Microarray analysis of oral cancer samples

Humberto Ortiz-Zuazaga

April 27, 2011

1 Introduction

Bioconductor [4] is a set of R packages for analysis of biological data, with an emphasis on microarray and other high-throughput datasets.

This example will use standard `affy` [3] and `limma` [5] commands to analyze the workshop dataset. Bioconductor has extensive help, which you can access in many ways. One simple way is to type `?foo` where you want help on the object called "foo". You can open an interactive browser interface to the help system by typing `help.start()`. In the browser, you can look at the documentation for the installed packages to find help on `limma` and `affy`.

```
> library(limma)
> library(affy)
```

2 Reading the data

A simple text file with tab separated columns can describe the microarray samples. In our case the first 6 samples are positive for HPV, and the remaining 5 samples are negative. These are labeled "pos" and "neg" in the targets file.

```
> targets <- readTargets("targets.txt")
> targets
```

```
                    FileName Target
1   OC-1_(HuGene-1_0-st-v1).CEL    pos
2   OC-5_(HuGene-1_0-st-v1).CEL    pos
3   OC-6_(HuGene-1_0-st-v1).CEL    pos
4   OC-7_(HuGene-1_0-st-v1).CEL    pos
5   OC-8_(HuGene-1_0-st-v1).CEL    pos
6  OC-10_(HuGene-1_0-st-v1).CEL    pos
7  OC-11_(HuGene-1_0-st-v1).CEL    neg
8  OC-12_(HuGene-1_0-st-v1).CEL    neg
9  OC-13_(HuGene-1_0-st-v1).CEL    neg
10 OC-14_(HuGene-1_0-st-v1).CEL    neg
11 OC-15_(HuGene-1_0-st-v1).CEL    neg
```

```
> ab <- ReadAffy(filenames = targets$FileName)
```

`ab` will contain the AffyBatch, with the raw expression values for each probe in each sample, with additional information on the probes and samples.

3 Normalization and pre-processing

We can use the `rma` command to normalize and summarize the probes for each feature. Prior to the summarization, each feature is represented with four probes. After the normalization and summarization routine, we have a single expression value for each feature in each sample.

```
> probeNames(ab)[1:10]
```

```
 [1] "7892501" "7892501" "7892501" "7892501" "7892502" "7892502" "7892502"
 [8] "7892502" "7892503" "7892503"
```

```
> eset <- rma(ab)
```

```
Background correcting
Normalizing
Calculating Expression
```

```
> featureNames(eset)[1:10]
```

```
 [1] "7892501" "7892502" "7892503" "7892504" "7892505" "7892506" "7892507"
 [8] "7892508" "7892509" "7892510"
```

A boxplot shows the distribution of expression values before (Figure 1) and after (Figure 2) the normalization.

```
> boxplot(ab)
```

```
> boxplot(exprs(eset))
```

4 Experimental design

The experiment has a simple design, each sample is labeled in the targets file with the target it was hybridized with. This information can be used to constuct a design matrix that identifies each group.

```
> f <- factor(targets$Target, levels = c("pos", "neg"))
> design <- model.matrix(~0 + f)
> colnames(design) <- c("pos", "neg")
> design
```

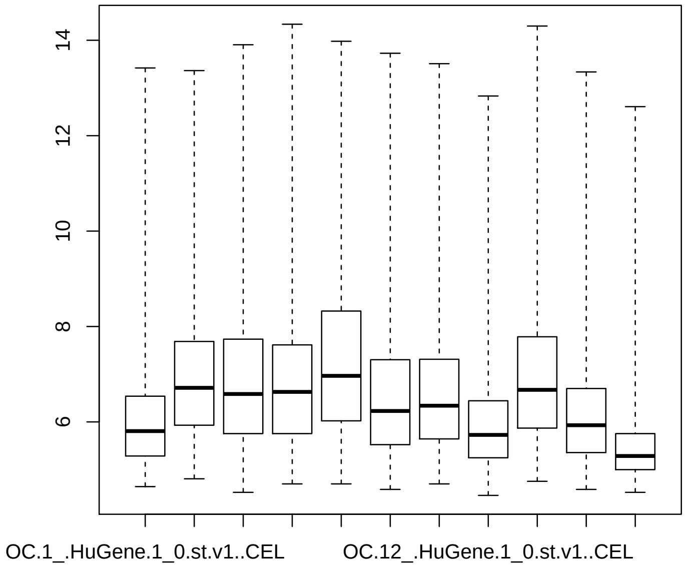

Figure 1: Box plot before normalization

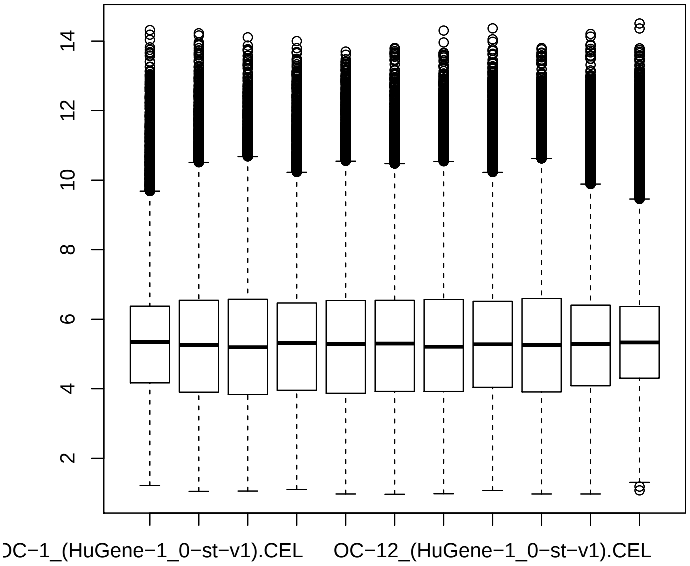

Figure 2: Box plot after normalization

```
   pos neg
1    1   0
2    1   0
3    1   0
4    1   0
5    1   0
6    1   0
7    0   1
8    0   1
9    0   1
10   0   1
11   0   1
attr(,"assign")
[1] 1 1
attr(,"contrasts")
attr(,"contrasts")$f
[1] "contr.treatment"
```

We can fit a model that has a mean for each group, and test if the group means are different. The `eBayes` function computes an empirical Bayes factor, pooling the variances from all the genes to estimate significance.

```
> cont.matrix <- makeContrasts(posvsneg = pos - neg, levels = design)
> cont.matrix
```

```
      Contrasts
Levels posvsneg
   pos        1
   neg       -1
```

```
> fit <- lmFit(eset, design)
> fit2 <- contrasts.fit(fit, cont.matrix)
> fit.b <- eBayes(fit2)
```

5 Reporting the results

We now have a model fit that estimates the log ratios between the positive and negative samples. An MA plot (Figure 3) summarizes the fit. The y axis plots M, the log ratio of expression in the positive and negative coefficients. The x axis plots the A, or average log intensity of each gene.

```
> plotMA(fit.b)
```

The fit also has an estimate of the Bayes factor, the log odds of differential expression for each gene. A plot of the B vs log ratios is called a volcanoplot (see Figure 4).

```
> volcanoplot(fit.b)
```

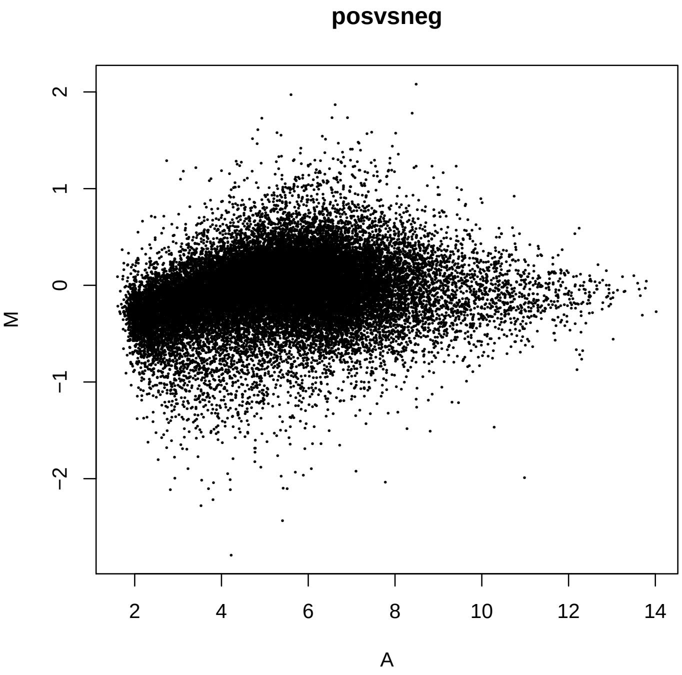

Figure 3: MA plot

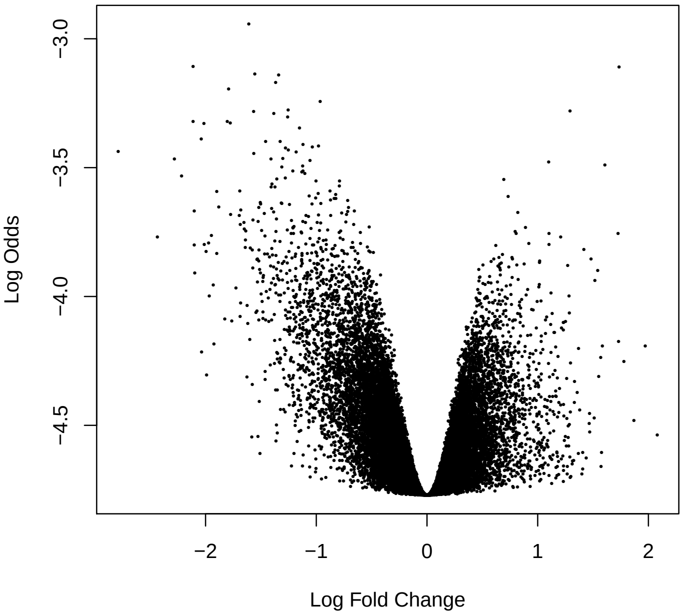

Figure 4: Volcano plot

Another way to report the results is exporting a table with the most significant features. Estimated p-values using a number of multiple testing corrections can be computed, in this case we use the Benjamini & Hochberg correction. [1]

```
> topTable(fit.b, adjust = "BH")

           ID     logFC  AveExpr         t      P.Value adj.P.Val         B
974   7893495 -1.609189 2.850864 -7.485772 6.893302e-06 0.2227984 -2.943503
13025 7987464 -2.112633 2.822780 -6.148559 4.716153e-05 0.4252392 -3.107950
12861 7985571  1.735895 6.550553  6.142491 4.760069e-05 0.4252392 -3.108858
9536  7951865 -1.554120 5.275033 -5.967527 6.232155e-05 0.4252392 -3.135787
2763  7895321 -1.339255 3.022994 -5.932751 6.578373e-05 0.4252392 -3.141315
3547  7896127 -1.366318 3.090447 -5.756009 8.681311e-05 0.4676477 -3.170342
1694  7894231 -1.791860 4.267852 -5.615846 1.085106e-04 0.5010245 -3.194516
14747 8006296 -0.963775 3.030733 -5.353550 1.659613e-04 0.6396166 -3.242671
301   7892809 -1.253387 2.922941 -5.182932 2.199338e-04 0.6396166 -3.276156
1429  7893959  1.291958 7.529752  5.162851 2.274056e-04 0.6396166 -3.280214
```

References

[1] Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B,* 57, 289–300.

[2] Gautier, L., Cope, L., Bolstad, B. M., and Irizarry, R. A. (2004). affy—analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics* 20, 3 (Feb. 2004), 307-315.

[3] R. Gentleman, V. J. Carey, D. M. Bates, B.Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, and others Bioconductor: Open software development for computational biology and bioinformatics (2004). *Genome Biology*, Vol. 5, R80

[4] Smyth, G. K. (2005). Limma: linear models for microarray data. In: 'Bioinformatics and Computational Biology Solutions using R and Bioconductor'. R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, W. Huber (eds), Springer, New York, pages 397–420.