EMBO03 Lab4: Introduction to Bioconductor `affy` Package

Sandrine Dudoit, Robert Gentleman, and Rafael Irizarry

March 14, 2003

In this lab, we demonstrate the main functions in the `affy` package for pre-processing Affymetrix microarray data. For a more detailed introduction, consult the package vignettes which can be listed by the command `openVignette("affy")`. A demo can also be accessed by `demo(affy)`. A number of sample datasets are available in the package; to list these, type `data(package="affy")`. To load the package

```
> library(affy)
```

The function `ReadAffy` is available for reading CEL files. However, in this lab we will work mainly with the `Dilution` dataset, which is included in the package. For a description of `Dilution`, type `? Dilution`. To load this dataset

```
> data(Dilution)
```

One of the main classes in `affy` is the `AffyBatch` class. For details on this class consult the help file, `? AffyBatch`; methods for manipulating instances of this class are also described in the help file. Other classes include `ProbeSet` (PM and MM intensities for individual probe sets), `Cdf` (information contained in a CDF file), and `Cel` (single array cel intensity data). 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
notes=
```

```
> annotation(Dilution)
```

```
[1] "hgu95av2"
```

For a description of the target samples hybridized to the arrays

```
> 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 micro
               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 arrays and rows to individual probes on the array. To obtain the matrix of intensities for all four arrays

```
> e <- exprs(Dilution)
> nrow(Dilution) * ncol(Dilution)
```

```
[1] 409600
```

```
> dim(e)
```

```
[1] 409600      4
```

You can access probe-level PM and MM intensities using

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

```
[1] 201800      4
```

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

```
           20A   20B   10A   10B
1000_at1 468.8 282.3 433.0 198.0
1000_at2 430.0 265.0 308.5 192.8
1000_at3 182.3 115.0 138.0  86.3
1000_at4 930.0 588.0 752.8 392.5
1000_at5 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, you can use subsetting commands for `AffyBatch` objects

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

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

```
> dil1
```

```
AffyBatch object
size of arrays=640x640 features (3204 kb)
cdf=HG_U95Av2 (12625 affyids)
number of samples=1
number of genes=12625
annotation=hgu95av2
notes=
```

```
> cel1 <- Dilution[[1]]
> class(cel1)
```

```
[1] "Cel"
```

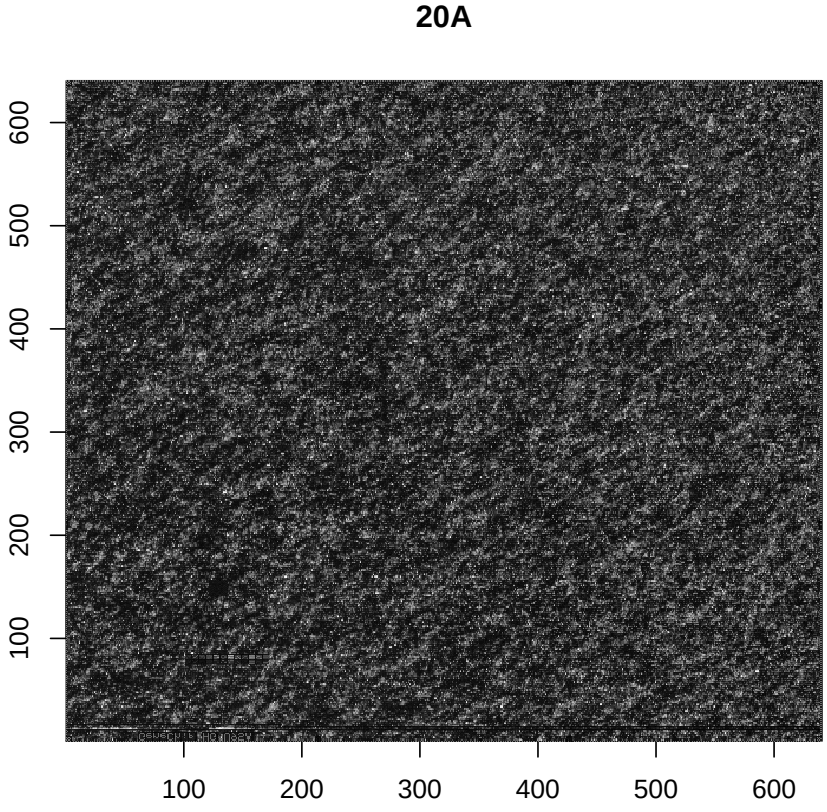
```
> cel1
```

```
Cel object
name=20A
cdfName=HG_U95Av2
intensity=640 x 640 (3200 kb)
masked= 0 %
outliers= 0 %
history=
```

One of the main functions for reading in Affymetrix data is `ReadAffy`. It reads in data from `CEL` and `CDF` files and creates objects of class `AffyBatch`. Using `ReadAffy(widget=TRUE)` provides widgets for interactive data input.

To produce a spatial image of probe log intensities and probe raw intensities

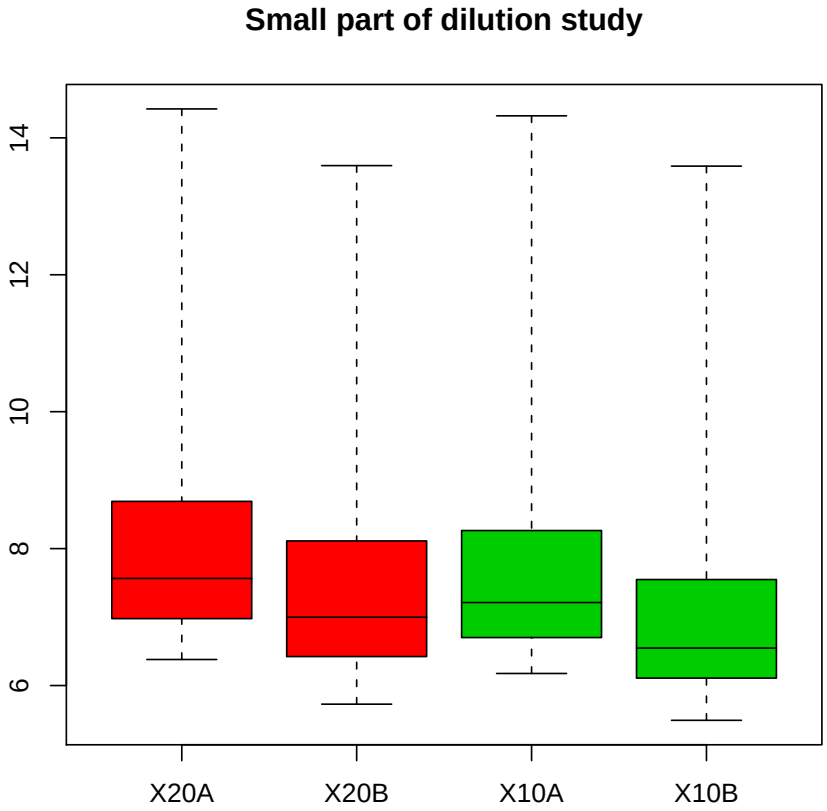
```
> image(Dilution[1])
```



```
> image(cel1)
```

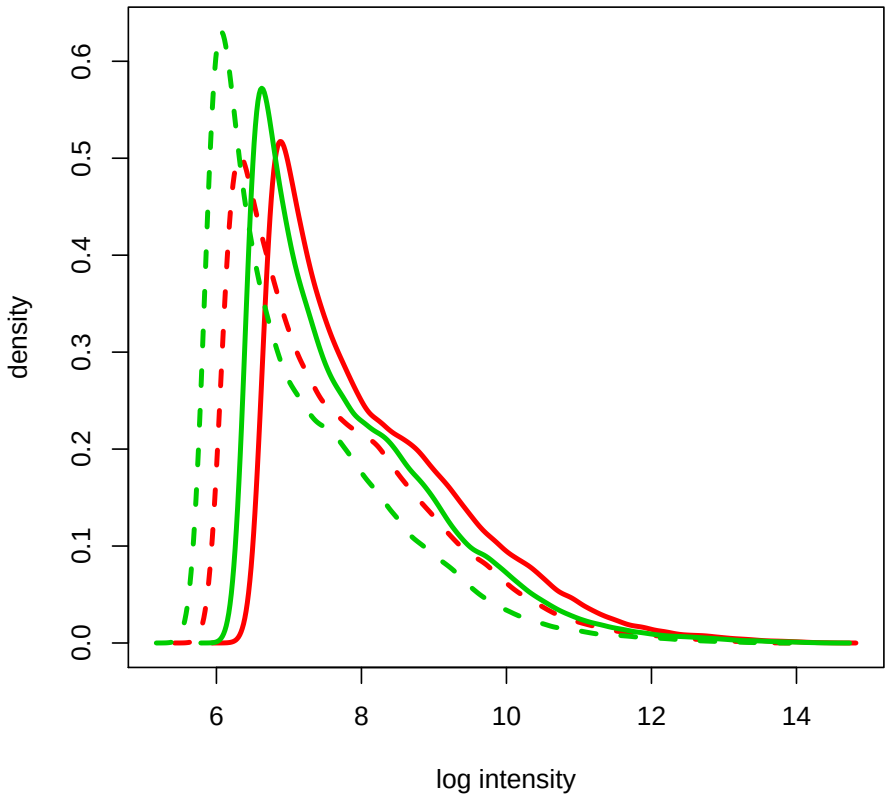
To produce boxplots of probe log intensities

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



To produce density plots of probe log intensities

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

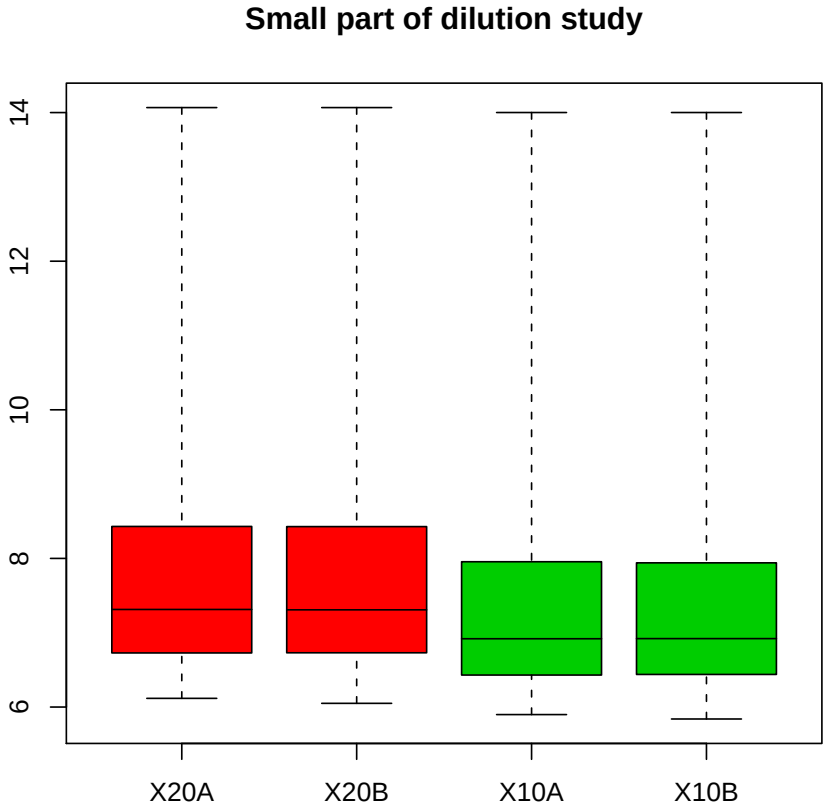


The boxplots and density plots show that the Dilution data needs normalization. As described in the dataset help file and in the `phenoData` slot (`pData(Dilution)`), two concentrations of mRNA were used and, for each concentration, two scanners were used. From the plots, we note that scanner effects seem stronger than concentration effects (different colors). Arrays that should be the same are different; arrays that should be different are similar. Because different mRNA concentrations were used, we perform probe-level normalization within concentration groups.

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

Notice how the boxplot now looks better.

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



The `affy` package provides implementations for a number of methods for background correction, probe-level normalization (e.g., quantile, curve-fitting (Bolstad et al., 2002)), and computation of expression measures (e.g., MAS 4.0, MAS 5.0, MBEI (Li & Wong, 2001), RMA (Irizarry et al., 2003)). To list available methods for `AffyBatch` objects

```
> bgcorrect.methods

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

> normalize.AffyBatch.methods

[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 normalization function is `expresso`. You can select pre-processing methods interactively using widgets by typing `expresso(Dilution, widget=TRUE)`. The function operates on objects of class `AffyBatch` and returns objects of class `exprSet`. `rma` provides a more efficient implementation of Robust Multi-array Average (RMA). We don't normalize because we already did above.

```
> rmaDil <- rma(normDil, normalize = FALSE)
> class(rmaDil)
```

Data packages for 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 cels. CDF environments for HGU95Av2 and HGU133A chips are 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

> gnames[1:5]

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

> get(gnames[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
```

You can also use the `indexProbes`, `pmindex`, and `mmindex` functions to get 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], xy = TRUE)

$"1000_at"
        x   y
 [1,] 400 560
 [2,] 545 186
 [3,] 531 506
 [4,] 618 350
 [5,] 460 490
 [6,] 409 546
 [7,] 485 312
 [8,] 549 334
 [9,] 579 370
[10,] 499 466
[11,] 504 442
[12,] 483 440
[13,] 398 546
[14,] 353 466
[15,] 254 496
[16,] 229 632

> 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
```

Having access to PM and MM data can be useful. Let's look at a plot of PM vs. MM

```
> plot(mm(Dilution[1]), pm(Dilution[1]), pch = ".", log = "xy")
> abline(0, 1, col = "red")
```