limma, Affymetrix, RMA, Independent filtering

- Journal club signup
- Some project ideas
- Affymetrix arrays + RMA → robust linear regression
- Some of the basics of mapping (mostly practical, theory comes next week)

Project ideas: Consulting/Research

1. **Long-read RNA-seq**: in-house dataset, compare a new protocol (longer fragments sequenced with MiSeq) of RNA-seq data to existing protocols using a qPCR independent truth dataset. Involves running various algorithms, comparing them to truth and to existing datasets.
2. **Male-female differential expression analysis** (RNA-seq): non-model insect for which the *de novo* assembled transcriptome is currently being built. In principle, it would be a standard differential expression analysis.
3. **Splicing changes in cancer** (RNA-seq): dataset from Unispital Zurich (~40 patients). Would augment standard differential expression analyses with recurrent splice changes. Apply 1 or 2 methods.
4. **Myeloid cell classification**. Partial re-analysis of Becher et al. 2014 http://www.ncbi.nlm.nih.gov/pubmed/25306126 (e.g. Figure 5)

Project ideas: some papers used in past years

1. http://www.ncbi.nlm.nih.gov/pubmed/23029029 -- take the raw reads, map to genome, count reads by gene, do a differential expression analysis between cell types
2. http://www.ncbi.nlm.nih.gov/pubmed/22988256 -- recreate some of the analyses (some real datasets, some simulations) that compared normalization methods
3. http://www.ncbi.nlm.nih.gov/pubmed/22965124 -- recreate some of the analyses that compared RNA-seq to microarrays
4. http://www.ncbi.nlm.nih.gov/pubmed/21824971 -- using some independent simulated RNA-seq, add a new method to an existing comparison

Affymetrix probe design

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

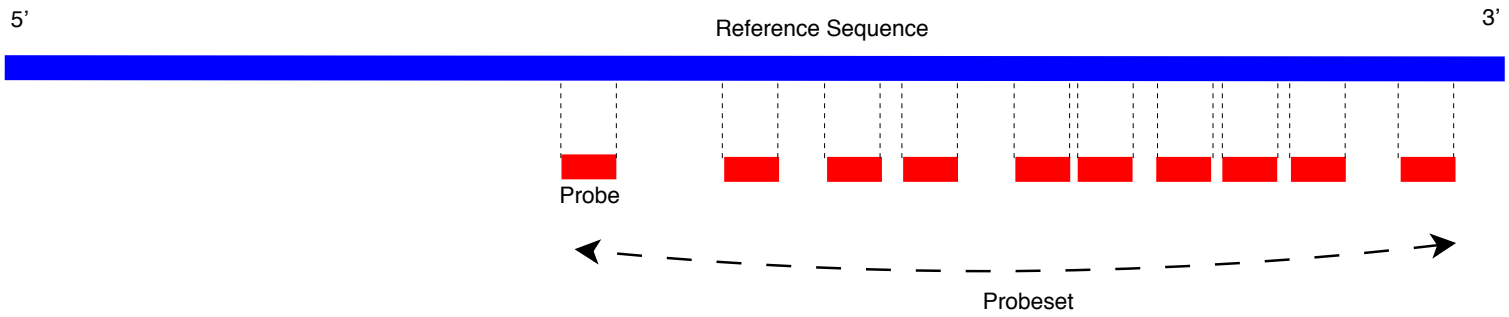


Figure 1.1: Multiple probes interrogating the sequence for a particular gene make up probesets.

Figure 1.2: Pefect Match and Mismatch Probes.

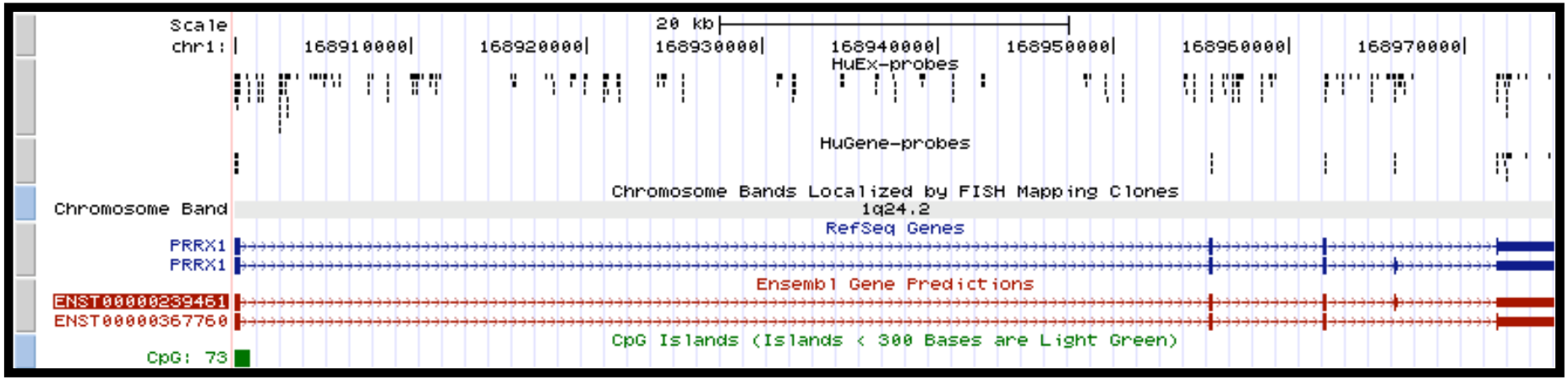
Latest Affymetrix design: “whole transcript” arrays

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

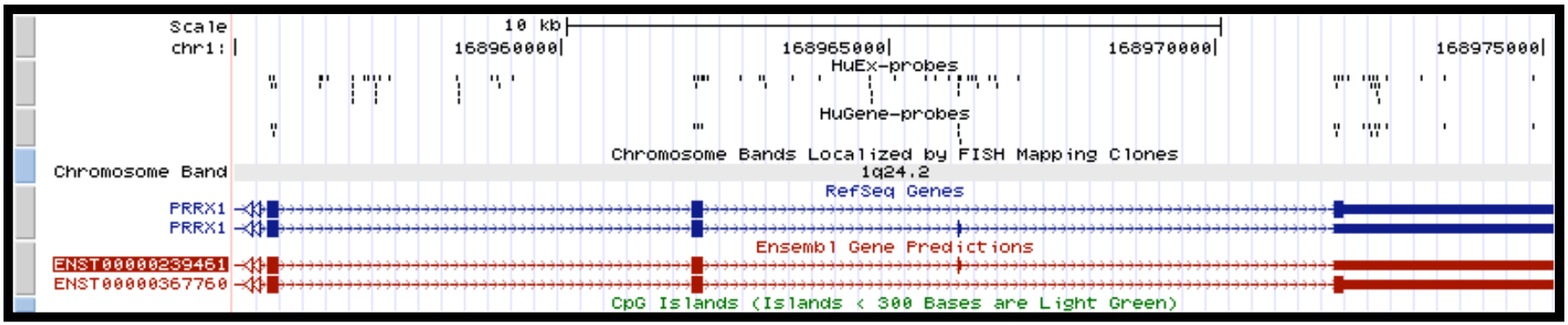
No more mismatch probes.

Reference Sequence

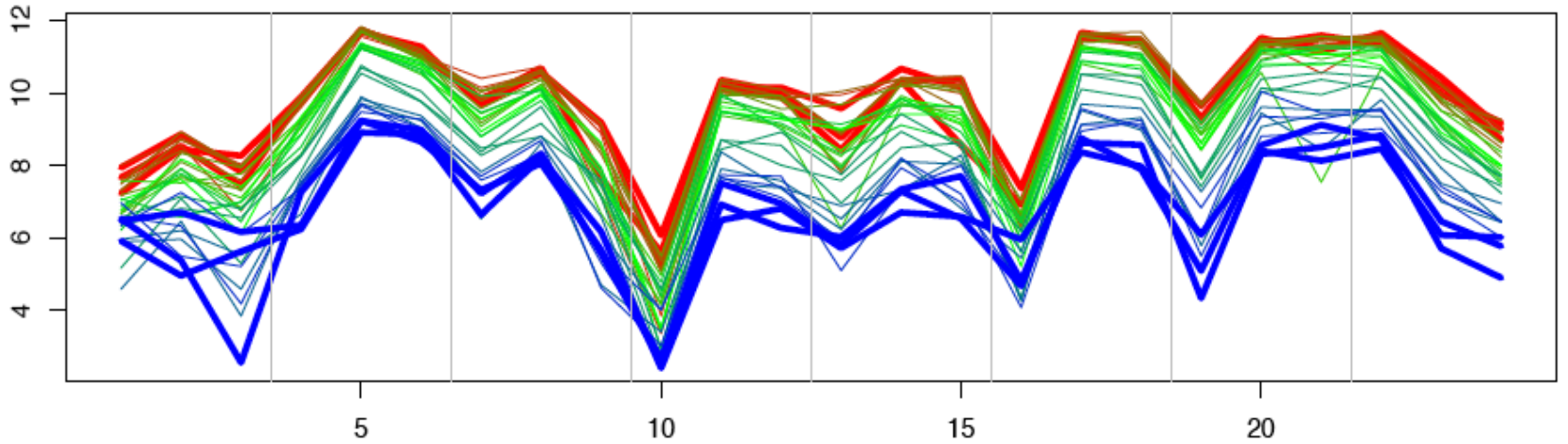
- HuExon: *Human **Exon** 1.0 ST* (~40 probes per gene, 4 probes per “exon”, annotated and predicted transcripts)
- HuGene: *Human **Gene** 1.0 ST* (~25 probes per gene, annotated genes only)
- NEW in 2013: HTA (Human Transcriptome Array): updated content + junction probes



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)

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

θ_i expression index
ϕ_j probe-specific affinity
ε_{ij} noise component

RMA, GCRMA

Robust multichip analysis (RMA)

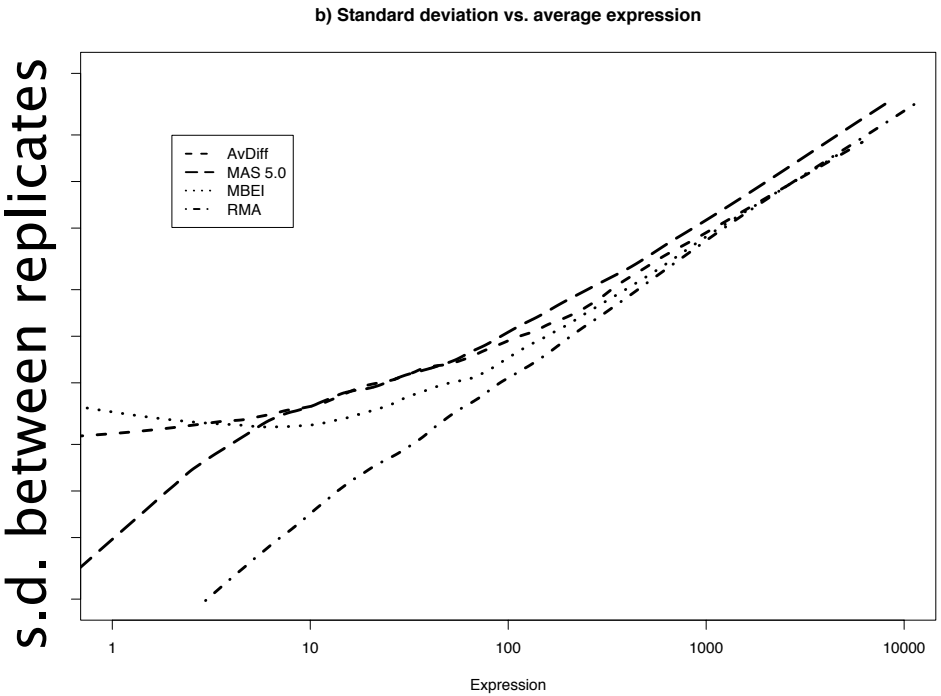
Exploration, normalization, and summaries of high density oligonucleotide array probe level data

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

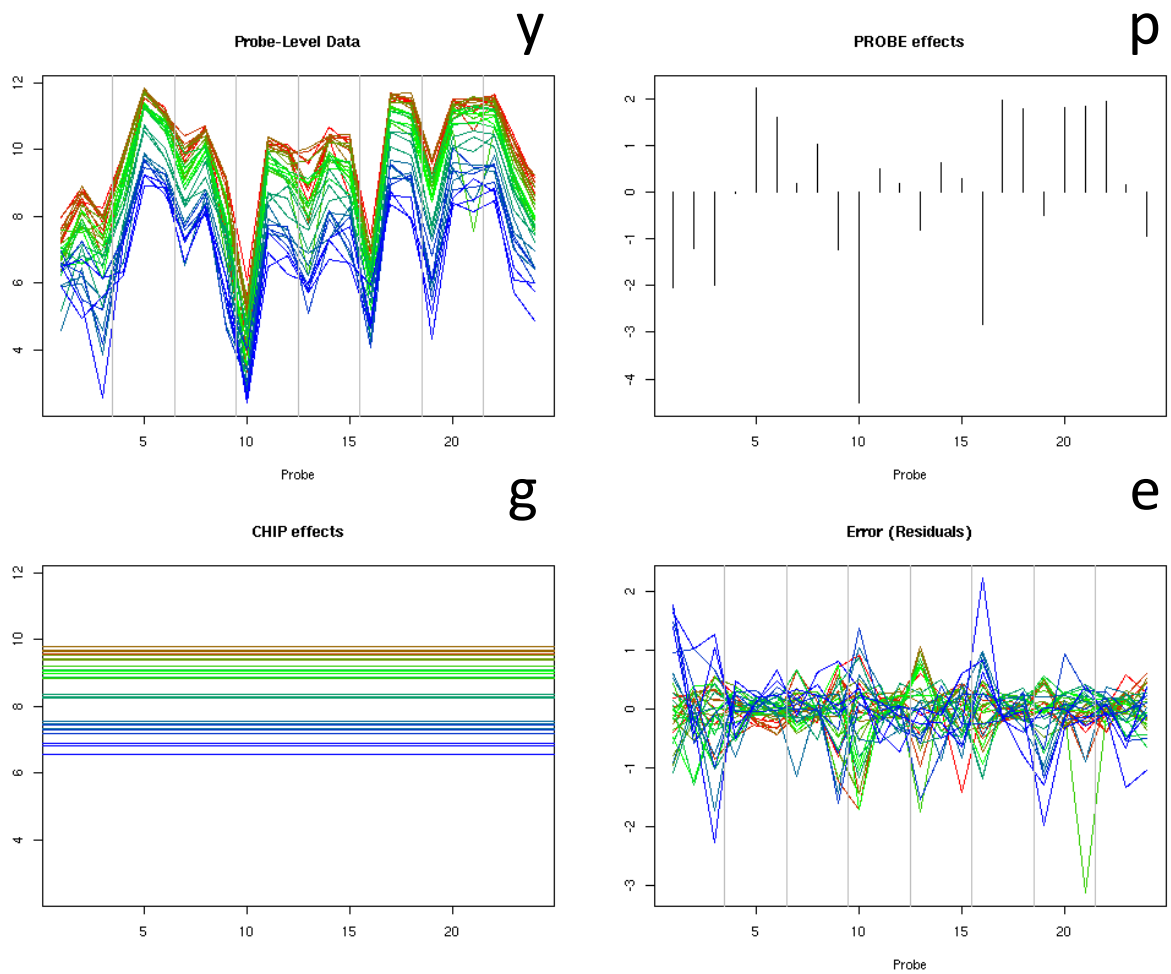
Biostatistics 2003

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:

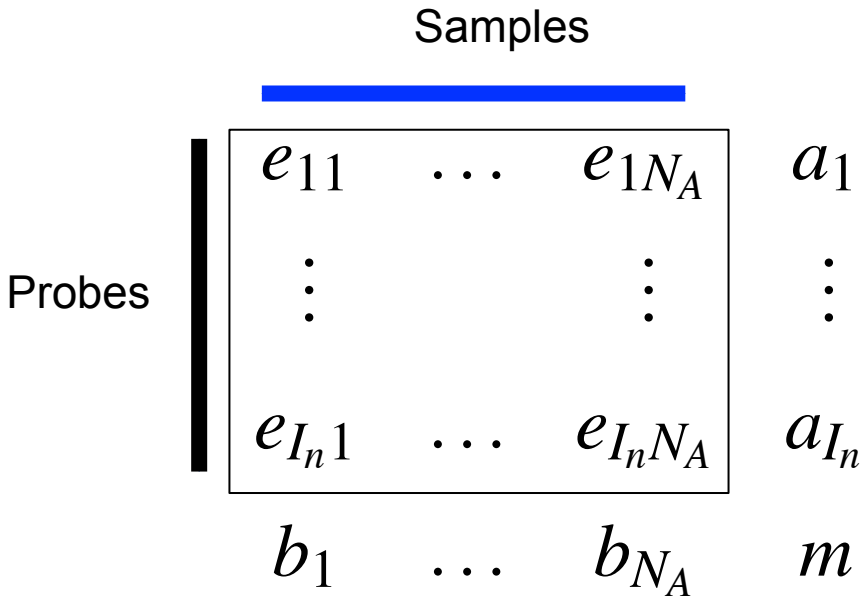
$$y_{ik} = g_i + p_k + e_{ik}$$

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

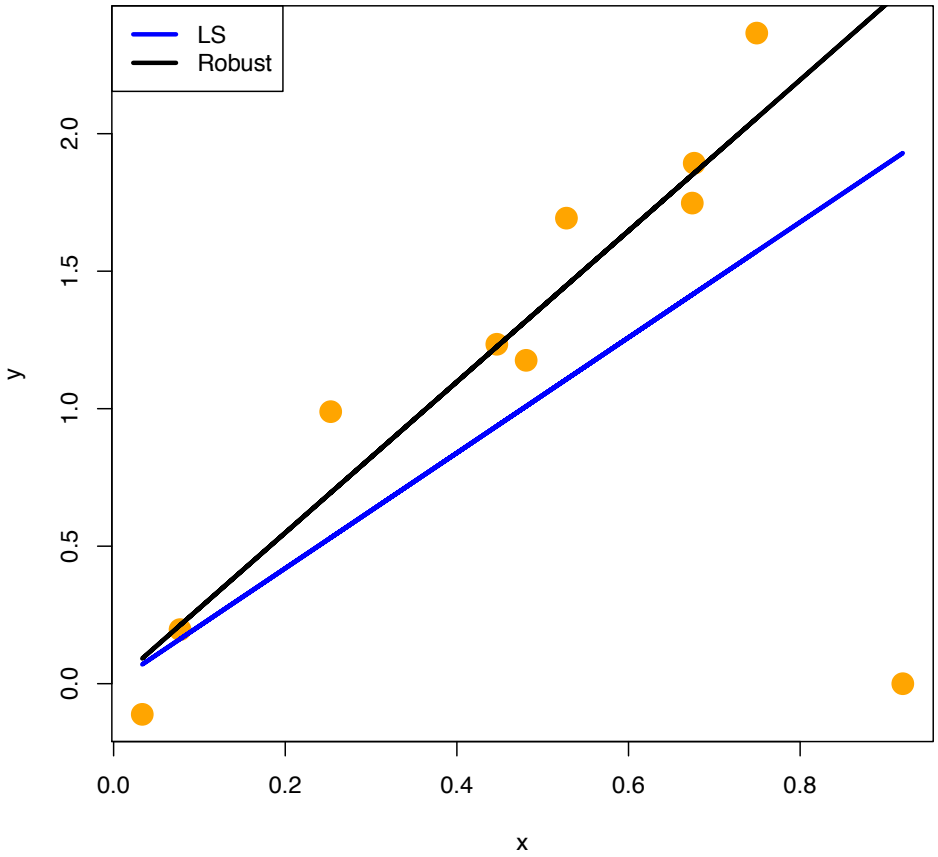



```
pe <- rnorm(11)
ce <- rnorm(8)+8
z <- outer(pe,ce,"+") +
     rnorm(length(pe)*length(ce),sd=.5)
e <- z
m <- a <- b <- 0
niter <- 3

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:

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

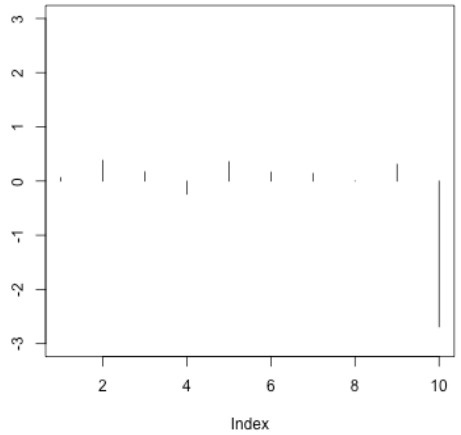
with:

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

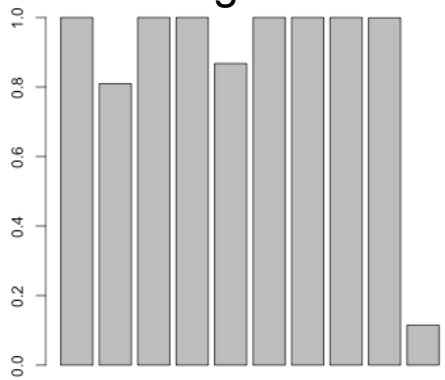
http://en.wikipedia.org/wiki/Iteratively_reweighted_least_squares

Robust regression – mechanics of iteratively reweighted least squares

Residuals



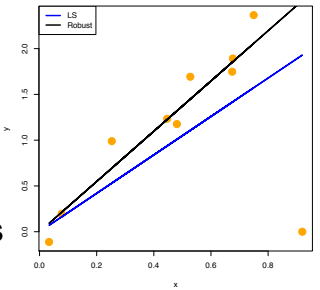
Weights



Sketch of IRLS:

Calculate initial estimates of parameters

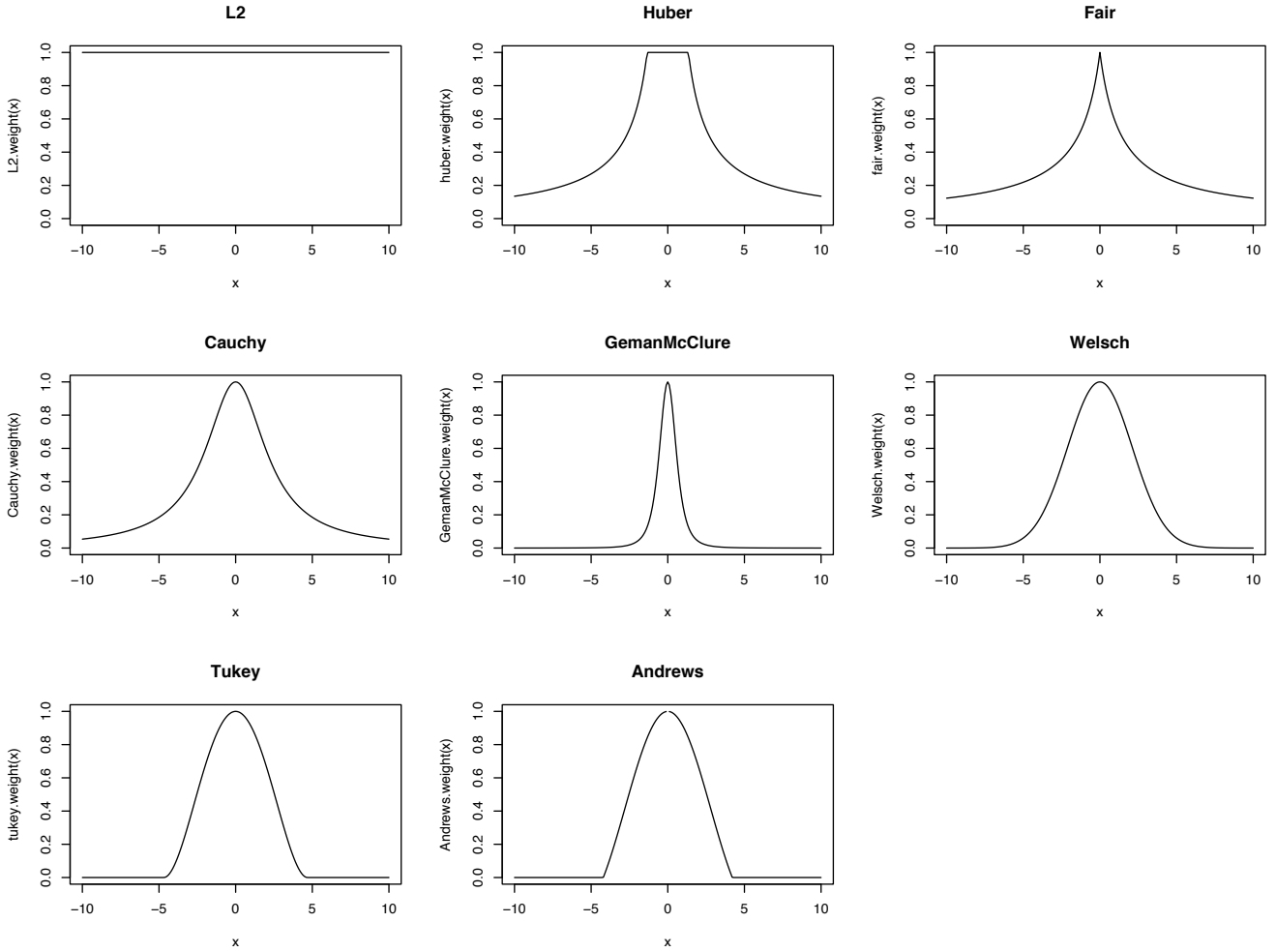
Repeat until very little change:
- Calculate residuals
- Using standardized residuals, weight observations
- Re-estimate parameters




```
# 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

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

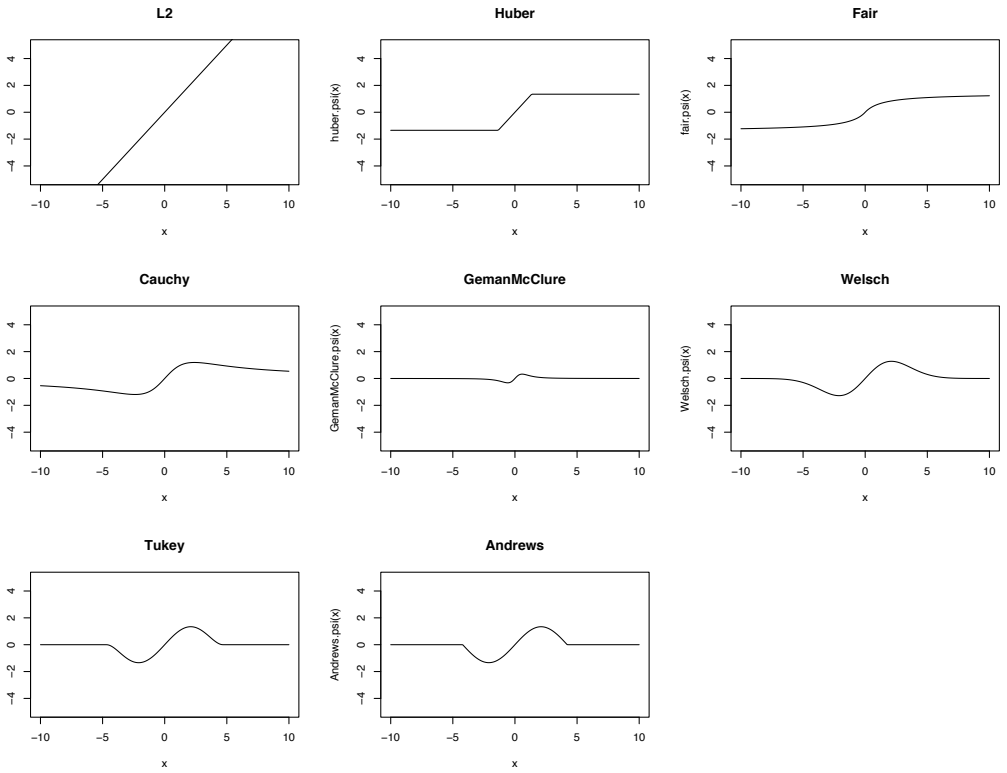
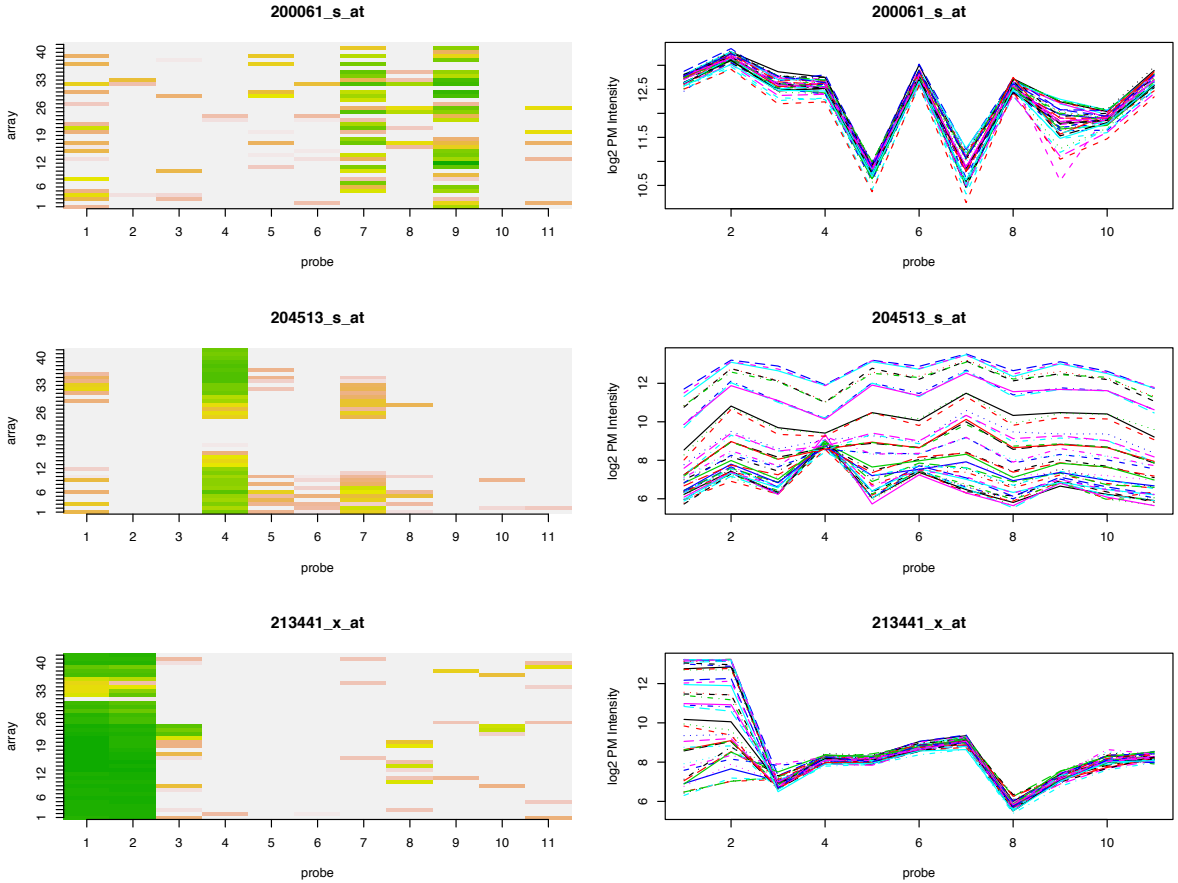


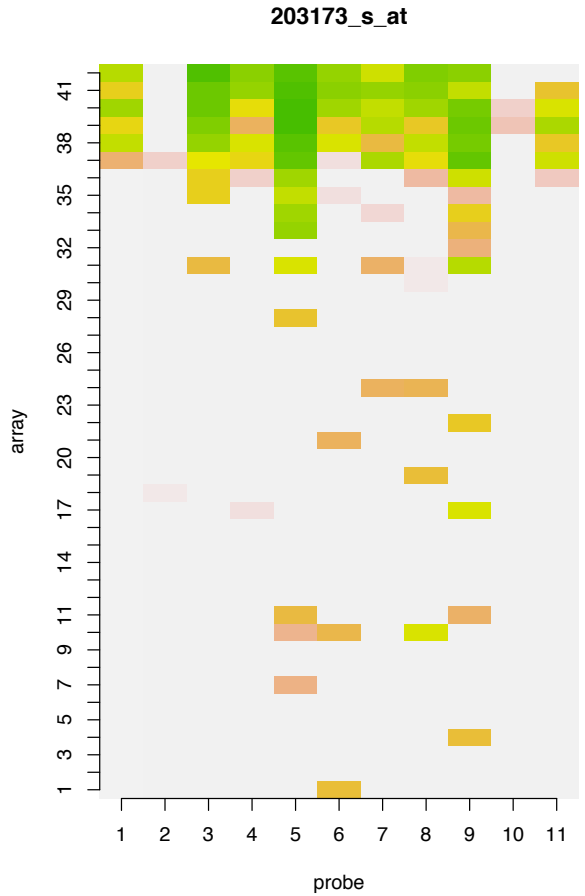
Figure 4.2: The ψ functions for some common M-estimators.

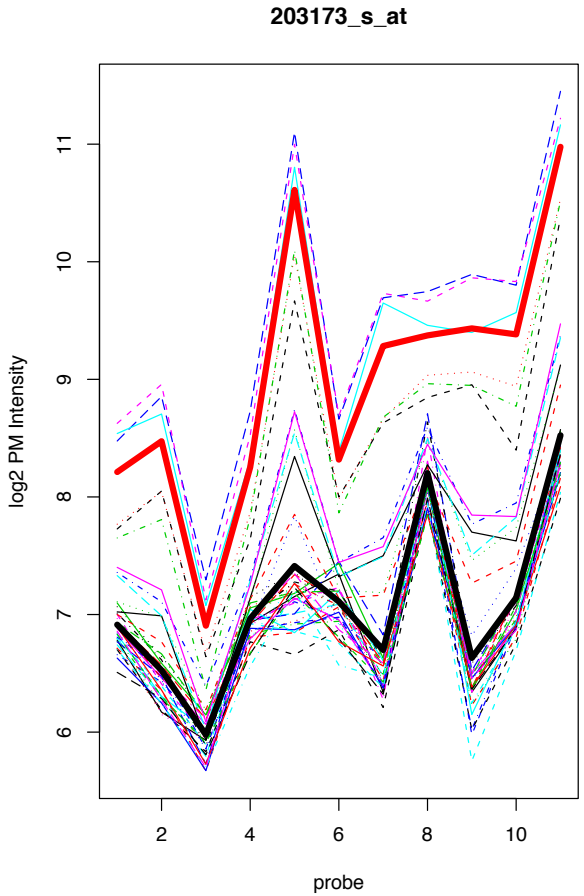
Robust regression leads to various quality assessment metrics



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
```

University of Zurich[UZH]

Institute of Molecular Life Sciences

MAPPING MILLIONS OF SHORT "READS" ONTO THE GENOME/TRANSCRIPTOME

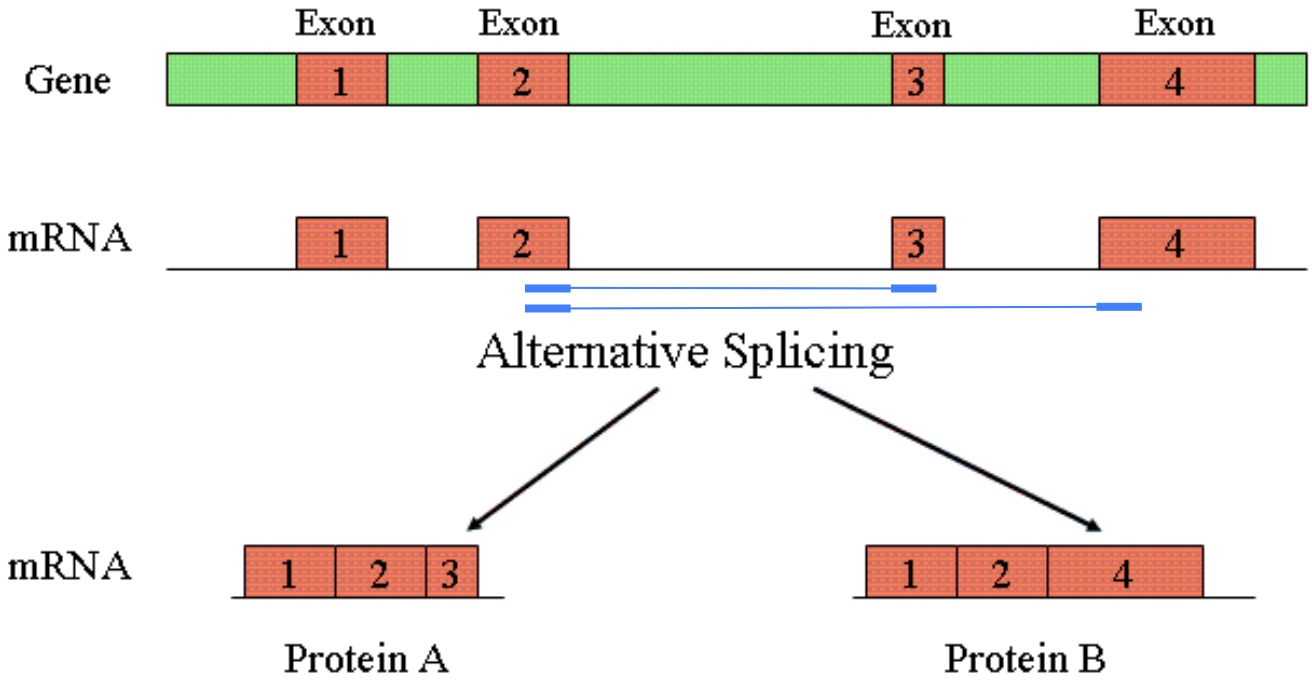
Mark D. Robinson, Statistical Genomics, IMLS

In a nutshell: Mapping reads onto genomes/ transcriptomes

- Computationally intense problem: DNA sequencing instruments produce millions-billions of short “reads” from genome (or cDNA population, etc.)
- Genomes are large: Human 3.3B characters
- Need to allow for mismatches (e.g. genetic polymorphisms, sequencing errors)
- Repetitive regions (report multiple equally good alignments?)
- BLAST – was workhorse of sequence alignment for many years (slightly different application: best substring)
- Additional challenges: RNA, paired-end reads, quality scores, bisulphite treatment, etc.

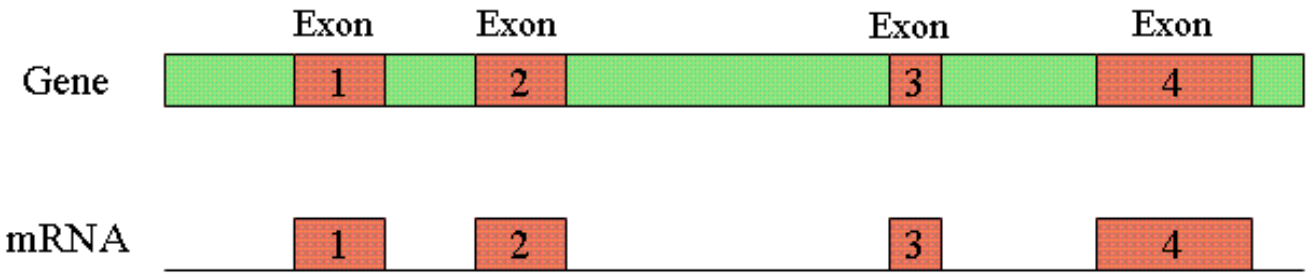
http://www.ncbi.nlm.nih.gov/Class/MLACourse/Modules/MolBioReview/alternative_splicing.html

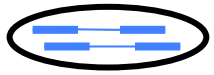
Mapping cDNA (from RNA) reads: needs a "splice-aware" aligner



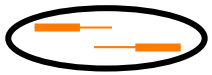
Solution: if you know where they are, add the junction sequence to the genome

Paired-end reads





Paired-end reads: Sequence both ends of DNA fragment



Single-end reads: Sequence one end of DNA fragment

Paired-end reads: costs more, but gives additional information (exact length of fragment, can map reads where one of pair hits repetitive region)

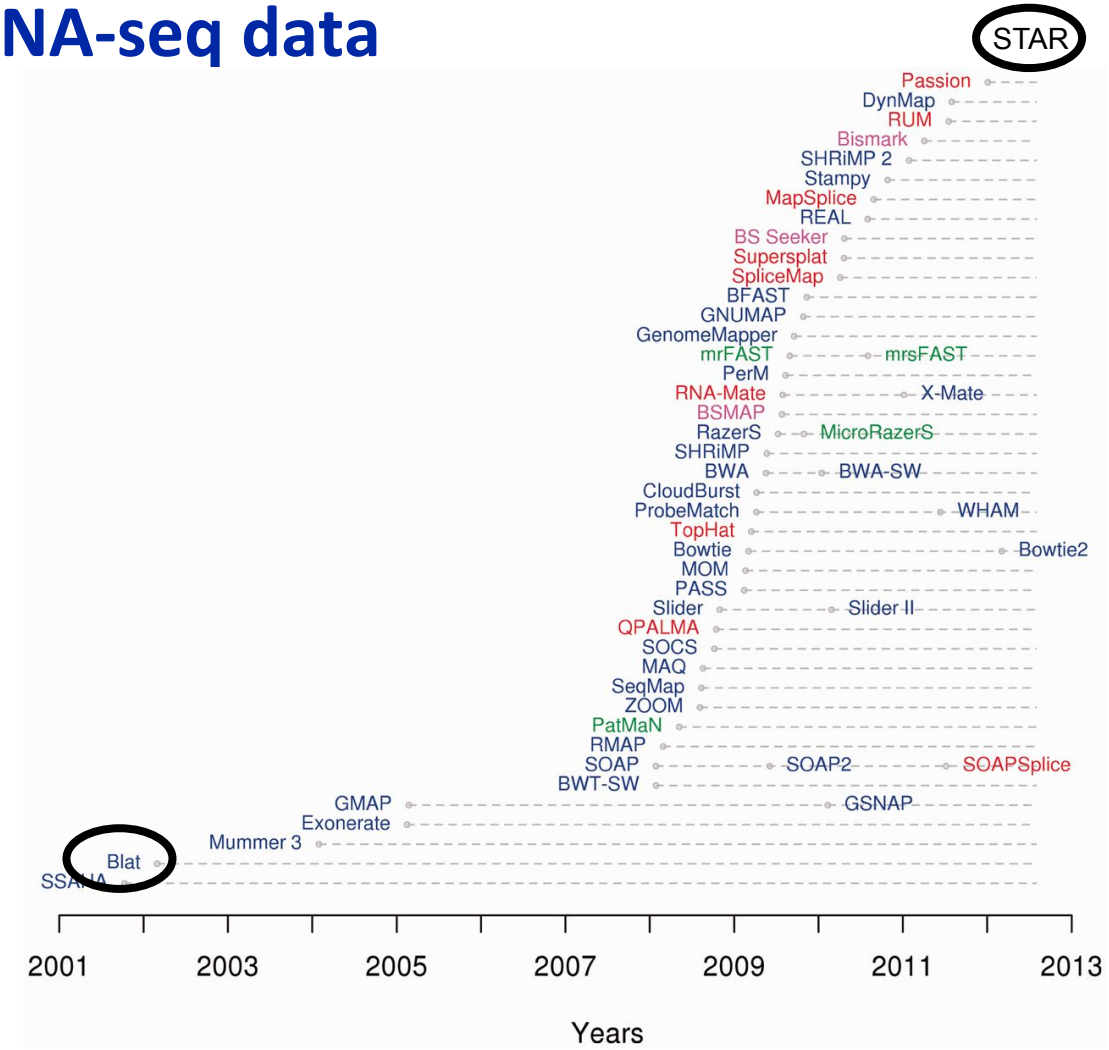
FASTA versus FASTQ formats

```
>ERR030881.107 HWI-BRUNOP16X_0001:2:1:13663:1096
ATCTTTTGTGGCTACAGTAAGTTCAATCTGAAGTCAAAACCAACCAATTT
>ERR030881.311 HWI-BRUNOP16X_0001:2:1:18330:1130
TCCATACATAGGCCTCGGGGTGGGGGAGTCAGAAGCCCCCAGACCCTGTG
```

```
@ERR030881.107 HWI-BRUNOP16X_0001:2:1:13663:1096#0/1
ATCTTTTGTGGCTACAGTAAGTTCAATCTGAAGTCAAAACCAACCAATTT
+
5.544,444344555CC?CAEF@EEFFFFFFFFFFFFFFFFEFFFEFFF
@ERR030881.311 HWI-BRUNOP16X_0001:2:1:18330:1130#0/1
TCCATACATAGGCCTCGGGGTGGGGGAGTCAGAAGCCCCCAGACCCTGTG
+
GFFFGFFBFCHHHHHHHHHHIHEEE@@@=GHGHHHHHHHHHHHHHHHHH
```

1 read

The world of mapping algorithms: focus here on RNA-seq data



Fonseca et al. 2012

University of Zurich[UZH]

Institute of Molecular Life Sciences

Exercises: Mapping reads to genomes/ transcriptomes, counting features

- Spot check mapping with BLAT
- Familiarize with the STAR pipeline → later: kallisto
- Understand the common file types
- Mapping a small example (2M bases of chr19 as the “genome”)

Mark D. Robinson, Statistical Genomics, IMLS

A small example

hs_ch19_subset.fa: Sequence from human chr19 from 2,000,000-4,000,000

→ imagine a genome that is only 2M bases long

galaxy_small.fastq: subset of reads from a real dataset

chr19_rescaled.gff: Table of annotation

→ Tell the aligner where the junctions should be

Pipeline + Filetypes

FASTA/FASTQ → SAM/BAM → "count" table

raw reads	aligned reads	matrix

Annotation is stored in:
GTF/GFF1/GFF2/GFF3

http://www.broadinstitute.org/igv/GFF

Getting help / info

STAR (RNA-seq):

https://code.google.com/p/rna-star/

IGV:

http://www.broadinstitute.org/software/igv/home

For DNA: bowtie2, bwa mem, …

IGV


chr19
11 kb
1,004 kb
1,006 kb
1,008 kb
1,010 kb
1,012 kb
1,014 kb
tophat2_mapped_s.bam Coverag
[0 - 3223]
tophat2_mapped_s.bam
chr19_rescaled.gff3
"NM_198969"
"NM_198970"
"NM_001130"