RNA Degradation and NUSE Plots

Austin Bowles
STAT 5570/6570
April 22, 2011
References

- Sections 3.4 and 3.5.1 of Bioinformatics and Computational Biology Solutions Using R and Bioconductor (Gentleman et al., 2005)
Notes 2.1 – Quality Checks

- Even after preprocessing to remove “noise” and make arrays comparable, some arrays may be “beyond correction.”

- Visual checks of microarray quality (from Notes 2.1)
  - Array Images
  - Boxplots/Histograms
  - MA Plot
  - PLM Images/Residual Images

- Also:
  - RNA Degradation
  - Normalized Unscaled Standard Error (NUSE) Plots
Why RNA Degradation?

• Once RNA has reached the end of its useful life (i.e. it has participated in protein synthesis) it is “degraded” by cellular enzymes.

• Some arrays might have been prepared using a sample with “bad” RNA
  –RNA that has been degraded past the point of providing useful information.

• We need a way to check whether or not our arrays have “good” RNA.
5’ to 3’ Ordering of Probesets

- A gene is represented by a probeset on a microarray.

- Each probeset consists of about 11 different PM probes.

- For each probeset on an array, the individual probes are numbered sequentially from the 5’ end of the transcript to the 3’ end.
RNA Degradation in Probesets

- Due to the specific mechanisms of RNA degradation, probe intensities should be systematically lower towards the 5’ end of a transcript than towards the 3’ end.

- This fact can be exploited for analyzing expression array data.
RNA Degradation for One Probeset

• Even with minimal degradation, we should see an upward trend in expression levels as probe number increases.

• The slope of this trend depends on the probeset and the extent of the degradation.

• We need a way to look at all probesets on an array.
The “AffyRNAdeg” Function

\[ Y_{ij} = \text{the log transformed intensity for the } j^{th} \text{ probe in the } i^{th} \text{ probeset.} \]

\[ Y_j = \text{the average log intensity at the } j^{th} \text{ position, taken over all probesets in an array.} \]

- Plotting the \( Y_j \) vs. \( j \) shows a linear 3’/5’ trend, even in an experiment with “good” RNA.
library(affy); library(affydata)
abatch.raw <- ReadAffy(celfile.path="F:\R\Data\GSE5425")
hw.rnadeg.1 <- AffyRNAdeg(abatch.raw[,1])
plotAffyRNAdeg(hw.rnadeg.1,"neither")
legend(0,40,"Array 1",lty=1,col=4,lwd=2)
Rescaling

- To make these plots comparable across arrays, we first rescale them so that the standard errors of the rescaled means are approximately 1.

\[ \hat{\sigma}_j = \text{standard deviation for probes at position } j \text{ in all } N \text{ probesets} \]

\[ \hat{\sigma}_j / \sqrt{N} \approx \text{standard error for mean intensity at position } j \]

\[ \frac{Y_j}{\hat{\sigma}_j / \sqrt{N}} = \text{rescaled mean with standard error } \approx 1 \]
The lines have been shifted from the original data for a clearer view, but the slopes remain unchanged.

plotAffyRNAdeg(hw.rnadeg)
What to look for:

• Slopes deviating from the group pattern.

• Remember Array #3 from our homework?
cols <- rep(4,16); cols[3]<-2
plotAffyRNAdeg(hw.rnadeg,cols=cols)
legend(0,60,c("Array 3","Others"),lty=1,col=c(2,4),lwd=2)
summaryAffyRNAdeg(hw.rnadeg)

<table>
<thead>
<tr>
<th>Array ID</th>
<th>Slope</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSM118665.CEL</td>
<td>4.38e+00</td>
<td>1.12e-08</td>
</tr>
<tr>
<td>GSM118666.CEL</td>
<td>4.21e+00</td>
<td>2.52e-10</td>
</tr>
<tr>
<td>GSM118667.CEL</td>
<td>5.52e+00</td>
<td>4.43e-12</td>
</tr>
<tr>
<td>GSM118668.CEL</td>
<td>4.58e+00</td>
<td>1.31e-08</td>
</tr>
<tr>
<td>GSM118669.CEL</td>
<td>4.04e+00</td>
<td>3.07e-11</td>
</tr>
<tr>
<td>GSM118671.CEL</td>
<td>4.22e+00</td>
<td>1.17e-08</td>
</tr>
<tr>
<td>GSM118673.CEL</td>
<td>4.01e+00</td>
<td>1.07e-09</td>
</tr>
<tr>
<td>GSM118674.CEL</td>
<td>5.25e+00</td>
<td>3.62e-10</td>
</tr>
<tr>
<td>GSM118677.CEL</td>
<td>4.14e+00</td>
<td>3.96e-08</td>
</tr>
<tr>
<td>GSM118679.CEL</td>
<td>3.92e+00</td>
<td>3.70e-09</td>
</tr>
<tr>
<td>GSM118681.CEL</td>
<td>4.00e+00</td>
<td>4.53e-08</td>
</tr>
<tr>
<td>GSM118682.CEL</td>
<td>3.77e+00</td>
<td>1.01e-09</td>
</tr>
<tr>
<td>GSM118684.CEL</td>
<td>4.95e+00</td>
<td>2.61e-10</td>
</tr>
<tr>
<td>GSM118686.CEL</td>
<td>4.16e+00</td>
<td>1.64e-08</td>
</tr>
<tr>
<td>GSM118687.CEL</td>
<td>3.58e+00</td>
<td>9.56e-09</td>
</tr>
<tr>
<td>GSM118689.CEL</td>
<td>4.98e+00</td>
<td>1.27e-10</td>
</tr>
</tbody>
</table>

**Array #3**
Example: ALLMLL Data

- RNA Degradation Plots can help identify physical artifacts, but should not be used exclusively.
library(ALLMLL); data(MLL.B)
ALLdata <- MLL.B[,c(1:6,13,14)]
ALL.rnadeg <- AffyRNAdeg(ALLdata)
cols <- rep(4,8); cols[2]=2
plotAffyRNAdeg(ALL.rnadeg, cols=cols)
legend(0,17,c("Array 2","Others"),lty=1,col=c(2,4),lwd=2)

• Instead, we should look for derivations that suggest an
  array was prepared with “bad” RNA.

library(AmpAffyExample)
data(AmpData)
AmpData
#Array S3 had issues in lecture notes
sampleNames(AmpData) <- c("S1","S2","S3","A1","A2","A3")
AmpData.rnadeg <- AffyRNAdeg(AmpData)
par(mfrow=c(1,1))
plotAffyRNAdeg(AmpData.rnadeg, col=c(2,2,2,4,4,4))
legend(0,60,sampleNames(AmpData),lty=1,col=c(2,2,2,4,4,4),lwd=2)
Example: AmpAffyExample Data

- The steeper slopes for the A samples suggest that the RNA degradation has advanced further than for the S samples.
How steep is too steep?

• Unfortunately, different array types have different “normal” slopes. Experience will give the user a sense of what is typical for the different arrays.

• Instead, it is important to check for agreement within studies.
  • If the arrays in a study have similar slopes, then within gene comparisons may still be valid.
  • If the arrays have different degrees of RNA degradation (i.e. different slopes), extra bias is introduced into the experiment.
NUSE Plots

- Normalized Unscaled Standard Error Plots.

- Remember the Probe Level Model:
  \[ Y_{ijk} = \mu_{ik} + \alpha_{jk} + \varepsilon_{ijk} \]
  - Log-scale expression level for gene k on array i
  - Probe Affinity Effect

- The R function `fitPLM` fits the Probe Level Model and other functions are available for accessing its output.
Standard Error Estimates

We’re interested in the Standard Error Estimates, \(SE(\hat{\mu}_{ik})\), obtained by the PLM fit.

```r
Pset1 <- fitPLM(abatch.raw)
# NOTE: rmaPLM(abatch.raw) will not return standard errors
se(Pset1)[1:6,1:3]
```

<table>
<thead>
<tr>
<th></th>
<th>GSM118665.CEL</th>
<th>GSM118666.CEL</th>
<th>GSM118667.CEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1415670_at</td>
<td>0.10637111</td>
<td>0.11596710</td>
<td>0.11225729</td>
</tr>
<tr>
<td>1415671_at</td>
<td>0.11275118</td>
<td>0.12974944</td>
<td>0.11359738</td>
</tr>
<tr>
<td>1415672_at</td>
<td>0.11465433</td>
<td>0.13025307</td>
<td>0.11655682</td>
</tr>
<tr>
<td>1415673_at</td>
<td>0.12740979</td>
<td>0.14139465</td>
<td>0.12500363</td>
</tr>
<tr>
<td>1415674_a_at</td>
<td>0.08401818</td>
<td>0.09562223</td>
<td>0.08614656</td>
</tr>
<tr>
<td>1415675_at</td>
<td>0.12846968</td>
<td>0.14957198</td>
<td>0.12846968</td>
</tr>
</tbody>
</table>

Since variability differs between genes, we standardize these standard errors so that the median SE across arrays is 1 for each gene.
\[
\text{NUSE}(\hat{\mu}_{ik}) = \frac{\text{SE}(\hat{\mu}_{ik})}{\text{med}_i(\text{SE}(\hat{\mu}_{ik}))}
\]

- We visualize these NUSE values with a boxplot. This can be done with `boxplot()` or ` NUSE()`.

- When examining these plots, look for arrays with boxes that are significantly elevated or more spread out than other arrays. These indicate lower quality arrays.

```r
par(mfrow=c(1,2))
cols <- c(brewer.pal(8, "Set3"), brewer.pal(8, "Dark2"))
boxplot(abatch.raw[,1:16], col=cols, names=1:16,
    xlab = "Array number", ylab = "Intensities",
    main="Raw Boxplot")
NUSE(Pset1, col=cols, names=1:16, xlab = "Array number",
    ylab = "Intensities", main="NUSE Plot")
```
HW Data

Raw Boxplot

NUSE Plot
Pset2 <- fitPLM(ALLdata)
cols <- brewer.pal(8,"Set3")
boxplot(ALLdata,col=cols,names=1:8,xlab="Array number", ylab="Intensities", main="Raw Boxplot")
NUSE(Pset2,col=cols,names=1:8,xlab="Array number", ylab="Intensities", main="NUSE Plot")
Comparing median NUSE with Affymetrix quality metrics for the ALLMLL dataset. Array 2 is indicated on each plot.

Figure 3.8 courtesy of Course Text (Gentleman et al., 2005)
Summary

• RNA Degradation
  - Helps identify arrays with “bad” RNA.
  - Can sometimes help identify arrays with physical artifacts.
  - Samples with similar degradation can be used for valid comparisons.

• NUSE Plots
  - Can help identify lower quality arrays.
  - Easier than Affymetrix quality standards to distinguish problematic arrays.