

Analysis of Genomic and Proteomic Data

Affymetrix[©] Technology and Preprocessing Methods

Practicals

Benjamin Haibe-Kains

February 17, 2005

1 R and Bioconductor

You must have installed R (available from <http://www.r-project.org>) and the Bioconductor packages (available from <http://www.bioconductor.org>). For the further exercises, you have to install the *affy* and the *limma* libraries. If you have problem about basic R functions (matrix manipulations, function writing, etc), you can consult the R manuals available from <http://cran.r-project.org/manuals.html>.

Launch R and load the *affy* library (this command will load automatically the dependences).

```
> library(affy)
Loading required package: Biobase
Loading required package: tools
Welcome to Bioconductor
  Vignettes contain introductory material. To view,
  simply type: openVignette()
  For details on reading vignettes, see
  the openVignette help page.
Loading required package: reposTools
```

Don't worry about the error messages.

1.1 Help

Here are 3 ways to access help under R :

- you can search a specific keyword with the function *help.search(pattern)*.

```
> help.search("rma")
```

- you can access to the help page of a specific function by the function `?<function>`.

```
> ?just.rma
```

- you can access to help (in pdf format) for all Bioconductor loaded libraries by typing

```
> openVignette()
Please select (by number) a vignette
1: affy primer
2: affy: Built-in Processing Methods
3: affy: Custom Processing Methods (HowTo)
4: affy: Automatic downloading of cdfenvs (HowTo):
5: affy: Import Methods (HowTo)
6: HOWTO access a repository
7: HOWTO set up a simple R repository
8: Biobase Primer
9: Howto Bioconductor
10: HowTo HowTo
11: esApply Introduction
Selection:
```

2 Read the CEL Files

You can download the CEL and the demographics files from http://www.ulb.ac.be/di/map/bhaibeka/bioinfo_courses/cel_demo.zip. It is a ZIP archive. Decompress it and put all the CEL files in the `path_to_cel_files` directory and the demographics file in the `path_to_demographics` directory.

```
> fn <- dir(path="path_to_cel_files", pattern="*.CEL", full.names=TRUE)
> phenod <- read.csv("path_to_demographics/demo.csv")
> phenod <- new("phenoData", pData=phenod, varLabels=list("samplename",
  "celname", "class"))
> abatch <- read.affybatch(filename=fn, phenoData=phenod, verbose=TRUE)
```

You can print information about your `AffyBatch` object.

```
> print(abatch)
AffyBatch object
size of arrays=712x712 features (19809 kb)
cdf=HG-U133B (22645 affyids)
number of samples=5
number of genes=22645
annotation=hgu133b
> print(sampleNames(abatch))
> print(geneNames(abatch))
```

You can see that your CEL files coming from hgu133b chips which contain $712 * 712$ probes. These probes will be summarized in 22645 probe sets.

2.1 Probe Intensities

The `affyBatch` class is just an extension of the `ExprSet` class (see the presentation¹ for details). So, you can access to the matrix containing all the intensities at the probe level with the function `intensity(object)`. The matrix contains one probe per row and one chip per column. You can access at the PM or the MM intensities for one or several probe sets with the function `probeset(object, genenames)`.

```
> my.probesets <- c("234372_at", "234954_at")
> r <- probeset(abatch, genenames=my.probesets)
> r[[1]]@id
>r[[1]]@pm
>r[[1]]@mm #similar for r[[2]]
```

2.2 Phenotype Information

As the `ExprSet` class, the `AffyBatch` class has a `phenoData` slot containing the information of the microarray experiments. You can access and modify this information with the function `pData(object)`. This information is a matrix.

```
> pData(abatch)
  samplename celname class
1         1_B 1_B.CEL     1
2         2_B 2_B.CEL     0
3         3_B 3_B.CEL     0
4         4_B 4_B.CEL     1
5         5_B 5_B.CEL     0
```

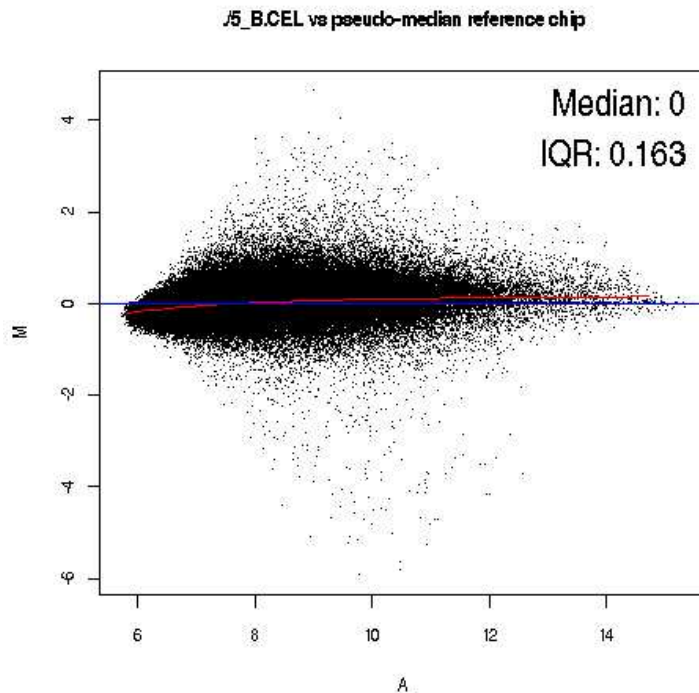
3 MA Plots of the Raw Data

You can display an MA plots of the raw data (data at the probe level without any correction) with the function `MAplot(object)`.

```
> MAplot(abatch)}
```

It will take some time !

¹The presentation is available from http://www.ulb.ac.be/di/map/bhaibeka/bioinfo_courses/affy_tech_hkb.pdf.



4 RMA procedure

You can perform the RMA procedure with :

- only background correction and gene expression summarization (log2)

```
> eset.bkg <- rma(abatch, background=TRUE, normalize=FALSE, verbose=TRUE)
Background correcting
Calculating Expression
```

- background correction, normalization and gene expression summarization (log2)

```
> eset.bkg.norm <- rma(abatch, background=TRUE, normalize=TRUE, verbose=TRUE)
Background correcting
Normalization
Calculating Expression
```

Remark Because of the high memory consumption of an AffyBatch object, you can use the function *just.rma()*.

- only background correction and gene expression summarization (log2)

```
> eset.bkg <- just.rma(filename=fn, background=TRUE, normalize=FALSE, verbose=TRUE)
```

- background correction, normalization and gene expression summarization (log2)

```
> eset.bkg.norm <- just.rma(filename=fn, background=TRUE, normalize=TRUE,
  verbose=TRUE)
```

4.1 Gene Expression

As previously mentioned for the `AffyBatch` class, you can access to the gene expressions with the function `exprs(object)`. The matrix contains the observed expression levels. The columns represent patients or cases and rows represent genes.

```
> exprs(abatch)[1:3,1:3] #display only the gene expressions of the first
  three probe sets for the first three patients
> my.probesets <- c("234372_at", "234954_at")
> gapd <- exprs(abatch)[my.probesets,]
> apply(gapd, 1, mean)
> apply(gapd, 1, var)
```

5 MA plots of the Expression Set

You can display the MA plots of the normalized data (gene expression level) :

- only background correction and gene expression summarization (log2)

```
> library(limma)
> par(mfrow=c(1,2), pty="s")
> plotMA(eset.bkg)
```

- background correction, normalization and gene expression summarization (log2)

```
> plotMA(eset.bkg.norm)
```

If you want to generate a pdf of your figure, it is easy by calling the function `pdf(file)`.

```
> pdf("maplots.pdf")
> par(mfrow=c(1,2), pty="s")
> plotMA(eset.bkg)
> plotMA(eset.bkg.norm)
> dev.off()
```

Good Work !