

MBI Lab2: Introduction to the Bioconductor R Package *affy*

Sandrine Dudoit, Ben Bolstad, Robert Gentleman, and Rafael Irizarry

September 18, 2004

Short course	Statistical Methods and Software for the Analysis of Microarray Experiments
Location	Mathematical Biosciences Institute Ohio State University, Columbus, OH
Instructors	Sandrine Dudoit, Division of Biostatistics, UC Berkeley Nicholas P. Jewell, Division of Biostatistics, UC Berkeley
Date	September 20–24, 2004
Website	www.stat.berkeley.edu/~sandrine/Docs/Talks/MBI04/mbi.html

1 Getting started

In this lab, we demonstrate the main functions in the *affy* package for pre-processing Affymetrix oligonucleotide chip data. For a more detailed introduction, consult the package vignettes which can be listed and accessed in PDF by the command `openVignette("affy")`. The `vExplorer` function, from the *tkWidgets* package, may be used to step through the code chunks interactively. A number of sample datasets are available in this and other related packages such as *affydata*; to list these, type `data(package="affy")`. To load the necessary packages,

```
> library(affy)
> library(affydata)
> library(hexbin)
```

One of the main functions for reading in Affymetrix data is `ReadAffy`. It reads in data from CEL files and creates objects of class *AffyBatch*. Using `ReadAffy(widget=TRUE)` provides widgets for interactive data input. However, in this lab we will work mainly with the *Dilution* dataset, which is included in the *affydata* package. For a description of *Dilution*, type `? Dilution`. To load the dataset,

```
> data(Dilution)
```

2 *affy* classes and methods

In order to deal with the complexity of microarray data, the *affy* and other Bioconductor packages have adopted the *class/method object-oriented programming* (OOP) paradigm proposed in J. M. Chambers (1998). *Programming with Data*. One of the main classes in *affy* is the *AffyBatch* class. For details on this class, please consult the help file, `class?AffyBatch`; methods for manipulating instances of this class are also described in the help file. Other classes include *ProbeSet*, for PM and MM intensities of individual probe-sets in one or more chips. The object *Dilution* is an instance of the class *AffyBatch*. Try the following commands to obtain information on this object.

```
> class(Dilution)
```

```
[1] "AffyBatch"
```

```
> slotNames(Dilution)
```

```
[1] "cdfName"      "nrow"         "ncol"         "exprs"        "se.exprs"
[6] "phenoData"    "description"  "annotation"   "notes"
```

```
> Dilution
```

AffyBatch object
size of arrays=640x640 features (12805 kb)
cdf=HG_U95Av2 (12625 affyids)
number of samples=4
number of genes=12625
annotation=hgu95av2

> *cdfName(Dilution)*

[1] "HG_U95Av2"

> *annotation(Dilution)*

[1] "hgu95av2"

> *description(Dilution)*

Experimenter name: Gene Logic
Laboratory: Gene Logic
Contact information: 708 Quince Orchard Road
Gaithersburg, MD 20878
Telephone: 1.301.987.1700
Toll Free: 1.800.GENELOGIC (US and Canada)
Facsimile: 1.301.987.1701

Title: Small part of dilution study
URL: <http://qolotus02.genelogic.com/datasets.nsf/>

A 68 word abstract is available. Use 'abstract' method.

> *notes(Dilution)*

[1] ""

> *nrow(Dilution)*

[1] 640

> *ncol(Dilution)*

[1] 640

For a description of the target samples hybridized to the chips,

> *phenoData(Dilution)*

phenoData object with 3 variables and 4 cases

varLabels

liver: amount of liver RNA hybridized to array in micrograms

sn19: amount of central nervous system RNA hybridized to array in micrograms

scanner: ID number of scanner used

```
> pData(Dilution)
```

```
      liver sn19 scanner
20A    20    0        1
20B    20    0        2
10A    10    0        1
10B    10    0        2
```

The `exprs` slot contains a matrix with columns corresponding to chips and rows to individual probes on the chip. To obtain the matrix of intensities for all four chips,

```
> e <- exprs(Dilution)
```

```
> nrow(Dilution) * ncol(Dilution)
```

```
[1] 409600
```

```
> dim(e)
```

```
[1] 409600      4
```

Probe-level PM and MM intensities can be accessed using the `pm` and `mm` methods.

```
> PM <- pm(Dilution)
```

```
> dim(PM)
```

```
[1] 201800      4
```

```
> PM[1:5, ]
```

```
      20A  20B  10A  10B
[1,] 468.8 282.3 433.0 198.0
[2,] 430.0 265.0 308.5 192.8
[3,] 182.3 115.0 138.0  86.3
[4,] 930.0 588.0 752.8 392.5
[5,] 171.0 128.0 152.3  97.8
```

To get the probe-set names (Affy IDs),

```
> gnames <- geneNames(Dilution)
```

```
> length(gnames)
```

```
[1] 12625

> gnames[1:5]

[1] "1000_at" "1001_at" "1002_f_at" "1003_s_at" "1004_at"

> nrow(e)/length(gnames)

[1] 32.44356
```

As with other microarray objects in Bioconductor packages, subsetting methods are provided for *AffyBatch* objects.

```
> dil1 <- Dilution[, 1]
> class(dil1)

[1] "AffyBatch"

> dil1

AffyBatch object
size of arrays=640x640 features (3205 kb)
cdf=HG_U95Av2 (12625 affyids)
number of samples=1
number of genes=12625
annotation=hgu95av2
```

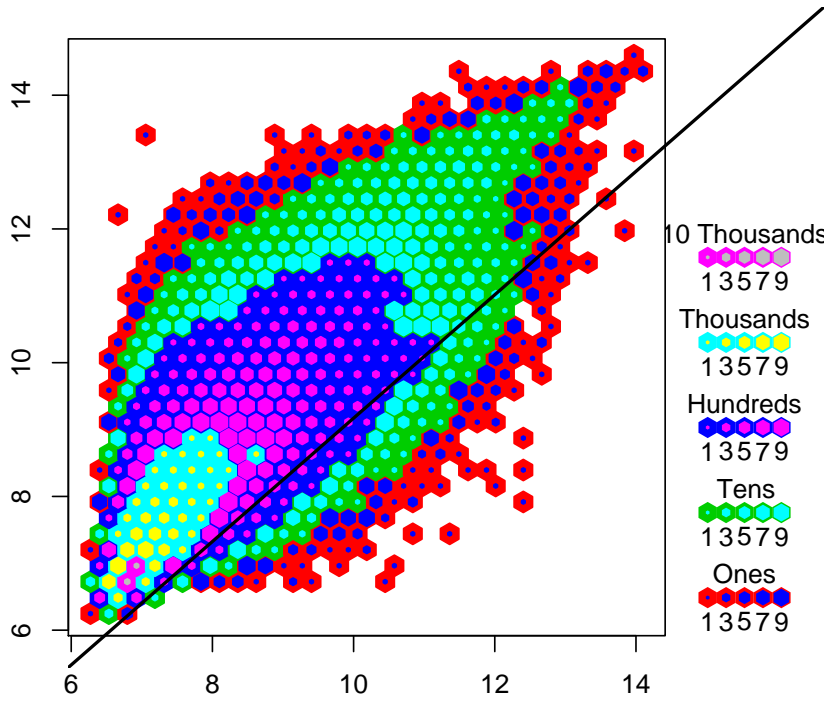
3 Diagnostic plots

To produce a *spatial image* of log base 2 probe intensities,

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

Hexagonal binning functions from the package *hexbin* may be used to explore the joint distribution of PM and MM intensities.

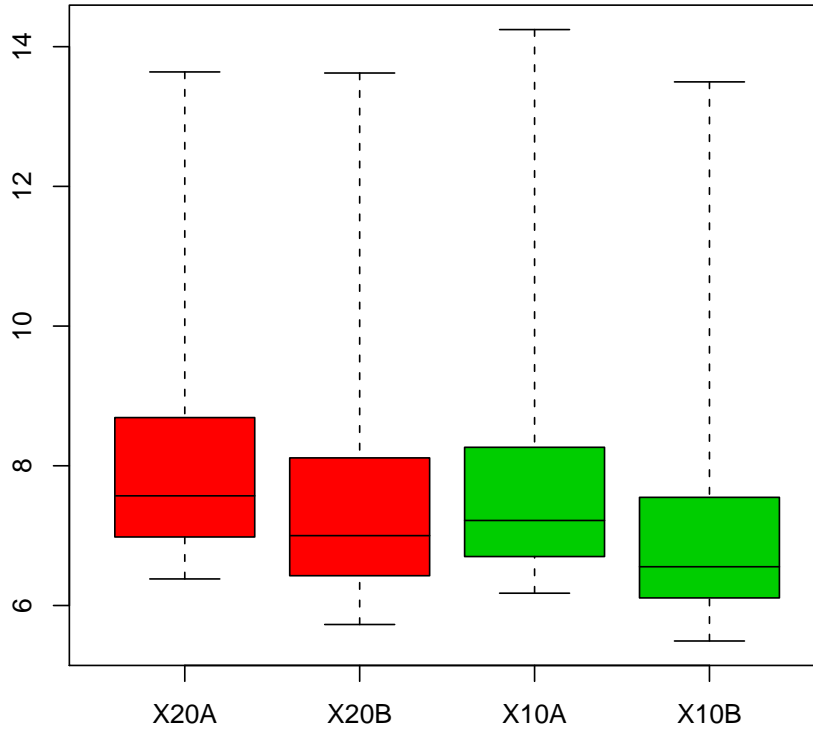
```
> hbin <- hexbin(log(mm(Dilution[, 1])), 2), log(pm(Dilution[, 1])),
+ 2))
> plot(hbin, style = "nested.lattice")
> abline(0, 1, lwd = 2)
```



To produce *boxplots* of log base 2 probe intensities,

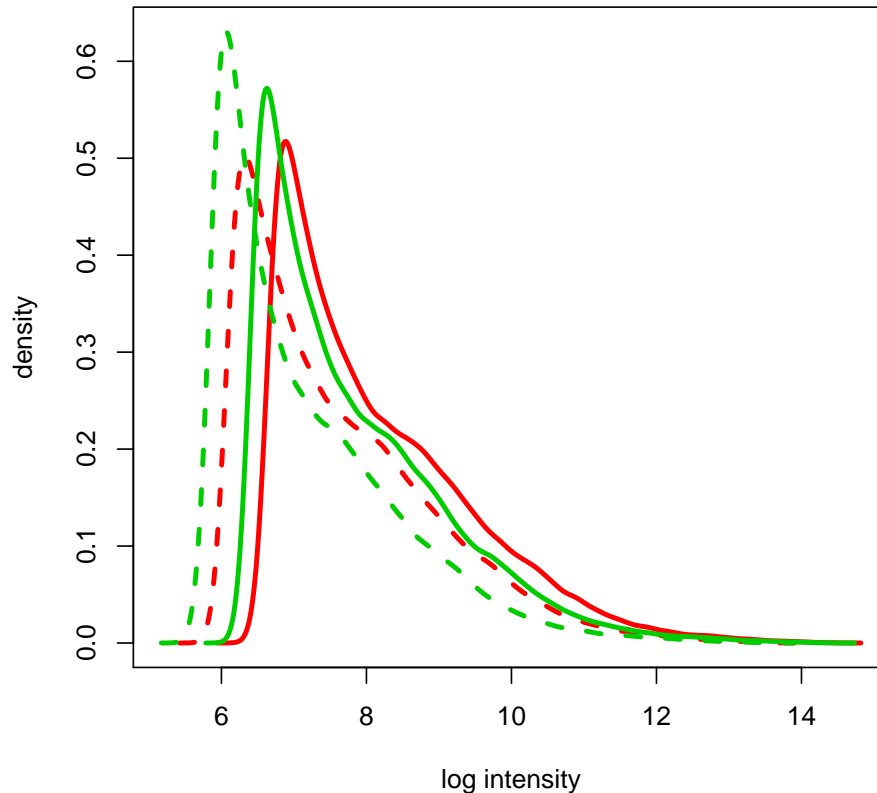
```
> boxplot(Dilution, col = c(2, 2, 3, 3))
```

Small part of dilution study



To produce *density plots* of log base 2 probe intensities,

```
> hist(Dilution, type = "l", col = c(2, 2, 3, 3), lty = rep(1:2,  
+ 2), lwd = 3)
```



The boxplots and density plots show that the `Dilution` data needs normalization. As described in the dataset help file and in the `pData` slot (`pData(Dilution)`), two concentrations of mRNA were used and, for each concentration, two scanners were used. From the plots, we note that scanner effects (different line types) seem stronger than concentration effects (different colors). In other words, chips that should be the same are different; chips that should be different are similar.

4 From probe-level intensities to expression measures

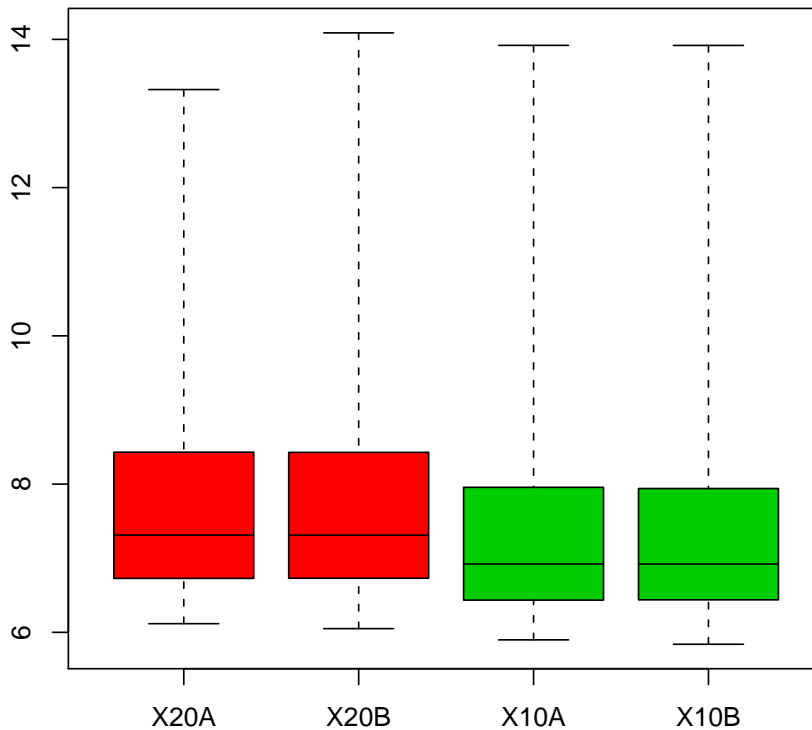
Because different mRNA concentrations were used, we perform normalization within concentration groups. The default procedure implemented in the `normalize` method is *probe-level quantile normalization*.

```
> Dil20 <- normalize(Dilution[, 1:2])
> Dil10 <- normalize(Dilution[, 3:4])
> normDil <- merge(Dil20, Dil10)
```

Notice how the boxplots now look better.

```
> boxplot(normDil, col = c(2, 2, 3, 3))
```

Batches. No description was supplied. The description of the two



We view the process of going from probe-level intensities to gene-level expression measures as a three-step procedure consisting of: (i) *background adjustment*; (ii) *normalization*; (iii) *summarization*. The *affy* package provides implementations for a number of methods for each of these steps: (i) background correction: e.g., none, MAS 5.0, convolution; (ii) normalization: e.g., probe-level quantile, cyclic loess, contrast loess; (iii) summarization: e.g., MAS 4.0, MAS 5.0, MBEI (Li & Wong, 2001), median polish for additive linear model (Irizarry et al., 2003). The *Robust Multichip Average* (RMA) method refers to the sequence: convolution background adjustment, probe-level quantile normalization, and median polish summarization for gene-specific additive models with probe and chip effects. These methods are discussed in detail in the vignette `builtin-Methods`, which can be viewed using `openVignette("affy")`. To list available methods for *AffyBatch* objects,

```
> bgcorrect.methods
```

```

[1] "mas" "none" "rma" "rma2"

> normalize.AffyBatch.methods

[1] "constant"          "contrasts"          "invariantset"      "loess"
[5] "qspline"           "quantiles"          "quantiles.robust"

> normalize.methods(Dilution)

[1] "constant"          "contrasts"          "invariantset"      "loess"
[5] "qspline"           "quantiles"          "quantiles.robust"

> pmcorrect.methods

[1] "mas"          "pmonly"        "subtractmm"

> express.summary.stat.methods

[1] "avgdiff"          "liwong"          "mas"          "medianpolish" "playerout"

```

The main user-level pre-processing function is `expresso`: it starts from raw probe-level intensities to produce gene-level expression measures. Specifically, the function operates on objects of class *AffyBatch* and returns objects of class *exprSet*. Pre-processing methods can be selected interactively using widgets by typing `expresso(Dilution, widget=TRUE)`. The function `rma` provides a more efficient implementation of the RMA procedure.

```
> rmaDil <- rma(Dilution)
```

```

Background correcting
Normalizing
Calculating Expression

```

```
> class(rmaDil)
```

```

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

```

5 CDF data packages

Data packages providing CDF information can be download from www.bioconductor.org. These packages contain *environment* objects which provide mappings between AffyIDs and matrices of probe locations, with rows corresponding to probe-pairs and columns to PM and MM cells. The CDF environment for the HGU95Av2 chip is already in the package. For information on the environment object ? `hgu95av2cdf`,

```
> annotation(Dilution)

[1] "hgu95av2"

> data(hgu95av2cdf)
> pnames <- ls(env = hgu95av2cdf)
> length(gnames)

[1] 12625

> pnames[1:5]

[1] "1000_at"  "1001_at"  "1002_f_at" "1003_s_at" "1004_at"

> get(pnames[1], env = hgu95av2cdf)

      pm      mm
[1,] 358160 358800
[2,] 118945 119585
[3,] 323731 324371
[4,] 223978 224618
[5,] 313420 314060
[6,] 349209 349849
[7,] 199525 200165
[8,] 213669 214309
[9,] 236739 237379
[10,] 298099 298739
[11,] 282744 283384
[12,] 281443 282083
[13,] 349198 349838
[14,] 297953 298593
[15,] 317054 317694
[16,] 404069 404709
```

The methods `indexProbes`, `pmindex`, and `mmindex` provide information on probe location.

```

> plocs <- indexProbes(Dilution, which = "both")
> plocs[[1]]

[1] 358160 118945 323731 223978 313420 349209 199525 213669 236739 298099
[11] 282744 281443 349198 297953 317054 404069 358800 119585 324371 224618
[21] 314060 349849 200165 214309 237379 298739 283384 282083 349838 298593
[31] 317694 404709

> pmindex(Dilution, genenames = gnames[1])

$"1000_at"
[1] 358160 118945 323731 223978 313420 349209 199525 213669 236739 298099
[11] 282744 281443 349198 297953 317054 404069

```

6 Other tools

In addition to CDF and probe metadata packages for specific Affymetrix chip series, Bioconductor provides a number of other packages for Affymetrix data analysis: *affyPLM*, probe-level models; *affycomp*, graphics toolbox for assessment of Affymetrix expression measures; *affydata*, sample Affymetrix datasets; *affylmGUI*, GUI for Affymetrix data analysis using the *limma* package; *affypdnn*, probe-dependent nearest-neighbor (PDNN) analyses; *altcdfenvs*, handling CDF environments; *annaaffy*, annotation tools for Affymetrix biological metadata; *germa*, background adjustment using sequence information (e.g., CG content); *makecdfenv*, CDF environment maker; *simpleaffy*, high-level functions, based on *affy* package; *vsu*, variance stabilization and calibration.