

Holly Zheng Bradley (version 1.0 30<sup>th</sup> June, 2008)

## Quality Assessment of Microarray Gene Expression Data

This tutorial will introduce how to perform quality assessment (QA) of microarray gene expression data. Since Affymetrix GeneChip arrays are the most commonly used platforms for gene expression experiments, we will focus on QA of data generated on Affymetrix arrays using two Bioconductor packages *simpleaffy* and *affyPLM*.

### You will learn about

- Why it is important to QA microarray gene expression data
- How to do quality assessment using Bioconductor packages *simpleaffy* and *affyPLM*
- How to interpret the QA results

### Contents

- 1 Why it is important to QA microarray gene expression data
- 2 How to install R and standard Bioconductor packages
- 3 How to perform data QA using Bioconductor *simpleaffy* package
- 4 How to evaluate RNA degradation using RNA degradation slope
- 5 How to perform QA in a batch mode
- 6 How to interpret *simpleaffy* results and RNA degradation slope
- 7 How to perform data QA using Bioconductor *affyPLM* package
- 8 How to interpret the *affyPLM* QA results

## 1 Why it is important to QA your microarray gene expression data

Gene expression levels measured by microarray experiments are obtained through an elaborated procedure which is subject to many potential variations. It is critical to do adequate QA to make sure the data is of high quality and is consistent and comparable for further analysis. In some cases, arrays are too bad to be corrected, even with normalization; these arrays should be removed from further analysis.

When the microarray data is taken from a public repository for integrating study, it is especially essential to do data QA since the quality of the data varies greatly from submitter to submitter. It is important to filter out bad data to maintain data integrity.

Data quality is a relative term here. After the QA procedure, we would like to identify microarray data with lower variability; outliers in the quality assessment metrics are to be removed to ensure the homogeneity of the data. For data integration, data generated by different laboratories should be evaluated using the same QC procedures and compared under the same QC scale. Only data with comparable QA metrics can be put together for further analysis.

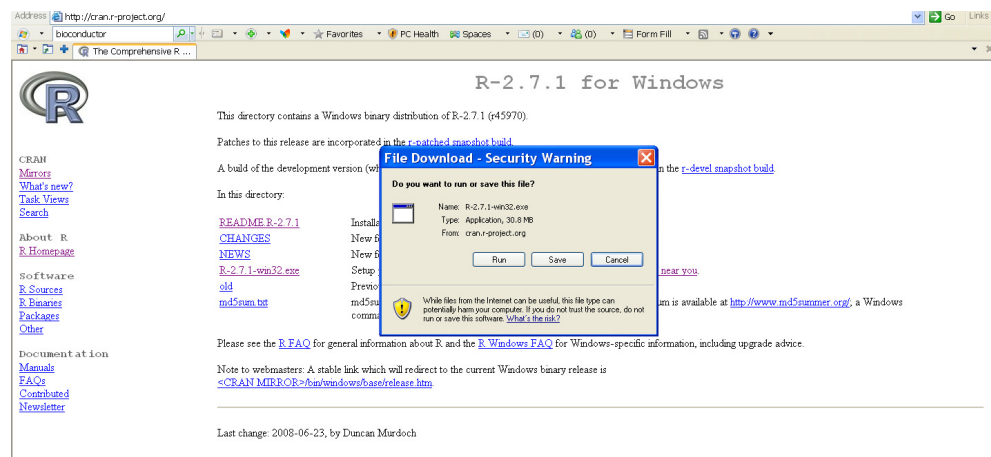
In the following sessions, you will learn how to perform QA on Affymetrix data using several Bioconductor packages.

## 2 How to install R and Standard Bioconductor packages

As a brief introduction, Bioconductor is an open source and open development software project to provide tools for the analysis and comprehension of genomic data. It is based primarily on programming language R, which is a free software environment for statistical computing and graphics. We first need to install R and standard Bioconductor packages on your machine.

### Install R

R can be downloaded from CRAN (Comprehensive R Archive Network) <http://www.r-project.org/>. On the CRAN site, find a mirror site nearest you. Use the latest version of R; make sure to choose the download designated for your preferred operating system. Then download the precompiled binary distributions of the base system. Refer to the installation instruction (the README files) on the CRAN website for more details.



### Install Standard Bioconductor packages

The installation can be accomplished by running the `biocLite.R` installation script. Start running R; in an R command window, type the following:

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

Make sure the computer is connected to internet during the installation. This installs the following packages: *affy*, *affydata*, *affyPLM*, *annaffy*, *annotate*, *Biobase*, *Biostrings*, *DynDoc*, *gcrma*, *genefilter*, *genefilter*, *hgu95av2.db*, *limma*, *marray*, *matchprobes*, *multtest*, *ROC*, *simpleaffy*, *vsu*, *xtable*, *affyQCReport*. The QA procedure we describe below will use the *simpleaffy* and *affyPLM* packages.

### 3 How to perform data QA using Bioconductor *simpleaffy* package

Once R and standard Bioconductor packages are installed, an Affymetrix QA metrics can be produced using package *simpleaffy*. Each CEL file is analyzed individually using *simpleaffy* and three QA metrics measurements are produced: average background, scale factors, and percent present. These QA measurements are recommended by Affymetrix to evaluate the quality of RNA samples and whether the labeling, hybridization and scanning procedures are done properly. High average background level would affect signal-to-noise ratio and is likely due to problems in RNA sample preparation. As for scale factor, the MAS 5.0 expression summary algorithm normalizes arrays by scaling them to a common value. If scale factors between arrays are large, then it is an indication that issues may occur when trying to compare between chips; present-and-absent calls can be used to flag genes as having been reliably detected. The percentage of present calls is used to provide an overall measure of quality. Large variations in present calls between similar samples can signal problems, especially when considered alongside scale factor and background level.

Here is how to obtain the QA metrics:

First load the *simpleaffy* package.

```
➤ library("simpleaffy")
```

Read a CEL file (using E-GEOD-1008-raw-cel-1508900008.cel as an example) and create an object *data*

```
➤ data <- ReadAffy("E-GEOD-1008-raw-cel-1508900008.cel")
```

Create Affymetrix QA metrics

```
➤ data.qc <- qc(data)
```

Obtain average background for the array:

```
➤ avgbg <- avbg(data.qc)
```

Obtain scale factor for the array:

```
➤ sfs <- sfs(data.qc)
```

Obtain percent present for the array:

```
➤ pp <- percent.present(data.qc)
```

To print out the data metrics values to file E-GEOD-1008-raw-cel-1508900008.cel.qc, do

```
➤ sink("E-GEOD-1008-raw-cel-1508900008.cel.qc")
```

```
➤ c("platform",cdfName(data))
```

- `c("affy_average_background",prettyNum(as.double(avgbg)))`
- `c("affy_scale_factor",prettyNum(sfs))`
- `c("affy_percent_present",prettyNum(as.double(pp)))`
- `sink()`

Here is an example output file E-GEOD-1008-raw-cel-1508900008.cel.qc

```
[1] "platform" "MG_U74Av2"  
[1] "affy_average_background" "580.3727"  
[1] "affy_scale_factor" "0.08205732"  
[1] "affy_percent_present" "43.70596"
```

## 4 How to evaluate RNA degradation using RNA degradation slope

A fourth QA measurement one can get for Affymetrix data is RNA degradation slope. For each probeset, it basically numbers individual probes sequentially from 5' end to the 3' end. If the probes towards the 3' end are systematically stronger than those towards the 5' end, the RNA degradation slope would be excessively big and indicating RNA degradation. Bioconductor has a package *AffyRNAdeg* that does the calculation:

- `RNAdeg <- AffyRNAdeg(data)`

To print the RNAdeg value out, do

- `sink("E-GEOD-1008-raw-cel-1508900008.cel.rna")`
- `c("RNAdegSlope",prettyNum(RNAdeg$slope))`
- `sink()`

For E-GEOD-1008-raw-cel-1508900008.cel, the RNA degradation slope is

```
[1] "affy_RNAdeg_slope" "0.0630135"
```

## 5 How to perform QA in a batch mode

Frequently, one needs to QA many CEL files and it is not efficient to run *simpleaffy* and *AffyRNAdeg* one CEL file a time. We have developed an R script that can be called for batch running of the QA processes described above. Please refer to the appendix for the script.

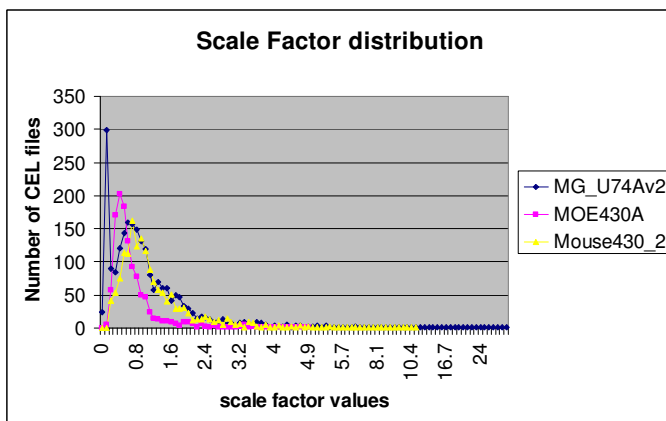
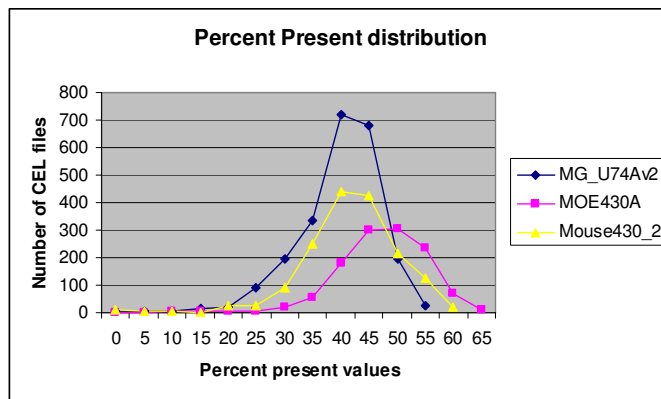
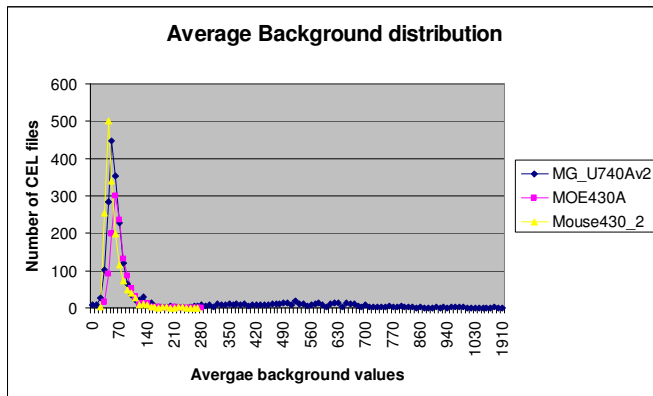
## 6 How to interpret the *simpleaffy* results and RNA degradation slope

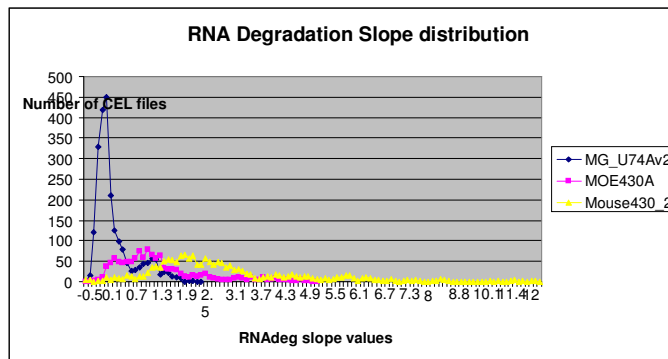
As of evaluating array data quality using the QA metrics, the basic guideline from Affymetrix is to make sure all arrays included in a study should be more or less comparable for their average background, scale factor values and

percent present values. Arrays with very high average background should be discarded from further analysis. For scale factors, the values should be within 3-fold of each other. Extremely low value of percent present is an indication of poor quality.

A typical RNA degradation slope varies by chip types. For high quality RNA, a slope of 0.5 is typical for HG-U95 and MG-U74 chips, while 1.7 is typical for HG-U133A chips. Slopes that are 2 fold or higher than these numbers indicate RNA degradation.

You may plot the QA metrics values on a diagram and decide what the cutoff should be used to remove outliers. Figures 1-4 are some example of QA metrics distribution. Based on the distribution, it is easier to determine the cutoff values.





Figures 1 -4. Distribution of Affymetrix QA metrics values and RNA degradation slope. The three lines with different colors represent three different datasets generated on different array platforms.

## 7 How to perform data QA using Bioconductor *affyPLM* package

In addition to the *simpleaffy* assessments, the CEL files can be further evaluated with probe level model (PLM). Bioconductor *affyPLM* package can be used to do PLM fitting for CEL files (Bolstad 2005). Unlike *simpleaffy* and *AffyRNAdeg*, in which CEL file is evaluated one at a time, *affyPLM* takes all CEL files within one dataset and generates a probe level model object. Numerous useful quality assessment tools have been derived from the PLM fitting procedure. Here we introduce how to use two of the methods – the Relative Log Expression (RLE) procedure and the Normalized Unscaled Standard Error (NUSE) procedure.

In a machine with large memory, load appropriate packages:

- `require(affy)`
- `require(simpleaffy)`
- `require(affyPLM)`

Read a list of CEL files:

- `files <- list.files("directory_name", "cel", full.names=TRUE)`

Generate data object:

- `data <- ReadAffy()`

PLM fitting on the *data* object:

- `plmStruct <- fitPLM(data)`

Calculate and print RLE and NUSE stats to files *rle.out* and *nuse.out*:

- `sink(./rle.out)`
- `RLE(plmStruct, type = "stats")`
- `sink()`

- `sink(/nuse.stats)`
- `NUSE(plmStruct, type = "stats")`
- `sink()`

Example output for a RLE QA run; the format of the output of a NUSE run is the same as this:

E-GEOD-1008-raw-cel-1508899808.cel		E-GEOD-1008-raw-cel-1508899818.cel
median	-0.09745975	0.04404993
IQR	0.59707186	0.45582381
E-GEOD-1008-raw-cel-1508899826.cel		E-GEOD-1008-raw-cel-1508899834.cel
median	0.0508637	0.07331429
IQR	0.4090146	0.48258093
E-GEOD-1008-raw-cel-1508899844.cel		E-GEOD-1008-raw-cel-1508899854.cel
median	-0.06030324	0.03250762
IQR	0.37609944	0.39949380

## 8 How to interpret the *affyPLM* QA results

RLE values are computed for each probeset by comparing the expression value on each array against the median expression value for that probeset across all arrays. Assuming that most genes do not change in expression levels across arrays, it means that in theory the RLE values for these arrays should be close to 0. The RLE values refer to the mean RLE values as shown in the above example. If you plot the RLE values for all arrays in a diagram such as Figure 5, a reasonable cutoff can be picked to remove arrays with poor quality.

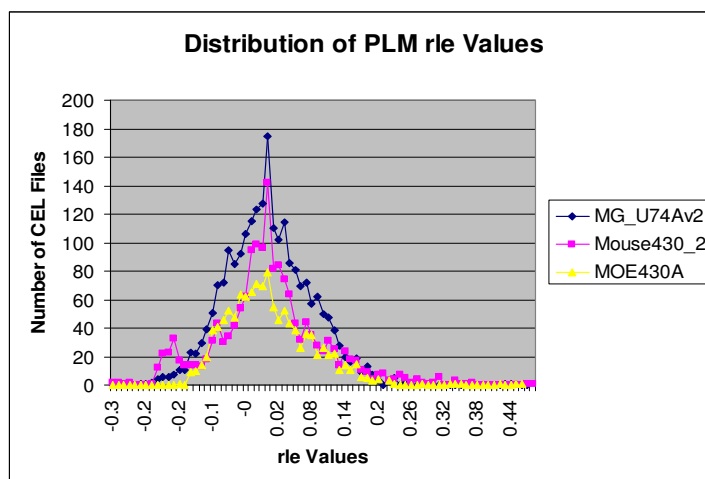


Figure 5. Distribution of *affyPLM* RLE values. The three lines with different colors represent three different datasets generated on different array platforms.

When NUSE is calculated, the standard error estimates obtained for each gene on each array from *fitPLM* are taken and standardized across arrays so that the median standard error for that gene is 1 across all arrays. This process accounts for differences in variability between genes. An array with elevated NUSE relative to the other arrays is typically of lower quality. The NUSE mean values can be plotted and a reasonable cutoff can be chosen. See Figure 6 for an example distribution plot of NUSE values for three different datasets.

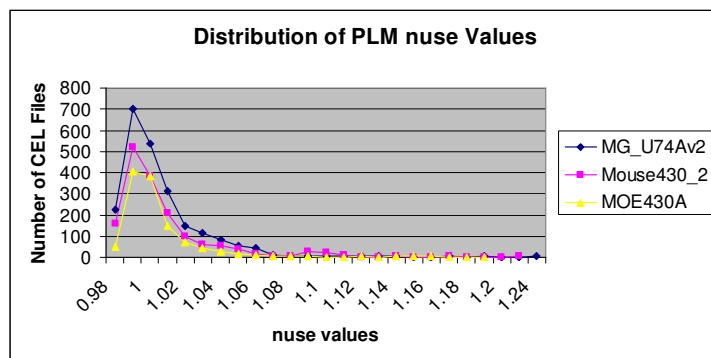


Figure 6. Distribution of *affyPLM* NUSE values. The three lines with different colors represent three different datasets generated on different array platforms.

## Appendix

### A R script that allows QA CEL files in batch mode

```
# Run using the following command:
# R CMD BATCH -input=<cel file> -output=<output file> <this script file>.
input<-FALSE;
output<-FALSE;

# This is a fairly generic options parsing loop, modified to store -input and -output values. Taken
# from https://sws.stat.iastate.edu/resources/programmingExamples/R/R-Command-Line.html
for (e in commandArgs()) {
  ta = strsplit(e,"=",fixed=TRUE);
  if(! is.na(ta[[1]][2])) {
    temp = ta[[1]][2];
    if(substr(ta[[1]][1],nchar(ta[[1]][1]),nchar(ta[[1]][1])) == "I") {
      temp = as.integer(temp);
    }
    if(substr(ta[[1]][1],nchar(ta[[1]][1]),nchar(ta[[1]][1])) == "N") {
      temp = as.numeric(temp);
    }
    assign(ta[[1]][1],temp);
    if ( ta[[1]][1] == "-input" ) {
      input = temp;
    }
    if ( ta[[1]][1] == "-output" ) {
      output = temp;
    }
  } else {
    assign(ta[[1]][1],TRUE);
  }
}

library('simpleaffy');
scorecel <- function(infile, outfile) {
  celdata<-ReadAffy(filename=infile);
  celqc<-qc(celdata);
  RNAdeg=AffyRNAdeg(celdata);
  celavgbg=avbg(celqc);
  perc=percent.present(celqc);
  # Each tag here will correspond to a qc type in the tracking DB. The
  # exception is "platform" which has its own table.
  capture.output(c("platform",cdfName(celdata)),
```

```
        file=outfile, append=FALSE);
capture.output(c("affy_average_background",prettyNum(as.double(celavgbg))),
              file=outfile, append=TRUE);
capture.output(c("affy_scale_factor",prettyNum(sfs(celqc))),
              file=outfile, append=TRUE);
capture.output(c("affy_percent_present",prettyNum(as.double(perc))),
              file=outfile, append=TRUE);
capture.output(c("affy_RNAdeg_slope",prettyNum(RNAdeg$slope)),
              file=outfile, append=TRUE);
}
scorecel(infile=input, outfile=output);
```

## Glossary

### Bioconductor

Bioconductor is an open source and open development software project to provide tools for the analysis and comprehension of genomic data. The website for Bioconductor is <http://www.bioconductor.org/>

### R

R is a free programming language and software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. The website is <http://www.r-project.org/>

### *simpleaffy*

simpleaffy is a [Bioconductor](#) package designed to provide a starting point for exploring Affymetrix data, and to provide functions for some of the most common tasks we found ourselves doing over and over again. It also provides access to many of the standard QC functions recommended for Affymetrix arrays.

### *affyPLM*

affyPLM is a package that extends and improves the functionality of the base affy package. Central focus is on implementation of methods for fitting probe-level methods and tools using these models. It also provides PLM based quality assessment tools. <http://www.bioconductor.org/packages/bioc/1.8/html/affyPLM.html>

### *ArrayExpress*

ArrayExpress is a public repository for transcriptomics data, which is aimed at storing [MIAME](#)- and [MINSEQE](#)-compliant data in accordance with [MGED](#) recommendations. The ArrayExpress Data Warehouse stores gene-indexed expression profiles from a curated subset of experiments in the repository.

[http://www.ebi.ac.uk/microarray-as/aer/#ae-main\[0\]](http://www.ebi.ac.uk/microarray-as/aer/#ae-main[0])

## Further reading

1. Claire L. Wilson and Crispin J. Miller (2005) *Simpleaffy*: a BioConductor package for Affymetrix Quality Control and data analysis. *Bioinformatics* 2005 21(18):3683-3685
2. Bolstad BM, Irizarry RA, Gautier L, and Wu Z. (2005) Preprocessing High-density Oligonucleotide Arrays in *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Gentleman R, Carey V, Huber W, Irizarry R, and Dudoit S. (Eds.), Springer, 2005.
3. Bolstad BM (2007) affPLM: Model based QC assessment of Affymetrix GeneChips. <http://www.bioconductor.org/packages/bioc/1.8/vignettes/affyPLM/inst/doc/QualityAssess.pdf>
4. Audrey Kauffmann of EBI is developing an R package which in combination with arrayQualityMetrics will allow direct QA of microarray data from ArrayExpress. arrayQualityMetrics is another QA package EBI developed; it can perform QA on arrays made by manufactures other than Affymetrix. We will introduce how to use these packages when the new one is released to Bioconductor.

### What to do next

Once you have read this tutorial, you might want to test your understanding by trying the QA processes using your own CEL files.