Biological Science Division - Center for Research Informatics

Analysis of Microarray data with R and Bioconductor

Instructors: Jorge Andrade, Ph.D. & Bao Riyue Ph.D.

January 17, 2013

INTRODUCTION:

This hands-on tutorial is focused on the analysis of Affymetrix microarray data using R and Bioconductor, this tutorial assumes that you have previous experience using R for data analysis.

THE DATA:

- Data: Down syndrome is caused by an extra copy of all or part of chromosome 21; it is the most common non-lethal trisomy in humans. The study used in this tutorial revealed a significant up-regulation of chromosome 21 genes at the gene expression level in individuals with Down syndrome; this dysregulation was largely specific to chromosome 21 only and not to any other chromosomes. This experiment was performed using the Affymetrix® GeneChip™ Human U133A arrays. It includes 25 samples taken from 10 human subjects and 4 different tissues.
- The raw data for this study is available as experiment number GSE1397 in the Gene Expression Omnibus: http://www.ncbi.nlm.nih.gov/geo/

Please download and unzip the data the link below:

ftp://logia.cri.uchicago.edu/tutorials/Jan2013/DownSyndromeData.zip

PREPROCESSING:

Preprocessing Affymetrix data is a well-defined process consisting of the following steps:

1. Importing the "raw" data in .CEL format and the PHENODATA;
2. Summarize the expression values per each probe set.

Summarizing expression values is constituted of the following steps:

1. BACKGROUND CORRECTION;

2. NORMALIZATION;
3. SUMMARIZATION.

All these operations are supported in the Bioconductor package *affy*.

Preliminary operations

We will start by defining our working directory:

```
> setwd("C:/Users/jandrade/Documents/DownSyndrome")
> getwd()
[1] "C:/Users/jandrade/Documents/DownSyndrome"
```

The folder *DownSyndrome* contains twenty five *.CEL* raw data expression files and the file: *phenod.txt* with a list of file names and the associated phenotype information.

Installing *affy* and *affycoretools* packages under R environment:

```
> source("http://www.bioconductor.org/biocLite.R")
> biocLite("affy")
> biocLite("affycoretools")
```

Loading *affy* and *affycoretools* packages into our R environment:

```
> library(affy)
> library(affycoretools)
```

Reading all *.CEL (*.cel) files in your current working directory and storing them into the AffyBatch object *mydata*

```
> mydata <- ReadAffy()
```

Where *mydata* is the new `AffyBatch` object that will be created at the end of the process operated by the function `ReadAffy`.

Importing the phenotype data:

```
> pData(mydata)<-read.table("phenod.txt", header=T,
row.names=1, sep="\t")
```

The phenotype object contains all the information about the samples of our dataset. It is always important to keep track off all the information about our samples, since they can strongly affect the final results.

Visualizing the phenotype data:

```
> pData(mydata)
```

```
                                          diagnosis     tissue
Down Syndrome-Astrocyte-1478-1-U133A.CEL  DownSyndrom  Astrocyte
Down Syndrome-Astrocyte-748-1-U133A.CEL   DownSyndrom  Astrocyte
Down Syndrome-Cerebellum-1218-1-U133A.CEL DownSyndrom Cerebellum
Down Syndrome-Cerebellum-1389-1-U133A.CEL DownSyndrom Cerebellum
Down Syndrome-Cerebellum-1478-1-U133A.CEL DownSyndrom Cerebellum
Down Syndrome-Cerebrum-1218-1-U133A.CEL   DownSyndrom Cerebellum
Down Syndrome-Cerebrum-1389-1-U133A.CEL   DownSyndrom Cerebellum
Down Syndrome-Cerebrum-1478-1-U133A.CEL   DownSyndrom Cerebellum
Down Syndrome-Cerebrum-847-1-U133A.CEL    DownSyndrom Cerebellum
Down Syndrome-Heart-1218-1-U133A.CEL      DownSyndrom      Heart
Down Syndrome-Heart-1478-1-U133A.CEL      DownSyndrom      Heart
Normal-Astrocyte-1479-1-U133A.CEL              Normal  Astrocyte
Normal-Astrocyte-1521-1-U133A.CEL              Normal  Astrocyte
Normal-Cerebellum-1390-1-U133A.CEL             Normal Cerebellum
Normal-Cerebellum-1411-1-U133A.CEL             Normal Cerebellum
Normal-Cerebellum-1521-1-U133A.CEL             Normal Cerebellum
Normal-Cerebrum-1390-1-U133A.CEL               Normal Cerebellum
Normal-Cerebrum-1390-2-U133A.CEL               Normal Cerebellum
Normal-Cerebrum-1411-1-U133A.CEL               Normal Cerebellum
Normal-Cerebrum-1411-2-U133A.CEL               Normal Cerebellum
Normal-Cerebrum-1521-1-U133A.CEL               Normal Cerebellum
Normal-Cerebrum-1521-2-U133A.CEL               Normal Cerebellum
Normal-Cerebrum-1565-1-U133A.CEL               Normal Cerebellum
Normal-Heart-1390-1-U133A.CEL                  Normal      Heart
Normal-Heart-1411-1-U133A.CEL                  Normal      Heart
```

Exploring the affybatch:

Typing just the name of the *Affybatch* we have created by importing the data, will give us a summary about the dataset.

```
> mydata
```

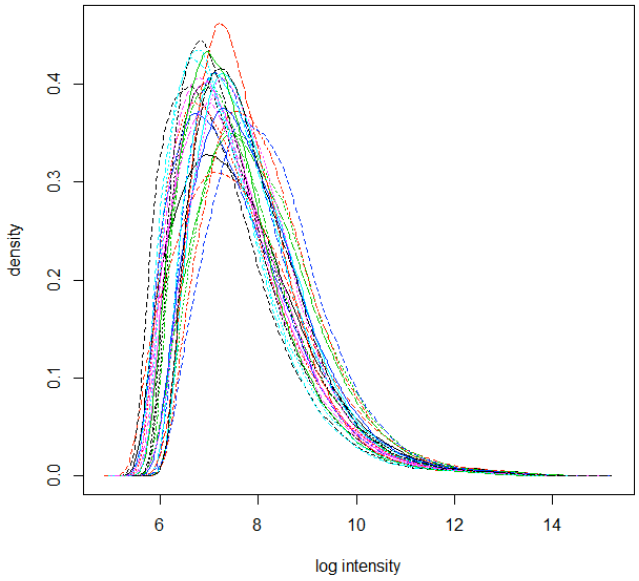
```
AffyBatch object
size of arrays=712x712 features (29 kb)
cdf=HG-U133A (22283 affyids)
number of samples=25
number of genes=22283
annotation=hgu133a
notes=
```

PRELIMINARY EXPLORATORY ANALYSIS

It is possible to visually explore the dataset using different methods, and this analysis will allow us to easily identify outliers and/or other problems with the samples.

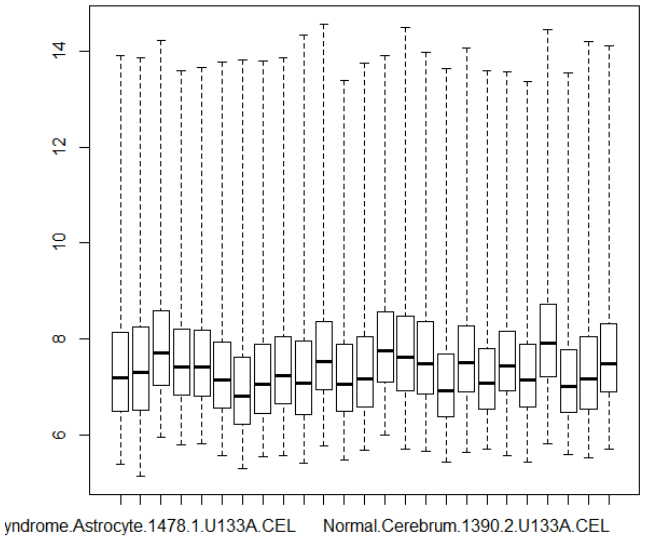
We can produce a density/intensity *histogram*.

```
> hist(mydata)
```



The *boxplot* is another way to visualize the distribution of the intensities in each array of the dataset.

```
> boxplot(mydata)
```



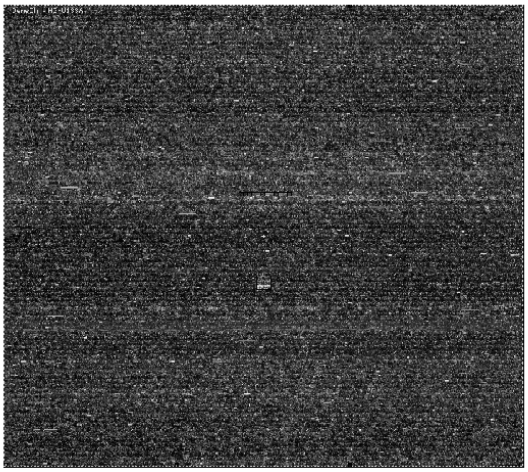
These two graphics are showing us:

1. There is no evident outlier in the data set
2. The distribution of the intensities in each array (boxplot) illustrates the need of a normalization step.

It is also possible to visualize the *image* representing each CEL file (in this example, we visualize the first slide of the dataset only). In this way, it is possible to pinpoint technical problems occurring eventually only to one region of the array.

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

Down Syndrome-Astrocyte-1478-1-U133A.CEL



SUMMARIZING THE EXPRESSION VALUES

In *affy* package there are several methodologies available to *correct background*, *normalize* and *summarize* expression values per each probe set of the dataset.

Here we will use the RMA and MAS5 methods that are implemented into ready made functions.

In order to apply the RMA method, we can type:

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

This will create a normalized and background corrected set of expression values using the RMA method. The generated data are stored as ExpressionSet class in the *eset* object.

We can visualize a summary of this new object:

```
> eset

ExpressionSet (storageMode: lockedEnvironment)
assayData: 22283 features, 25 samples
  element names: exprs
protocolData
  sampleNames: Down Syndrome-Astrocyte-1478-1-U133A.CEL Down Syndrome-
Astrocyte-748-1-U133A.CEL ... Normal-Heart-1411-1-U133A.CEL (25 total)
  varLabels: ScanDate
  varMetadata: labelDescription
```


```
phenoData
  sampleNames: Down Syndrome-Astrocyte-1478-1-U133A.CEL Down Syndrome-
Astrocyte-748-1-U133A.CEL ... Normal-Heart-1411-1-U133A.CEL (25 total)
  varLabels: diagnosis tissue
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu133a
```

For the MAS5 method we can type:

```
> eset1 <- mas5(mydata)
```

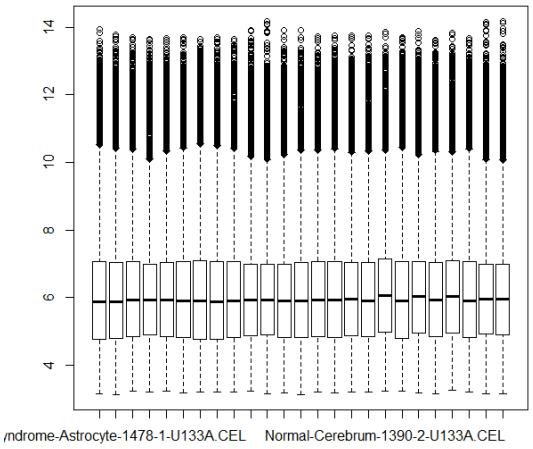
MAS5 normalizes each array independently and sequentially; RMA as the name suggests (robust multi-array) uses a multi-chip model. The functions rma and mas5 are implemented in C++ language and for this reason they are quite fast, however MAS5 is considerable slower when compare with RMA.

To save the expression values in an Excel file we can use:

```
> write.exprs(eset, file="Expression_values.xls")
```

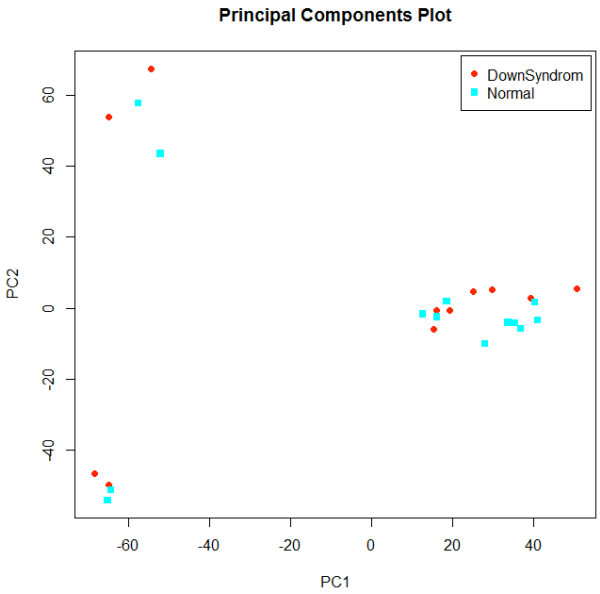
EXPLORATORY ANALYSIS

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



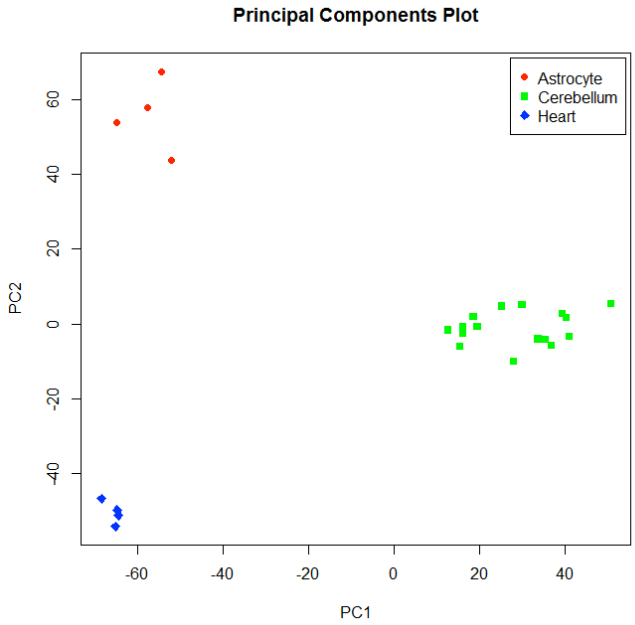
Pricipal Component Analysis:

```
> plotPCA(eset, groups = as.numeric(pData(mydata)[,1]), groupnames =
levels(pData(mydata)[,1]))
```



PC1 shows no evident grouping by the factor *diagnosis*, next we would like to evaluate if there is any grouping by the factor *tissue*:

```
> plotPCA(eset, groups = as.numeric(pData(mydata)[,2]), groupnames =
levels(pData(mydata)[,2]))
```



Expression is apparently grouped by factor *tissue*.

ANALYSIS OF DIFFERENTIAL GENE EXPRESSION

We will use the *limma* (Linear Models for Microarray Data) package from Bioconductor to perform the analysis of diferential gene expression:

```
> #source("http://bioconductor.org/biocLite.R")
> #biocLite("limma")

> library(limma)
```

Creating a two groups compare design matrix:

```
> Group<- factor(pData(mydata)[,1 ] , levels = levels(pData(mydata)[,1]))
> design<- model.matrix(~Group)
```

Fiting a linear model for each gene in the expression set *eset* given the design matrix:

```
> fit <-lmFit(eset,design)
```

Now we are going to calculate the differential expression by empirical Bayes shrinkage of the standard errors towards a common value, by computing the moderated t-statistics, moderated F-statistic, and log-odds:


```
> fit<-eBayes(fit)
```

Creating a table with the top 50 most statistically significant differentially expressed genes between the groups classified by corrected p-value:

```
> tab <- topTable(fit, coef = 2, adjust = "fdr", n = 50)
> write.table(tab ,file="DEG.xls",row.names=F, sep="\t")
```

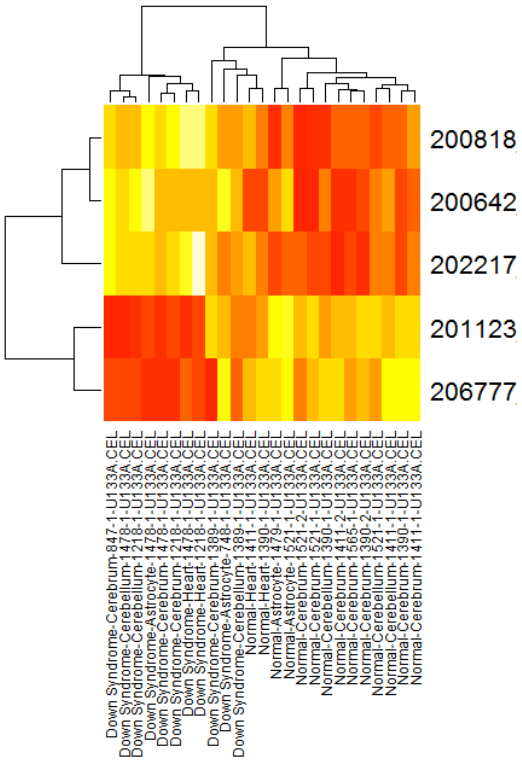
Next, we are going to select those genes that have adjusted p-values below 0.05, to create a list a small number of highly significant genes:

```
> selected  <- p.adjust(fit$p.value[, 2]) <0.05
> esetSel <- eset [selected, ]
> esetSel

ExpressionSet (storageMode: lockedEnvironment)
assayData: 5 features, 25 samples
  element names: exprs
protocolData
  sampleNames: Down Syndrome-Astrocyte-1478-1-U133A.CEL Down Syndrome-Astrocyte-748-1-
U133A.CEL ... Normal-Heart-1411-1-U133A.CEL (25
    total)
  varLabels: ScanDate
  varMetadata: labelDescription
phenoData
  sampleNames: Down Syndrome-Astrocyte-1478-1-U133A.CEL Down Syndrome-Astrocyte-748-1-
U133A.CEL ... Normal-Heart-1411-1-U133A.CEL (25
    total)
  varLabels: diagnosis tissue
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu133a
```

Next we are going to create a heatmap of the expression of highly significant genes:

```
> heatmap(exprs(esetSel))
```



ANNOTATION

The final step in this analysis will be the annotation of the DEGs. First we are going to create a new expresionset *eset2* with selected top 50 DEGs:

```
> eset2 <- eset [tab[,1]]
```

To verify witch platform was used for the data:

```
> eset@annotation
[1] "hgu133a"
```

We will now install and load the libraries we need for the annoation:


```
> biocLite("hgu133a.db")
> biocLite("annotate")
> biocLite("R2HTML")
> library(hgu133a.db)
> library(annotate)
> library(R2HTML)
```

To list the objects available in this annotation package we can use:

```
> ls("package:hgu133a.db")
```

```
[1] "hgu133a"              "hgu133a.db"           "hgu133a_dbconn"       "hgu133a_dbfile"       "hgu133a_dbInfo"
[6] "hgu133a_dbschema"     "hgu133aACCNUM"        "hgu133aALIAS2PROBE"   "hgu133aCHR"           "hgu133aCHRLENGTHS"
[11] "hgu133aCHRLOC"        "hgu133aCHRLOCEND"     "hgu133aENSEMBL"       "hgu133aENSEMBL2PROBE" "hgu133aENTREZID"
[16] "hgu133aENZYME"        "hgu133aENZYME2PROBE"  "hgu133aGENENAME"      "hgu133aGO"            "hgu133aGO2ALLPROBES"
[21] "hgu133aGO2PROBE"      "hgu133aMAP"           "hgu133aMAPCOUNTS"     "hgu133aOMIM"          "hgu133aORGANISM"
[26] "hgu133aORGPKG"        "hgu133aPATH"          "hgu133aPATH2PROBE"    "hgu133aPFAM"          "hgu133aPMID"
[31] "hgu133aPMID2PROBE"    "hgu133aPROSITE"       "hgu133aREFSEQ"        "hgu133aSYMBOL"        "hgu133aUNIGENE"
[36] "hgu133aUNIPROT"
```

We are now going to extract the feature names from *eset2* that contains the selected genes of interest:

```
> ID <- featureNames(eset2)
```

And look up the Gene Symbol, Name, and Ensembl Gene ID for each of those IDs:

```
> Symbol <- getSYMBOL(ID, "hgu133a.db")
> Name <- as.character(lookUp(ID, "hgu133a.db", "GENENAME"))
> Ensembl <- as.character(lookUp(ID, "hgu133a.db", "ENSEMBL"))
```

For each Ensembl ID (if we have it), we will now create a hyperlink that goes to the Ensembl genome browser:

```
> Ensembl <- ifelse(Ensembl=="NA", NA,
                  paste("<a href='http://useast.ensembl.org/Homo_sapiens/Gene/Summary?g=",
                        Ensembl, "'>", Ensembl, "</a>", sep=""))
```

And make a temporary data frame with all those identifiers:

```
> tmp <- data.frame(ID=ID, Symbol=Symbol, Name=Name, Ensembl=Ensembl, stringsAsFactors=F)

> tmp[tmp=="NA"] <- NA  # The stringsAsFactors makes "NA" characters. This fixes that problem.
```

We are now going to write out an HTML file with clickable links to the Ensembl Genome Browser, and .txt file with gene list:

```
> HTML(tmp, "out.html", append=F)
> write.table(tmp ,file="target.txt",row.names=F, sep="\t")
```

Please browse the created out.html file and try the links, as expected most of the detected significantly DEGs are located in chromosome 21.

For questions or comments please contact:

- Jorge Andrade, Ph.D.: jandrade@bsd.uchicago.edu
- Bao Riyue, Ph.D.: rbao@bsd.uchicago.edu