Working with Affymetrix data: estrogen, a 2x2 factorial design example

June 2004

Robert Gentleman, Wolfgang Huber

1.) Preliminaries. To go through this exercise, you need to have installed R, the packages `Biobase`, `affy`, `hgu95av2.db`, `hgu95av2cdf` and `vsn`.

```
library("affy")
library("estrogen")
library("vsn")
library("genefilter")
```

2.) Load the data.

a. Find the directory where the example cel files are. The directory path should end in `.../R/library/estrogen/extdata`.

```
system.file("extdata", package="estrogen")
## [1] "/tmp/Rtmp17wilI/Rinst34363429a53146/estrogen/extdata"
datadir <- function(x)
    file.path(system.file("extdata", package="estrogen"), x)
```

The function `system.file` here is used to find the subdirectory `extdata` of the estrogen package on your computer's harddisk. To use your own data, set `datadir` to the appropriate path instead.

b. The file `estrogen.txt` contains information on the samples that were hybridized onto the arrays. Look at it in a text editor. Again, to use your own data, you need to preprare a similar file with the appropriate information on your arrays and samples. To load it into a `phenoData` object

```
pd <- read.AnnotatedDataFrame(datadir("estrogen.txt"),
                              header = TRUE, sep = "", row.names = 1)
pData(pd)
##               estrogen time.h
## low10-1.cel    absent     10
## low10-2.cel    absent     10
## high10-1.cel  present     10
## high10-2.cel  present     10
## low48-1.cel    absent     48
## low48-2.cel    absent     48
## high48-1.cel  present     48
## high48-2.cel  present     48
```

`phenoData` objects are where the Bioconductor stores information about samples, for example, treatment conditions in a cell line experiment or clinical or histopathological characteristics of tissue biopsies. The `header` option lets the `read.phenoData` function know that the first line in the file contains column headings, and the `row.names` option indicates that the first column of the file contains the row names.

c. Load the data from the CEL files as well as the phenotypic data into an `AffyBatch` object.

```
a <- ReadAffy(filenames = datadir(rownames(pData(pd))),
              phenoData = pd,
              verbose = TRUE)
```

```
a
## Warning:  replacing previous import 'AnnotationDbi::tail' by 'utils::tail'
when loading 'hgu95av2cdf'
## Warning:  replacing previous import 'AnnotationDbi::head' by 'utils::head'
when loading 'hgu95av2cdf'
```

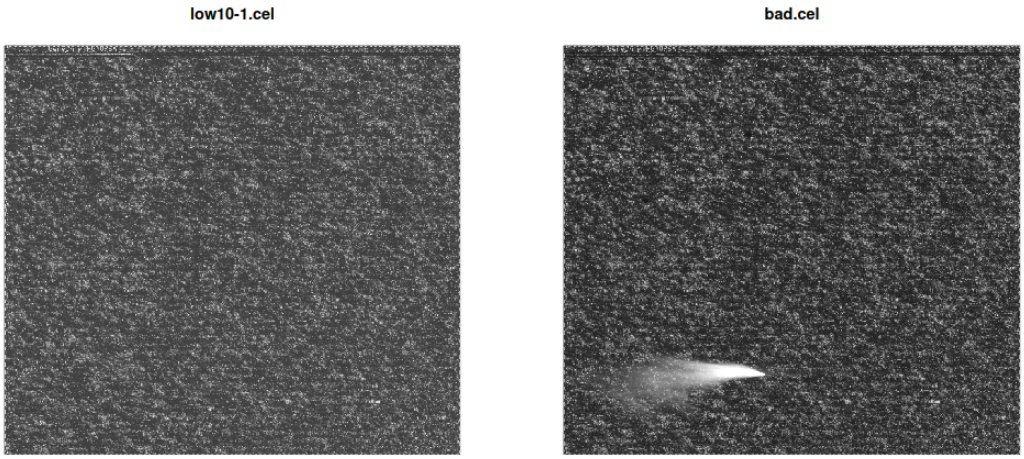


Figure 1: see Exercise 4.

```
##
## AffyBatch object
## size of arrays=640x640 features (22 kb)
## cdf=HG_U95Av2 (12625 affyids)
## number of samples=8
## number of genes=12625
## annotation=hgu95av2
## notes=
```

3.) Normalization. Now we can use the function `vsnrma` to normalize the data and calculate expression values.

```
x <- vsnrma(a)
```

```
x
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 12625 features, 8 samples
##   element names: exprs
## protocolData
##   sampleNames: low10-1.cel low10-2.cel ... high48-2.cel (8 total)
##   varLabels: ScanDate
##   varMetadata: labelDescription
## phenoData
##   sampleNames: low10-1.cel low10-2.cel ... high48-2.cel (8 total)
##   varLabels: estrogen time.h
##   varMetadata: labelDescription
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: hgu95av2
```

4.) Looking at the CEL file images. The `image` function allows us to look at the spatial distribution of the intensities on a chip. This can be useful for quality control. Fortunately, all of the 8 celfiles that we have just loaded do not show any remarkable spatial artifacts (see Fig. 1).

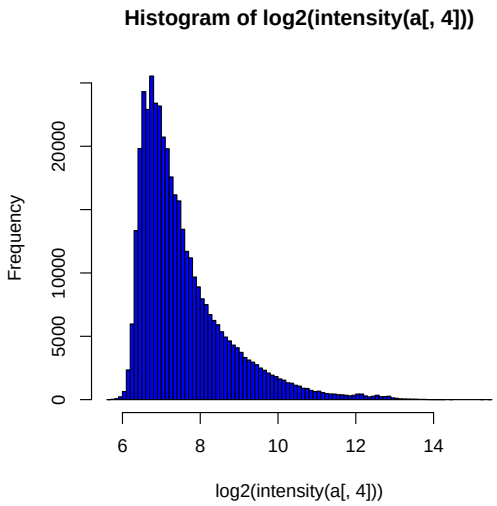


Figure 2: see Exercise 5.

But we have another example:

Note that in these images, row 1 is at the bottom, and row 640 at the top.

5.) Histograms. Another way to visualize what is going on on a chip is to look at the histogram of its intensity distribution. Because of the large dynamical range ($O(10^4)$), it is useful to look at the log-transformed values (see Fig. 2):

6.) Boxplot. To compare the intensity distribution across several chips, we can look at the boxplots, both of the raw intensities `a` and the normalized probe set values `x` (see Fig. 3):

In the commands above, note the different syntax: `a` is an object of type `AffyBatch`, and the `boxplot` function has been programmed to know automatically what to do with it. `exprs(x)` is an object of type `matrix`. What happens if you do `boxplot(x)` or `boxplot(exprs(x))`?

```
class(x)
```

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

```
class(exprs(x))
```

```
## [1] "matrix" "array"
```

7.) Scatterplot. The scatterplot is a visualization that is useful for assessing the variation (or reproducibility, depending on how you look at it) between chips. We can look at all probes, the perfect match probes only, the mismatch probes only, and of course also at the normalized, probe-set-summarized data: (see Fig. 4):

Why are the arrays that were made at $t = 48$h much brighter than those at $t = 10$h? Look at histograms and scatterplots of the probe intensities from chips at 10h and at48h to see whether you can find any evidence of saturation, changes in experimental protocol, or other quality problems. Distinguish between probes that are supposed to represent genes (you can access these e.g. through the functions `pm()`) and control probes.

8.) Heatmap. Select the 50 genes with the highest variation (standard deviation) across chips. (see Fig. 5):

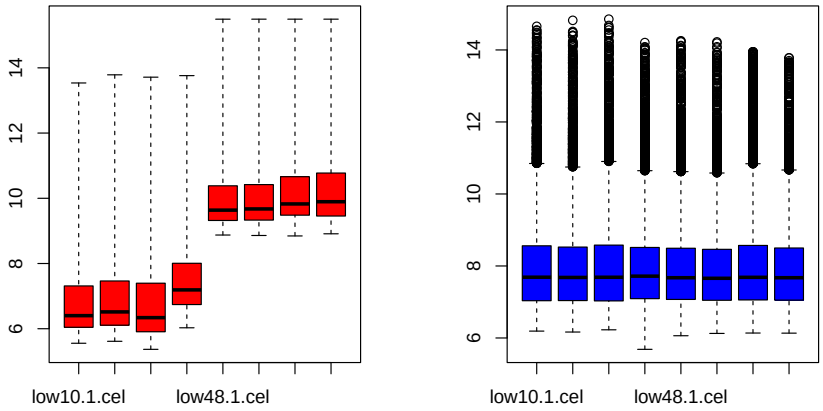


Figure 3: see Exercise 6.

```
rsd <- rowSds(exprs(x))
sel <- order(rsd, decreasing=TRUE)[1:50]
```

9.) ANOVA. Now we can start analysing our data for biological effects. We set up a linear model with main effects for the level of estrogen (`estrogen`) and the time (`time.h`). Both are factors with 2 levels.

```
lm.coef <- function(y)
  lm(y ~ estrogen * time.h)$coefficients
eff <- esApply(x, 1, lm.coef)
```

For each gene, we obtain the fitted coefficients for main effects and interaction:

```
dim(eff)
```

```
## [1]     4 12625
```

```
rownames(eff)
```

```
## [1] "(Intercept)"            "estrogenpresent"        "time.h"
## [4] "estrogenpresent:time.h"
```

```
affyids <- colnames(eff)
```

Let's bring up the mapping from the vendor's probe set identifier to gene names.

```
library(hgu95av2.db)
```

```
## Loading required package:  AnnotationDbi
## Loading required package:  stats4
## Loading required package:  IRanges
## Loading required package:  S4Vectors
##
## Attaching package:  'S4Vectors'
## The following object is masked from 'package:utils':
##
##     findMatches
```

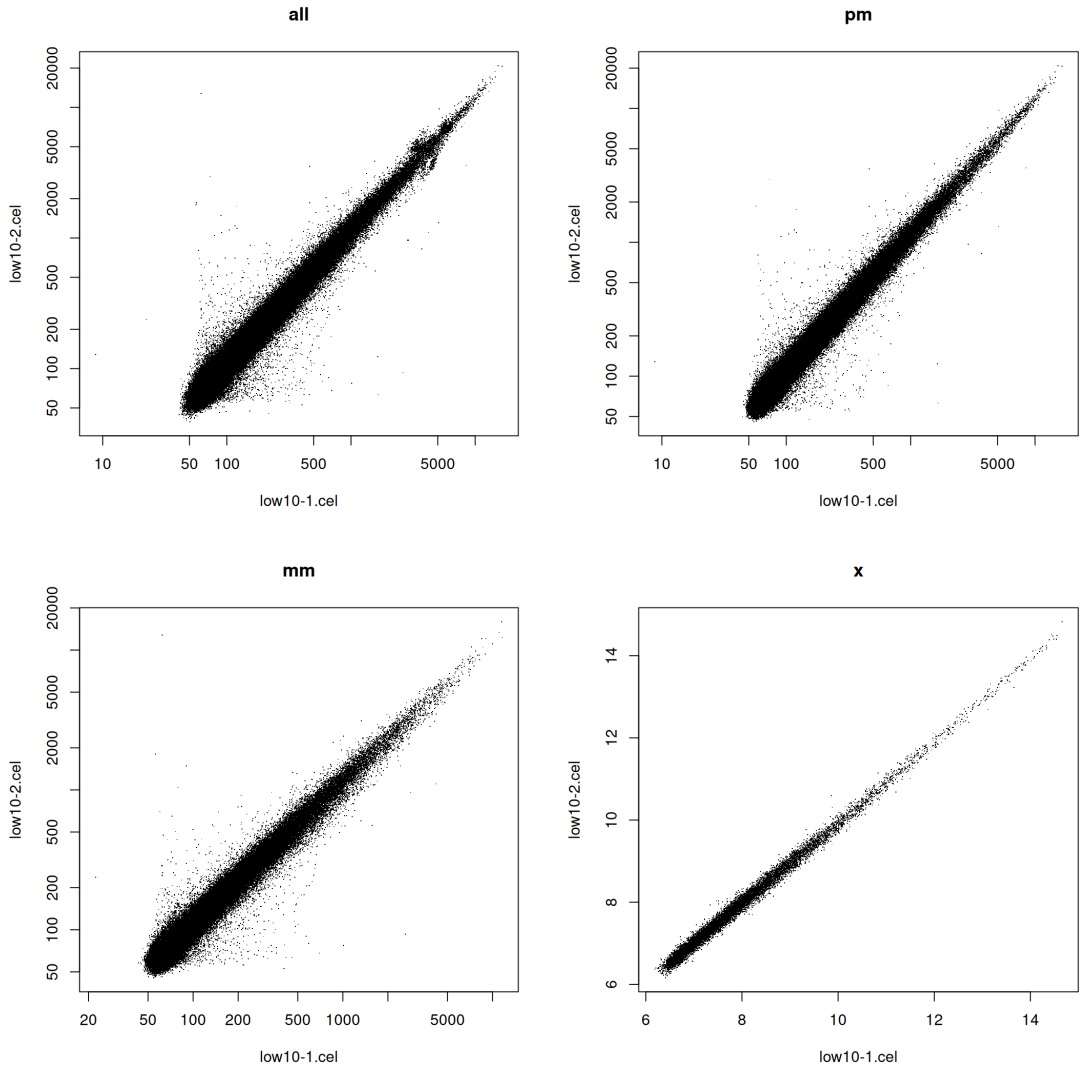


Figure 4: see Exercise 7.

```
## The following objects are masked from 'package:base':
##
##     I, expand.grid, unname
## Loading required package:  org.Hs.eg.db
##
##
ls("package:hgu95av2.db")
##  [1] "hgu95av2"                "hgu95av2.db"             "hgu95av2ACCNUM"
##  [4] "hgu95av2ALIAS2PROBE"     "hgu95av2CHR"             "hgu95av2CHRLENGTHS"
##  [7] "hgu95av2CHRLOC"          "hgu95av2CHRLOCEND"       "hgu95av2ENSEMBL"
## [10] "hgu95av2ENSEMBL2PROBE"   "hgu95av2ENTREZID"        "hgu95av2ENZYME"
## [13] "hgu95av2ENZYME2PROBE"    "hgu95av2GENENAME"        "hgu95av2GO"
## [16] "hgu95av2GO2ALLPROBES"    "hgu95av2GO2PROBE"        "hgu95av2MAP"
```

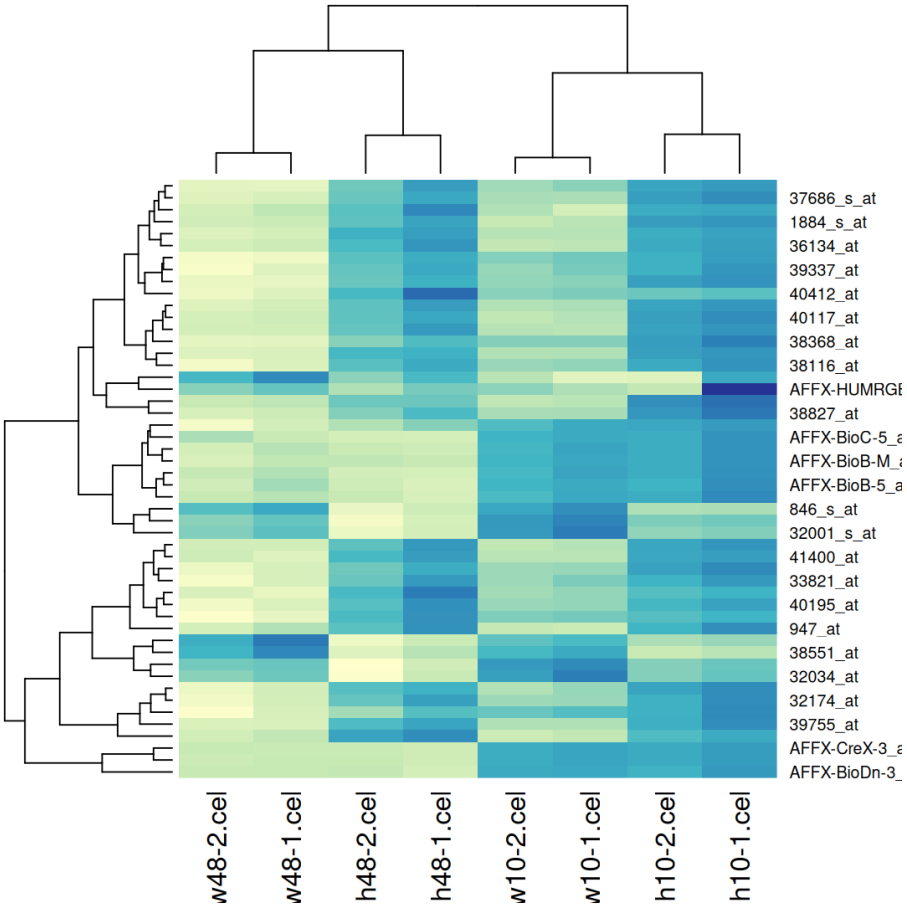
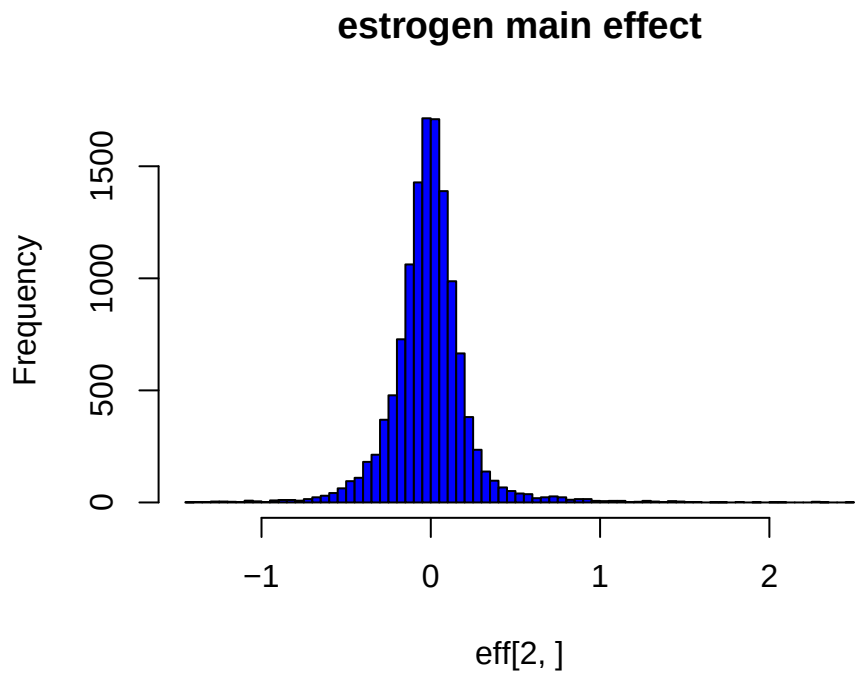


Figure 5: see Exercise 8.

```
## [19] "hgu95av2MAPCOUNTS"     "hgu95av2OMIM"          "hgu95av2ORGANISM"
## [22] "hgu95av2ORGPKG"        "hgu95av2PATH"          "hgu95av2PATH2PROBE"
## [25] "hgu95av2PFAM"          "hgu95av2PMID"          "hgu95av2PMID2PROBE"
## [28] "hgu95av2PROSITE"       "hgu95av2REFSEQ"        "hgu95av2SYMBOL"
## [31] "hgu95av2UNIPROT"       "hgu95av2_dbInfo"       "hgu95av2_dbconn"
## [34] "hgu95av2_dbfile"       "hgu95av2_dbschema"
```

Let's now first look at the **estrogen main effect**, and print the top 3 genes with largest effect in one direction, as well as in the other direction. Then, look at the **estrogen:time interaction**.

```
hist(eff[2,], breaks=100, col="blue", main="estrogen main effect")
```



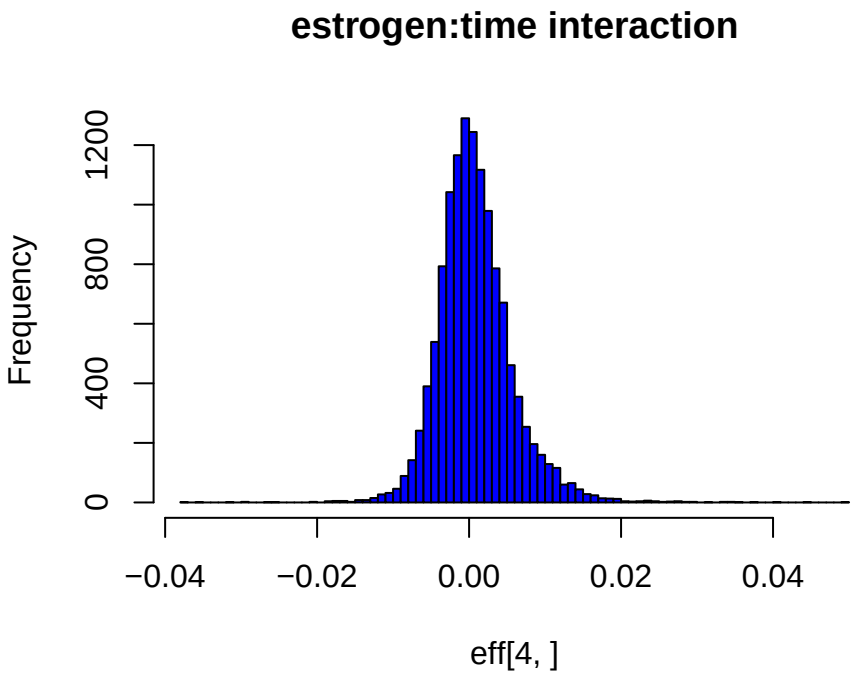

```
lowest <- sort(eff[2,], decreasing=FALSE)[1:3]
mget(names(lowest), hgu95av2GENENAME)
```

```
## $`37294_at`
## [1] "BTG anti-proliferation factor 1"
##
## $`846_s_at`
## [1] "BCL2 antagonist/killer 1"
##
## $`36617_at`
## [1] "inhibitor of DNA binding 1"
```

```
highest <- sort(eff[2,], decreasing=TRUE)[1:3]
mget(names(highest), hgu95av2GENENAME)
```

```
## $`910_at`
## [1] "thymidine kinase 1"
##
## $`31798_at`
## [1] "trefoil factor 1"
##
## $`1884_s_at`
## [1] "proliferating cell nuclear antigen"
```

```
hist(eff[4,], breaks=100, col="blue", main="estrogen:time interaction")
```




```
highia <- sort(eff[4,], decreasing=TRUE)[1:3]
mget(names(highia), hgu95av2GENENAME)
```

```
## $`1651_at`
## [1] "ubiquitin conjugating enzyme E2 C"
##
## $`40412_at`
## [1] "PTTG1 regulator of sister chromatid separation, securin"
##
## $`1945_at`
## [1] "cyclin B1"
```