

---

# Quality Checks for Microarray Data

---

Utah State University – Spring 2014  
STAT 5570: Statistical Bioinformatics  
Notes 2.1

---

# References & Reminder

- Chapter 3 of Bioconductor Monograph (course text)
- (Same issues here recur in other technologies)

---

# Recall Background Correction and Normalization

- Why background correction? -  
to remove “noise” and “local artifacts”
- Why normalization? -  
to make arrays more comparable
- What is the intended effect of both steps? -  
to make spot-level data comparable -  
so differences are biologically meaningful

---

# Is it enough?

- Even after background correction and normalization-
    - some arrays are “beyond correction”
  - How to assess this?
    - Sometimes obvious -
      - array image has glaring problem
    - Sometimes more subtle -
      - a “local” problem on one array
  - Rely on – graphical checks
-

---

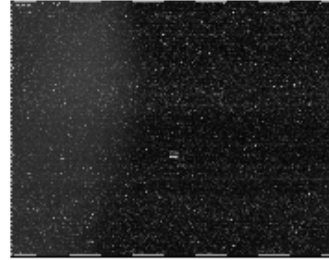
# Example data: ALL MLL

- A large acute lymphoblastic leukemia (ALL) study using the HGU133A and HGU133B (human) arrays
- A subset of the ALL data is provided in the MLL.B AffyBatch object
- Let's just look at 8 arrays

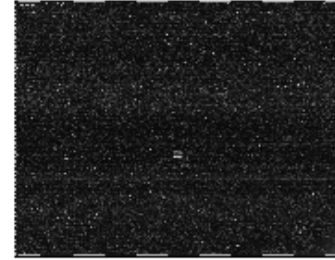
**Array 1**



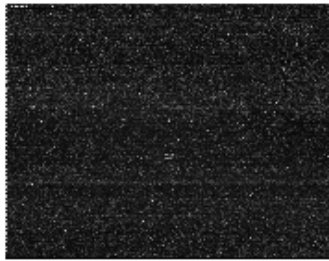
**Array 2**



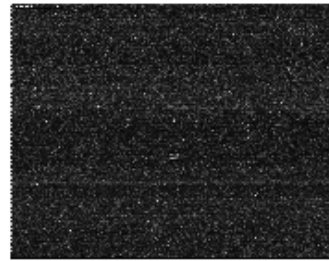
**Array 3**



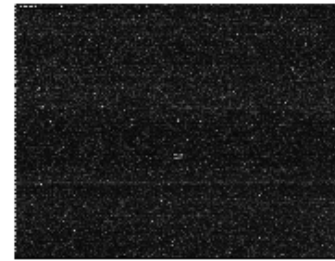
**Array 4**



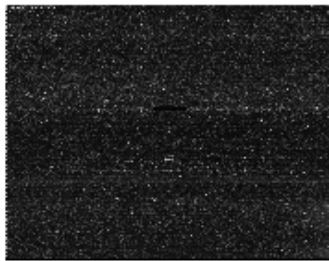
**Array 5**



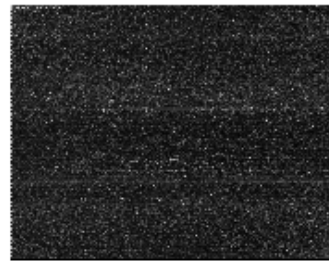
**Array 6**



**Array 7**



**Array 8**

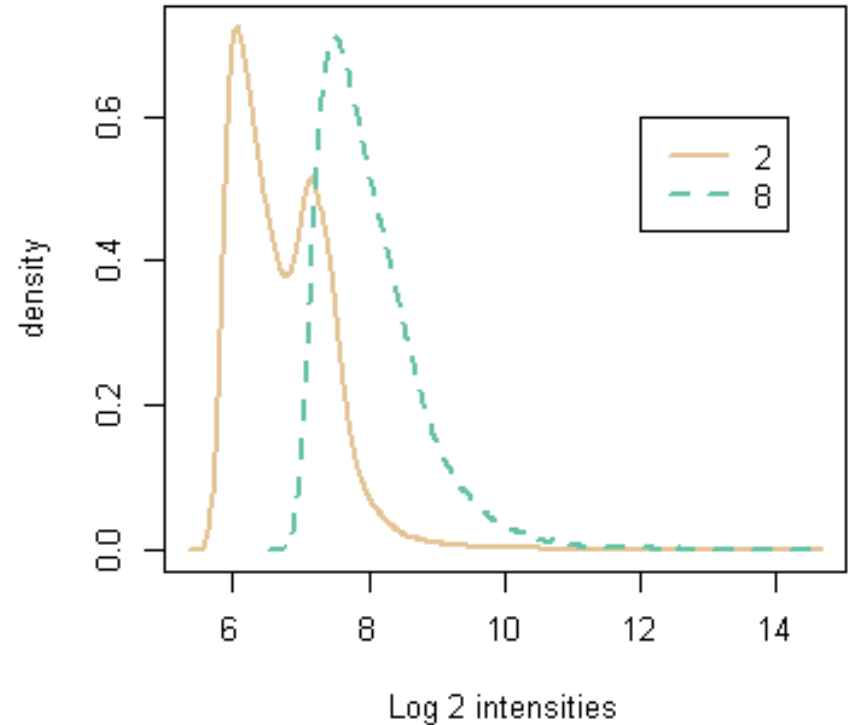
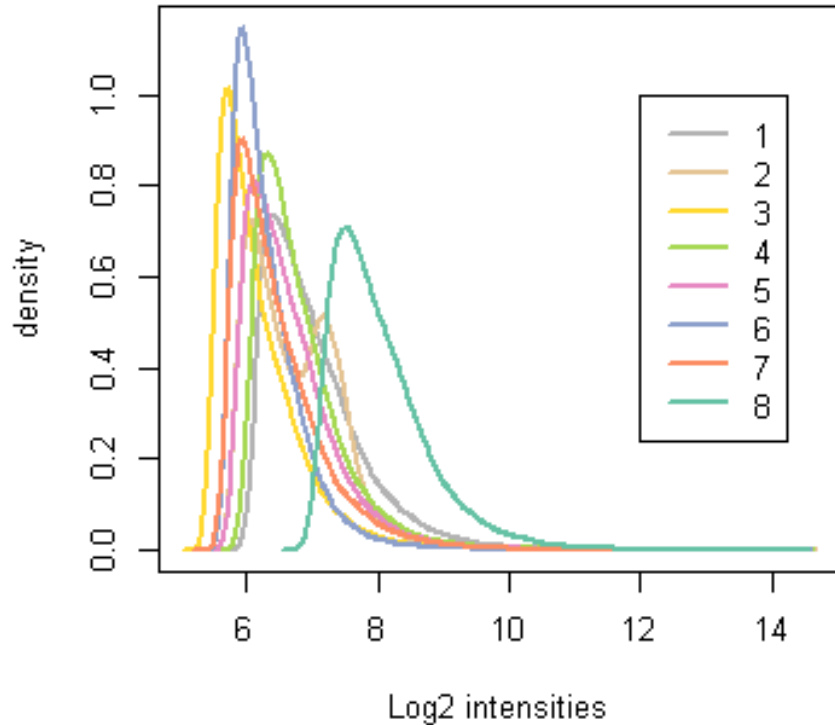


# Image – compare arrays (#2?)

```
library(affy)
library(ALLMLL)
data(MLL.B)
Data <- MLL.B[,c(1:6,13,14)]

par(mfrow=c(3,3))
for(i in 1:8)
{
  image(Data[,i], main=paste('Array',i),
        cex.main=2)
}
```

# [Smoothed] Histograms



Bimodality suggests -  
a spatial artifact

# Histograms – compare arrays (#2!)

```
library(RColorBrewer)
par(mfrow=c(2,2))
cols <- rev(brewer.pal(8, "Set2"))

hist(Data,col=cols, lty=1,
      xlab="Log2 intensities",lwd=2)
legend(12,1,1:8,lty=1,col=cols,lwd=2)

hist(Data[,c(2,8)],col=cols[c(2,8)],
      lty=c(1,2),xlab="Log 2 intensities",lwd=2)
legend(12,0.6,c(2,8),lty=c(1,2),
      col=cols[c(2,8)],lwd=2)
```

# MA plot – compare arrays

- MA plot:  $M=Y-X$  vs.  $A=0.5(Y+X)$ 
  - Rotate and scale Y vs. X scatterplot
  - For log-scale expression on arrays Y and X:
    - $M = \log \text{ fold-change (Y vs. X)}$
    - $A = \text{average expression}$
- For comparing multiple arrays, create a “pseudo-array” reference by taking the median for each probe across all arrays
- Loess curve:  
locally weighted polynomial regression

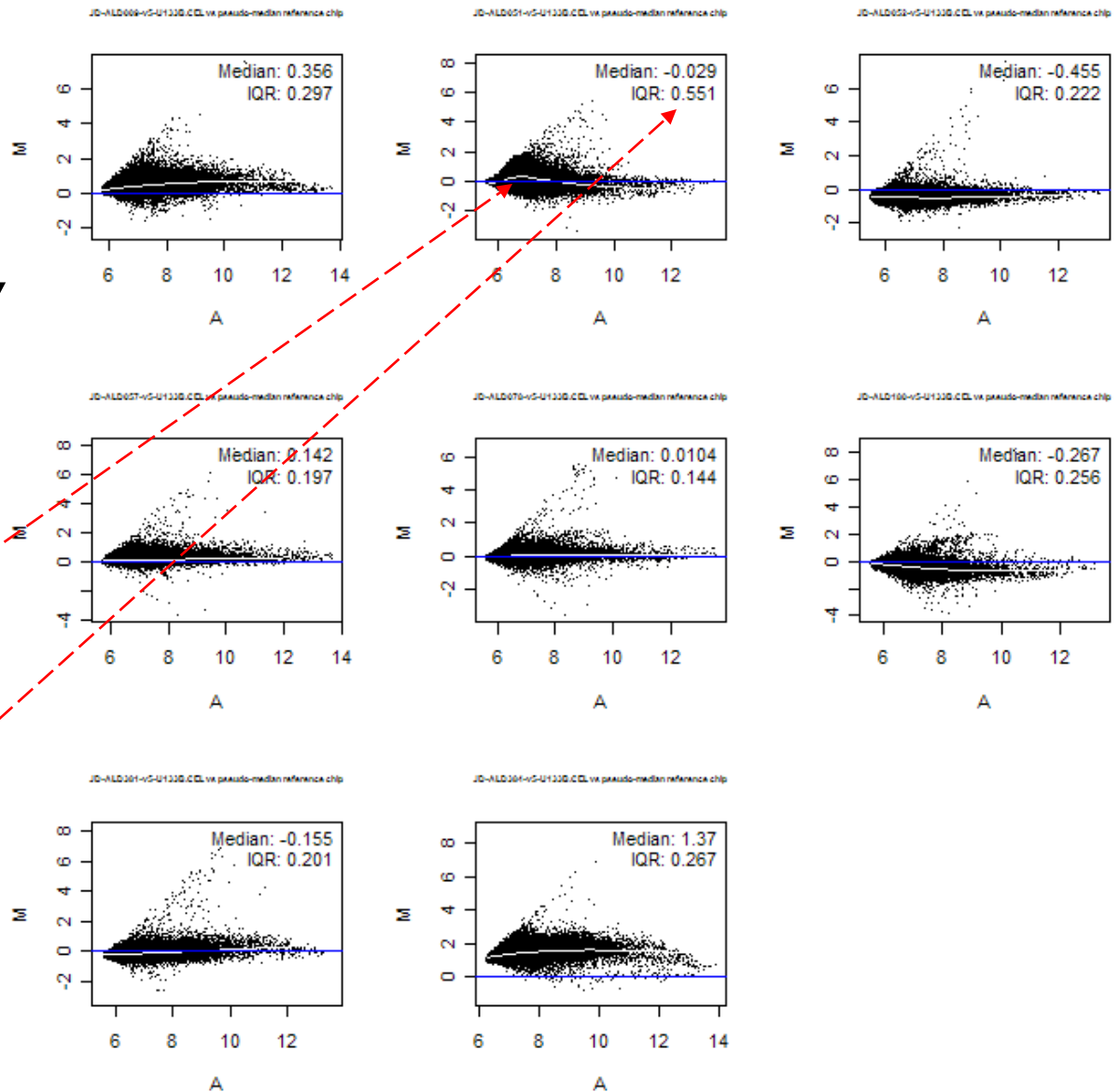
# MA plot

```
par(mfrow=c(3,3))  
MAplot(Data,  
  loess.col='white',  
  cex=1,  
  cex.main=0.5)
```

```
# this can take  
# a few minutes
```

Quality problems  
most apparent when:

- Loess line oscillates much
- M-variability is much greater than other arrays



# PLM Image

- Probe Level Model
- Recall RMA model (Notes 1.4):

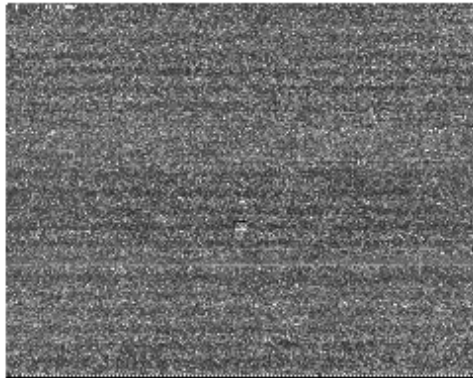
$$Y_{ijk} = \underbrace{\mu_{ik}} + \underbrace{\alpha_{jk}} + \varepsilon_{ijk}$$

Probe affinity effect

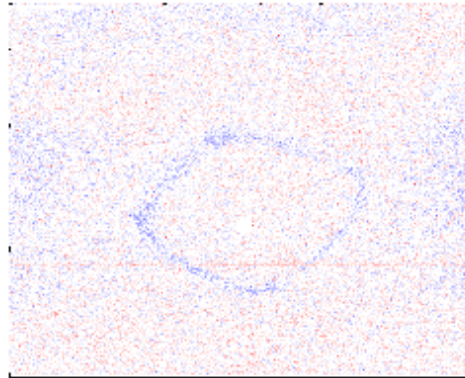
↑  
Log-scale expression level for gene k on array i

- Use robust measure to estimate model parameters
- To identify quality problems, look at residuals

**default image**



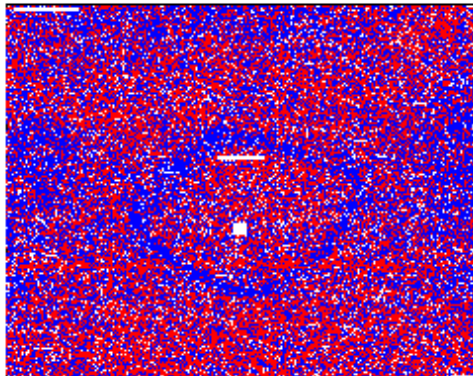
**5\_Std\_TB\_75-6**



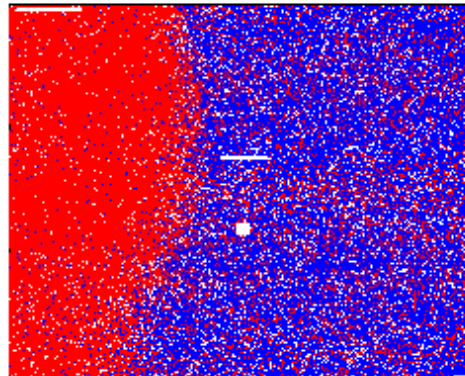
Look for:

- Substantial artifacts
- Systematic patterns

**5\_Std\_TB\_75-6**



**JD-ALD051-v5-U133B.CEL**



# PLM Image – look at residuals

```
library(affyPLM)
library(AmpAffyExample)
data(AmpData)
par(mfrow=c(2,2))

# Fit RMA PLM to data; could also use fitPLM function
Pset1 <- rmaPLM(AmpData)
image(AmpData[,3],main='default image', cex.main=2)
image(Pset1, type="resids", which=3)
image(Pset1, type="sign.resids",which=3)

# Fit RMA PLM to other data; look at problem array
Pset2 <- rmaPLM(Data)
image(Pset2, type="sign.resids",which=2)

# Could also consider type="pos.resids"
# or type="neg.resids"
```

---

# Summary

- Use graphical checks to look at microarray data quality
  - Image
  - Histogram
  - MA plot
  - PLM (residual) image
- Also consider:
  - Boxplots
  - RNA degradation (3'/5' ratios)
  - Normalized Unscaled Standard Error (NUSE) plot