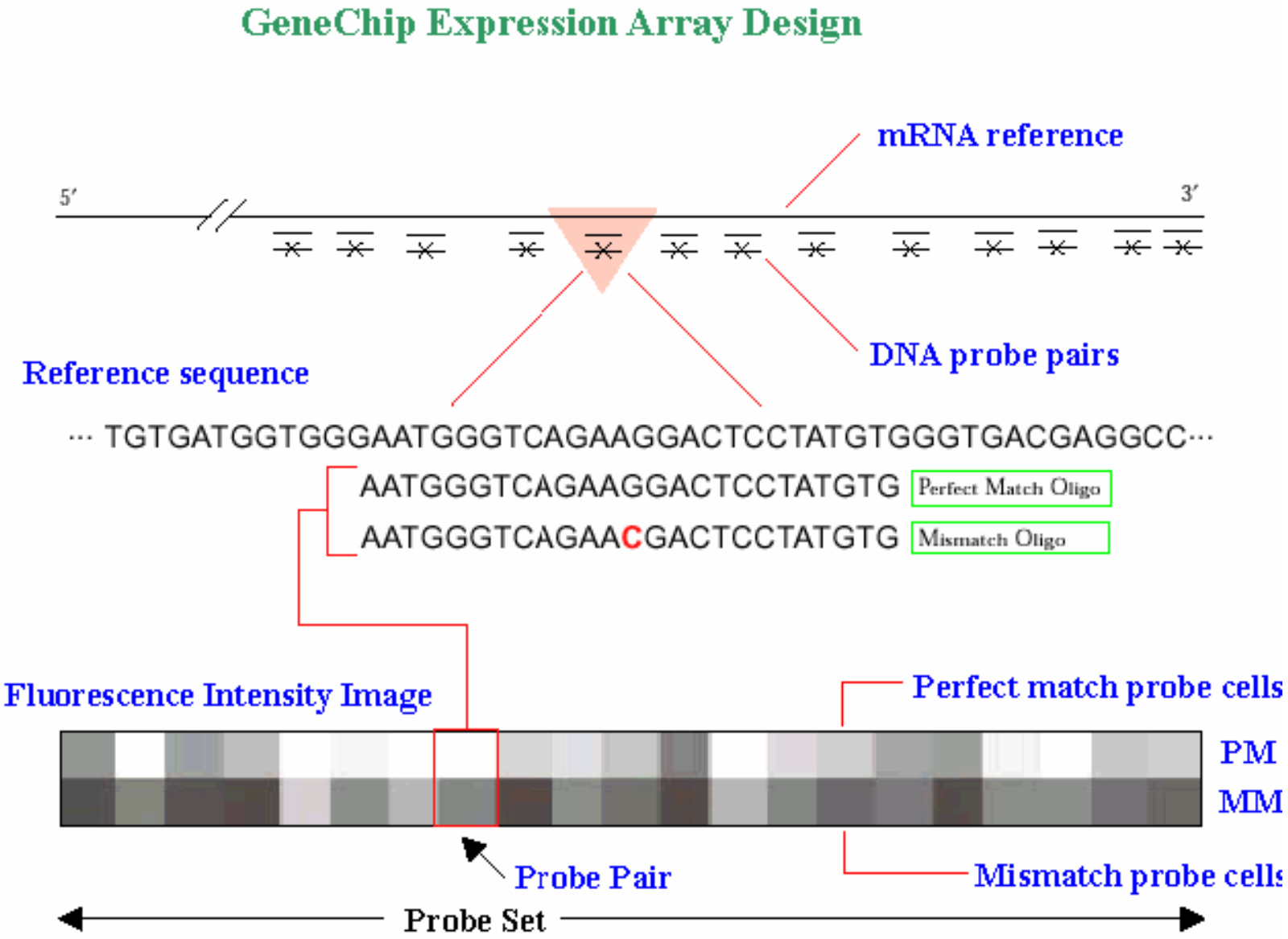
Microarray Data Analysis (VI)

Preprocessing (ii): High-density Oligonucleotide Arrays

High-density Oligonucleotide Array



High-density Oligonucleotide Array

- **PM (Perfect Match):** The perfect match probe has a sequence exactly complimentary to the particular gene.
- **MM (Mismatch):** The mismatch probe differs from the perfect match probe by a single base substitution at the center base position.

Affymetrix GeneGhips

.dat file: a huge image file

.cel file: cell intensity file

.cdf file: probe information denote which probe belongs to which probe set and whether the probe is a PM or MM

Preprocessing for Affymetrix

probe-level data **preprocessing** → genomic-level data

- Image analysis
- Data import
- Background adjustment
- Normalization
- Summarization: for each probe set compute a single number to represent gene expression
- Quality assessment

Affymetrix GeneGhips

.dat file: need not import

.cel file: *ReadAffy* function in the *affy* package

.cdf file: automatically load by ReadAffy

Data Import

- *ReadAffy* (*affy*): the data imported is an object of class *AffyBatch*.
  - ReadAffy() reads all the CEL files in the current working directory
  - ReadAffy(filenames=c(*fnames1,fnames2,…, fnamesk*)) reads a specific set of CEL files.
  - list.celfiles() used to show the CEL files that are located in the current working directory

Data Import

```
> list.celfiles()
 [1] "JD-ALD009-v5-U133B.CEL" "JD-ALD051-v5-U133B.CEL"
 [3] "JD-ALD052-v5-U133B.CEL" "JD-ALD057-v5-U133B.CEL"
 [5] "JD-ALD078-v5-U133B.CEL" "JD-ALD180-v5-U133B.CEL"
 [7] "JD-ALD226-v5-U133B.CEL" "JD-ALD232-v5-U133B.CEL"
 [9] "JD-ALD237-v5-U133B.CEL" "JD-ALD244-v5-U133B.CEL"
[11] "JD-ALD294-v5-U133B.CEL" "JD-ALD380-v5-U133B.CEL"
[13] "JD-ALD381-v5-U133B.CEL" "JD-ALD384-v5-U133B.CEL"
[15] "JD-ALD385-v5-U133B.CEL" "JD-ALD420-v5-U133B.CEL"
[17] "JD-ALD421-v5-U133B.CEL" "JD-ALD431-v5-U133B.CEL"
[19] "JD-ALD433-v5-U133B.CEL" "JD-ALD520-v5-U133B.CEL"
> Data = ReadAffy()
```

Learn more about the probe-level data

cdfName extracts the type of GeneChip

```
> cdfName(Data)
[1] "HG-U133B"
```

sampleNames geneNames probeNames

```
> gg=geneNames(Data)
trying URL 'http://bioconductor.org/packages/2.0/data/annotation/bin/win$
Content type 'application/zip' length 1748313 bytes
opened URL
downloaded 1707Kb

package 'hgu133bcdf' successfully unpacked and MD5 sums checked

The downloaded packages are in
        C:\Documents and Settings\Li-yu D Liu\Local Settings\Temp\Rtmp89$
updating HTML package descriptions
> pp=probeNames(Data)
> length(pp)
[1] 249502
> length(gg)
[1] 22645
```

Learn more about the probe-level data

pm(Data): all of the pm intensities

mm(Data): all of the mm intensities

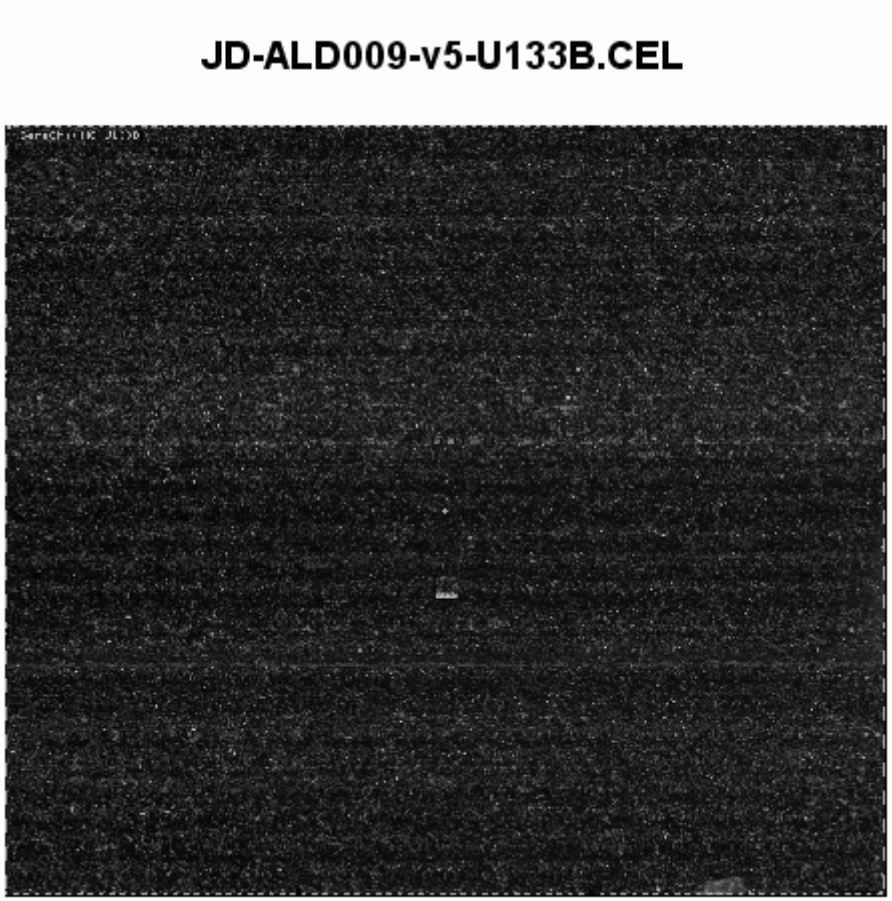
exprs(Data): all pm and mm intensities

pm(Data, “200000_s_at”): the pm intensities of 11 probes for Gene 200000_s_at.

Learn more about the probe-level data

- Visualization:

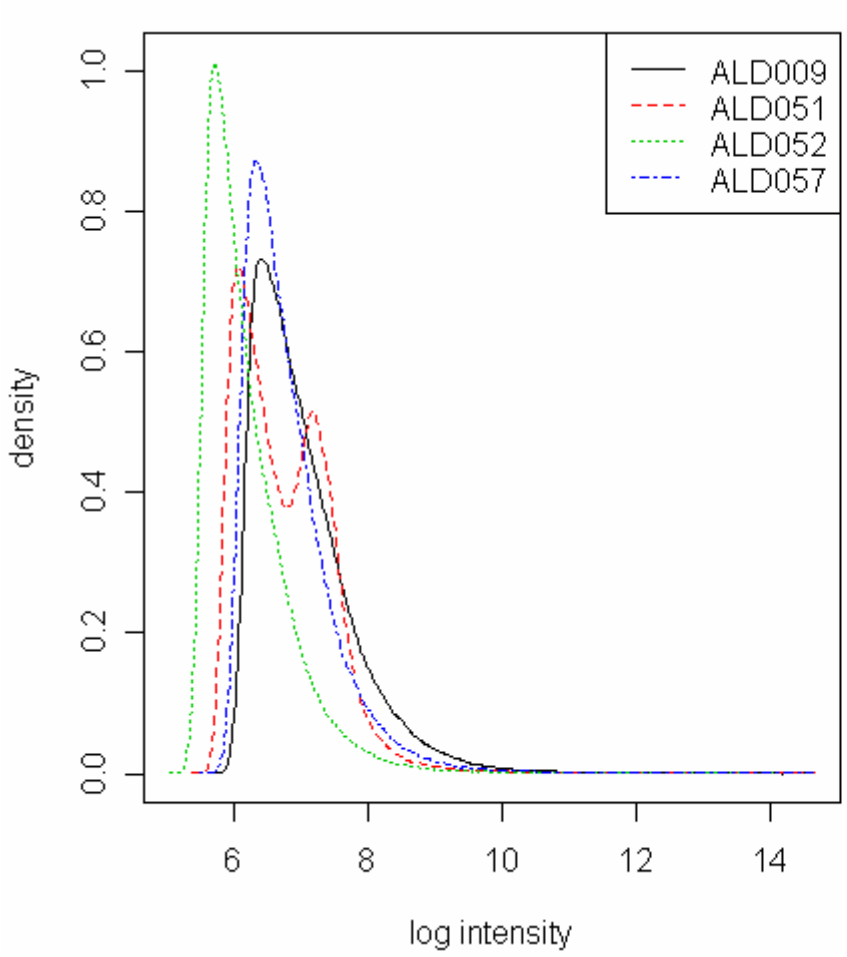
```
> image(Data[,1])
```



Learn more about the probe-level data

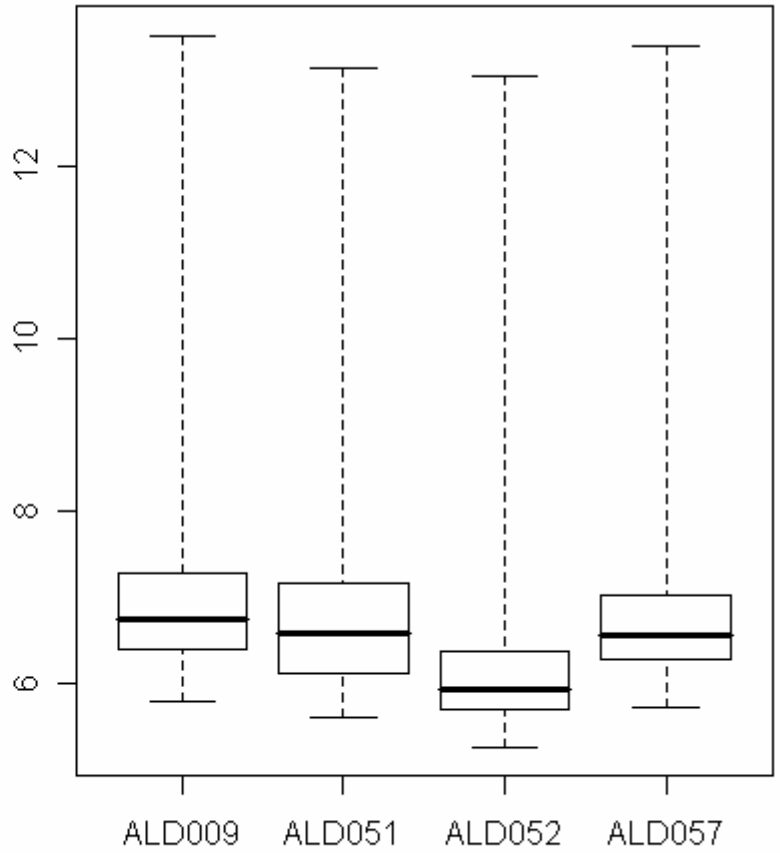
Densities of pm's

> hist(Data[,1:4])



Boxplots of pm's

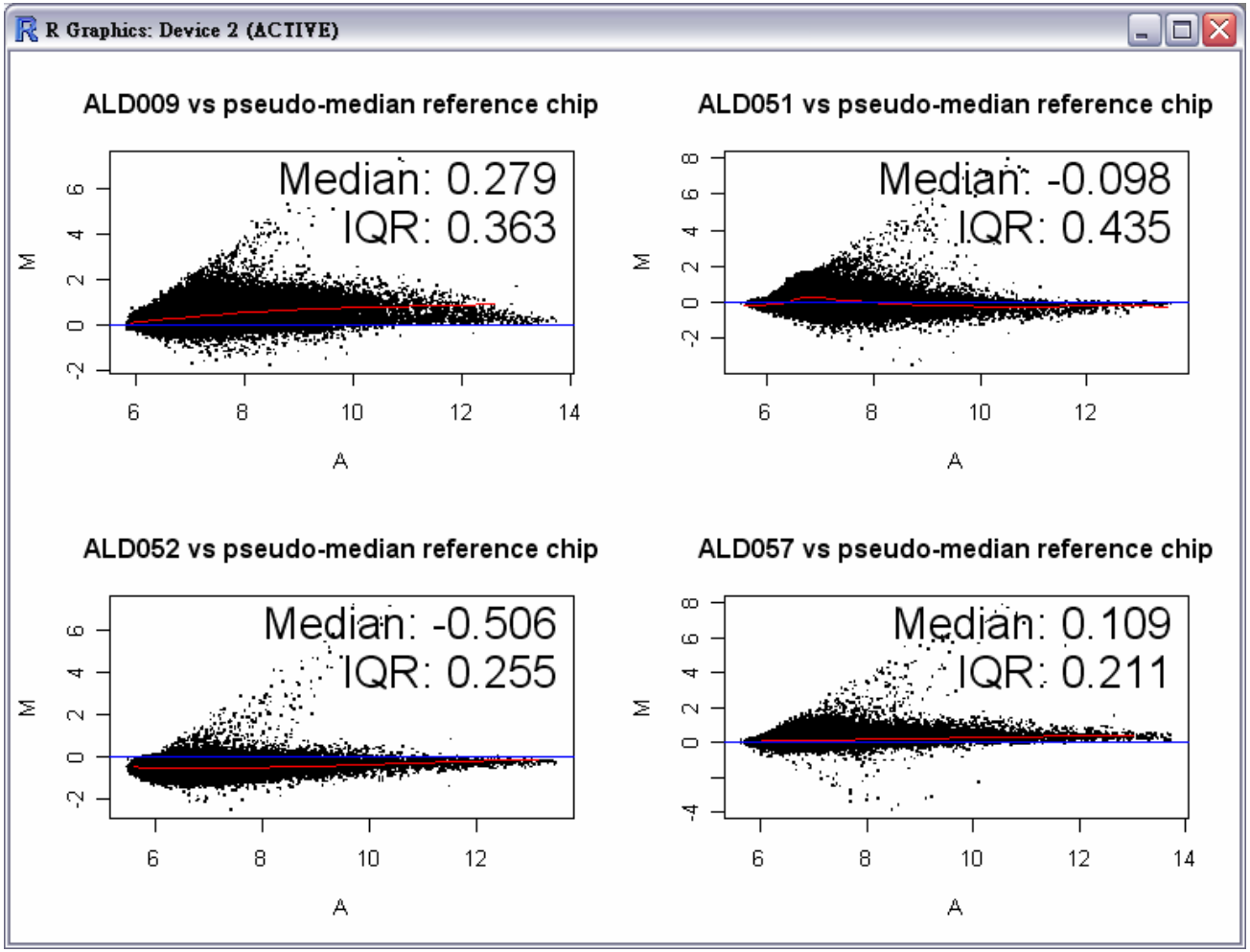
> boxplot(Data[,1:4])



Learn more about the probe-level data

MAplots of pm's

> MAplot(Data[,1:4])



Preprocessing for Affymetrix

probe-level data —**preprocessing**→ genomic-level data

- Image analysis
- Data import
- Background adjustment
- Normalization
- Summarization: for each probe set compute a single number to represent gene expression
- Quality assessment

Preprocessing for Affymetrix

Background / PM adjustment

- PM-MM
- MAS 5.0
- RMA
- GC-RMA

```
> bgcorrect.methods
[1] "mas"  "none" "rma"  "rma2"
```

Normalization

- Constant scaling
- Contrasts
- Invariant set
- Cyclic loess
- Quantile

```
> normalize.methods(Data)
[1] "constant"
[2] "contrasts"
[3] "invariantset"
[4] "loess"
[5] "qspline"
[6] "quantiles"
[7] "quantiles.robust"
```

Background Adjustment

- **Direct subtraction: PM-MM**
  MAS4.0, dChip, MAS5.0

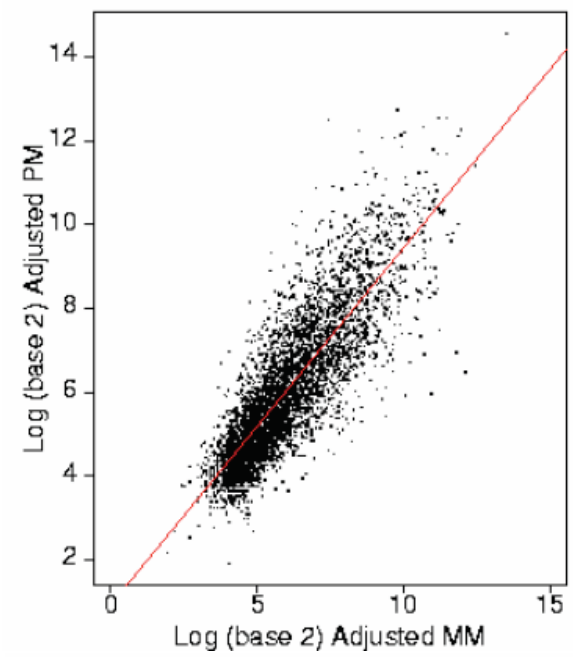
  Assume the following deterministic model:

  PM = O + N + S **(O: optical noise, N: non-specifi binding, S: signal)**

  MM = O + N

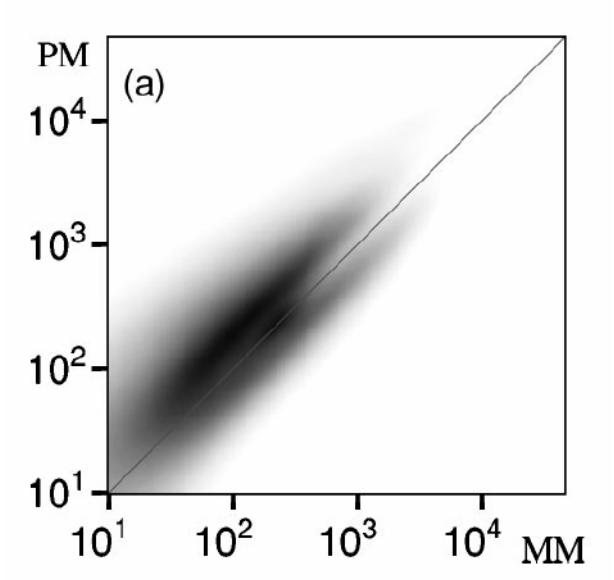
  => PM – MM = S > 0

  Is it true?



MM does not measure background noise of PM

- Yeast sample hybridized to human chip
- If MM measures non-specific binding of PM well, PM≅MM.
- R^2 only 0.5.



Many MM > PM

- 86 HG-U95A human chips, human blood extracts
- Two fork phenomenon at high abundance
- 1/3 of probes have MM > PM

Background Adjustment

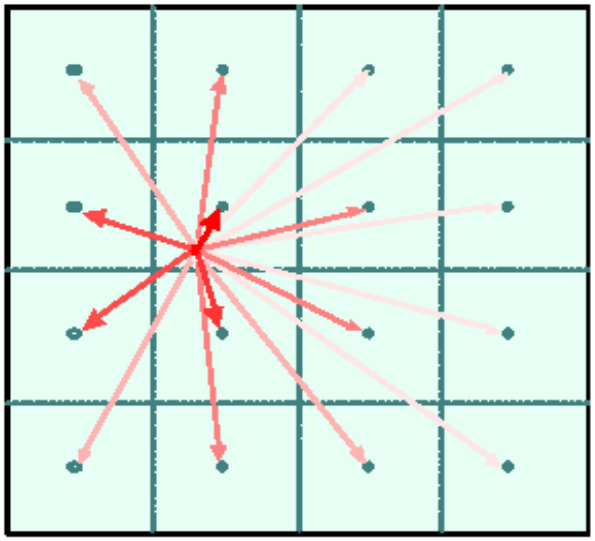
Reasons MM should not be used:

1. MM contain non-specific binding information but also include signal information and noise
2. The non-specific binding mechanism not well-studied.

❖ **Ignore MM ⇒ PM only**

Background Correction – MAS 5.0

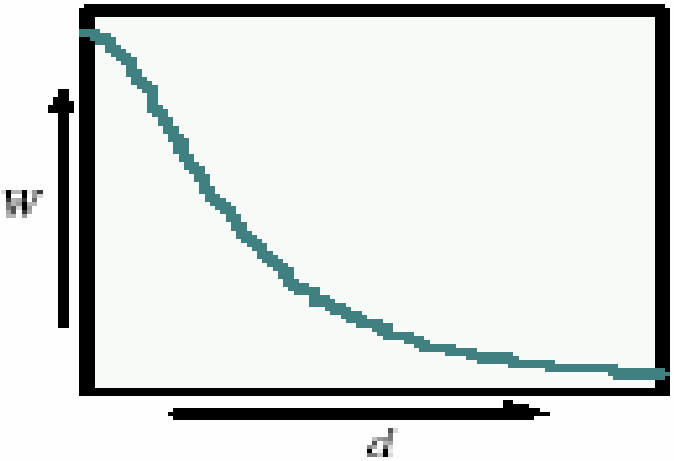
- Divide array into *K* zones (default *K* = 16)
  - Lowest 2% of the intensities in zone *k* are used to compute background B_k
  - Standard deviation for lowest 2% is chosen as noise N_k of zone *k*



background / noise adjustment:

B(x,y) = weighted average of the B_k,
N(x,y) = weighted average of the N_k,

where the weights depend on the distance between (x,y) and the centers of the regions.



(Affymetrix, 2002)

Background Correction – MAS 5.0

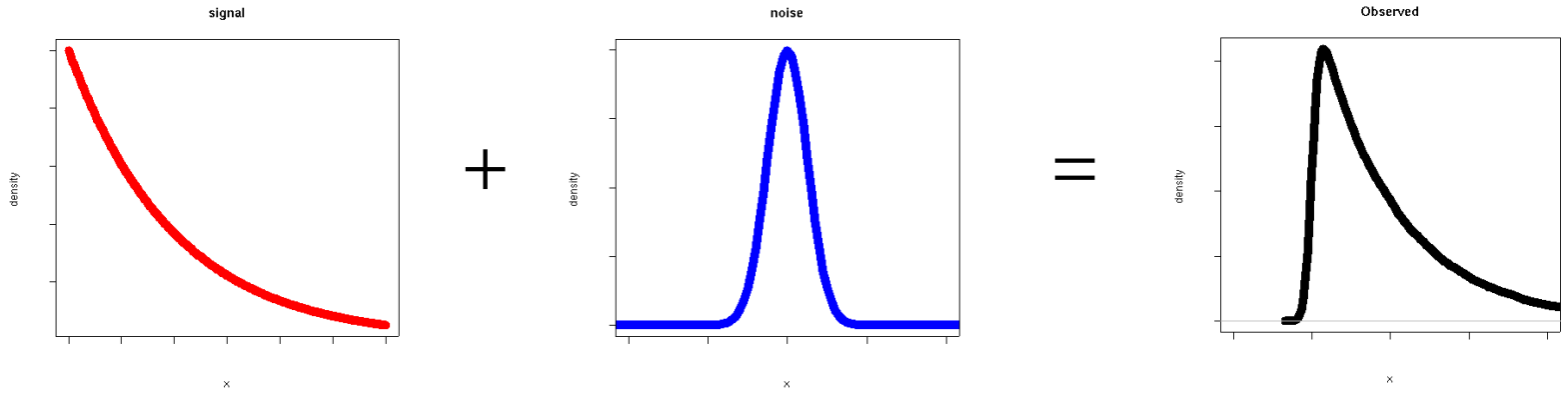
- Background corrected intensity:

$$A(x,y) = \max\left(I(x,y) - B(x,y),\ \text{NoiseFrac} * N(x,y)\right)$$

where NoiseFrac = 0.5 by default

Background Correction – RMA

- Y = observed PM intensity
- Model: Y is the sum of
  - true signal S and
  - background signal B
- $Y = S + B$, where $S \sim Exp(\alpha)$, $B \sim N(\mu,\sigma^2)$, $S \perp B$



(Speed, 2002)

Background Correction – RMA

- RMA (Robust Multiarray Average):

  Correct for background by replacing Y with E(S | Y = y)

  – Estimate μ, σ, α from data

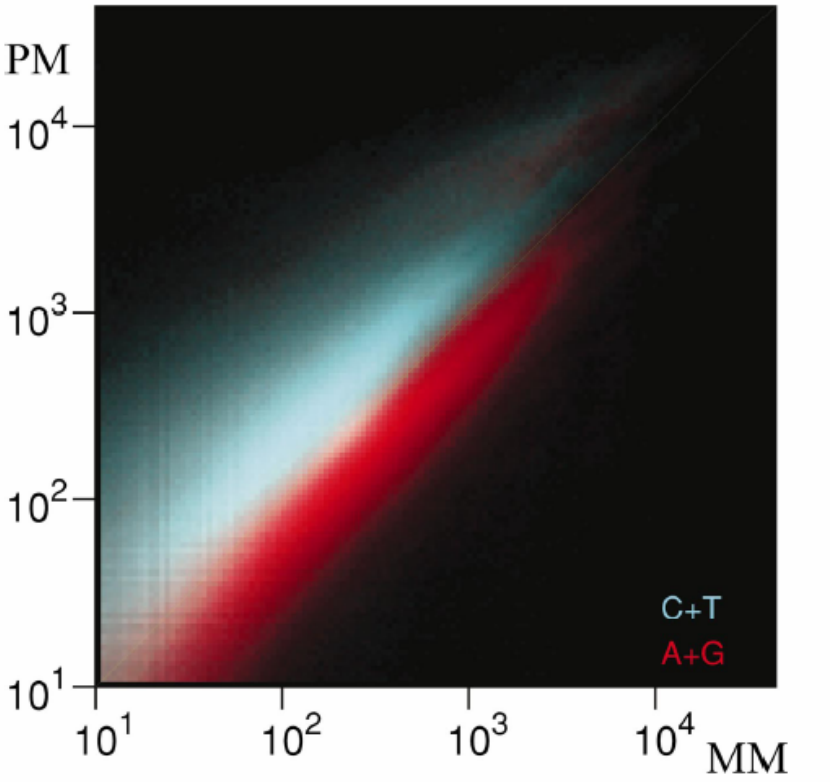
  – Let a = s - μ - $\sigma^2\alpha$, b = σ

$$\mathrm{E}(S \mid Y = y) = a + b \frac{\phi\left(\frac{a}{b}\right) - \phi\left(\frac{s-a}{b}\right)}{\Phi\left(\frac{a}{b}\right) - \Phi\left(\frac{s-a}{b}\right) - 1}$$

where $\phi =$ pdf of N(0,1), $\Phi =$ cdf of N(0,1)

Background Correction – GCRMA

- RMA ignores the different propensities of probes to undergo non-specific binding. Hence the background is often underestimated.



95% of (MM>PM) have purine (A, G) in the middle base.

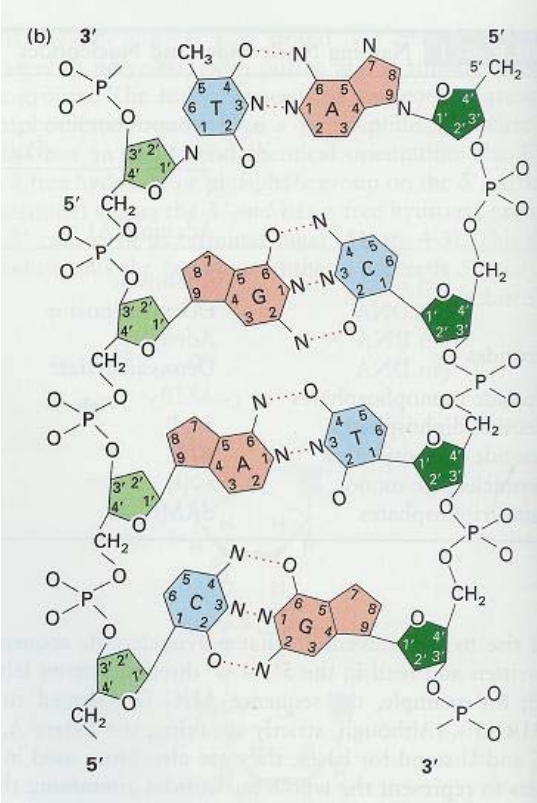
AATGGGTCAGAAGGACTCCTATGTG

AATGGGTCAGAACGACTCCTATGTG

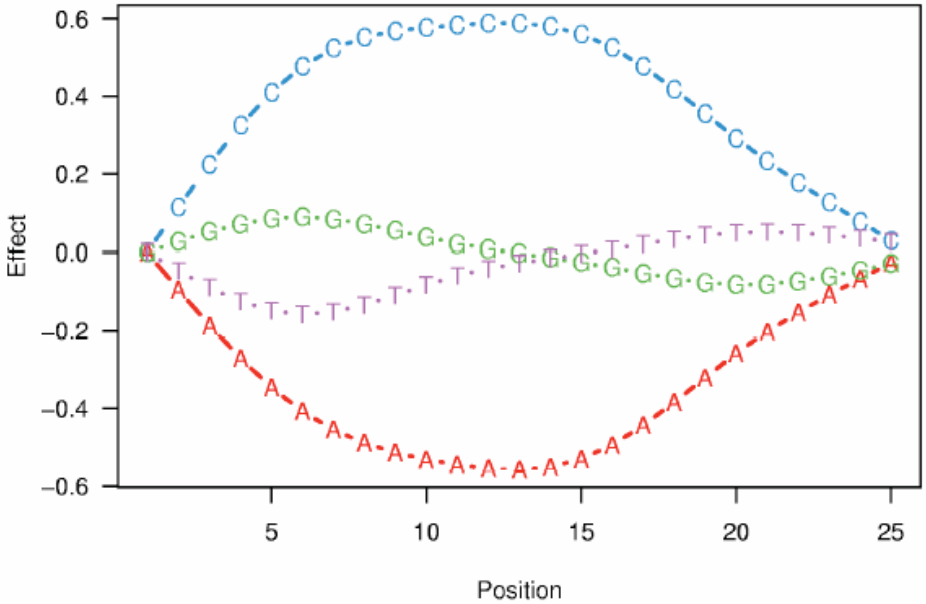
Naef & Magnasco, 2003

Background Correction – GCRMA

- GCRMA uses the sequence information to compute an *affinity* measure and adjust the background accordingly.



G-C has three hydrogen bonds. (stronger)
A-T has two hydrogen bonds. (weaker)



R: Background Adjustment

> bgc = bg.correct(Data, *method*)

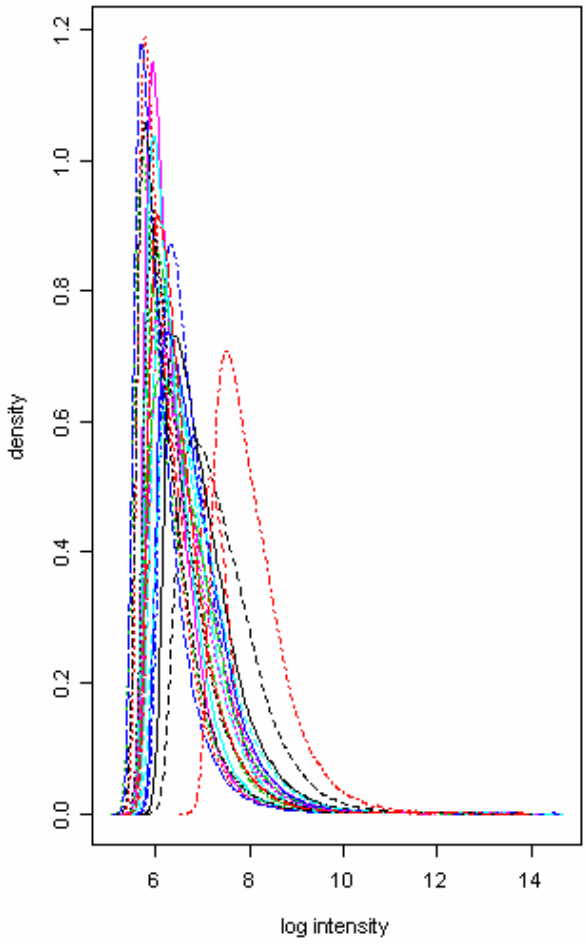
```
> bgcorrect.methods
[1] "mas"  "none" "rma"  "rma2"
```

> library(gcrma)

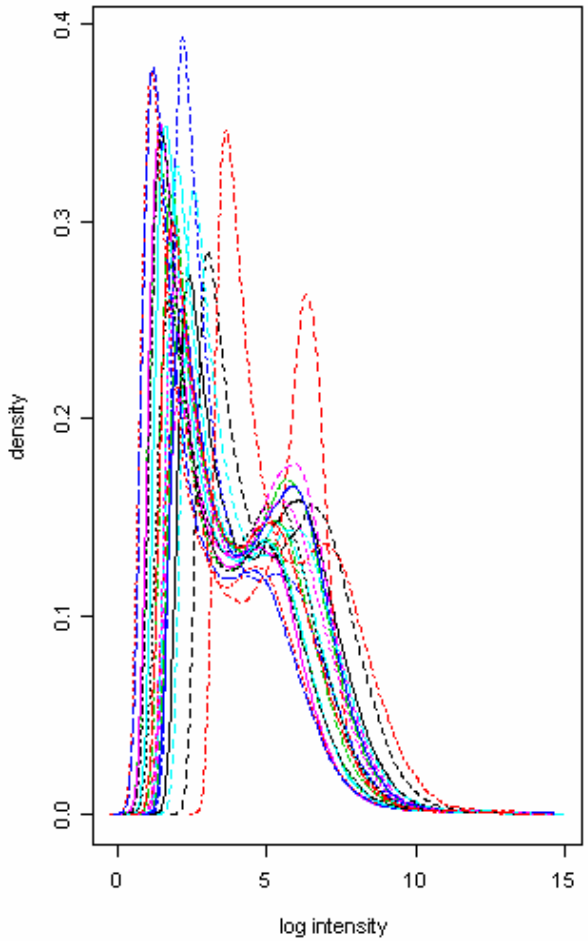
> bgc.gcrma = gcrma(Data)

includes bg.correct, normalization and summarization

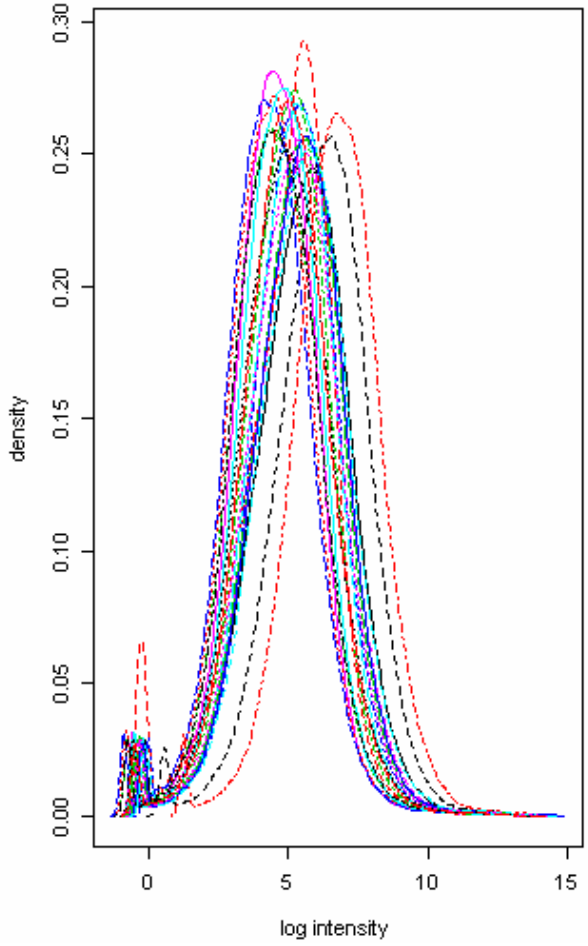
hist(Data)



hist(bgc.rma)



hist(bgc.mas)



Preprocessing for Affymetrix

Background / PM adjustment

- PM-MM
- MAS 5.0
- RMA
- GC-RMA

```
> bgcorrect.methods
[1] "mas"  "none" "rma"  "rma2"
```

Normalization

- Constant scaling
- Contrasts
- Invariant set
- Cyclic loess
- Quantile

```
> normalize.methods(Data)
[1] "constant"
[2] "contrasts"
[3] "invariantset"
[4] "loess"
[5] "qspline"
[6] "quantiles"
[7] "quantiles.robust"
```

Constant Scaling

- Distributions on each array are scaled to have identical mean (mean of the reference array).

Suppose array 1 is the reference array,

$$x'_{gs} = \frac{x_{\cdot 1}}{x_{\cdot s}} x_{gs}$$

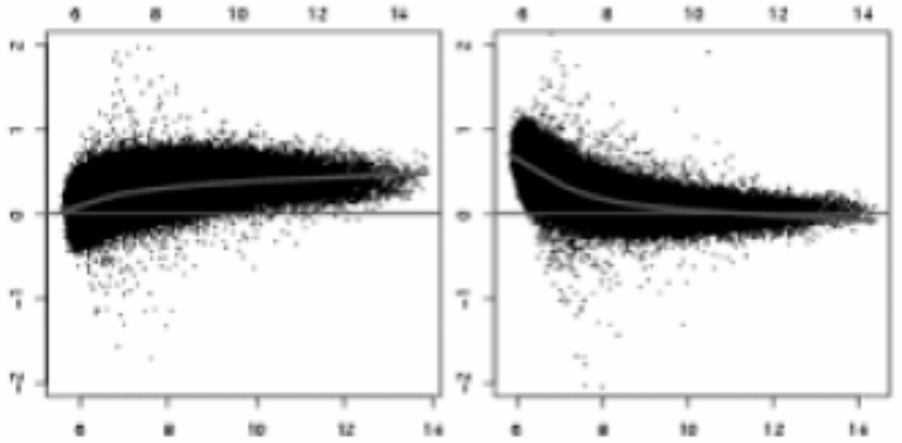
Invariant Set

- Select a baseline array (default is the one with median average intensity).
- For each “treatment” array, identify a set of genes that have ranks conserved between the baseline and treatment array. This set of rank-invariant genes are considered non-differentially expressed genes.
- Each array is normalized against the baseline array by fitting a non-linear normalization curve (loess) of invariant-gene set.

Cyclic Loess

(Yang et al., 2002)

- Using all genes to fit a non-linear normalization curve at the M-A plot scale.
- Perform normalization between arrays pairwisely, repeating the entire process until converegnce.
- Has been extended to perform normalization globally without selecting a baseline array but then is time-consuming.



Quantile Normalization

0.974[4]	0.341[3]	0.411[2]	-0.951[1]
1.857[5]	-0.036[2]	0.634[3]	-0.055[2]
-0.386[3]	-1.461[1]	0.885[4]	0.196[3]
-0.539[2]	0.368[4]	-0.530[1]	0.742[4]
-0.862[1]	2.634[5]	2.340[5]	2.277[5]

= (-0.862 – 1.461 – 0.951)/3

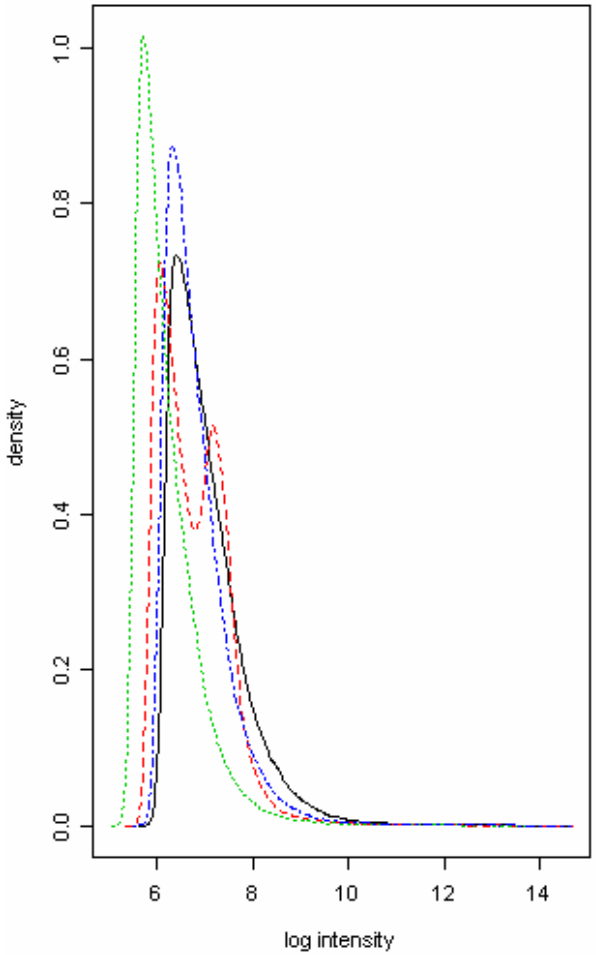
0.742[4]	0.196[3]	-0.055[2]
2.277[5]	-0.055[2]	0.196[3]
0.196[3]	-0.951[1]	0.742[4]
-0.055[2]	0.742[4]	-0.951[1]
-0.951[1]	2.277[5]	2.277[5]

R: Normalization

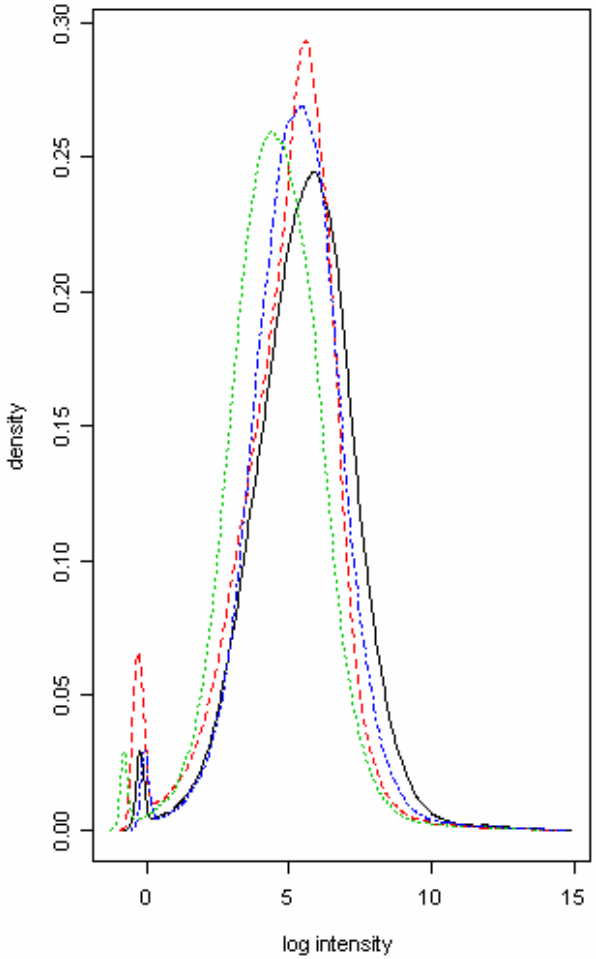
> normc = normalize(Data, *method*)

```
> normalize.methods(Data)
[1] "constant"
[2] "contrasts"
[3] "invariantset"
[4] "loess"
[5] "qspline"
[6] "quantiles"
[7] "quantiles.robust"
```

hist(Data[,1:4])



hist(bgc.mas[,1:4])



hist(normc.quantiles
.after.bgc[,1:4])