Lecture 9

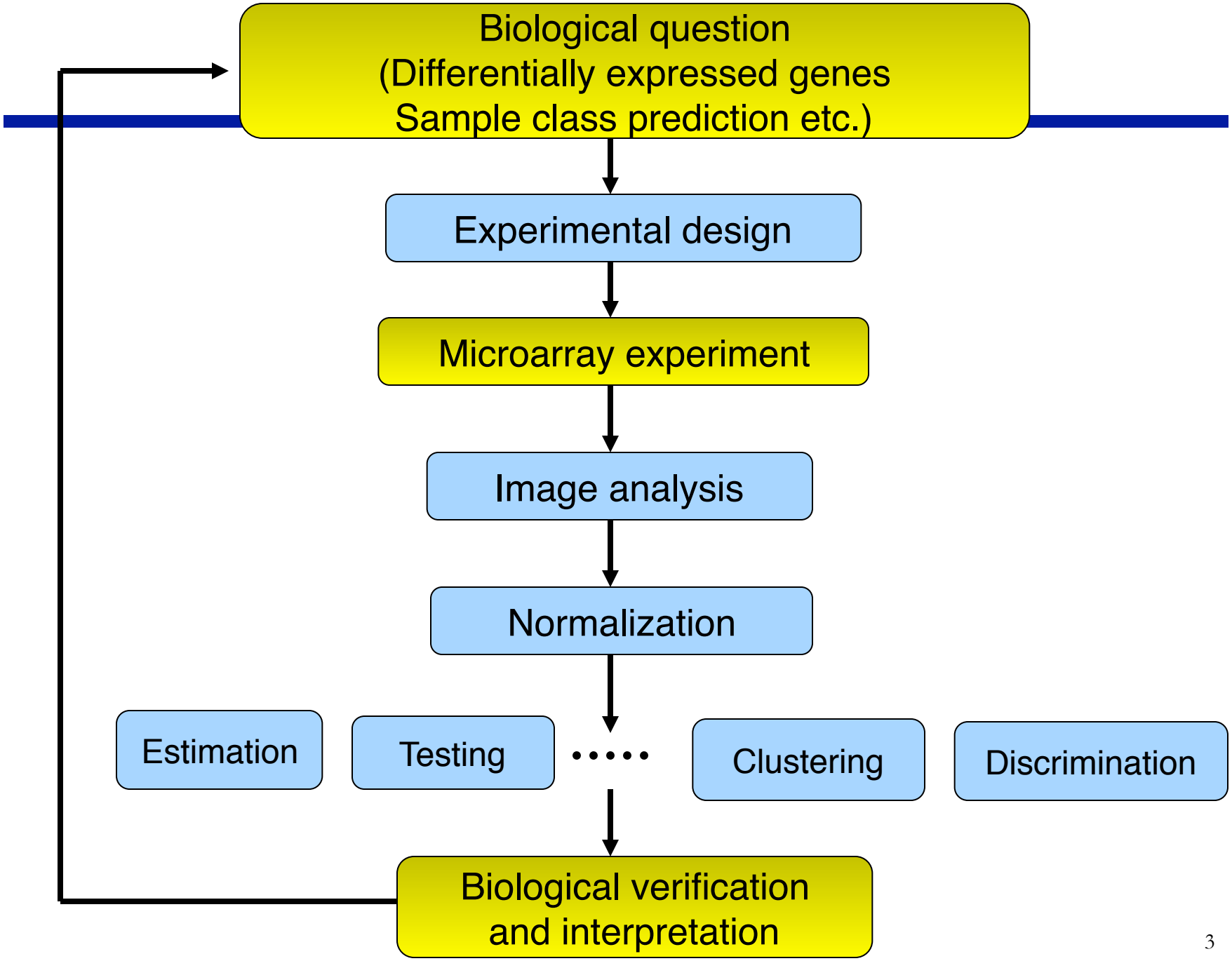
Preprocessing for Microarray Data (Affymetrix array)

MCB 416A/516A
Statistical Bioinformatics and Genomic Analysis

Prof. Lingling An
Univ of Arizona

Outline

- Why data preprocessing, i.e., signal processing?
- Preprocessing Affymetrix array data


Biological question
(Differentially expressed genes
Sample class prediction etc.)
Experimental design
Microarray experiment
Image analysis
Normalization
Estimation
Testing
•••••
Clustering
Discrimination
Biological verification
and interpretation

Microarray signal processing workflow

Communication with biologist/clinician: biological question

Choice of samples

Choice of protocol

Choice of probes and arrays

Allocation of replicates

Experimental design

Microarray experiment

Images

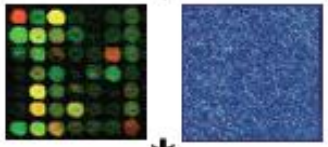


Image analysis

Quality control

Background correction

Quantification

Normalization

Summarization

Signal processing (pre-processing)

Expression matrix

	Array 1	Array 2	Array 3
Gene 1	10.05	9.58	9.76
Gene 2	4.12	4.16	4.05
Gene 3	6.05	6.04	6.08

Estimation

Testing

Annotation

Clustering

Discrimination

Higher-level analysis

Communication with biologist/clinician: biologically interpretable result, verification

Data preprocessing

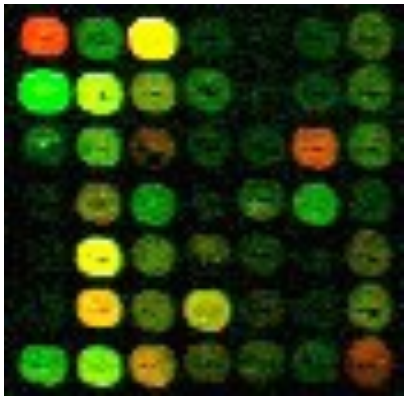
- Images to meaningful data
- What to do with “meaningful” data? -

—— Need to obtain meaningful quantitative measure of gene expression from probe-level data -

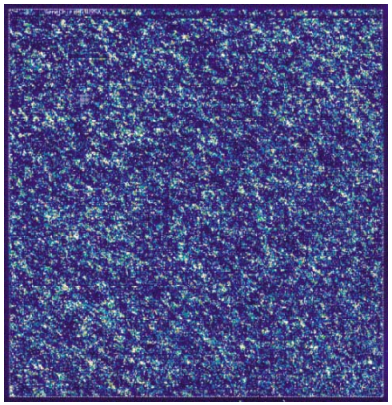
- Make measure “meaningful” by removing artificial and technical sources of variability
- look at what’s there because of biology

Image analysis

- Map region of the chip to a probe and convert pixel intensities in *numeric* expressions for each probe
  - This is a crucial step in the analysis pipeline



cDNA array



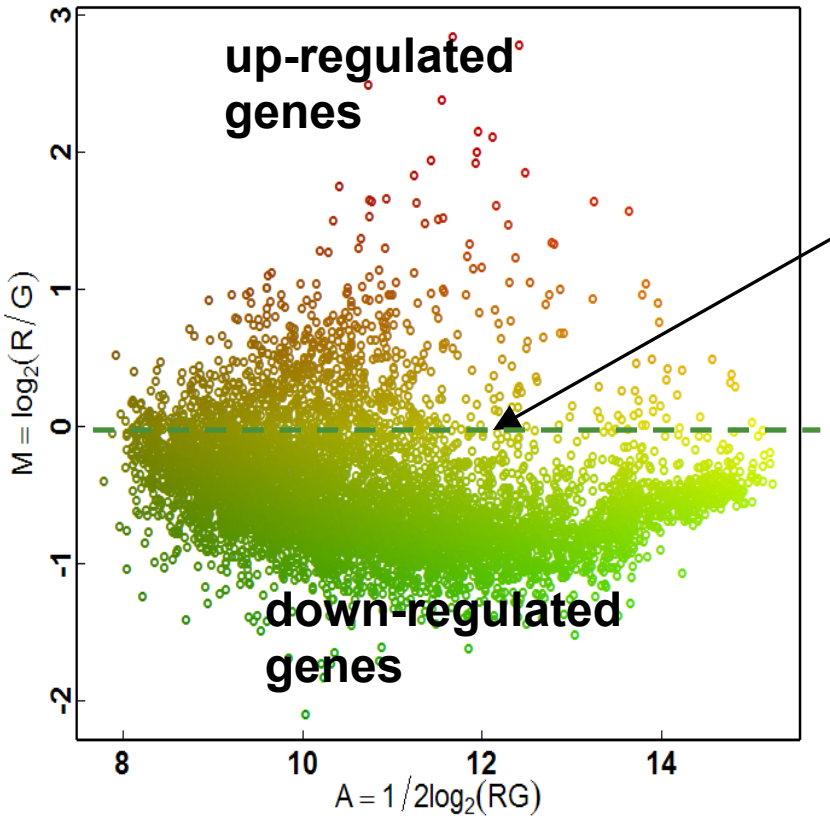
Affymetrix array

Scatter plot: M vs A (recommended)

Note: M vs A is basically a rotation of the $\log_2 R$ vs $\log_2 G$ scatter plot.

Why: Now the quantity of interest, i.e. the *fold change*, is contained in *one variable*, namely M!

If M > 0, up-regulated.
If M < 0, down-regulated.



non-differentially expressed genes are now along the horizontal line:

$M = 0$

$\log_2 R - \log_2 G = 0$

$R = G$

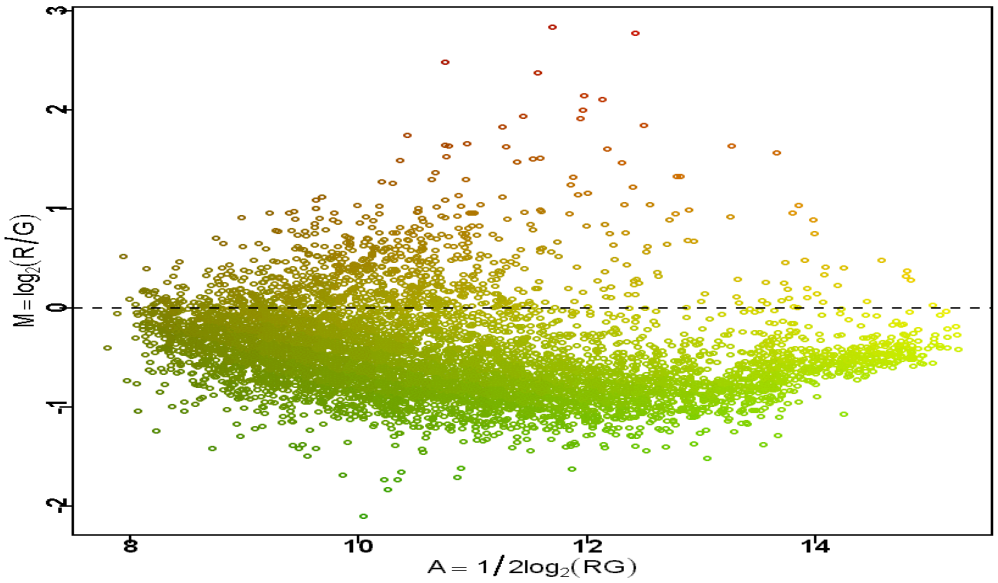
Transformed data $\{(M,A)_i\}$:

$M = \log_2(R) - \log_2(G)$ (minus)

$A = \frac{1}{2} \cdot [\log_2(R) + \log_2(G)]$ (add)

Normalization

- **Expectation**: Most genes are non-differentially expressed, i.e. most of the data points should be around M=0.
- **Idea**: draw various exploratory plots to see if this assumption is met, e.g., M vs A plot



Normalization -2

Result: We commonly observe something else:
Measured value = *real value* + *systematic errors* + *noise*

Correction: If so, *normalize* the data such that the expectations are met:
Corrected value = *real value* + ~~*systematic errors*~~ + *noise*

Normalization -3

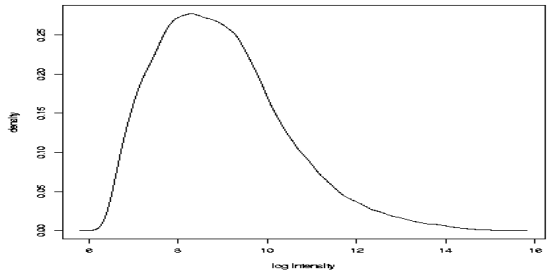
- The experimental goal is to identify biological variation (expression changes between samples)
- Technical variation can hide the real data
- Unavoidable systematic bias should be recognized and corrected – the process referred to as normalization
- Normalization is necessary to effectively make comparisons between chips – and sometimes within a single chip

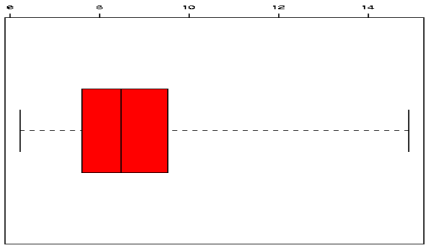
Normalization assumptions and approaches

- Some genes exhibit constant mRNA levels:
  – Housekeeping genes
- The level of some mRNAs are known:
  – Spike-in controls
- The total of all mRNA remains constant:
  – Global median and mean; Lowess
- The distribution of expression levels is constant
  – quantile

Assessing data - Some plots

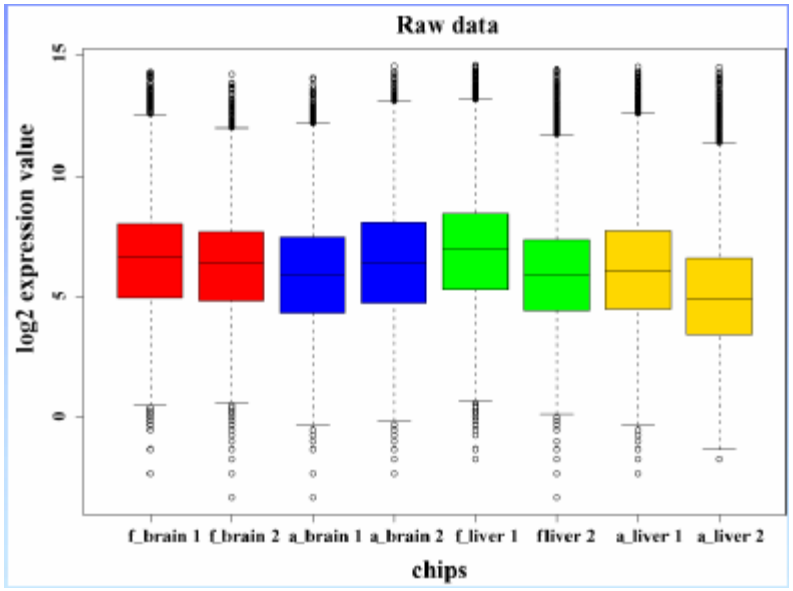
- Plots are useful tools to analyze both, raw and normalized microarray data.
- We use them to:
  - —— Unravel artifacts in the raw data which are not due to biological reasons.
  - —— Assess whether the normalization steps have succeeded in correcting them.

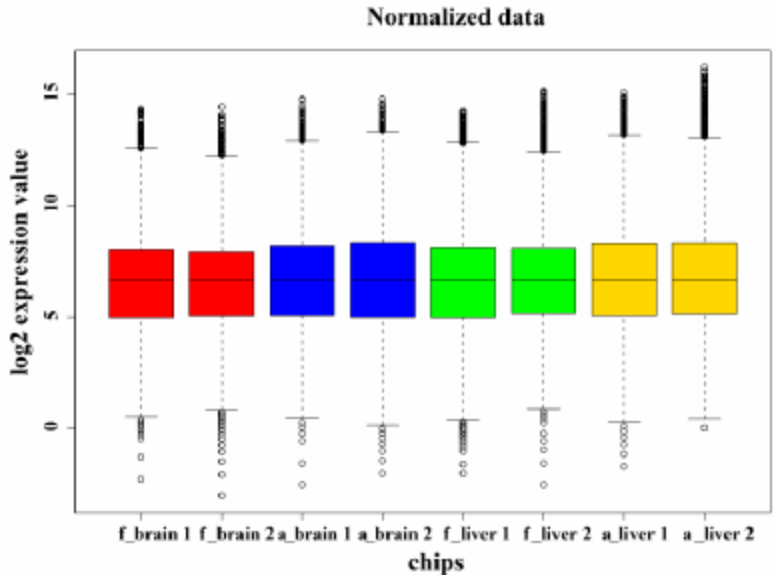




Global median normalization

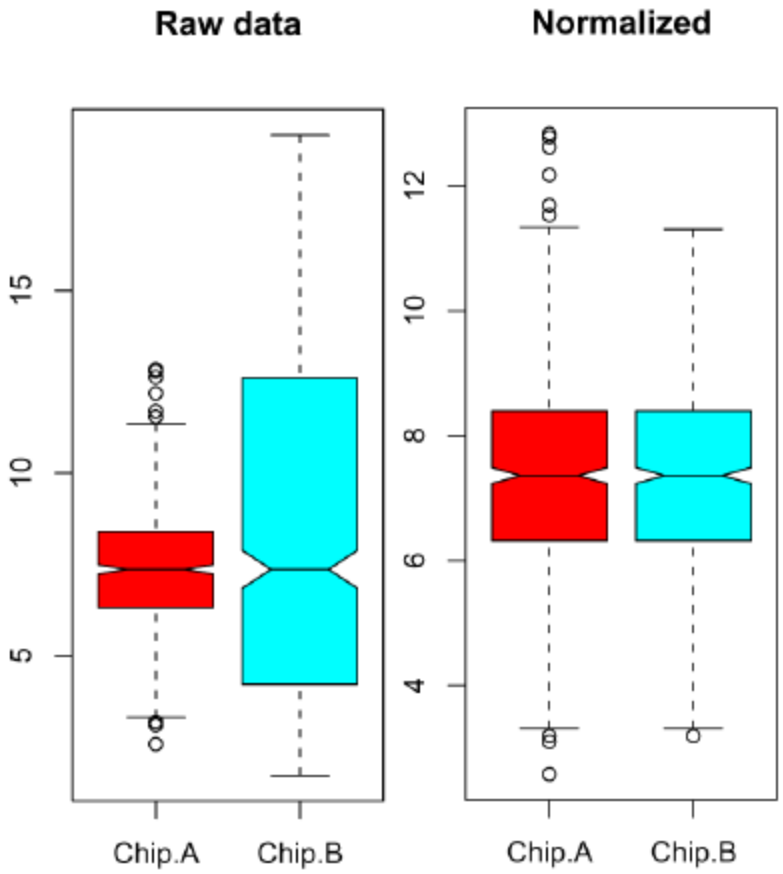
- Procedure: Transform all expression values to produce a constant median
- More robust than using the mean





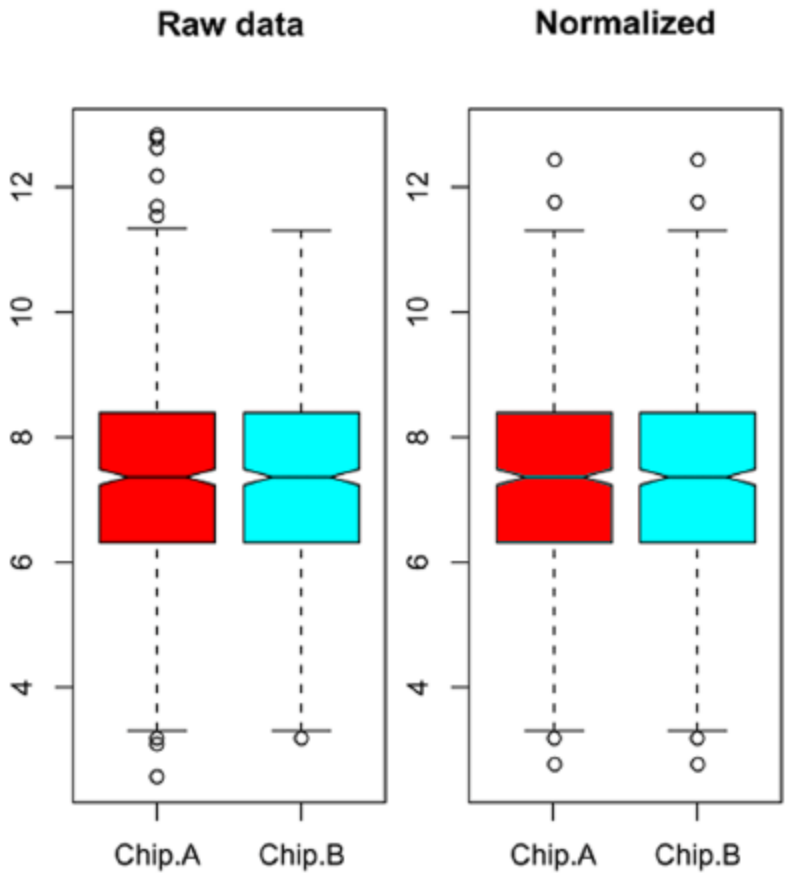
Variance normalization

- Different chips may have the same median or mean but still very different standard deviations
- If we assume the chips should have common standard deviations, they may be transformed in that manner



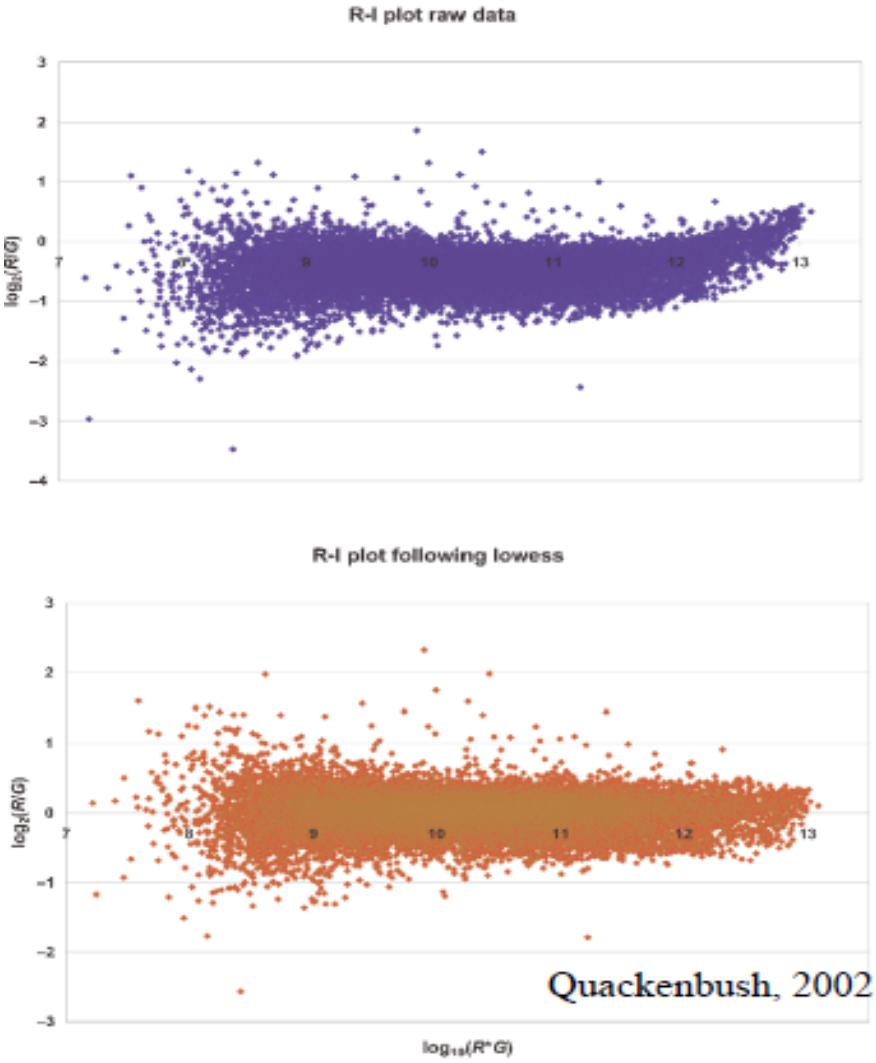
Quantile normalization

- Different chips may have the same standard deviation but different distributions
- If we assume the chips should have common distributions, they may be transformed in that manner



Lowess normalization

- Some 2-color arrays exhibit a systematic intensity-dependent bias
- As a result, the normalization factor needs to change with spot intensity
- Lowess(locally weighted scatterplot smoothing) uses local regression to address this



Local normalization

- Sometimes global within-array normalization may not correct all systematic unwanted variation
  - Examples: print tip differences, degradation in chip regions, thumbprints
- Local normalization adjusts intensities according to chip geography
- It's best to avoid technologies that require these "excessive" transformations

Normalization - summary

- Normalization removes technical variation and improves power of comparisons
- The assumption(s) you make determine the normalization technique to use
- Always look at all the data before and after normalization
- Spike-in controls can help show which method may be best

Pre-processing microarray data

- The core of the Bioconductor functionality is provided by the Biobase package which is loaded into a R session:

```
>library(Biobase)
```

- Note: If you need to install this package use the code below:

```
>source("http://www.bioconductor.org/biocLite.R")
>biocLite("Biobase")
```

Note: only need to install it once!

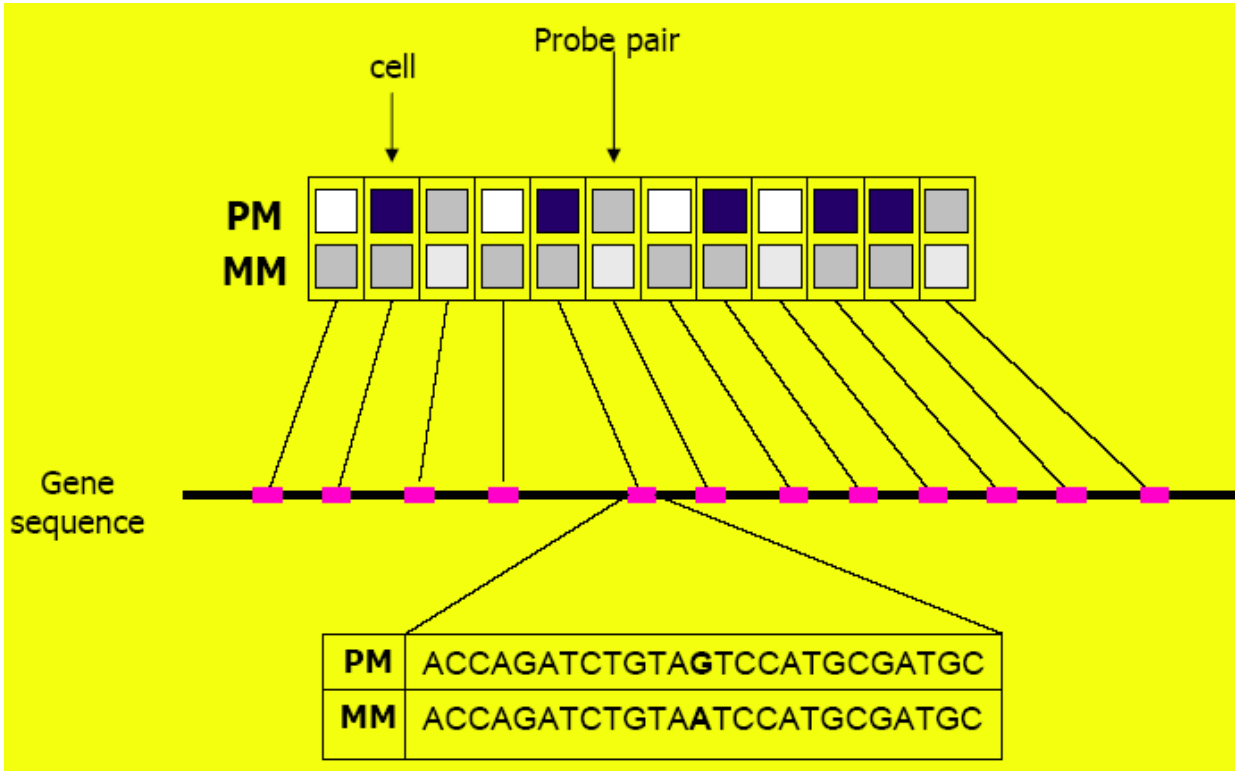
Pre-processing packages

- affy: Affymetrix oligonucleotide chips.
- marray, limma: Spotted cDNA microarrays.
- vsn: Variance stabilization for both types of arrays.
  - Reading in intensity data, diagnostic plots, normalization, computation of expression measures.
- The packages start with very different data structures, but produce similar objects of class exprSet.
- One can then use other Bioconductor and R packages, e.g., mva, genefilter, geneplotter, for more analysis for the data.

Structure of Affymetrix data

- Raw image data from Affymetrix arrays are stored in .DAT files
- The intensity data generated by processing .DAT file are saved as .CEL files. These are imported into an *AffyBatch object* by using *ReadAffy* function.
- `> library (affy)`
- `> Data <- ReadAffy()`
- Warning: This command reads all .CEL files in the working directory and returns the probe-level data in object of class.

Affymetrix GeneChip® technology



Probe: an oligonucleotide of 25 base-pairs, i.e., a 25-mer.
Perfect match (PM): A 25-mer complementary to a reference sequence of interest gene
Mismatch (MM): same as PM but with base change for the middle (13th) base
Probe-pair: a (PM,MM) pair.
Probe set: a collection of probe-pairs (11 to 20) related to a common gene

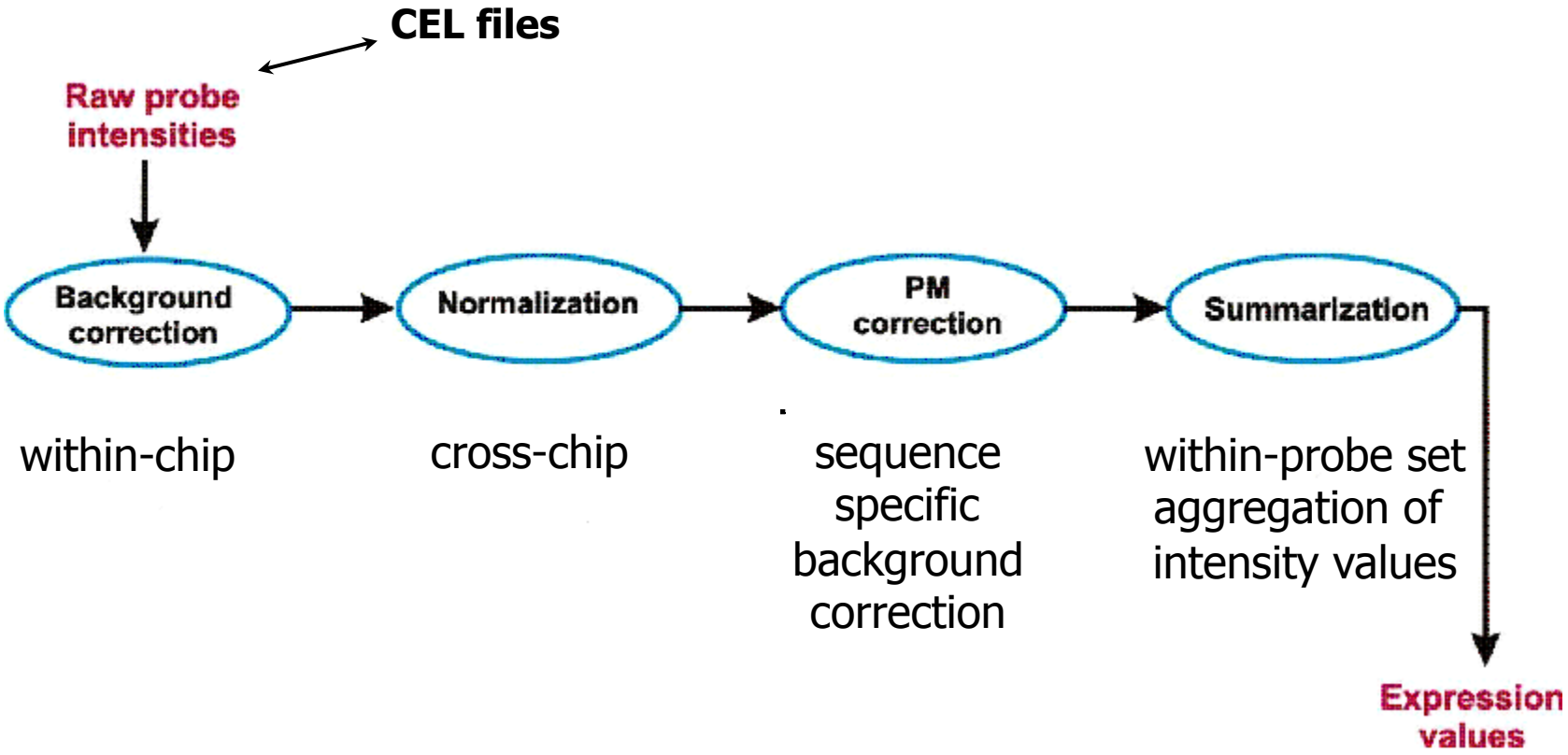
Quality Assessment

- Chip images - 2D spatial color images of log intensities (AffyBatch,Cel).
- Boxplots/density plot of log intensities
- MA plots – scatter plot with fitted curves
- RNA degradation
- Absent/present calls: Low percentages indicate low quality and the percentages should be similar to one another.

Pre-processing Affymetrix data: affy package

- Background correction
  - Adjust for random noise
- Normalization
  - Calibrate measurements of different arrays
  - Which probes / probesets are used?
- How to treat PM and MM values?
  - Adjust for non-specific RNA binding
- Summarization
  - For each probeset summarize levels of corresponding PM and MM probes to a single expression measure

Preprocessing for Affymetrix GeneChips®



How do start?

- Install package:

  >library(affy)

  ### Note: you need to install it if it's your first time to use it:

  source("http://www.bioconductor.org/biocLite.R")
  biocLite("affy")

- Two ways for changing the current directory to D:\MCB
  - Click "file" and then "change dir" then browse D drive and then MCB folder
  - Or setwd("D:\MCB")
- Run the following script to download our example data (6 .CEL files) to your current directory

  >source("http://eh3.uc.edu/affy/downloadAffy.R")
- Check what files are in D:/MCB

  >dir()

How do my data look? -- explore probe level data

- Load CEL files into R

  ```
  > harvard.rawData = ReadAffy()
  ```

- Take a first look at the experiment data

  ```
  >harvard.rawData (only see the description information)
  ```

```
   AffyBatch object
size of arrays=640x640 features (18 kb)
cdf=HG_U95Av2 (12625 affyids)
number of samples=6
number of genes=12625
annotation=hgu95av2
```

- Plot an image of an array

  ```
  image(harvard.rawData[,1])
  ```

- We can grab a glimpse of the PM (Perfect Match) and MM (Mis-Match) probe intensities for any particular probe set by specifying the probe set name as the second parameter in the following input:
- `> geneNames(harvard.rawData)[1:10]`
- `> pm(harvard.rawData, "100_g_at")`
- `> mm(harvard.rawData, "100_g_at")`

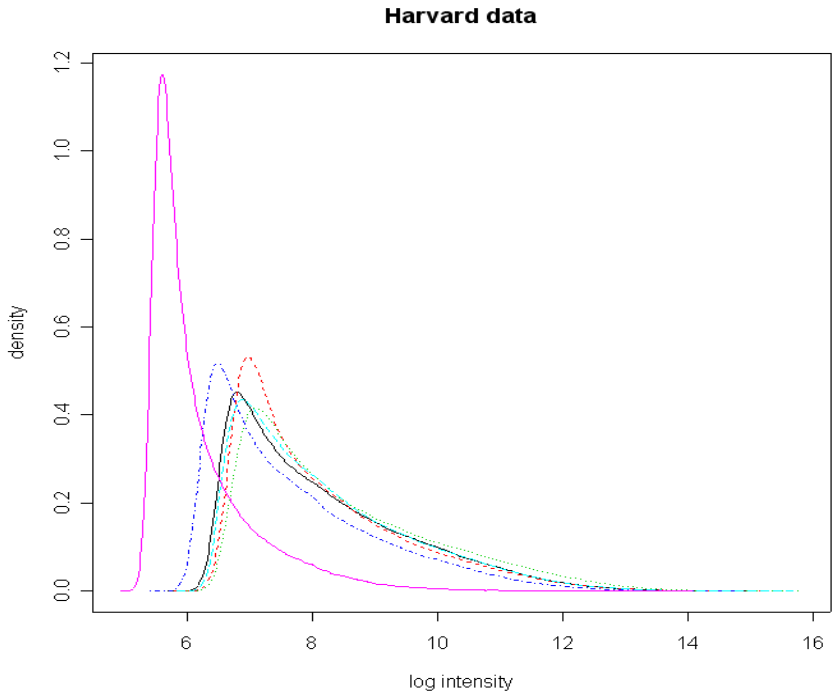
```
> pm(harvard.rawData, "100_g_at")
```

	adeno1.CEL	adeno2.CEL	adeno3.CEL	normal1.CEL	normal2.CEL	normal3.CEL
100_g_at1	222.0	218.0	358.3	244.0	321.5	63.0
100_g_at2	1313.0	939.8	1452.3	862.8	948.0	178.3
100_g_at3	1844.5	1546.0	2168.0	1306.0	1450.0	286.0
100_g_at4	176.8	142.0	213.0	108.0	199.0	49.0
100_g_at5	604.3	790.3	1093.3	437.3	632.3	89.0
100_g_at6	177.8	165.3	220.0	152.0	215.5	54.3
100_g_at7	2182.5	2087.0	3321.5	1801.5	2188.0	283.0
100_g_at8	992.0	1332.5	2033.0	790.0	957.0	135.0
100_g_at9	285.0	267.0	394.3	204.0	298.5	61.0
100_g_at10	204.0	201.8	282.3	210.3	295.5	67.3
100_g_at11	502.8	649.0	885.5	473.0	919.3	102.0
100_g_at12	777.0	743.3	1191.0	680.0	858.3	121.0
100_g_at13	204.0	274.3	417.3	184.5	273.0	56.0
100_g_at14	1750.3	4966.0	2573.0	860.0	2865.5	118.0
100_g_at15	3216.0	6202.5	3723.0	1495.5	3216.0	203.0
100_g_at16	129.8	139.8	213.8	126.0	132.0	60.0

Diagnostic Tools – density plot

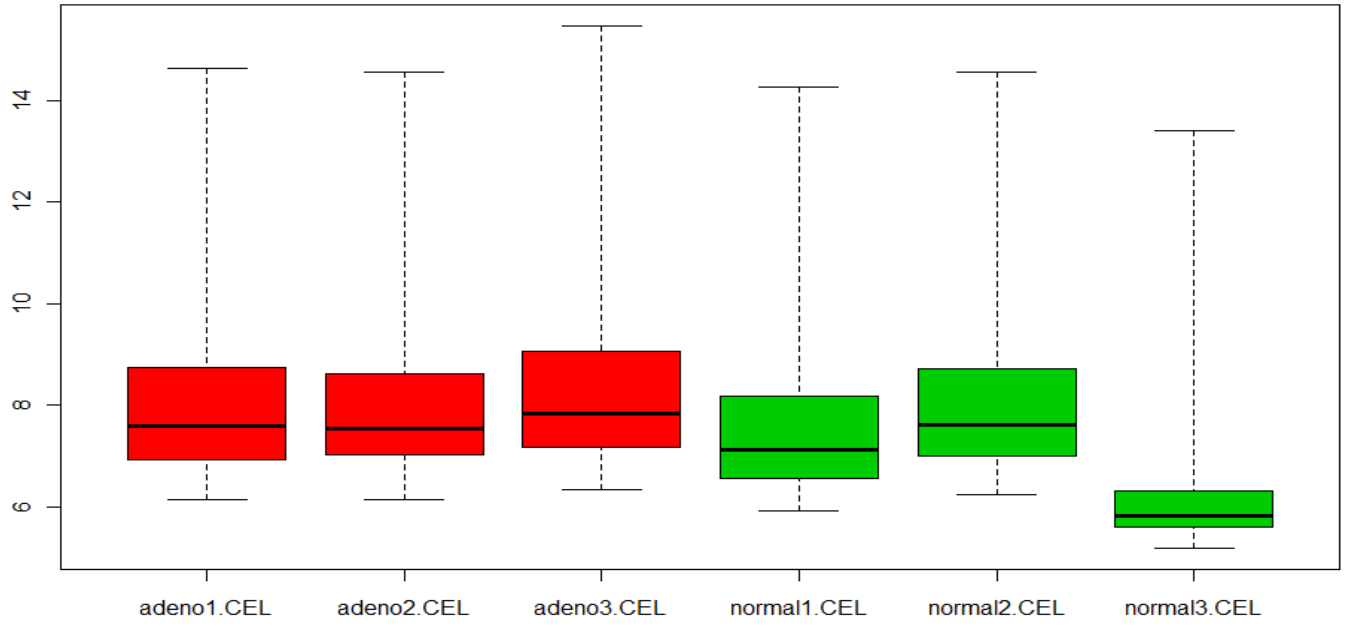
- Plot the log intensity distribution – each curve shows the distribution of each array

```
> hist(harvard.rawData, main =
"Harvard data")
```



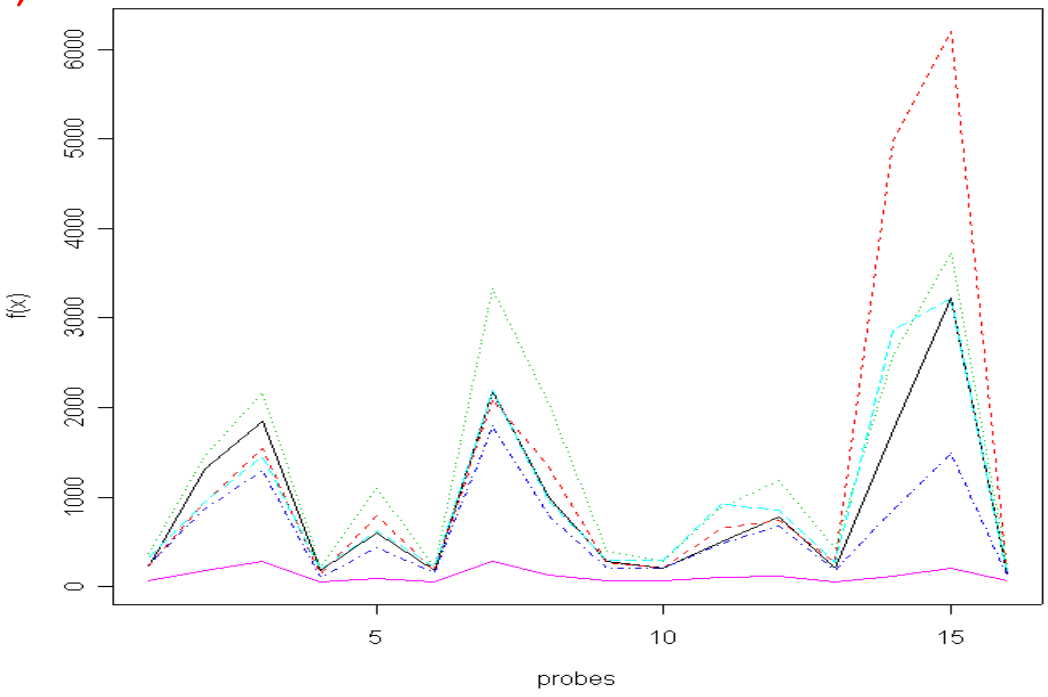
Diagnostic Tools – boxplot

- boxplot the log intensity - each box shows the distribution of expression values of each array
- `>boxplot(harvard.rawData, col=c(2,2,2,3,3,3))`



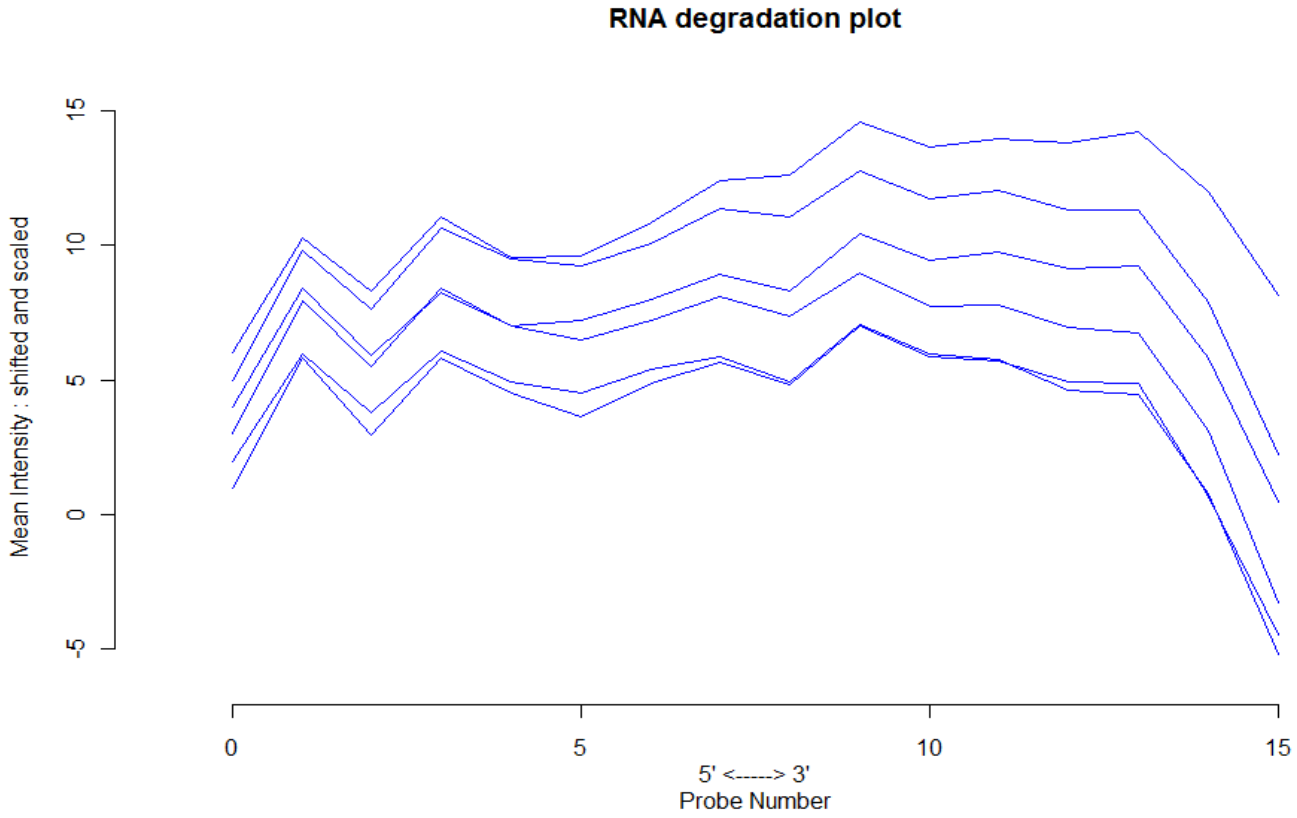
Diagnostic Tools -- Plot a probe set

- `>plot(probeset(harvard.rawData, geneNames(harvard.rawData)[1])[[1]])`



Diagnostic Tools - RNA degradation

- RNA is degraded from the 5' end of a sequence, therefore intensities of probes at the 3' end of a probeset are higher than those at the 5' end.
- The degradation plot shows the (shifted and scaled) mean intensity for each position within a probeset
- High slopes indicate degradation
- More important than the slope is the agreement between arrays




```
> RNAdeg<-AffyRNAdeg(harvard.rawData)
> plotAffyRNAdeg(RNAdeg)
```

Diagnostic Tools: present calls

```
>library(simpleaffy)
> h.qc=qc(harvard.rawData)
> avbg(h.qc)  ##(quality control for average background across arrays)
  adeno1.CEL  adeno2.CEL  adeno3.CEL normal1.CEL normal2.CEL normal3.CEL
   84.16235    92.25578    94.81327    68.61812    88.12901    40.52041
> percent.present(h.qc)
adeno1.CEL.present  adeno2.CEL.present  adeno3.CEL.present
normal1.CEL.present normal2.CEL.present normal3.CEL.present

    38.35248            38.15446            36.41188
    42.32079            38.55050            29.94059
```

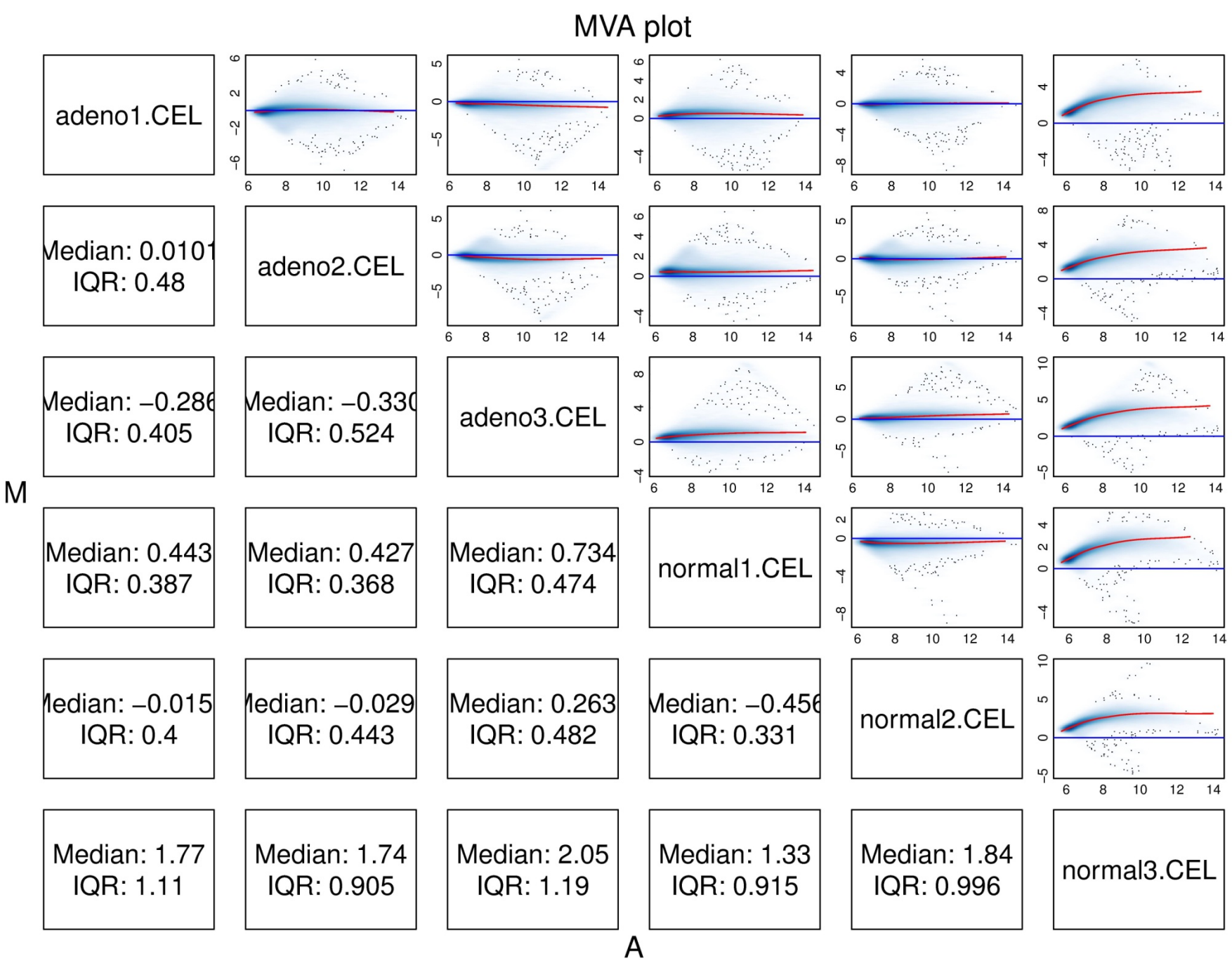
Diagnostic Tools -- MvA plots

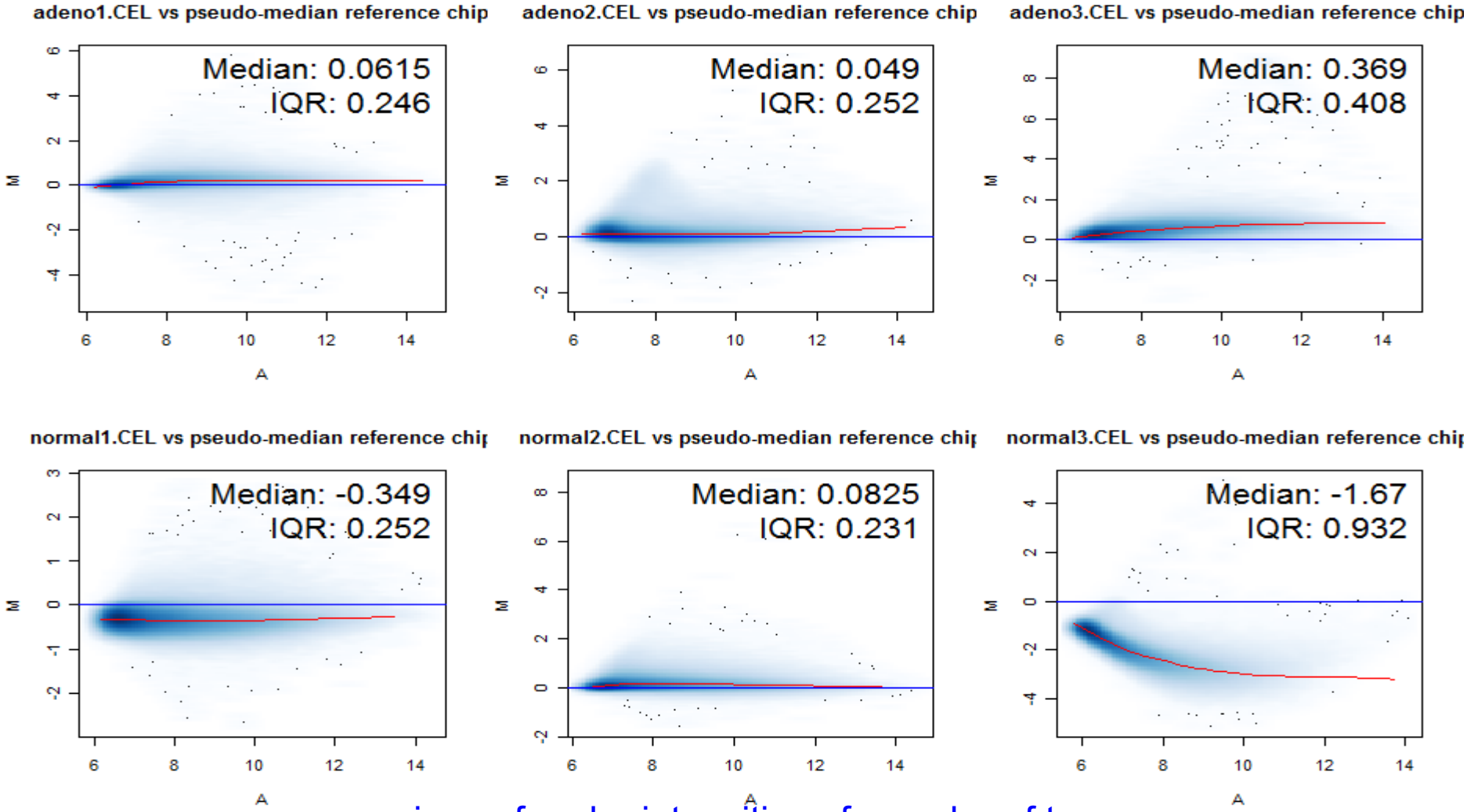
```
>MAplot(harvard.rawData,
plot.method="smoothScatter", pair=TRUE)
> par(mfrow=c(2,3)) ## note: layout of 2X3 plots
> MAplot(harvard.rawData,
plot.method="smoothScatter") - use the
reference array
```

$M_{ijg} = \log_2(PM_{ijg}) - \log_2(PM_{*jg})$
Difference between array i and a reference array $*$

$A_{ijg} = [\log_2(PM_{ijg}) + \log_2(PM_{*jg})]/2$
Average intensity

where PM_{*jg} is the probe-wise median over all arrays

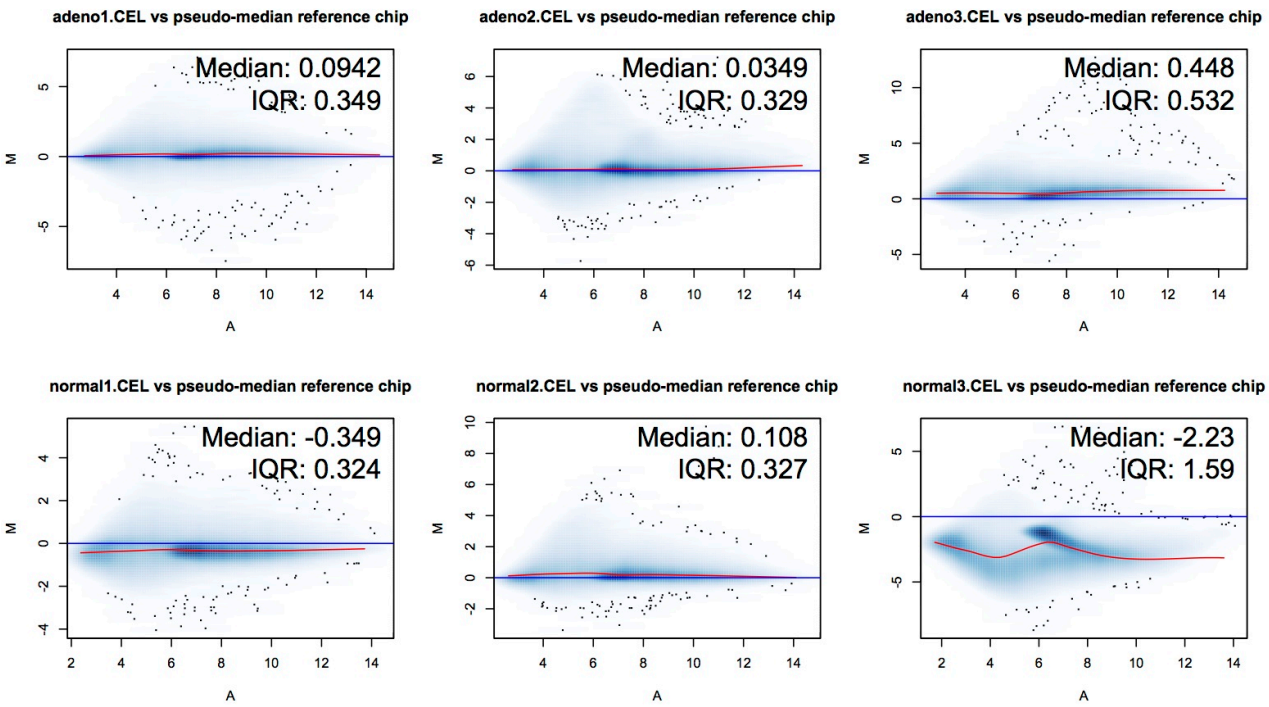

MVA plot
adeno1.CEL
adeno2.CEL
adeno3.CEL
normal1.CEL
normal2.CEL
normal3.CEL
Median: 0.0101
IQR: 0.48
Median: –0.286
IQR: 0.405
Median: –0.330
IQR: 0.524
Median: 0.443
IQR: 0.387
Median: 0.427
IQR: 0.368
Median: 0.734
IQR: 0.474
Median: –0.015
IQR: 0.4
Median: –0.029
IQR: 0.443
Median: 0.263
IQR: 0.482
Median: –0.456
IQR: 0.331
Median: 1.77
IQR: 1.11
Median: 1.74
IQR: 0.905
Median: 2.05
IQR: 1.19
Median: 1.33
IQR: 0.915
Median: 1.84
IQR: 0.996
M
A



comparison of probe intensities of x_i and y_i of two arrays. Points around M=0?

Preprocess1: Background correction

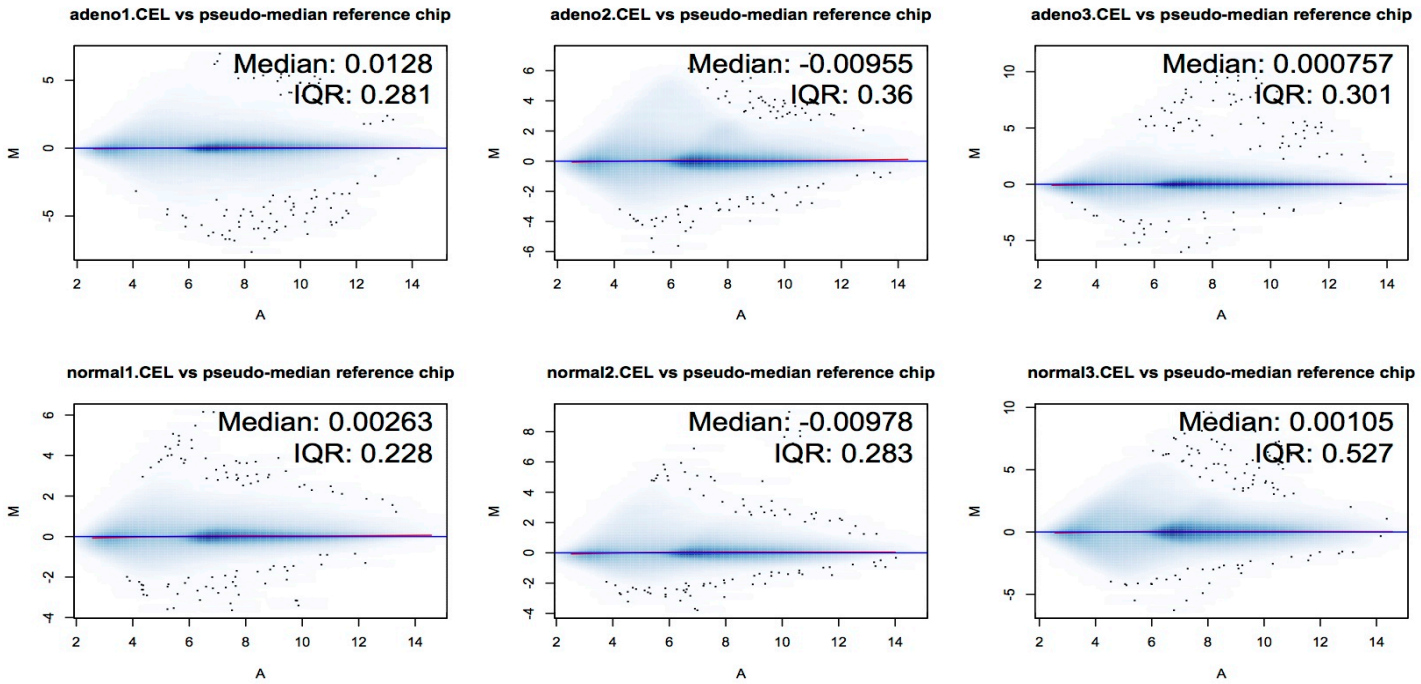
```
>harvard.rmaBG <- bg.correct(harvard.rawData, "rma")
>par(mfrow=c(2,3))
>MAplot(harvard.rmaBG, plot.method="smoothScatter")
```



Other background correction methods, options “mas”, “none”

Preprocess 2: Normalization

```
>harvard.rmaNorm <- normalize(harvard.rmaBG, "quantiles")
>par(mfrow=c(2,3))
>MAplot(harvard.rmaNorm,plot.method="smoothScatter")
```



other normalization options, such as:
"constant", "contrasts", "invariantset", "loess", "qspline", "quantiles.robust"

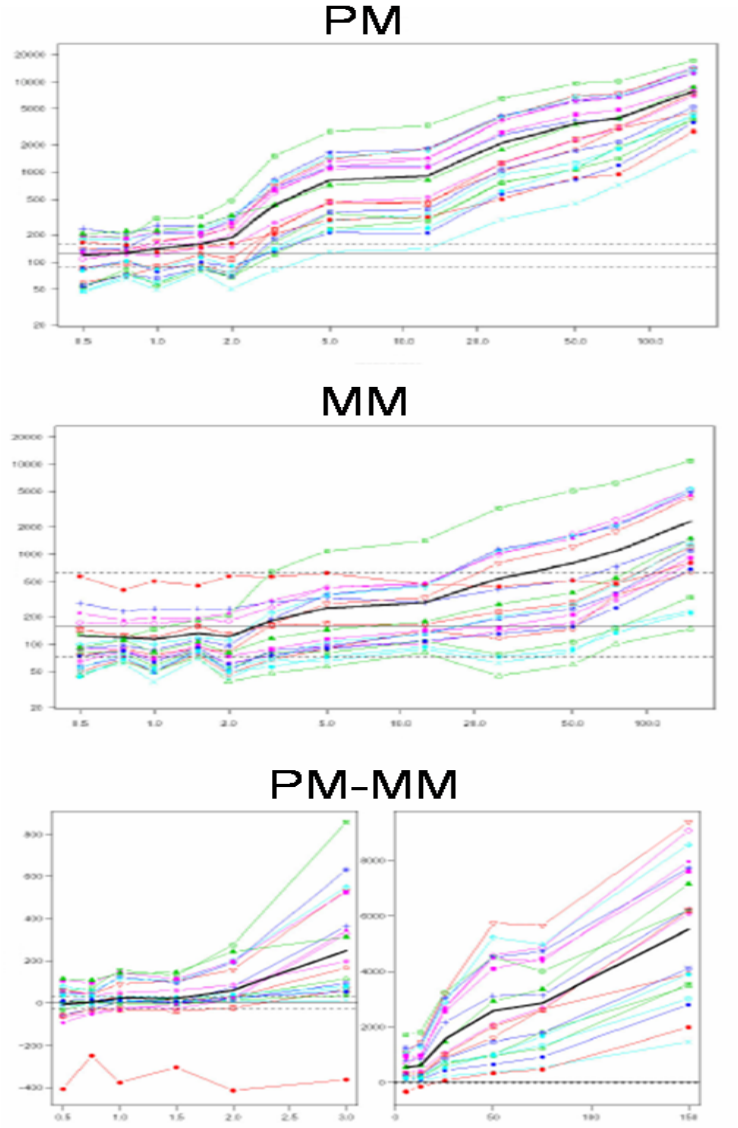
Preprocess 3: PM correction

```
Harvard.rmaPMcorr <-
pmcorrect.pmonly(harvard.rmaNorm)
```

other correction methods are mas and subtractmm

Arguments against PM - MM

- Difference is more variable. Is there a gain in bias to compensate for the loss of precision?
- Subtraction of MM is not strong enough to remove probe effects
- MM detects signal as well as PM

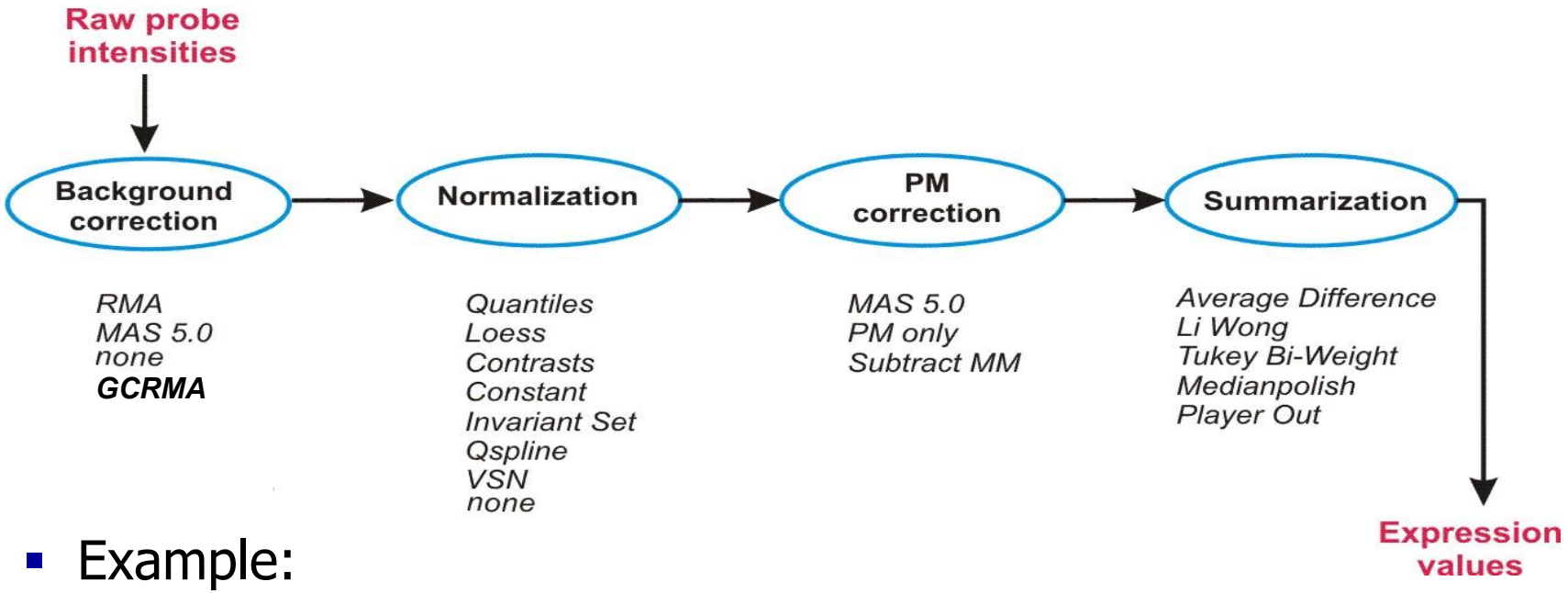


Preprocess 4: Summary methods

- expression level values are calculated from probe level data by methods:
  - avgdiff
  - liwong
  - mas
  - medianpolish (RMA)
  - playerout

  so that one gene is represented by one expression value now

But, "expresso" – does it all at once



- Example:

```
harvard.exprData <- expresso(harvard.rawData,
bgcorrect.method = "rma", normalize.method = "constant",
pmcorrect.method = "pmonly",  summary.method = "avgdiff")
```

`preprocessing` may be done one step at a time, examining effects of each step, just as what we showed

Two “standard” methods – wrapper of expresso

- MAS 5.0 (now GCOS/GDAS) by Affymetrix
- RMA by Speed group (UC Berkeley)

```
>harvard<-mas5(harvard.rawData)
>harvard<-rma(harvard.rawData)
```

Summary and outlook

- Gene expression microarray data is the result of a complex process of measuring and many different processing steps. Normalization is one important topic.
- There is no best normalization method. The selection of an appropriate method depends on the intention of a study.
  - But there is evidence that
    - MAS 5.0 is not a good idea
    - RMA is a much better alternative
    - Other, model-based approaches work well (e.g. GCRMA, VSN)
- It is important to balance between accuracy and precision (bias variance trade off)

Note:

- Affymetrix
  - — Normalization done across arrays
  - — After normalization, the expression data matrix shows absolute expression intensities.
- cDNA
  - — Normalization between two colors in an array.
  - — After normalization, the expression data matrix shows comparative expression intensities (log-ratios).

Other common functions in probe analysis

- probe level matrix

```
intensity(harvard.rawData)
```

- PM values

```
pm(harvard.rawData)[1:10,1]
```

- MM values

```
mm(harvard.rawData)[1:10,1]
```

- probe names

```
probeNames(harvard.rawData)[1:10]
```

- gene names

```
geneNames(harvard.rawData)[1:10]
```

- …