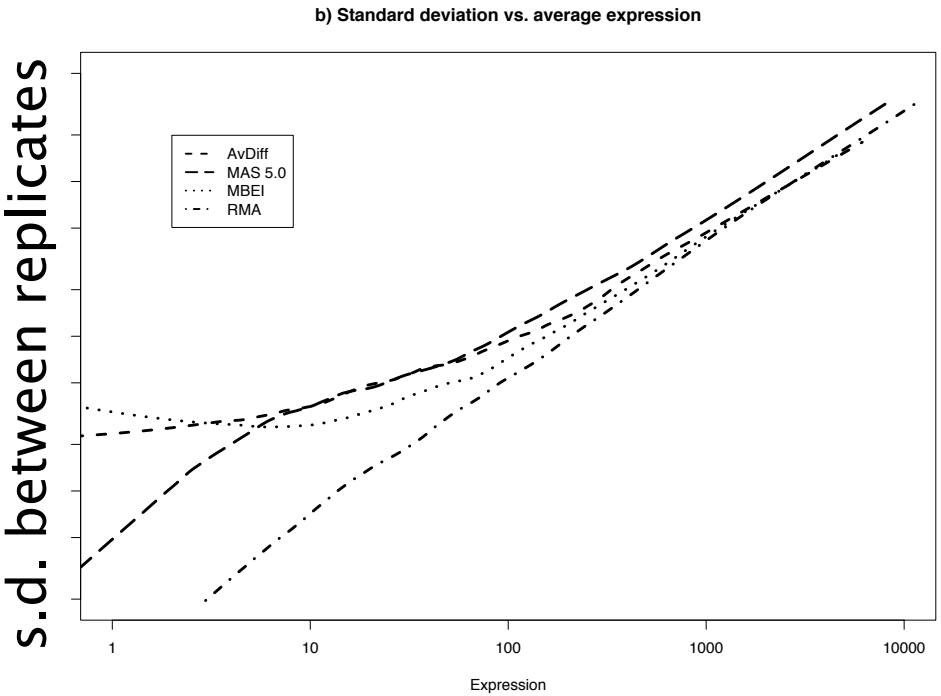
**Exploration, normalization, and summaries of high density oligonucleotide array probe level data**

RAFAEL A. IRIZARRY*
*Department of Biostatistics, Johns Hopkins University, Baltimore MD 21205, USA*
rafa@jhu.edu
BRIDGET HOBBS
*Division of Genetics and Bioinformatics, WEHI, Melbourne, Australia*
FRANCOIS COLLIN
*Gene Logic Inc., Berkeley, CA, USA*
YASMIN D. BEAZER-BARCLAY, KRISTEN J. ANTONELLIS, UWE SCHERF
*Gene Logic Inc., Gaithersburg, MD, USA*
TERENCE P. SPEED
*Division of Genetics and Bioinformatics, WEHI, Melbourne, Australia. Department of Statistics, University of California at Berkeley*

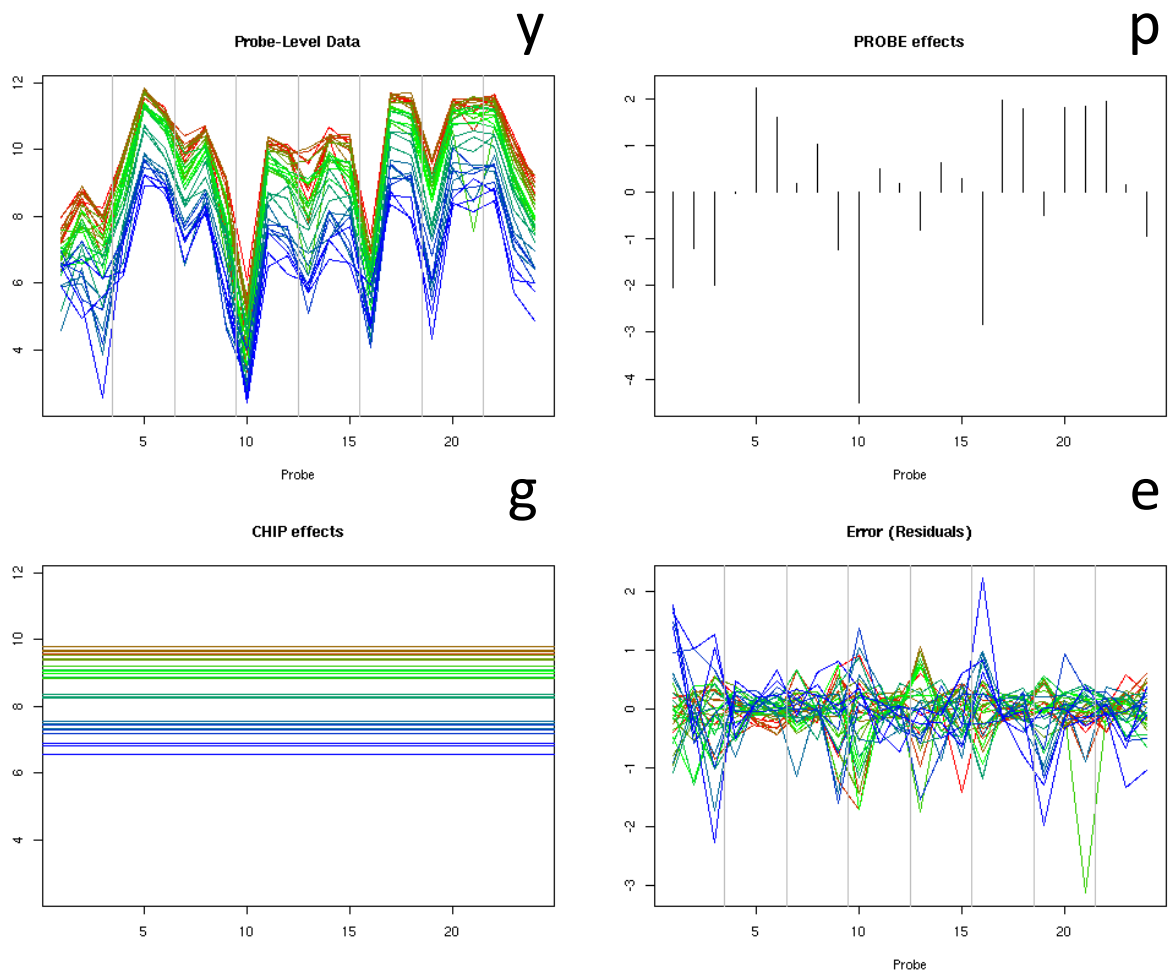
Biostatistics 2003

Encompasses 3 steps

- background correction
- normalization
- probe level model fit ("summarization")



# Linear model decomposes the probe-level data into PROBE effects and CHIP effects




Linear model:

$$y_{ik} = g_i + p_k + e_{ik}$$

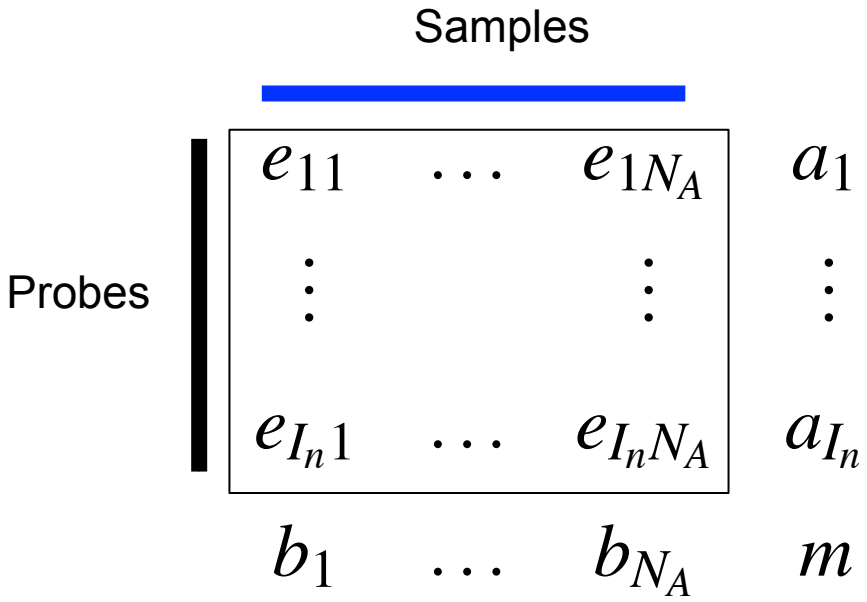
Robust Multichip Analysis (RMA) uses this model.
Irizarry et al. 2003, Biostatistics

Parameters are estimated robustly, meaning a small number of outliers have minimal effect



Tissue mixture dataset

# Fitting the model – median polish

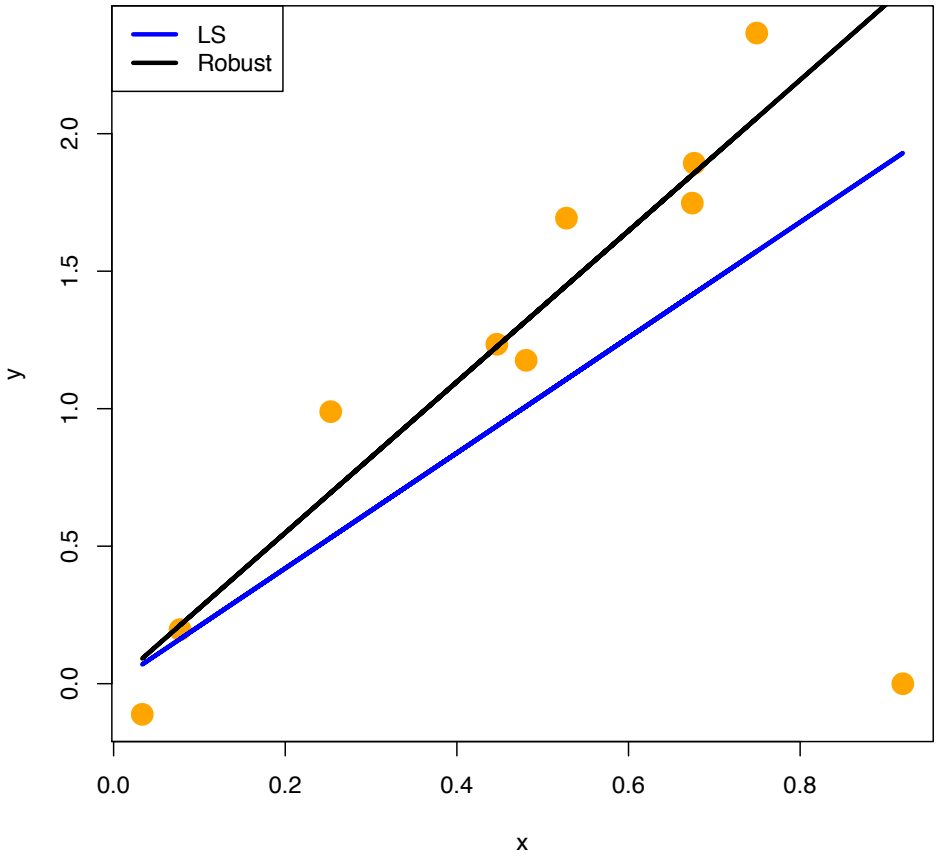



```
pe <- rnorm(11)
ce <- rnorm(8)+8
z <- outer(pe,ce,"+") +
     rnorm(length(pe)*length(ce),sd=.5)
e <- z
m <- a <- b <- 0
niter <- 3

for(i in 1:niter) {
  rm <- rowMedians(e)  # calc row medians
  e <- sweep(e,1,rm)   # subtract row medians
  a <- a + rm          # add row medians to a
  mb <- median(b)
  b <- b-mb
  m <- m+mb

  cm <- colMedians(e)  # calc col medians
  e <- sweep(e,2,cm)   # subtract col medians
  b <- b + cm          # add col medians to b
  ma <- median(a)
  a <- a-ma
  m <- m+ma
}

# a - "probe effects"
# m+b - "chip effects"
```

http://en.wikipedia.org/wiki/Ordinary_least_squares

# Robust regression – motivating example

```
library(MASS)

n <- 10
x <- runif(n)
y <- 3*x + rnorm(n,sd=.2)
y[which.max(y)] <- 0  # add in outlier

f <- lm(y~0+x)
fr <- rlm(y~0+x)

plot(x,y,pch=19,col="orange",cex=2)
lines(x,predict(fr),lwd=3)
lines(x,predict(f),lwd=3,col="blue")
legend("topleft",c("LS","Robust"),
       lwd=3,lty=1,col=c("blue","black"))
```




OLS = ordinary least squares

The OLS estimator is … optimal in the class of linear unbiased estimators when the errors are homoscedastic and serially uncorrelated … OLS provides minimum-variance mean-unbiased estimation when the errors have finite variances.

Has good properties, when the data is “nice”.

Replace:

$$\arg\min_\beta \sum_{i=1}^{n} (y_i - f_i(\beta))^2$$

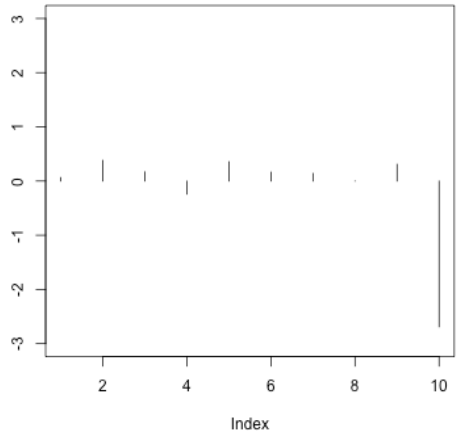
with:

$$\arg\min_\beta \sum_{i=1}^{n} w_i(\beta)(y_i - f_i(\beta))^2$$

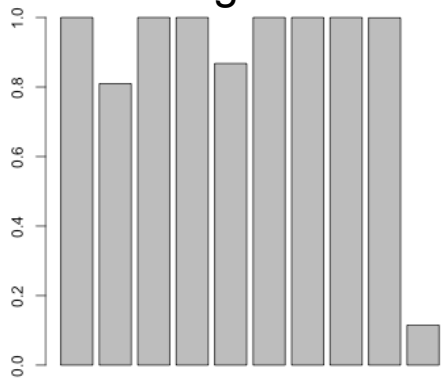
http://en.wikipedia.org/wiki/Iteratively_reweighted_least_squares

# Robust regression – mechanics of iteratively reweighted least squares

Residuals



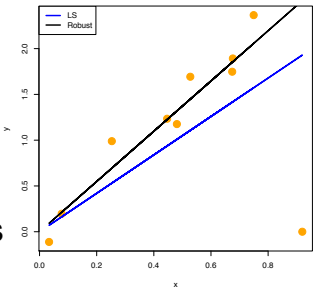

Weights




Sketch of IRLS:

Calculate initial estimates of parameters

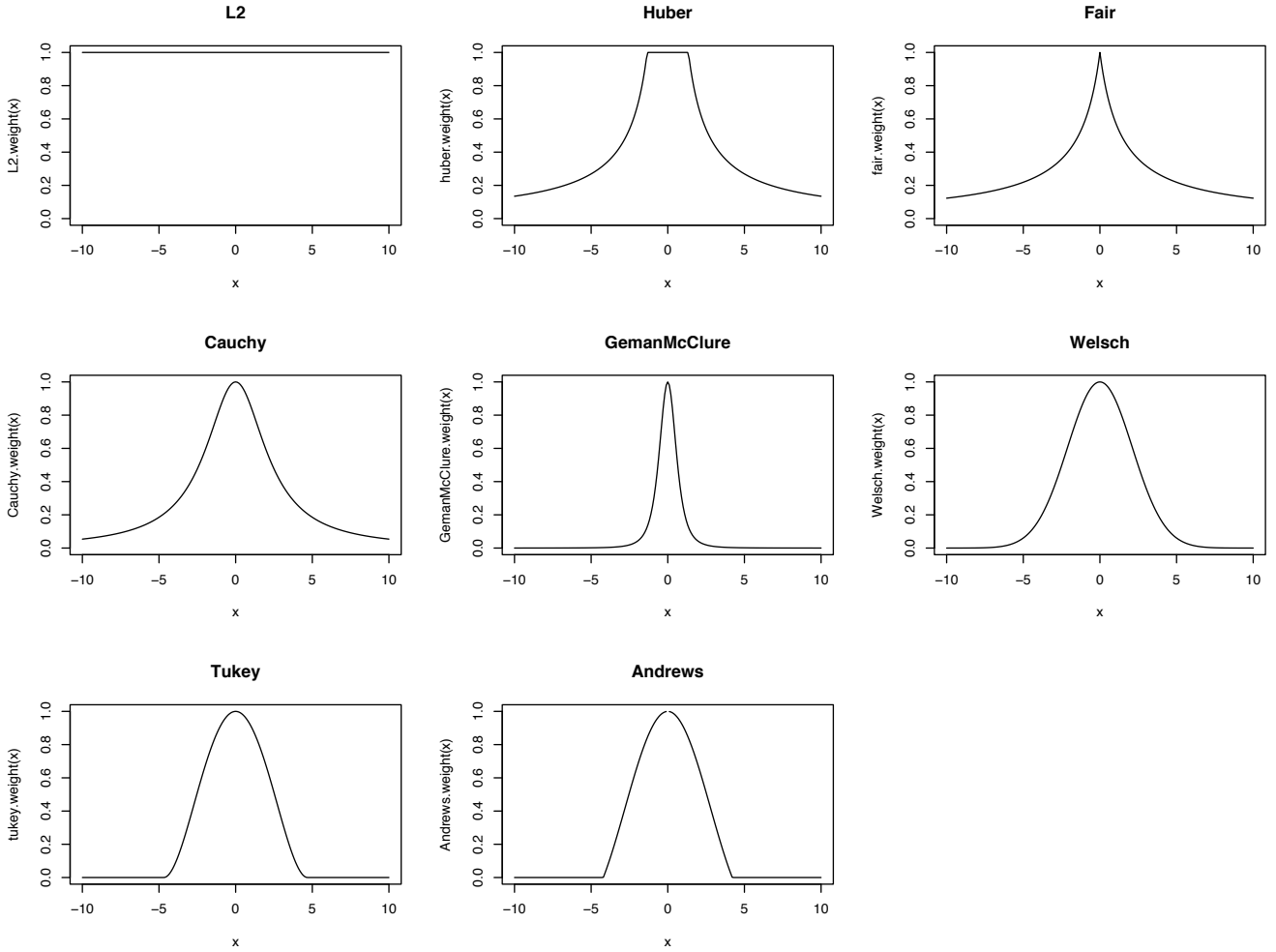
Repeat until very little change:
- Calculate residuals
- Using standardized residuals, weight observations
- Re-estimate parameters




```
# this construction only works for the
# 1-parameter no-intercept linear model
tukey <- function(r,k=1.345) {
  abs(r) < k + k/abs(r)*(abs(r)>k)
}

w <- 1
niter <- 2
b <- sum(w*y*x)/sum(w*x^2)

for(i in 1:niter) {
  r <- y-b*x
  w <- tukey( r/mad(r) )
  b <- sum(w*y*x)/sum(w*x^2)
}


par(mfrow=c(2,1))
plot(r,type="h",ylim=c(-3,3))
barplot(w)
```

← mad = median absolute deviation

# More details – weight functions (as function of standardized residuals)



# More details – weight functions (of normalized residuals)
# Concept: influence / bounded influence

The estimated standard error for our estimators is thus given by

$$\mathrm{SE}\left(\hat{\beta}_j^{(n)}\right) = \frac{1}{\sqrt{I_n}} \sqrt{\frac{\sum_{i=1}^{I_n} \psi\left(\frac{\log_2\left(y_{ij}^{(n)}\right) - \hat{\beta}_j^{(n)}}{s}\right)^2 / I_n}{\left(\sum_{i=1}^{I_n} \psi'\left(\frac{\log_2\left(y_{ij}^{(n)}\right) - \beta_j^{(n)}}{s}\right) / I_n\right)^2}}.$$

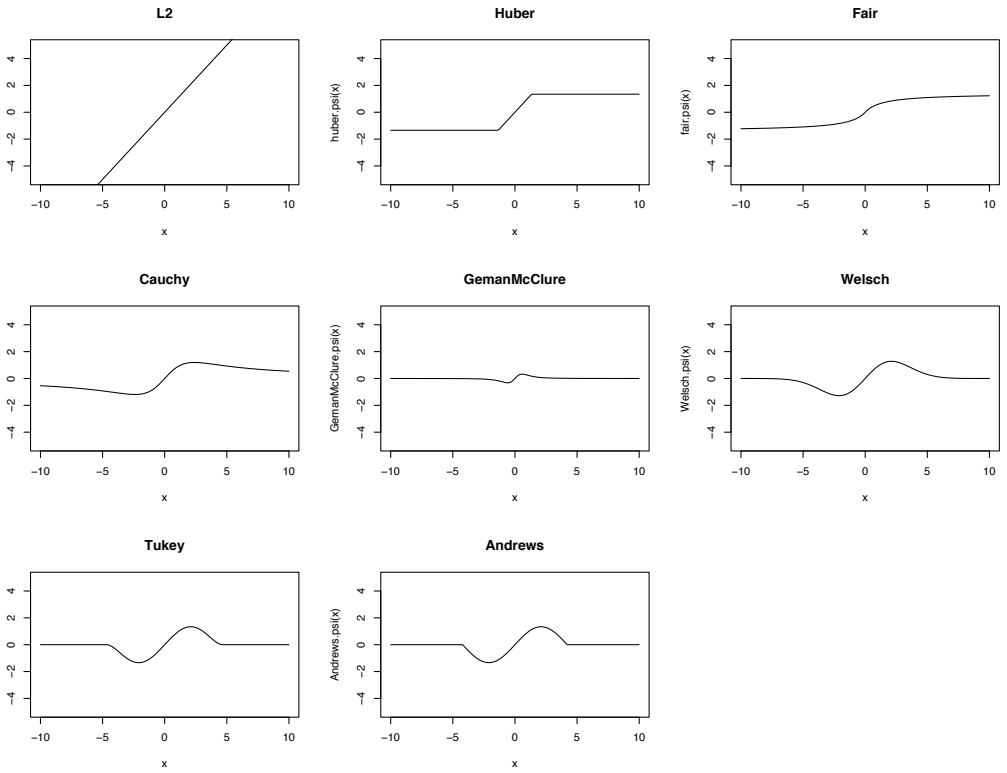
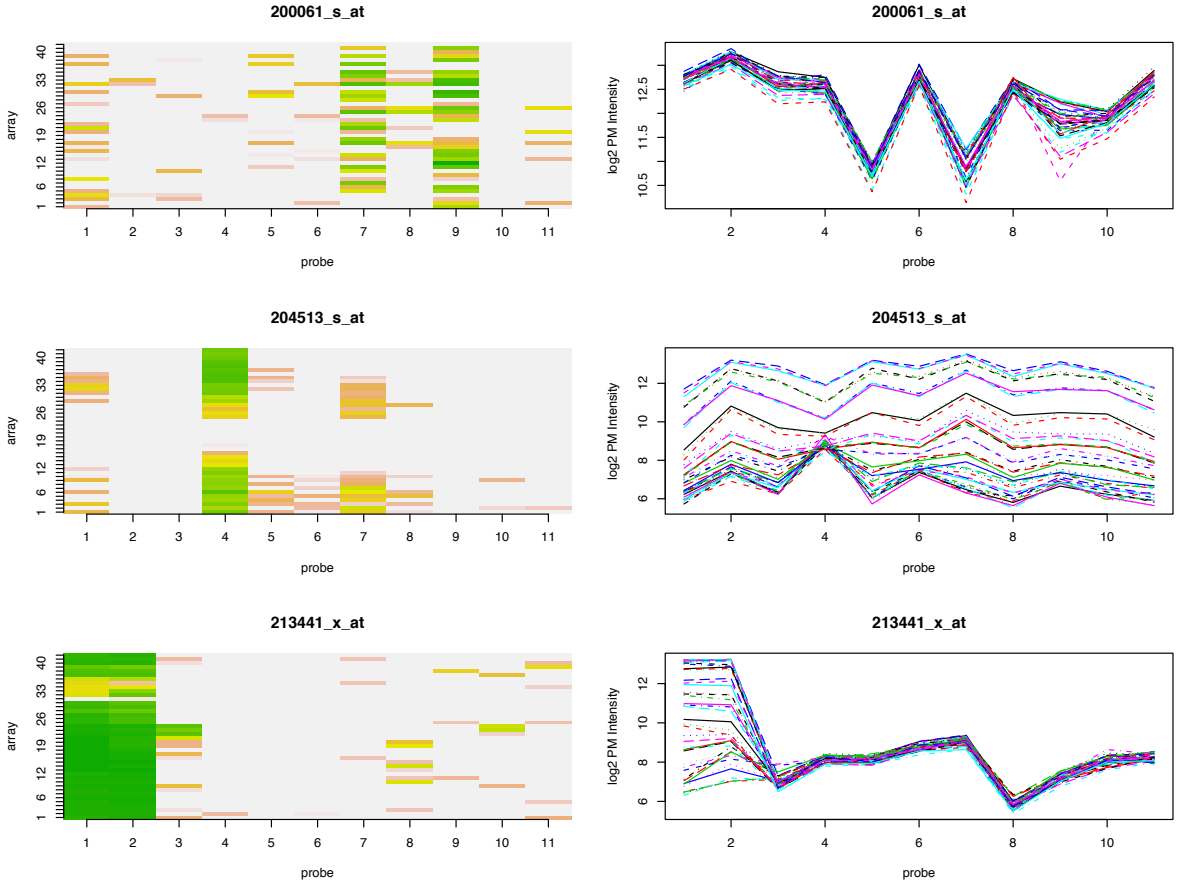


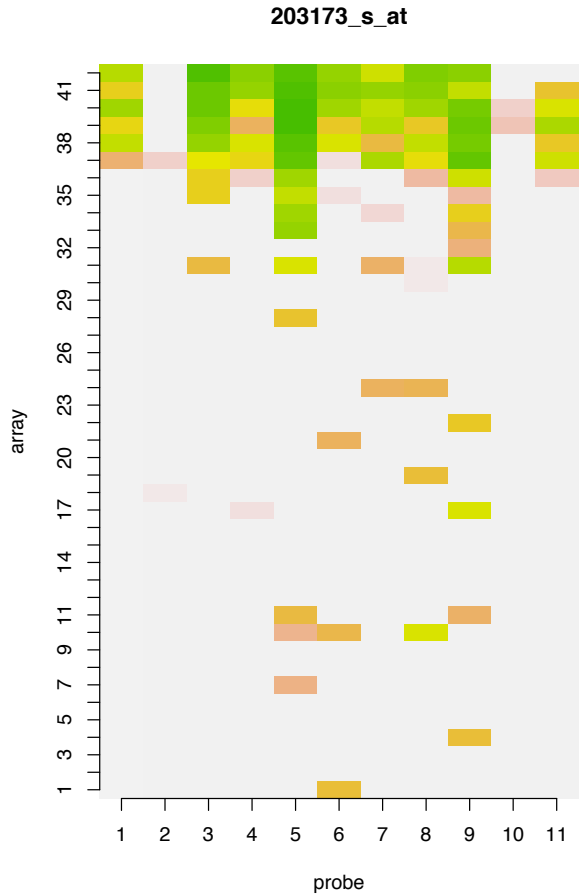

Figure 4.2: The $\psi$ functions for some common M-estimators.

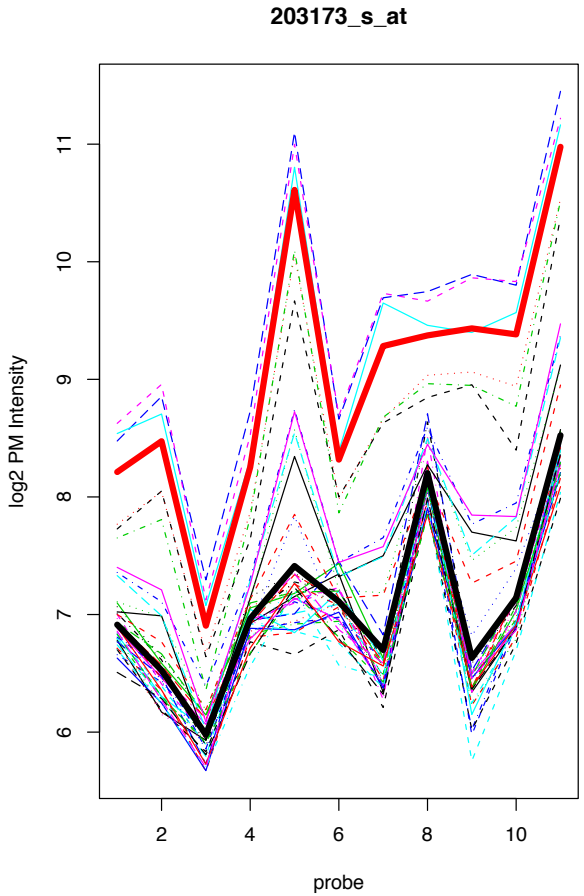
# Robust regression leads to various quality assessment metrics




Identifies poor performing probes

# Robust regression leads to various quality assessment metrics




Identifies poor performing samples

# Relate to limma objects

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix}$$

$E[y_1]=E[y_2]=\alpha_1$
$E[y_3]=E[y_4]=\alpha_2$
$E[y_5]=E[y_6]=\alpha_3$

$$\beta = C\alpha = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_2 - \alpha_1 \\ \alpha_3 - \alpha_2 \end{bmatrix}$$

```
> design
  alpha1 alpha2 alpha3
1      1      0      0
2      1      0      0
3      0      1      0
4      0      1      0
5      0      0      1
6      0      0      1
> cont.matrix <- makeContrasts(beta1="alpha2-alpha1",
                  beta2="alpha3-alpha2",levels=design)

> cont.matrix
         Contrasts
Levels    beta1 beta2
  alpha1     -1     0
  alpha2      1    -1
  alpha3      0     1

fit <- lmFit(y,design)

fit.c  <- contrasts.fit(fit, cont.matrix)
fit.c  <- eBayes(fit.c)

> head(round(y,2),3)
       [,1]   [,2]  [,3]  [,4]   [,5]   [,6]
[1,]  -1.62   1.49  2.50  1.57  -0.71   0.38
[2,]  -4.50  -4.95 -3.66 -7.83  -1.59   6.94
[3,] -10.17 -21.90 14.03  3.66 -12.21 -15.26

> head(round(fit$coef,2),3)
     alpha1 alpha2 alpha3
[1,]  -0.07   2.03  -0.16
[2,]  -4.73  -5.75   2.67
[3,] -16.04   8.85 -13.74

> head(round(fit.c$coef,2),3)
      Contrasts
       beta1  beta2
  [1,]  2.10  -2.20
  [2,] -1.02   8.42
  [3,] 24.89 -22.59
```