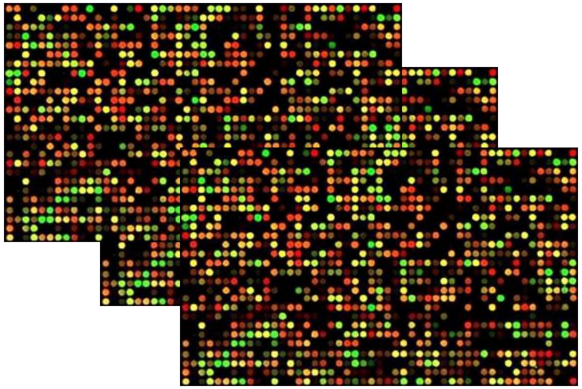
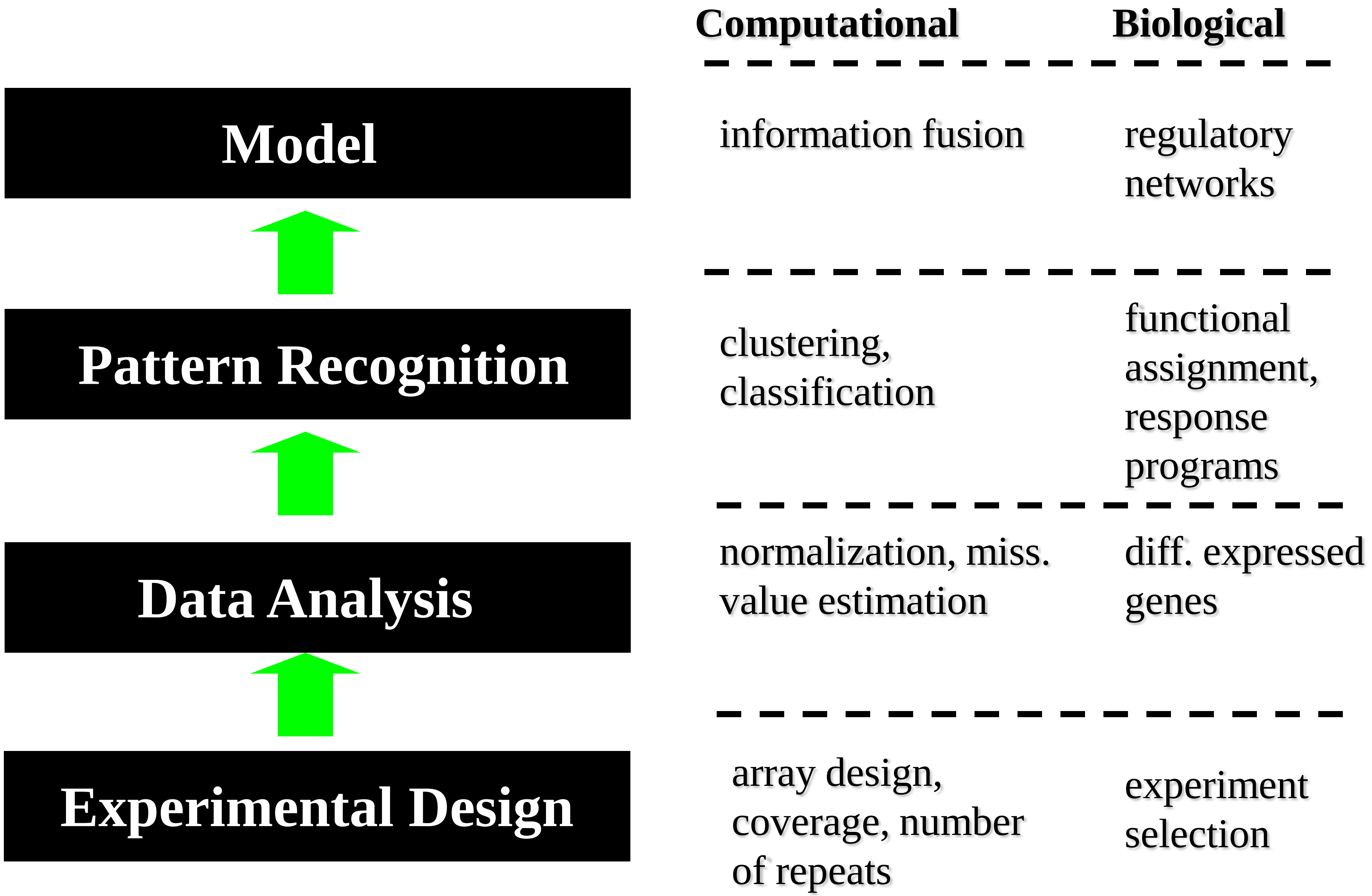
10-810 /02-710

Computational Genomics

Normalization



Gene Expression Analysis



	Computational	Biological
Model	information fusion	regulatory networks
Pattern Recognition	clustering, classification	functional assignment, response programs
Data Analysis	normalization, miss. value estimation	diff. expressed genes
Experimental Design	array design, coverage, number of repeats	experiment selection

Experiment design

A number of computational issues should be addressed:

- Which technology to use?
- Determining the number of replicates for each sample
- Sampling rates for time series experiments

Data analysis

- Alignment, read assignments
    - Discussed in NGS lecture
- Normalization
- Combining results from replicates
- Identifying differentially expressed genes
- Dealing with missing values
- Static vs. time series

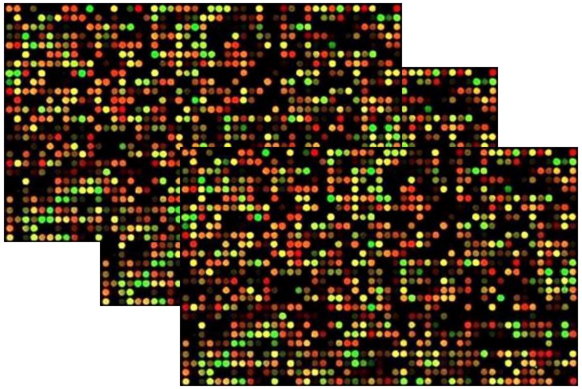
Data analysis

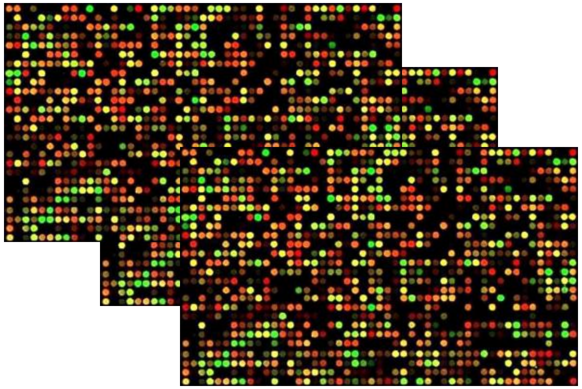
- Normalization
- Combining results from replicates
- Identifying differentially expressed genes
- Dealing with missing values
- Static vs. time series

Typical experiment: replicates

healthy

cancer





Technical replicates: same sample using multiple arrays / Seq

Clinical replicates: samples from different individuals

Many experiments have all both kinds of replicates

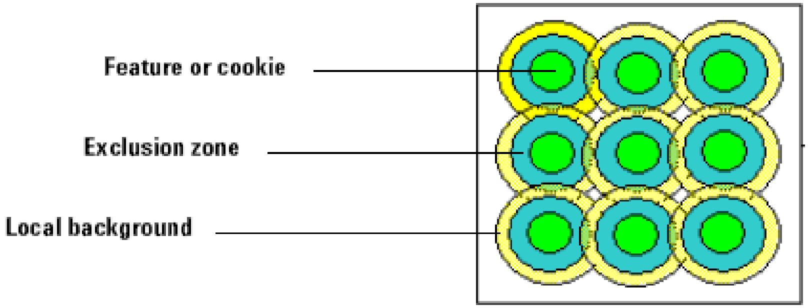
Normalization

- – Background Correction
- – Probe Set Summarization (Affymetrix)
- – Between Array Normalization – “Normalization”

Background Correction

- On Affymetrix major normalization software (e.g. MAS 5, dChip, RMA, gcRMA) use different approaches for background correction
- On Agilent the feature extraction software gives various options on ways to do background correction

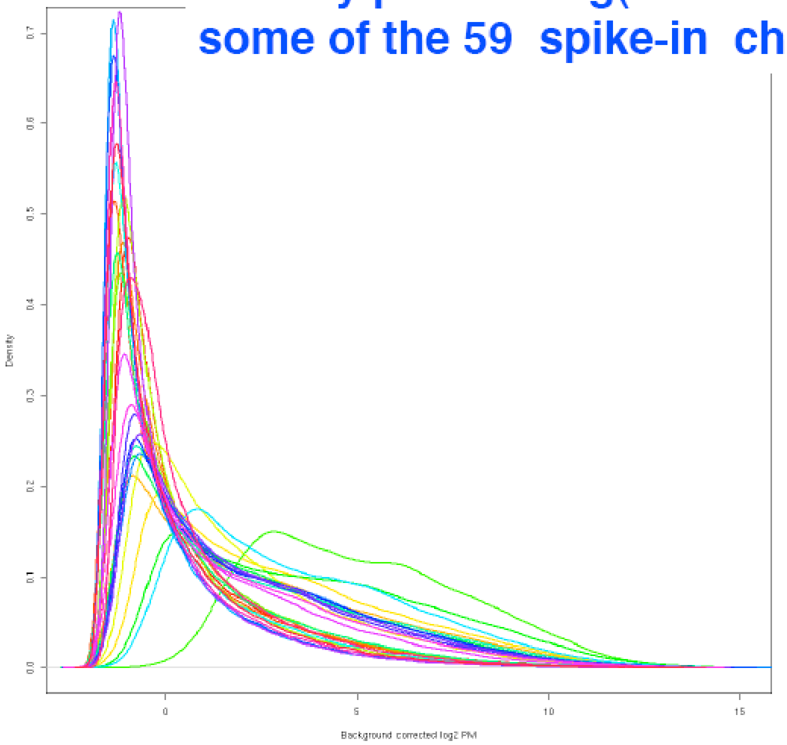
Image from Agilent User Manual



Normalization

- Background Correction
- Probe Set Summarization (Affymetrix)
- Between Array Normalization – “Normalization”

In theory these really should have the same distribution…

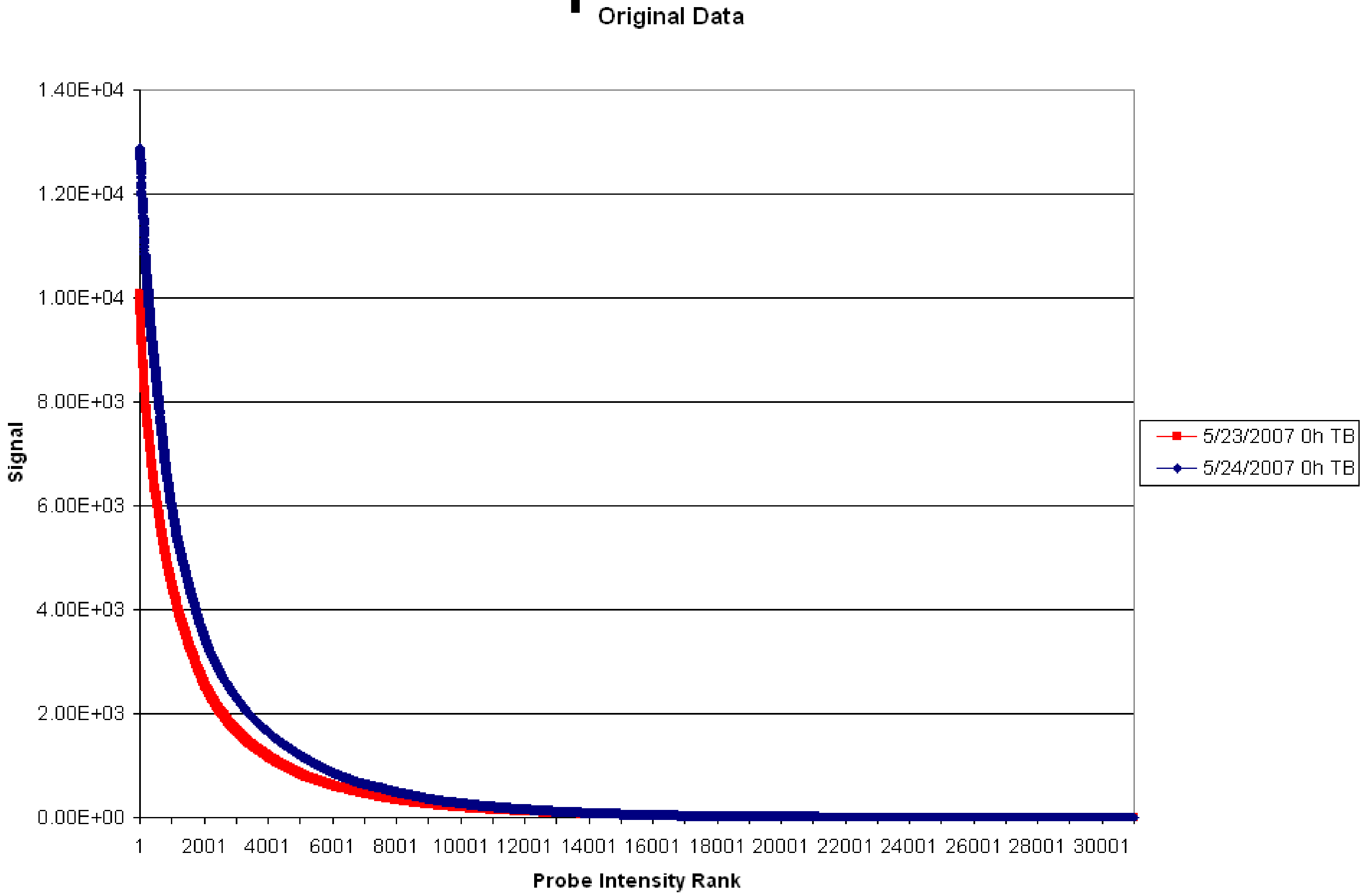


Density plots of log(PM-*BG) for some of the 59 spike-in chips

Image from Terry Speed's Slides

Ideally normalization should remove variation between microarray unrelated to the biology of interest, without removing the variation due to the biology of interest

Two repeats from Agilent mouse expression data



Normalization: Assumptions

- In order to normalize between arrays we need to make certain assumptions. Methods we will discuss will assume (with decreasing order of assumption strength):

  1. Values are *exactly* the same between the arrays (though different genes may be assigned different values in each array).

  2. Values are normally distributed with the same mean and variance across arrays.

  3. Some of the genes do not change between arrays and thus should have relatively similar values (rank invariant).

Three Major Approaches to Between Array Normalization

- Quantile Normalization → 1. Values are *exactly* the same between arrays (though different genes may be assigned different values in each array).

- Scale Factor Normalization → 2. Values are normally distributed with the same mean and variance across arrays.

- Invariant Set Normalization → 3. Some of the genes do not change between arrays and thus should have relatively similar values (rank invariant).

Three Major Approaches to Between Array Normalization

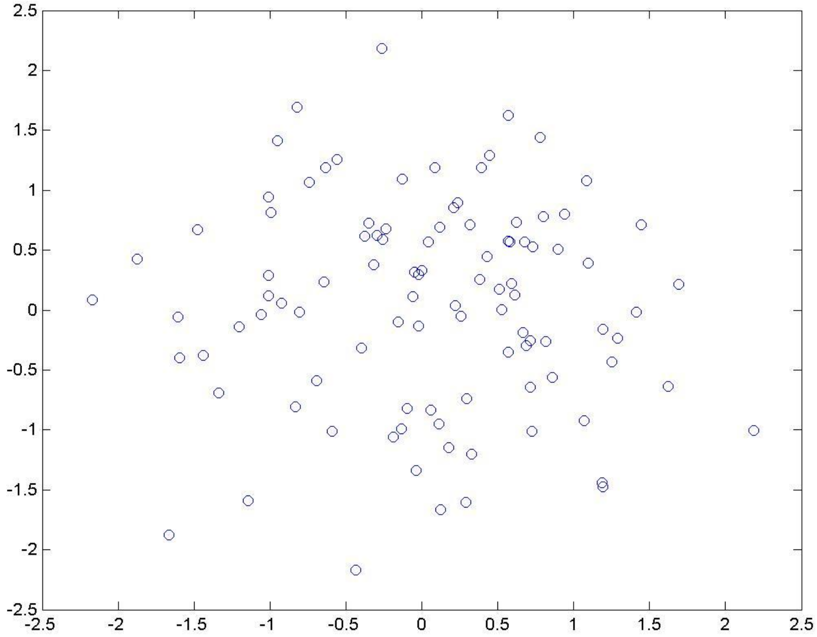
- Quantile Normalization → 1. Values are *exactly* the same between arrays (though different genes may be assigned different values in each array).

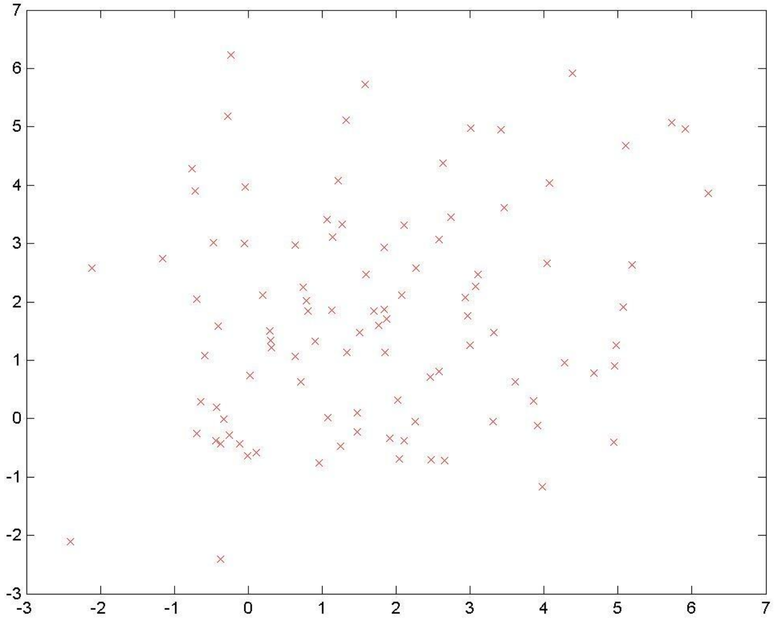
- Scale Factor Normalization → 2. Values are normally distributed with the same mean and variance across arrays.

- Invariant Set Normalization → 3. Some of the genes do not change between arrays and thus should have relatively similar values (rank invariant).

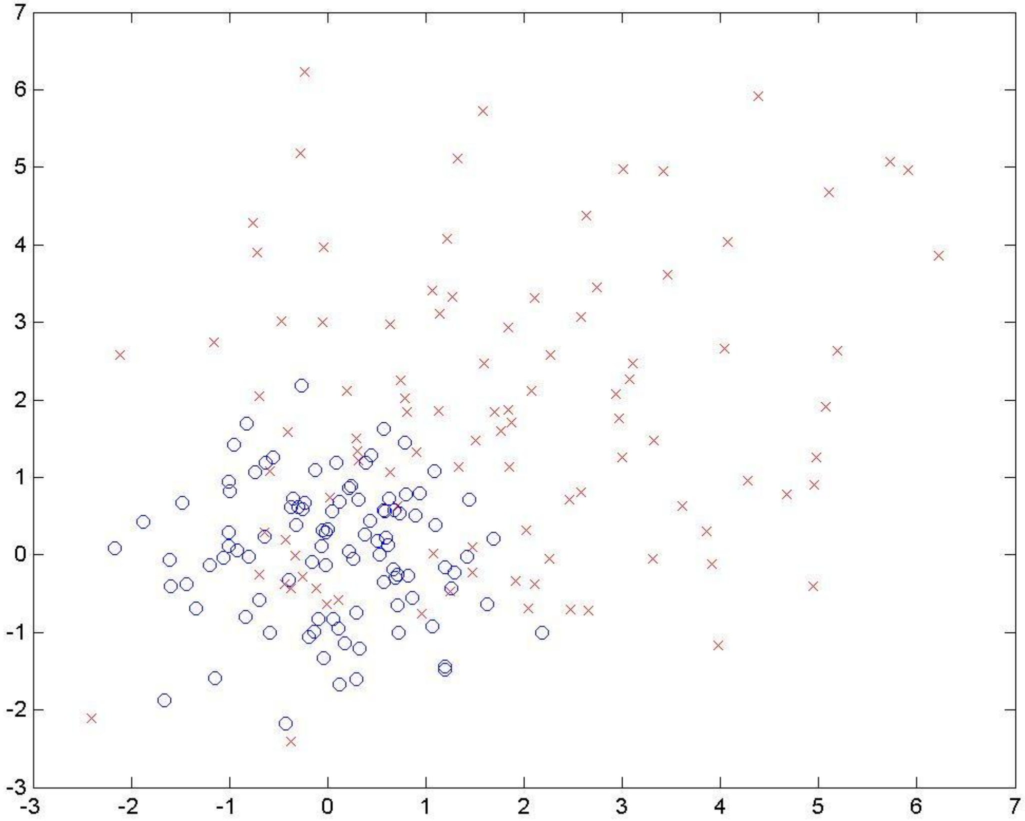
Normalizing across arrays

- Consider the following two sets of values:





Lets put them together …



Normalizing between arrays

The first step in the analysis of microarray data in a given experiment is to normalize *between* the different arrays.

- Simple assumption: mRNA quantity is the same for all arrays

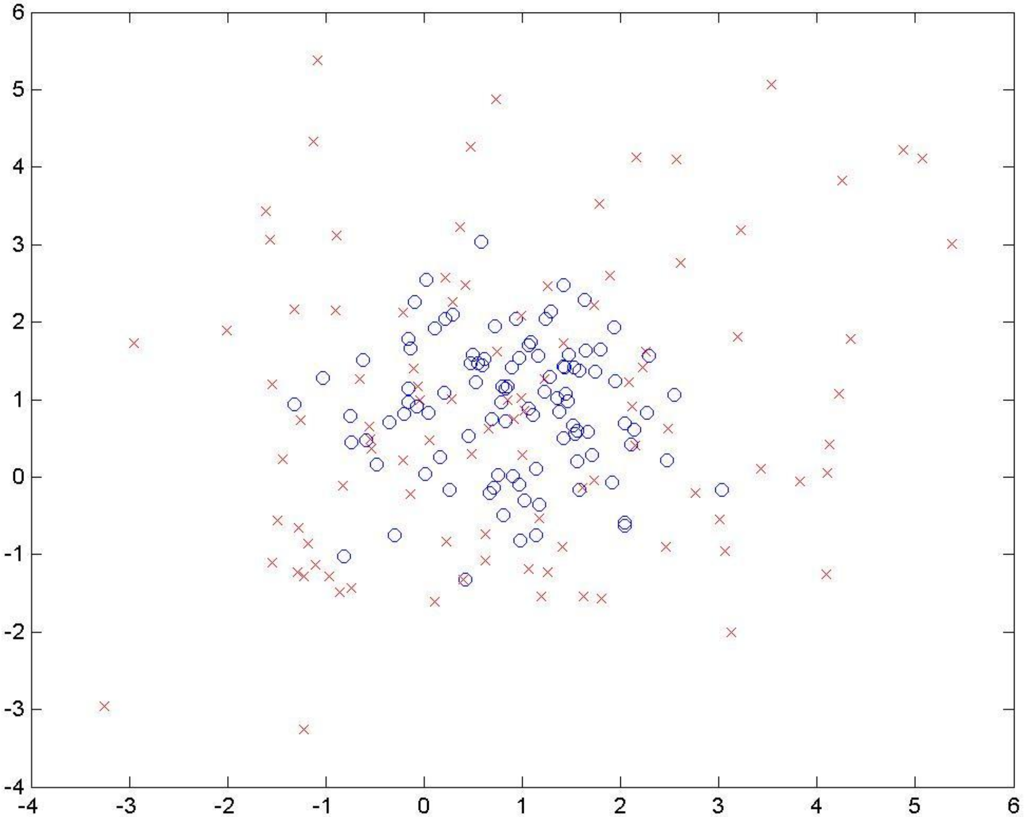
$$M^{j} = \frac{1}{n}\sum_{i=1}^{n} y_{i}^{j} \qquad M = \frac{1}{T}\sum_{j=1}^{T} M^{j}$$

- Where *n* and *T* are the total number of genes and arrays, respectfully. M^j is known as the ***sample*** mean

- Next we transform each value to make all arrays have the same mean:

$$\hat{y}_{i}^{j} = y_{i}^{j} - M^{j} + M$$

Normalizing the mean



Variance normalization

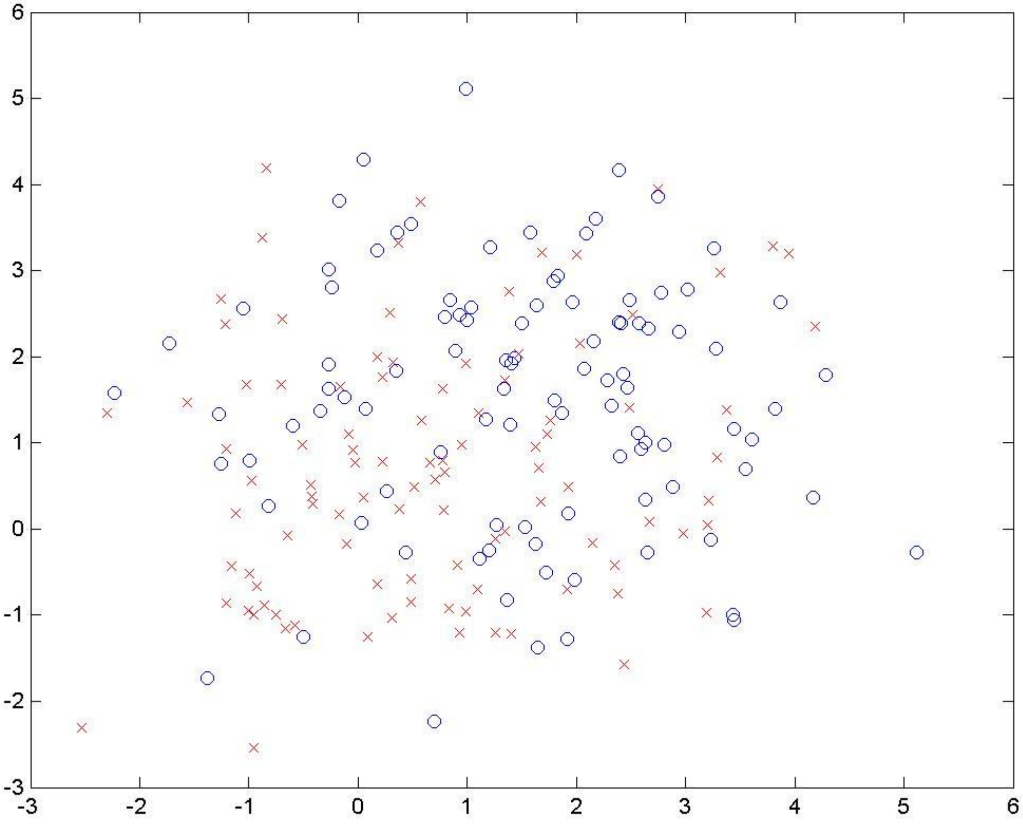
- In many cases normalizing the mean is not enough.
- We may further assume that the *variance* should be the same for each array
- Implicitly we assume that the expression distribution is the same for all arrays (though different genes may change in each of the arrays)

$$V^j = \frac{1}{n}\sum_{i=1}^{n}(y_i^j - M^j)^2 \quad V = \frac{1}{T}\sum_{j=1}^{T} V^j$$

- Here *V^j* is the sample variance.
- Next, we transform each value as follows:

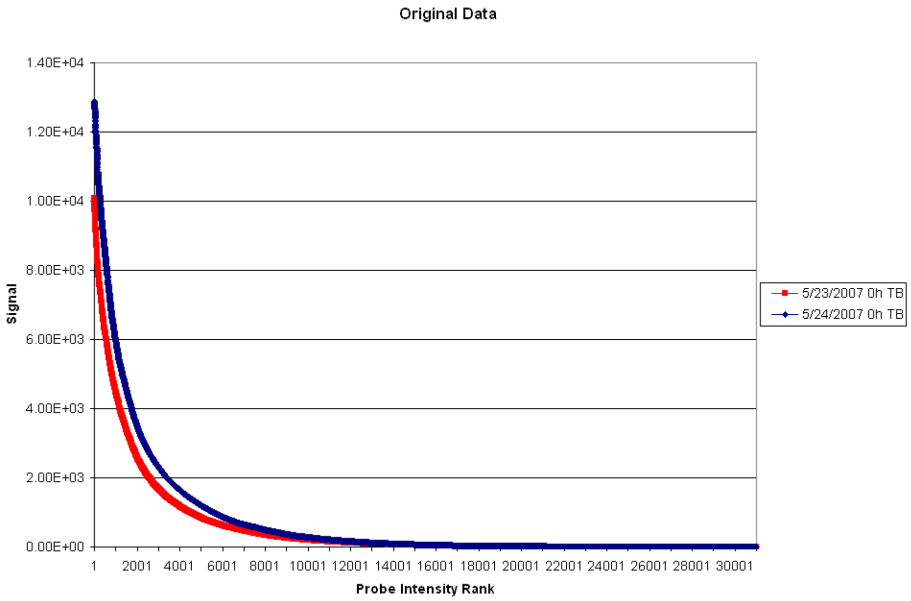
$$\hat{y}_i^j = \frac{\left(y_i^j - M^j\right)\sqrt{V}}{\sqrt{V^j}} + M$$

Normalizing mean and variance

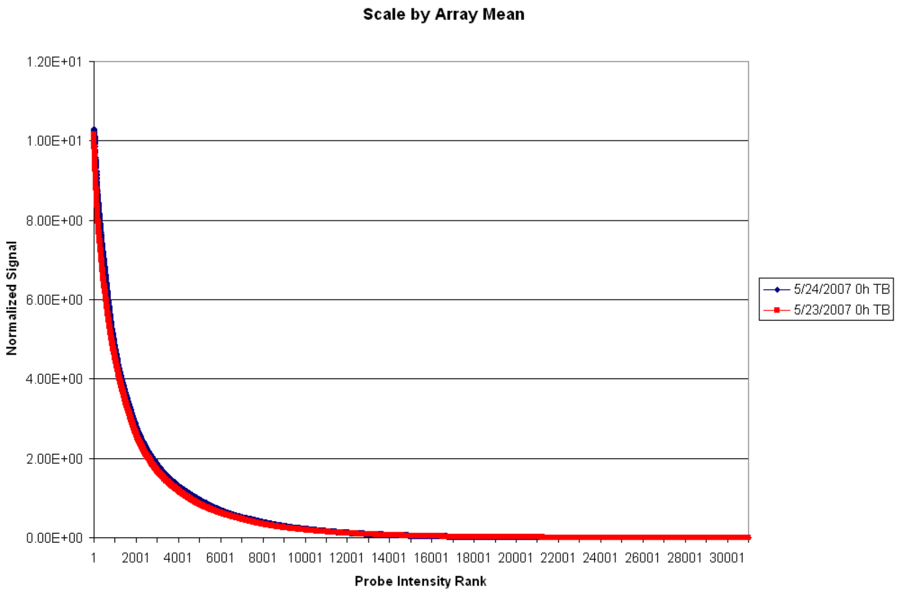


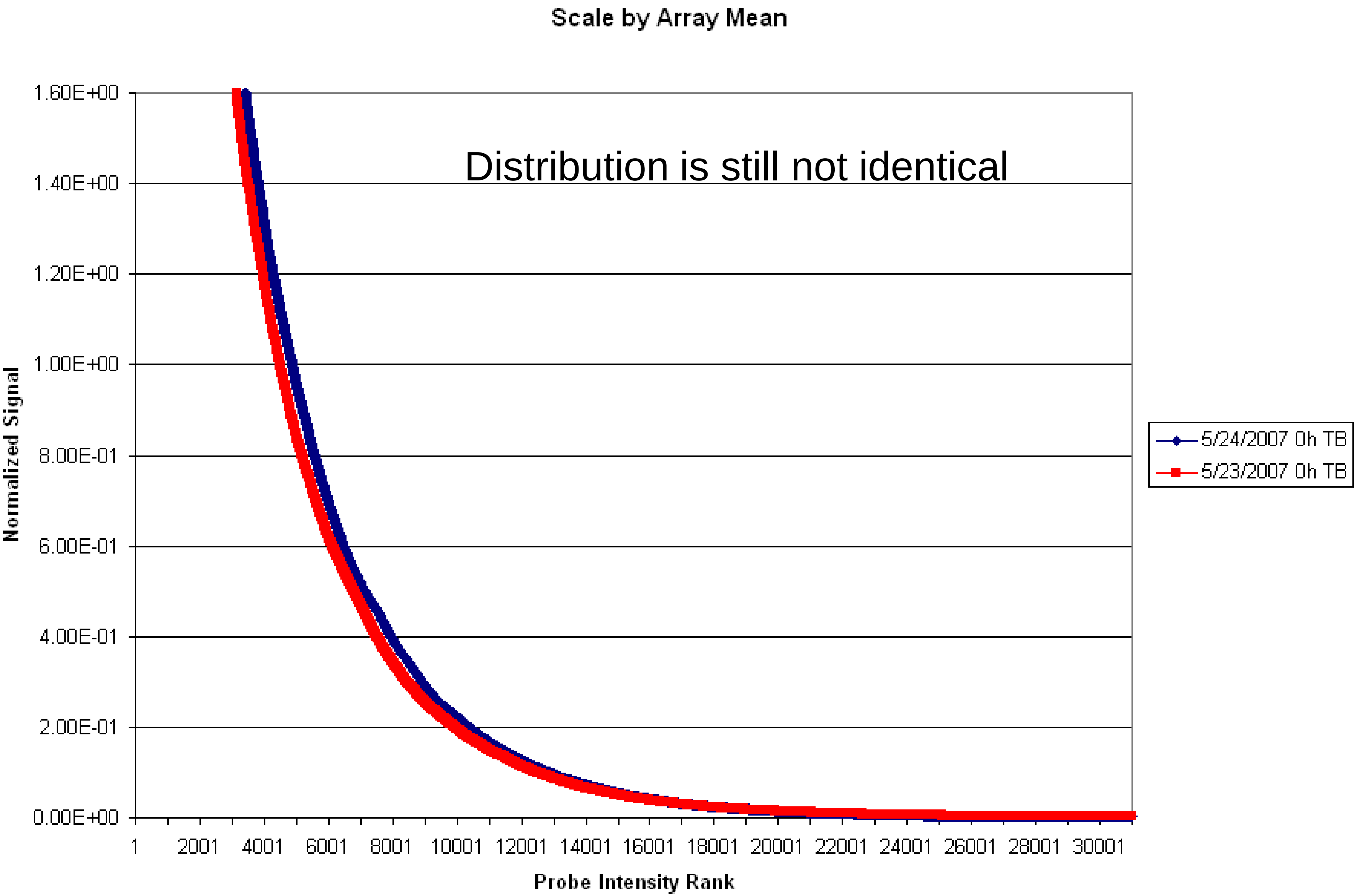
Mean Scaling

Before



After





Distribution is still not identical

Three Major Approaches to Between Array Normalization

- Quantile Normalization → 1. Values are *exactly* the same between arrays (though different genes may be assigned different values in each array).

- Scale Factor Normalization → 2. Values are normally distributed with the same mean and variance across arrays.

- Invariant Set Normalization → 3. Some of the genes do not change between arrays and thus should have relatively similar values (rank invariant).

Quantile Normalization

- Maps the distribution of expression values for every microarray experiment to the same target distribution
- This is the approach of Terry Speed’s group’s RMA software
  - Also an option in Agilent’s Gene Spring Software

Normalized distribution

- For each order all the values in the array
- Take the mean (or median) value in each position

A	B	C
3	8	5
6	4	8
7	2	2
2	9	1

Sort

A	B	C
2	2	1
3	4	2
6	8	5
7	9	8

Mean

Normalized
1.67
3
6.33
8

Three Major Approaches to Between Array Normalization

- Quantile Normalization → 1. Values are *exactly* the same between arrays (though different genes may be assigned different values in each array).

- Scale Factor Normalization → 2. Values are normally distributed with the same mean and variance across arrays.

- Invariant Set Normalization → 3. Some of the genes do not change between arrays and thus should have relatively similar values (rank invariant).

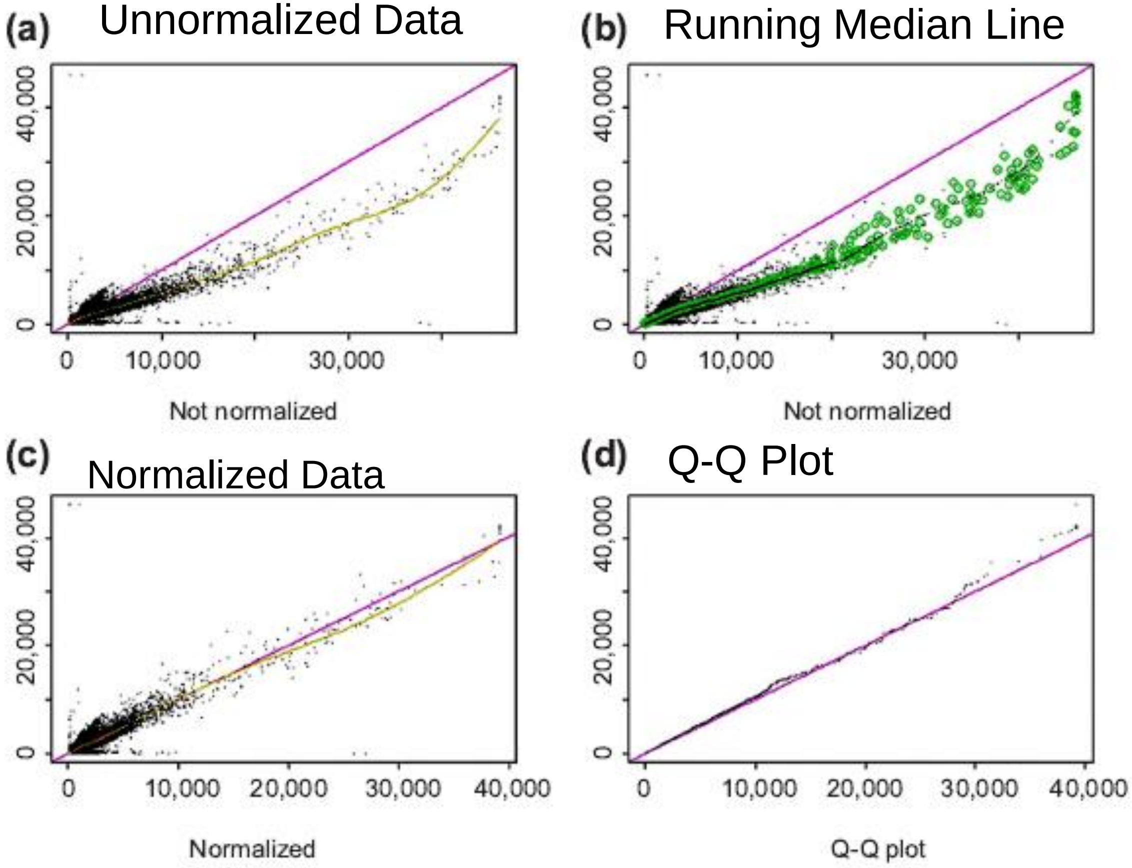
Invariant Set Normalization

- Normalization based on probes that do not change between data
  - Housekeeping Genes
  - Control Probes
  - Rank invariant genes
    - This is the approach of Wing Wong’s dChip software

dChIP Invariant Set Normalization

- Select one baseline array
  - Default Median Intensity
- All other arrays are normalized against baseline
- Computationally finds rank invariant probes
  - Different genes may be used for different arrays
- Running Median curve through rank invariant probes becomes the new “y=x” curve, all values adjust accordingly

dChIP on Replicate Data



Spline on All Unnormalized Data

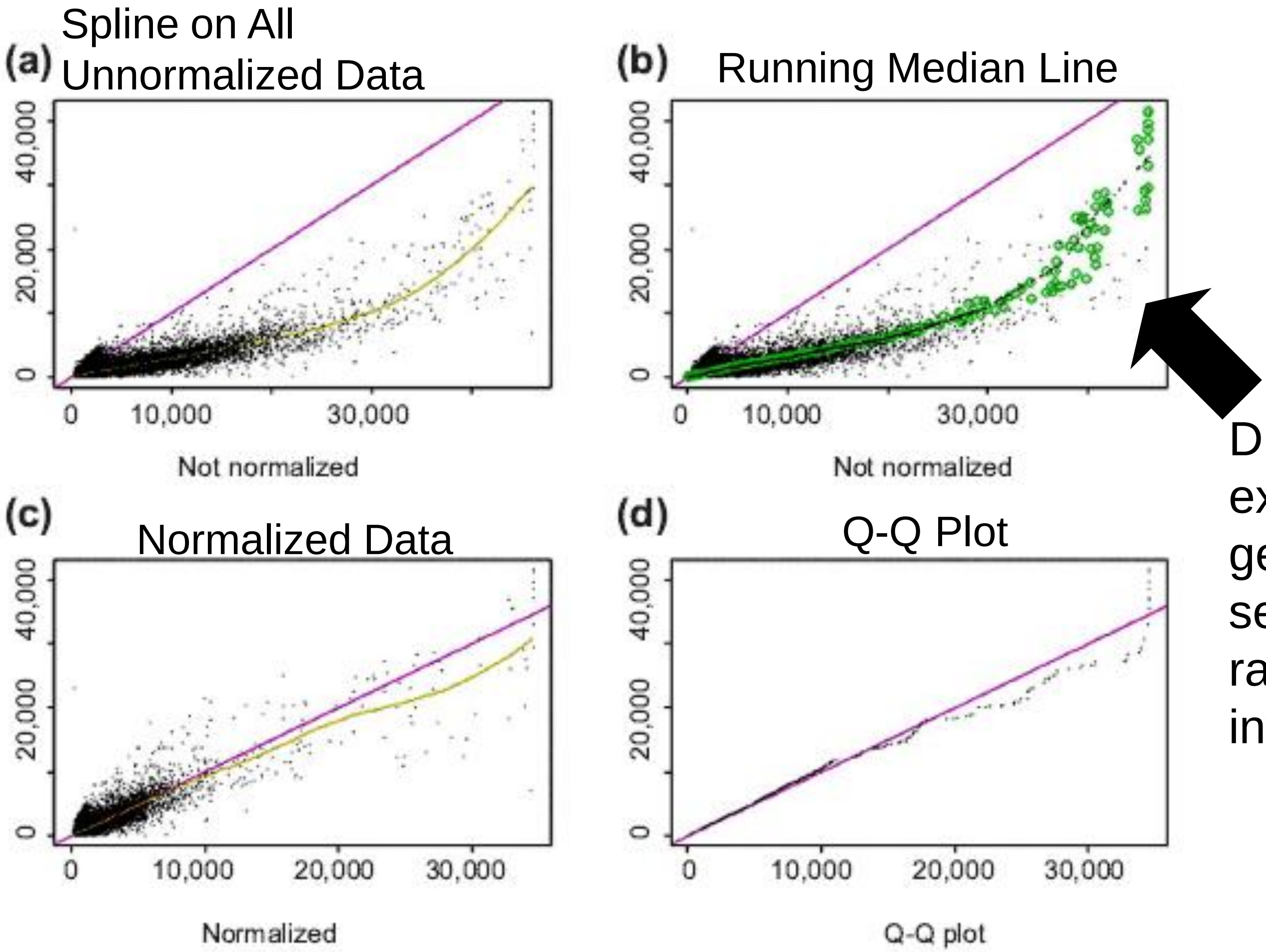
Running Median Line

Normalized Data

Q-Q Plot

Image From Li and Wong, Genome Biology 2001

dChIP on data with true differentially expressed genes



Spline on All Unnormalized Data

Running Median Line

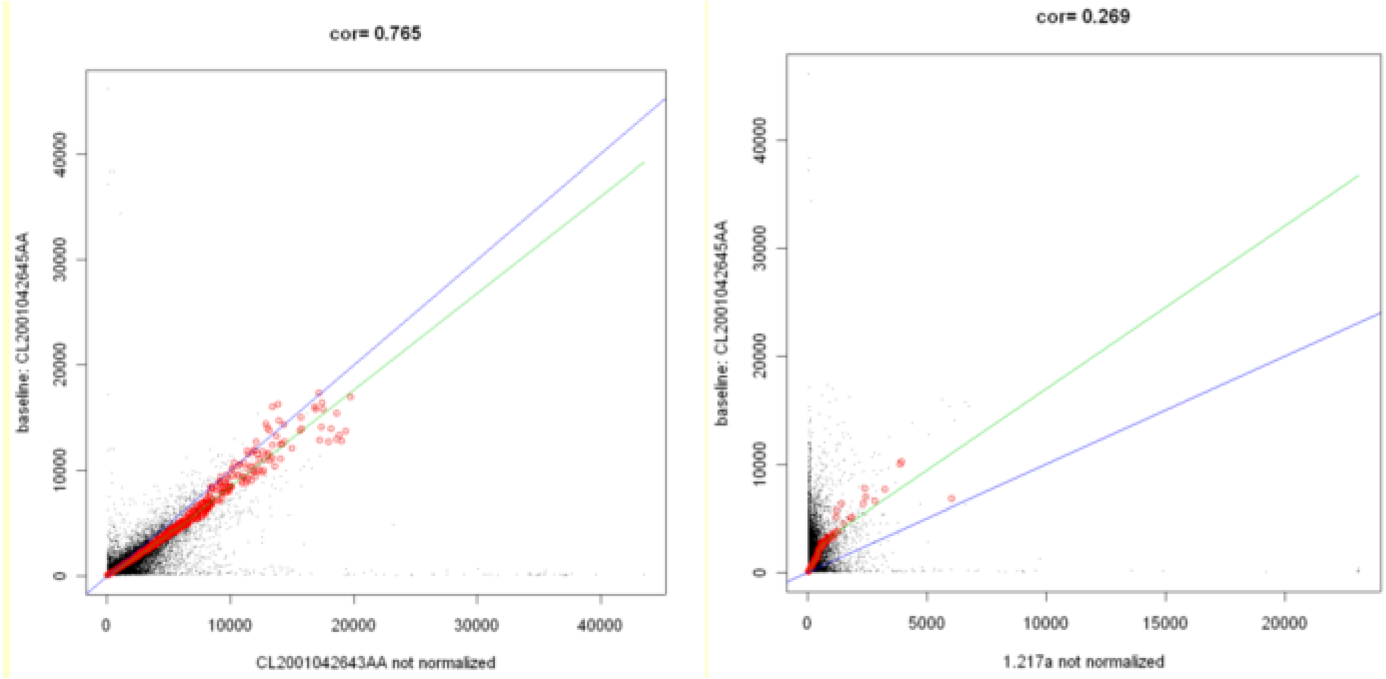
Normalized Data

Q-Q Plot

Differently expressed genes not selected as rank invariant

Image From Li and Wong, Genome Biology 2001

Normalizing Across Very Different Distributions can be Problematic



May want to normalize subsets from different tissues separately

Image From dChIP manual

RNA-Seq

- The techniques discussed above were developed for microarrays, but can also be used for Seq experiments.
- However, unlike array data which is based on image analysis / quantification, Seq data is discrete so other types of (non Gaussian) models may be more appropriate for this data.

Simple RPKM Normalization

- Proportion of reads: number of reads (n) mapping to an exon (gene) divided by the total number of reads (N), n/N.

- RPKM: Reads Per Kilobase of exon (gene) per Million mapped sequence reads, $10^9n/(NL)$,

  where L is the length of the transcriptional unit in bp (Mortazavi et al., *Nat. Meth.*, 2008).

Reads per Kilobase per Million mapped Reads (RPKM)

R = reads mapped to gene G

L =length of gene G

M=number of mapped reads obtained in experiment

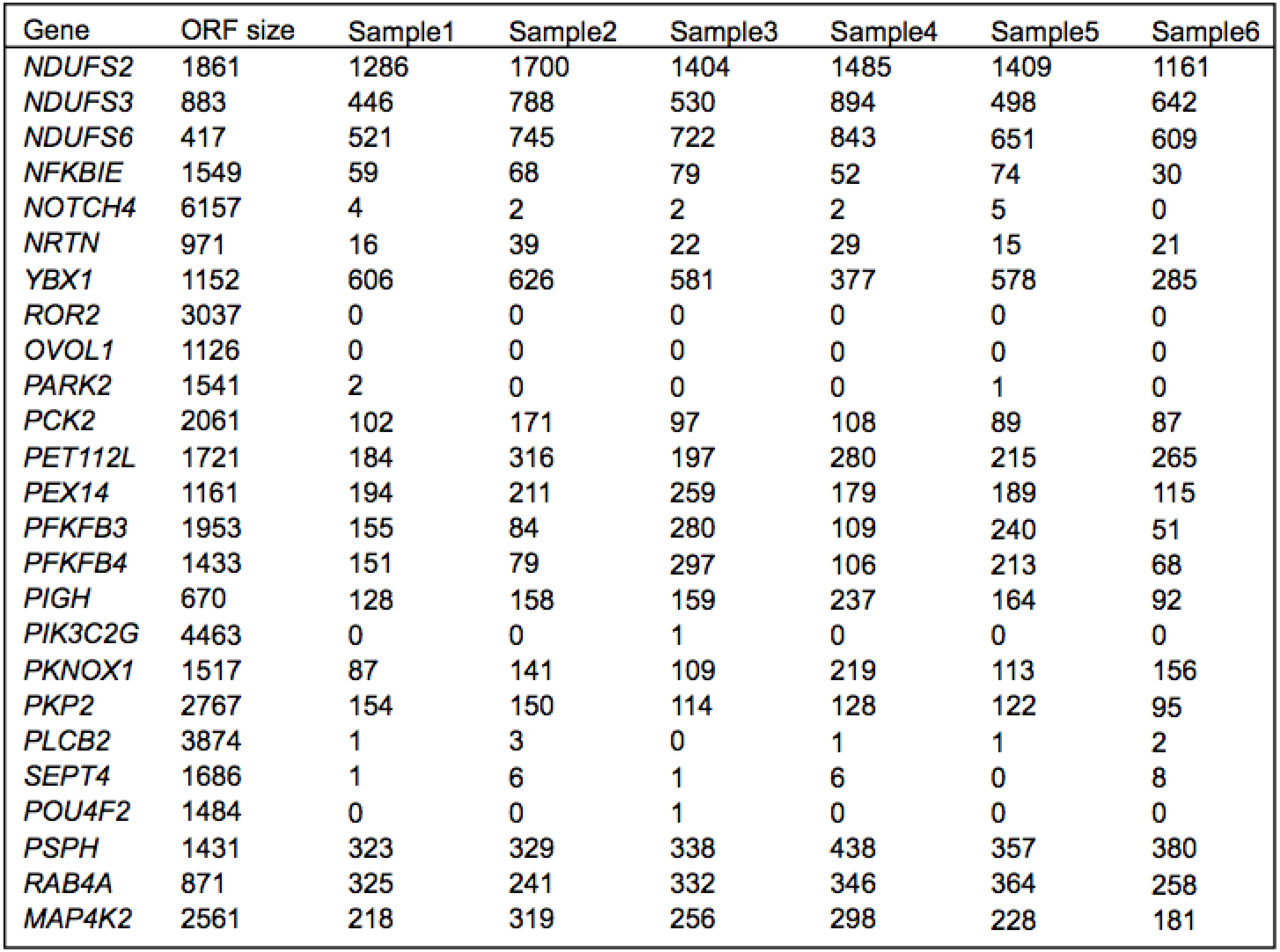
$$RPKM(G) = \boxed{\frac{R}{L}} \cdot \frac{\boxed{10^6 \cdot 10^3}}{M}$$

Example:

$$RPKM(G) = \frac{500}{5000} \cdot \frac{10^6 \cdot 10^3}{10^6} = 100$$

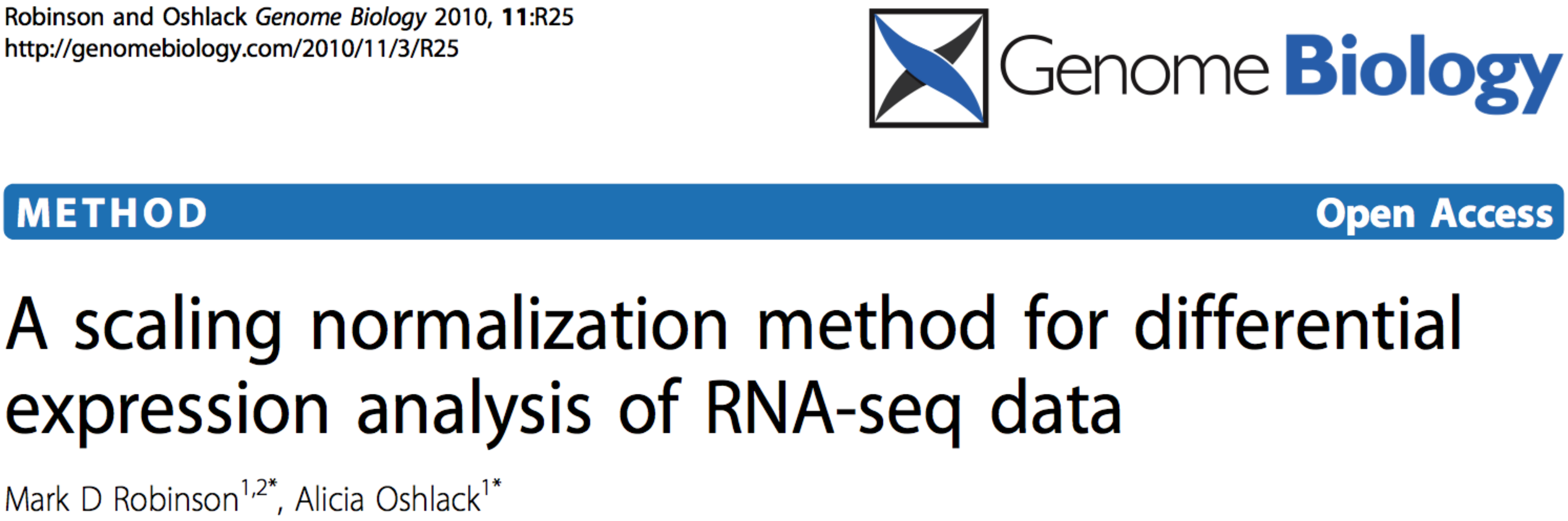
Thus, for gene G, 100 reads are found every 1 KB for 1 million mapped reads

Summarized Counts



Gene	ORF size	Sample1	Sample2	Sample3	Sample4	Sample5	Sample6
NDUFS2	1861	1286	1700	1404	1485	1409	1161
NDUFS3	883	446	788	530	894	498	642
NDUFS6	417	521	745	722	843	651	609
NFKBIE	1549	59	68	79	52	74	30
NOTCH4	6157	4	2	2	2	5	0
NRTN	971	16	39	22	29	15	21
YBX1	1152	606	626	581	377	578	285
ROR2	3037	0	0	0	0	0	0
OVOL1	1126	0	0	0	0	0	0
PARK2	1541	2	0	0	0	1	0
PCK2	2061	102	171	97	108	89	87
PET112L	1721	184	316	197	280	215	265
PEX14	1161	194	211	259	179	189	115
PFKFB3	1953	155	84	280	109	240	51
PFKFB4	1433	151	79	297	106	213	68
PIGH	670	128	158	159	237	164	92
PIK3C2G	4463	0	0	1	0	0	0
PKNOX1	1517	87	141	109	219	113	156
PKP2	2767	154	150	114	128	122	95
PLCB2	3874	1	3	0	1	1	2
SEPT4	1686	1	6	1	6	0	8
POU4F2	1484	0	0	1	0	0	0
PSPH	1431	323	329	338	438	357	380
RAB4A	871	325	241	332	346	364	258
MAP4K2	2561	218	319	256	298	228	181

TMM Normalization



Robinson and Oshlack *Genome Biology* 2010, **11**:R25
http://genomebiology.com/2010/11/3/R25

Genome **Biology**

METHOD **Open Access**

A scaling normalization method for differential expression analysis of RNA-seq data

Mark D Robinson[1,2*], Alicia Oshlack[1*]

Thought Experiment

- Suppose samples A and B are sequenced to the same depth, say 9000 reads
- 90 genes are expressed in A and B truly at the same level
- 10 genes are expressed at high levels in B but not in A, and no other genes are expressed
- Possible scenario
  - All 90 genes get about 100 reads for A
  - First 90 genes for B get about 50 reads, while the other 10 genes get about 450 reads each
  - It would appear that the first 90 are expressed twice as high in A as in B!
  - The reason for this result is that there is a fixed amount of sequencing real estate

Trim Mean of M (TMM) Solution

$$E[Y_{g,k}] = \frac{\mu_{g,k} L_g}{S_k} N_k$$

$$S_k = \sum_g \mu_{g,k} L_g$$

$Y_{g,k}$ – Observed counts of gene g in experiment k

μ_{gk} – True (unknown) expression level of gene g in experiment k

L_g – Length of gene g

N_k – Total number of reads for experiment k

Trim Mean of M (TMM) Solution

$$M_g = \log \frac{Y_{g,k} / N_k}{Y_{g,k'} / N_{k'}}$$

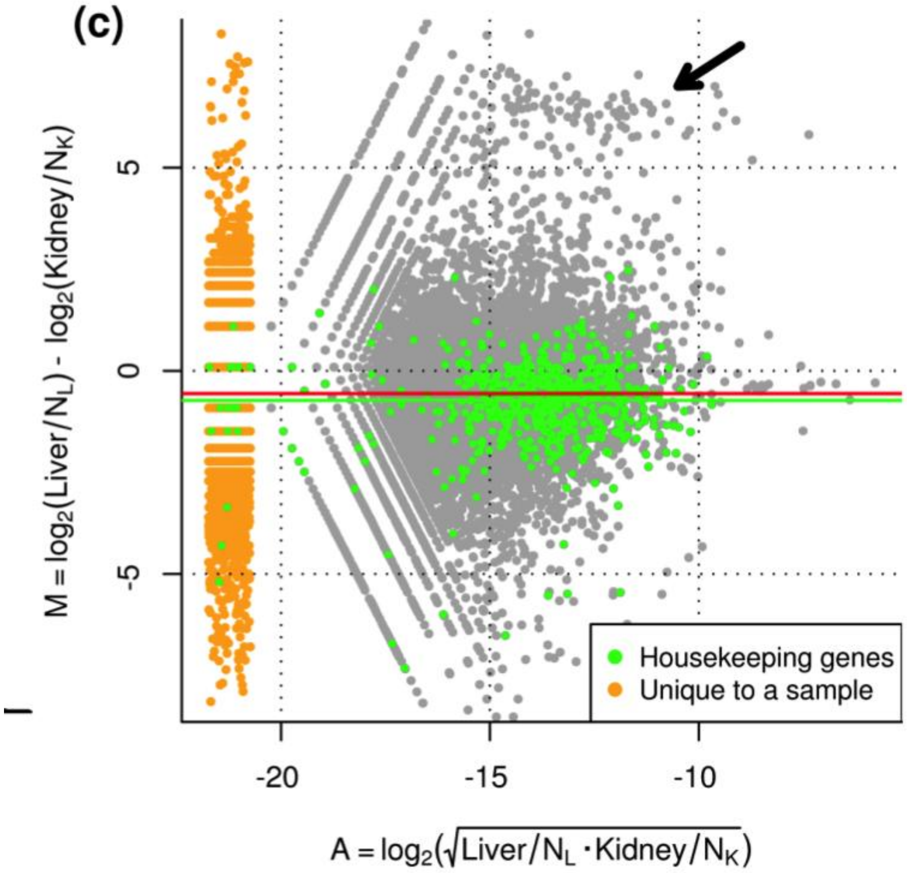
$$A_g = \log(Y_{g,k} / N_k) + \log(Y_{g,k'} / N_{k'})$$

- Trim off the genes with extreme M and A values (paper used 30% and 5%, respectively)
- Compute ratio based on all other genes and use it to normalize expression values
- Others normalize by 75th percentile

(Bullard et al., *BMC Bioinformatics*, 2010)

Does this remind you of a microarray method?

TMM Example



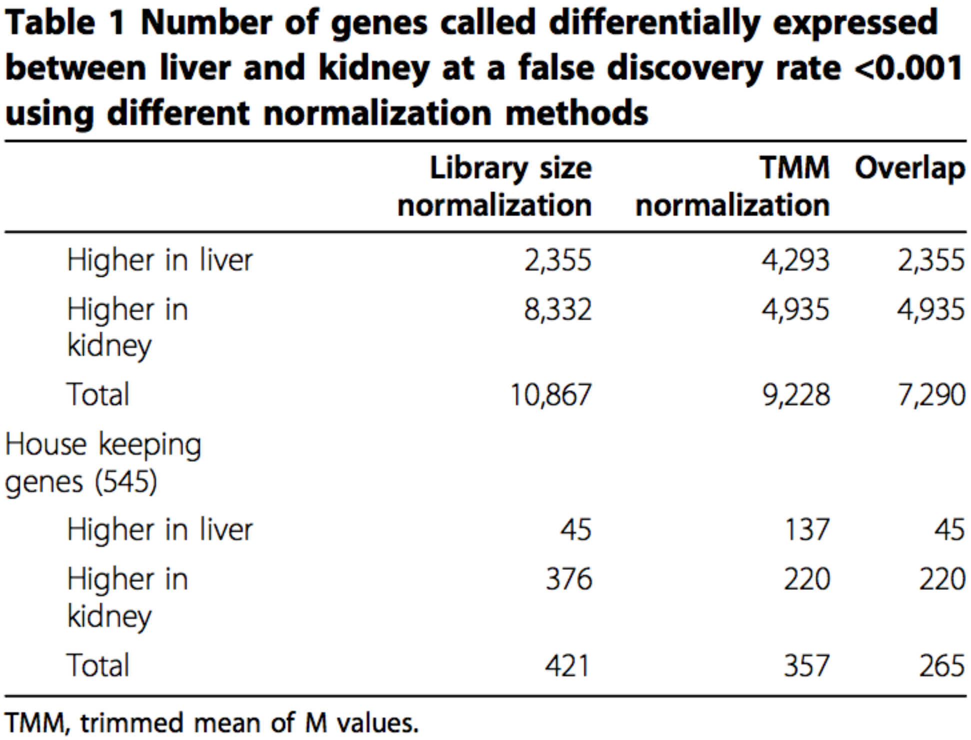


Table 1 Number of genes called differentially expressed between liver and kidney at a false discovery rate <0.001 using different normalization methods

	Library size normalization	TMM normalization	Overlap
Higher in liver	2,355	4,293	2,355
Higher in kidney	8,332	4,935	4,935
Total	10,867	9,228	7,290
House keeping genes (545)			
Higher in liver	45	137	45
Higher in kidney	376	220	220
Total	421	357	265

TMM, trimmed mean of M values.

Transformation

- While ratios are useful, they are not symmetric.
- If *R = 2*G* then R/G = 2 while G/R = ½
- This makes it hard to visualize the different changes
- Instead, we use a log transform, and focus on the *log ratio*:

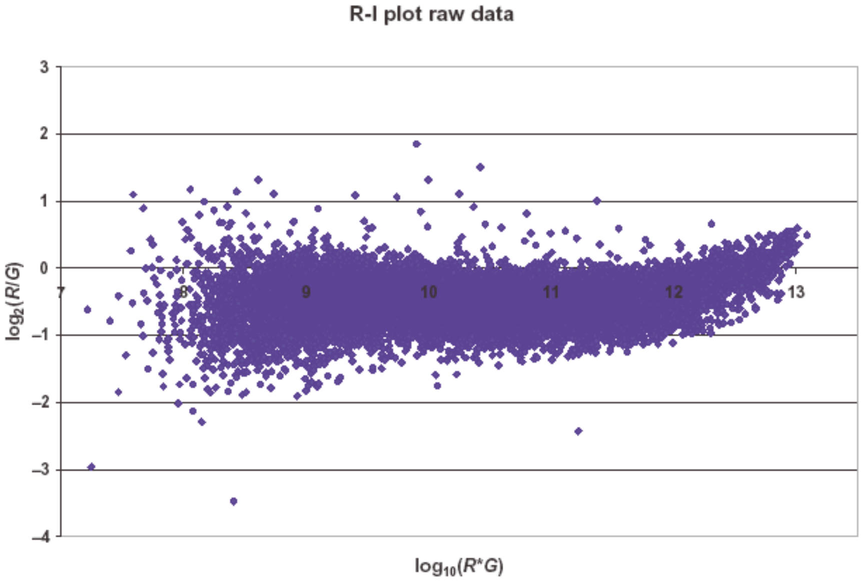
$$y_i = \log \frac{R_i}{G_i} = \log R_i - \log G_i$$

- Empirical studies have also shown that in microarray experiments the log ratio of (most) genes tends to be normally distributed

Advanced topics: Locally weighted regression

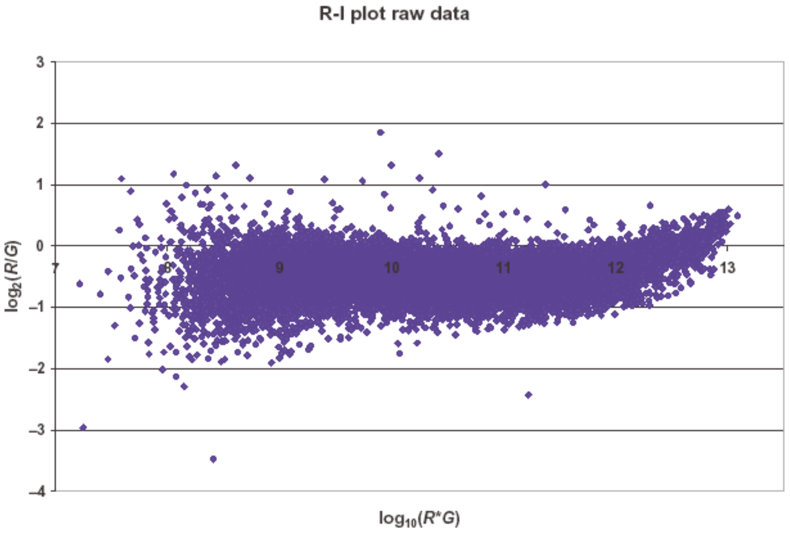
Normalizing between array: Locally weighted linear regression

- Normalizing the mean and the variance works well if the variance is independent of the measured value.
- However, this is not the case in gene expression.
- For microarrays it turns out that the variance is value dependent.



Locally weighted linear regression

- Instead of computing a single mean and variance for each array, we can compute different means and variances for different expression values.
- Given two arrays, *R* and *G,* or two channels on the same array*,* we plot on the *x* axis the (log) of their intensity and on the *y* axis their ratio
- We are interested in normalizing the average (log) expression ratio for the different intensity values



Computing local mean and variance

- Setting

$$m(x) = \frac{1}{k}\sum_{x_i = x} y_i \quad v(x) = \frac{1}{k}\sum_{x_i = x} (y_i - m(x))^2$$

may work, however, it requires that many genes have the same x value, which is usually not the case

- Instead, we can use a *weighted* sum where the weight is propotional to the distance of the point from *x*:

$$m(x) = \frac{1}{\sum_i w(x_i)} \sum_i w(x_i) y_i \quad v(x) = \frac{1}{\sum_i w(x_i)} \sum_i (w(x_i) y_i - m(x))^2$$

$$\hat{y}(x) = \frac{\left(y(x) - m(x)\right)\sqrt{V}}{\sqrt{v(x)}} + M$$

Determining the weights

- There are a number of ways to determine the weights
- Here we will use a Gaussian centered at *x*, such that

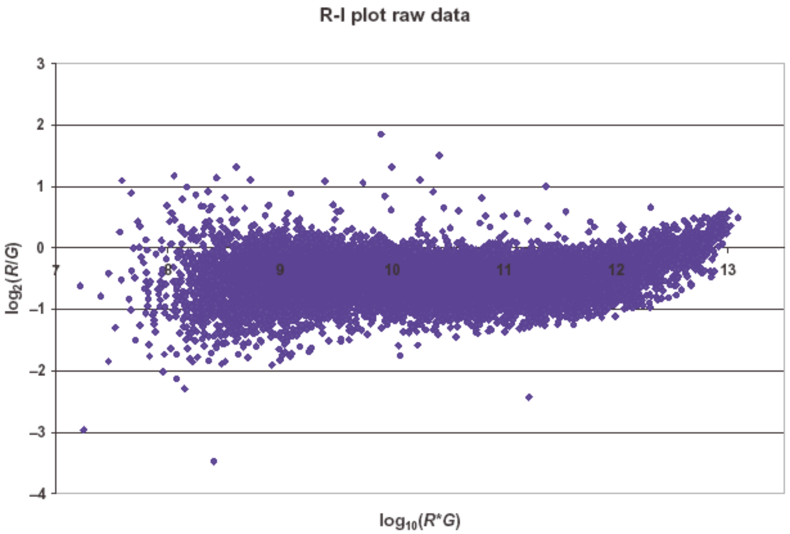
$$w(x_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-x_i)^2}{2\sigma^2}}$$

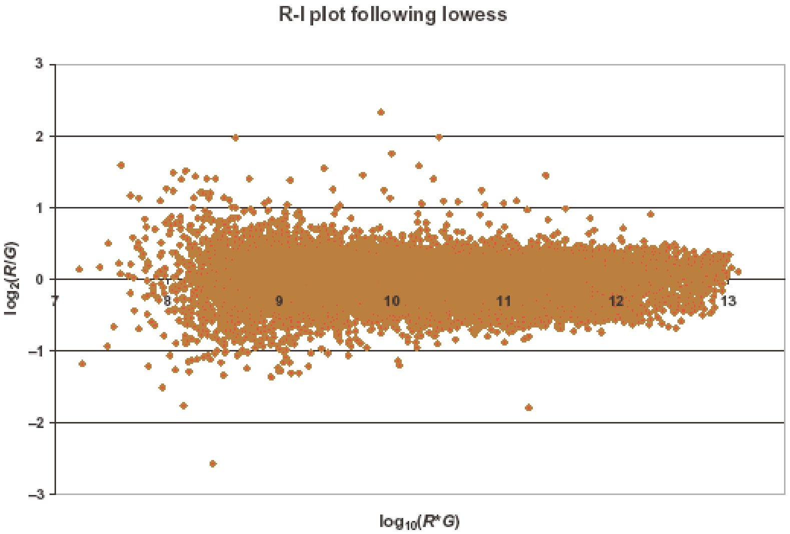
σ^2 is a parameter that should be selected by the user

Locally weighted regression: Results

Original values

normalized values





Projects

- Proposals (2 pages max) due on 3/18
- Work on your own or in teams of 2
- Can propose any project you want related to this class
- We provide a number of ideas on the projects webpage