# Lab: Microarray data quality metrics

Audrey Kauffmann
Wolfgang Huber

August 14, 2007

## 1 Introduction

The workshop will provide an overview of diagnostic plots and metrics for assessing the quality of microarray datasets. Participants will learn how to produce these plots and compute these metrics on different example datasets using the R/Bioconductor software environment. The use of the arrayQualityMetrics package allows the production of a report assessing the quality of the experiments. The input object of the function *arrayQualityMetrics* can be an *AffyBatch*, an *ExpressionSet* for non Affymetrix one channel assays, or a *NChannelSet* for dual channels assays. The quality metrics are still object of research and under development.

## 2 Quality report for AffyBatch objects

First load the arrayQualityMetrics package.

```
> library("arrayQualityMetrics")
```

### 2.1 Data import

```
> library("ALLMLL")
> data("MLL.A")
> MLL.A
```

```
AffyBatch object
size of arrays=712x712 features (10 kb)
cdf=HG-U133A (22283 affyids)
number of samples=20
number of genes=22283
annotation=hgu133a
notes=
```

*AffyBatch* is an S4 class and `MLL.A` is an instance of this class.

```
> class(MLL.A)
```

```
[1] "AffyBatch"
attr(,"package")
[1] "affy"
```

## 2.2 Report production

To produce a report, the function *arrayQualityMetrics* is called with the following arguments:

- *expressionset*: is an object of class *ExpressionSet*, *AffyBatch* or *NChannelSet*.
- *outdir*: is the directory in which the result files are created.
- *force*: if TRUE, if outdir already exists, it will be overwritten.
- *do.logtransform*: if TRUE, the data are log transformed before the analysis.
- *split.plots*: if the number of studied arrays is more than 50 it is adviced to define a number of experiments to represent on the density plots.

```
> arrayQualityMetrics(expressionset = MLL.A,
+                     outdir = "MLL",
+                     force = FALSE,
+                     do.logtransform = TRUE,
+                     split.plots = 10)
```

A report named `QMreport.html` is produced in the subdirectory *MLL*. It contains text illustrated by `.png` files. Each `.png` is linked to corresponding `.pdf` files in order to provide high quality images.

**Exercise 1**
*Open the report in a web browser and read it. In the first section of the report, which arrays have the worst MA plot? Which arrays have the best ones?*

**Solution 1**
*The worst MA plots are the ones of arrays 1, 7, 14 and 20. The best MA plots are the ones of arrays 3, 4, 9, 10, 12, 13, 16, 17, 18 and 19.*

**Exercise 2**
*Can you say from the second section if there are any arrays that are not homogeneous with the other ones? Is it consistent with your answers to the previous exercise?*

**Solution 2**
*The boxes of the array 1, 7, 14 and 20 are wider than the other ones, meaning that their variance is higher and the mean of the arrays 7 and 14 are larger than the overall means of the arrays. It is hard to see it but two arrays in the first group and two arrays in the second group on the density plots show a different distribution than the other ones. It is thus consistent with the MA plots.*

**Exercise 3**
*Is any of the clusters of the Section 3 of the report very different from the others? What can you say about the samples of this cluster?*

**Solution 3**
*There is one cluster that groups the arrays 1, 7, 14 and 20 which are the arrays with bad MA plots and heterogeneous with the other ones.*

**Exercise 4**
*What is your conclusion concerning the dependency between the variance and the mean shown in Section 4?*

**Solution 4**
*The red line is not horizontal but shows an up curve on the tail meaning that the higher the mean is, the higher the standard deviation is.*

**Exercise 5**
*Are there any array showing a spatial effect (Section 5)?*

**Solution 5**
*Fingerprint on the top of the array 7. Spatial distribution from bottom-right corner of the array 6. High density of high intensities showing a stain on the top of the array 13. High density of blue low intensities indicate a stain on the bottom of the array 20.*

**Exercise 6**
*What relevant information can you get from the Affymetrix specific plots of the Section 6?*

**Solution 6**
*The RNA degradation plot shows a normal profile and the experiments supposed to present quality problems do not show any peculiarities on the NUSE, RLE and qc plots. The affymetrix specific plots are performed after preprocessing so it seems that the preprocessing corrects the bias of the arrays 1, 7, 14 and 20.*

**Exercise 7**
*Could the problems of quality you have identified by reading this report be corrected and if so, how?*

**Solution 7**
*The preprocessing done for the Section 6 seems to be a good way to correct the problems. We can thus normalize the data and produce a report on the normalized data.*

# 3 Quality report for ExpressionSet objects

In the previous study, we produced a quality report on an *AffyBatch* object that contains raw data. In this section, we will produce a quality report on the same data after normalization with *rma* which produce an object of class *ExpressionSet*.

## 3.1 Data normalization

The *rma* function from the package affy computes the Robust Multichip Average. First, *rma* performs a background subtraction and a quantile normalization.

```
> rMLL.A = rma(MLL.A)

Background correcting
Normalizing
Calculating Expression

> class(rMLL.A)

[1] "ExpressionSet"
attr(,"package")
[1] "Biobase"

> show(rMLL.A)

ExpressionSet (storageMode: lockedEnvironment)
assayData: 22283 features, 20 samples
  element names: exprs
phenoData
  sampleNames: JD-ALD009-v5-U133A.CEL, JD-ALD051-v5-U133A.CEL, ..., JD-ALD520-v5
  -U133A.CEL  (20 total)
  varLabels and varMetadata description:
    sample: arbitrary numbering
featureData
  featureNames: 1007_s_at, 1053_at, ..., AFFX-r2-P1-cre-5_at  (22283 total)
  fvarLabels and fvarMetadata description: none
experimentData: use 'experimentData(object)'
Annotation: hgu133a
```

## 3.2 Report production

Now that the data are normalized, we can produce the report on the resulting *ExpressionSet*.

```
> arrayQualityMetrics(expressionset = rMLL.A,
+                     outdir = "rMLL",
+                     force = FALSE,
+                     do.logtransform = FALSE,
+                     split.plots = 10)
```

**Exercise 8**
*Open the report in a web browser and read it. In the first section of the report, which array(s) produced the worst MA plot? Is it the same observation than before normalization?*

**Solution 8**
*We still see that the arrays 1, 7, 14 and 20 have a worst MA plots but they are not as bad as before normalization.*

**Exercise 9**
*From the second section, is the homogeneity of the experiments better now?*

**Solution 9**
*The boxplots and density plots are homogeneous now.*

**Exercise 10**
*Do you see a cluster that is very distinct from the heatmap of Section 3?*

**Solution 10**
*There is no cluster with the "bad" arrays only anymore. The "bad" arrays are distributed among the other arrays.*

**Exercise 11**
*What is your conclusion concerning the dependency between the variance and the mean shown in Section 4?*

**Solution 11**
*The dependency between SD and mean has been corrected and is less important than before normalization.*

**Exercise 12**
*From your answers to the previous exercises, is there any improvment of the quality report as compared to before the normalization? If yes, which category of quality metrics are corrected by normalization?*

**Solution 12**
*The normalization improved the homogeneity between experiments, the between array comparison and the variance mean dependency. It also has slightly improved the individual array quality.*

# 4 Quality report for NChannelSet objects

In the cases of *ExpressionSet* and *NChannelSet*, some of the quality metrics provided by the package are performed using specific information about the features of the arrays. For an optimal use of the package, the data should be prepared accordingly to the following conventions.

## 4.1 Creating a NChannelSet

In this section, we use the *CCl4* example data set by Holger Laux, Timothy Wilkes, Amy Burrell and Carole Foy from LGC Ltd. in Teddington, UK. In the experiment, rat hepatocytes were treated either with carbon tetrachloride ($CCl_4$) or with DMSO. In the early $20^{th}$ century, $CCl_4$ was widely used as a dry cleaning solvent, as a refrigerant and in fire extinguishers, however, it was found to have multiple toxic and possible cancerogenous side-effects. The DMSO treatment served as negative control. Total RNA was hybridized to Agilent *Rat Whole Genome* microarrays. The arrays use a two-color labeling scheme (Cy3 and Cy5), and the experiment was done as a direct comparison with dye-swaps and 3 replicates each. The integrity of the RNA was quantified from the electrophoretic trace of the RNA samples by Agilent's RNA Integrity Number (RIN). The initial samples had a RIN of 9.7. To study the effect of RNA degradation, additional samples were generated by degrading the $CCl_4$ treated RNA sample with ribonuclease A, resulting in RINs of 5.0 and 2.5. The experimental design is described in more detail below. As an example, we will create a *NChannelSet* with the CCl4 data. We first have to load the needed libraries.

```
> library("Biobase")
> library("limma")
> library("CCl4")
> library("matchprobes")
```

### 4.1.1 Read the data and convert them into an RGList

The Genepix (`.gpr`) data files are in the `extdata` directory of the CCl4 package. If you have the package installed, we can locate them on your filesystem with the function *system.file*. If the files are somewhere else, please adapt the following assignment to `datapath`.

```
> datapath = system.file("extdata", package="CCl4")
> dir(datapath)
```

```
 [1] "013162_D_SequenceList_20060815.txt" "251316214319_auto_479-628.gpr"
 [3] "251316214320_auto_478-629.gpr"      "251316214321_auto_410-592.gpr"
 [5] "251316214329_auto_429-673.gpr"      "251316214330_auto_457-658.gpr"
 [7] "251316214331_auto_431-588.gpr"      "251316214332_auto_492-625.gpr"
 [9] "251316214333_auto_487-712.gpr"      "251316214379_auto_443-617.gpr"
[11] "251316214380_auto_493-682.gpr"      "251316214381_auto_497-602.gpr"
[13] "251316214382_auto_481-674.gpr"      "251316214384_auto_450-642.gpr"
[15] "251316214389_auto_456-694.gpr"      "251316214390_auto_456-718.gpr"
[17] "251316214391_auto_475-599.gpr"      "251316214393_auto_460-575.gpr"
[19] "251316214394_auto_463-521.gpr"      "samplesInfo.txt"
```

**Exercise 13**
*Use a text editor or a spreadsheet program to view these files. What does each of them contain?*

**Solution 13**
*There are 18 files with the extension* `.gpr`*. They contain the output of the image analysis, that is, the quantified red and green intensities for each feature on the arrays. The 18 files correspond to the 18 arrays. A description of what was hybridized to these arrays is in the file* `samplesInfo.txt`*.*

```
> p = read.AnnotatedDataFrame("samplesInfo.txt", path=datapath)
> p
> CCl4_RGList = read.maimages(files=sampleNames(p),
+    path = datapath,
+    source = "genepix",
+    columns = list(R = 'F635 Median', Rb = 'B635 Median',
+                   G = 'F532 Median', Gb = 'B532 Median'))
```

The function *read.maimages* from the limma package reads the `.gpr` files and builds an *RGList* object from it.

### 4.1.2 Build an NChannelSet from the RGList

Once the *RGList* object has been created, we can build an *NChannelSet*.
The *assayData* have 4 different slots corresponding to the red and green intensities and the red and green background intensities. In addition, the *phenoData* contain the information about the samples and the *featureData* include the features information. You can fill these slots with all the specificities you want to store in your *NChannelSet*. In the case of the use of *arrayQualityMetrics*, the optimal *NChannelSet* include specific *featureData* that are described in the following section.

**X and Y coordinates of the spots** To plot the images of the arrays, *arrayQualityMetrics* needs the coordinates of the spots on the chip. Two slots corresponding to the row and column numbers of the features are thus required in the *featureData*. These slots should be named "Row" and "Column". If the *NChannelSet* does not contain these slots, the images of the arrays will not be produced in the report.

```
>  CCl4_RGList$genes[95:105,]
```

```
    Block Row Column           ID            Name
95      1   1     95 A_44_P244495        AA819664
96      1   1     96 A_44_P138289    NM_001024787
97      1   1     97 A_44_P318805       XM_223584
98      1   1     98 A_44_P448307       XM_217432
99      1   1     99 A_44_P306486        AY387074
100     1   1    100 A_44_P559055        AA925039
101     1   1    101 A_44_P126261       XM_214443
102     1   1    102    (-)3xSLv1 NegativeControl
```

```
103     1   1    103 BrightCorner    BrightCorner
104     1   2      1 BrightCorner    BrightCorner
105     1   2      2    (-)3xSLv1 NegativeControl
```

By using the function *read.maimages*, the slot "genes" of the produced *RGList* automatically contains these coordinates if the source is "agilent", "genepix" or "imagene" or if the "annotation" argument is set.

**GC content of the reporters** If the GC content of the reporters is known, then it is possible to include it in the *featureData* of the *NChannelSet* under the column name "GC". Then a study of the GC content effect on intensities of the arrays can be performed. This information is not included in the `CCl4_RGList` data yet. If the GC content or the sequence of the reporters are available in the source data files, we can include it by using the argument "other.columns" of *read.maimages*. As it is not the case in this example, we have to proceed differently. The file with the sequences of the reporters is in the extdata directory of the package CCl4.

```
> seq = read.AnnotatedDataFrame("013162_D_SequenceList_20060815.txt",
+ path=datapath)
> if(any(duplicated(featureNames(seq))))
+   cat("IDs of the sequence file are not unique \n")
> bc = basecontent(seq$Sequence)
> GC = ((bc[,"C"]+bc[,"G"])/rowSums(bc))*100
> mt = match(featureNames(seq), CCl4_RGList$genes$ID)
> stopifnot(!any(is.na(mt)))
> fData = cbind(CCl4_RGList$genes,GC=rep(as.numeric("NA"),
+ nrow(CCl4_RGList$genes)))
> fData$GC[mt] = GC
```

**Mapping of the reporters** As a second part of the assessment of the platform quality, the report includes a study of the effect of the target mapping of the reporters. Thus a *featureData* slot named "HasTarget" should include logical "TRUE" if the reporter matches for a coding mRNA and "FALSE" if not. These information are not included in the `CCl4_RGList` data yet, but the slot "Name" of `CCl4_RGList$genes` give the RefSeq identifiers and we can use this to create the "HasTarget" slot.

```
> fData$hasTarget = (regexpr("^NM", CCl4_RGList$genes$Name) > 0)
```

**Building of the NChannelSet** Now that the *assayData* and *featureData* are ready, we can create the *NChannelSet*.

```
> featureData = new("AnnotatedDataFrame", data = fData)
> assayData = with(CCl4_RGList, assayDataNew(R=R, G=G, Rb=Rb, Gb=Gb))
```

```
> varMetadata(p)$channel=factor(c("G", "R", "G", "R"),
+                     levels=c(ls(assayData), "_ALL"))
> CCl4cfd <- new("NChannelSet",
+                assayData = assayData,
+                featureData = featureData,
+                phenoData = p)
```

**Normalization** We can normalize the data using the variance stabilization method available in the package vsn.

```
> library("vsn")
> nCCl4 = justvsn(CCl4cfd, subsample=2000)
> save(nCCl4, file = "nCCl4.RData")
```

## 4.2 Report production

First load the arrayQualityMetrics package.

```
> library("arrayQualityMetrics")
```

Then, you can execute the *arrayQualityMetrics* using the same arguments as seen in the Section 2.

```
> arrayQualityMetrics(expressionset = nCCl4,
+                     outdir = "CCl4")
```

**Exercise 14**
*Open the report in a web browser and read it. In the first section of the report, which arrays have the best MA plot? Which arrays have the worst? Is it consistent with what we know of the quality of the RNA hybridised to the arrays?*

**Solution 14**
*The worst MA plots are the ones of arrays 7, 11, 15 and 18. The best MA plots are the ones of arrays 1, 2, 5, 8, 9, 12 and 13. The worst MA plots are from bad RNA quality samples and the best MA plots are from medium or good RNA quality samples. It seems that the MA plots are consistent with the RNA quality.*

**Exercise 15**
*Are there any array showing a spatial effect (Section 2)?*

**Solution 15**
*The red channels are all good. Concerning the green channels there is an effect on spatial distribution of the intensities from the bottom of all the arrays and there are stains on the bottom of the arrays 5, 6, 7, 8, 14.*

**Exercise 16**
*Can you say from the third Section if there are any arrays that are not similar to the other ones?*

**Solution 16**
*The log(ratio) boxplots show wider boxes of the arrays 7, 11 and 18, meaning that their variance is higher but all the means of the arrays are similar.*

**Exercise 17**
*In Section 4, the platform quality is assessed. Does the GC content affect the intensities and log(ratio) (Figure 4)? What would you expect from Figure 5 and what can you conclude?*

**Solution 17**
*The higher the GC content is, the higher the variances and means intensities of the features are. But when the log(ratio) are built it corrects the effect and the GC content does not affect the variance and mean intensities of the features anymore. The distribution of the features which map for a coding mRNA should be lower on the y axis and shifted forward on the x axis as compared to the distribution of the unmapped features. Here it is not the case at all.*

**Exercise 18**
*Are the samples clustered in Section 5 according to the information you have about the RNA quality?*

**Solution 18**
*There is one cluster that groups the arrays 7, 11 and 18 and another cluster with the arrays 3, 4 and 15 which are the arrays of the bad RNA quality samples. The good and medium RNA quality samples are clustered in two subgroup without distinction between the quality of RNA.*

**Exercise 19**
*Does the variance depend on the mean (Section 6)?*

**Solution 19**
*The red line being almost horizontal, the variance does not depend on the mean.*

**Exercise 20**
*What can you conclude about the RNA quality effects?*

**Solution 20**
*The RNA quality of the samples seems to have an effect on the individual array quality and the comparison between arrays. However, it does not seem to have an effect on the spatial plots and the homogeneity between experiments. The effects of GC content and mapping of the features seem also to be independent of the RNA quality of the sample.*

```
> toLatex(sessionInfo())
```

- R version 2.6.0 Under development (unstable) (2007-08-14 r42499), `i686-pc-linux-gnu`
- Locale: `C`
- Base packages: base, datasets, grDevices, graphics, methods, splines, stats, tools, utils
- Other packages: ALLMLL 1.2.2, AnnotationDbi 0.0.88, Biobase 1.15.23, CCl4 1.0.4, DBI 0.2-3, EBImage 2.1.15, RColorBrewer 0.2-3, RSQLite 0.5-4, affy 1.15.7, affyPLM 1.13.6, affydata 1.11.2, affyio 1.5.1, annotate 1.15.2, arrayQualityMetrics 1.0.13, gcrma 2.9.1, genefilter 1.15.9, geneplotter 1.15.3, hgu133acdf 1.17.0, lattice 0.16-3, limma 2.11.9, matchprobes 1.9.10, preprocessCore 0.99.12, simpleaffy 2.11.2, survival 2.32, vsn 3.0.10
- Loaded via a namespace (and not attached): KernSmooth 2.22-21, grid 2.6.0