Mannheimia goes programming

Saturday, May 5, 2012

Agilent microarray data analysis in R

A month ago I started a collaboration with some researchers of my home university in Peru. They are doing some experiments with *Aspergillus niger* to observe its gene expression under different substrates and types of cultures. They are evaluating two substrates: lactose and xylose; and two types of culture: biofilms and submerged culture. Thus their experimental flasks were labeled something like this: CBL, CSL, CBX and CSX from their Spanish abbreviation: C for culture, B for biofilm, S for submerged, L for lactose and X for xylose.

As you may know already, R is a free statistical software that, among many other packages, has microarray packages developed by the Bioconductor group and collaborators. For the analysis of Agilent data that has been obtained from the proprietary *Agilent Feature Extraction* (AFE) image analysis algorithm, readers can refer to http://www.biomedcentral.com/1471-2164/12/64 and make use of the *AgiMicroRna* library in R. However, in this post I deal with Agilent data in a more general manner. It means that the code that will be presented a few lines below can be used for data obtained from AFE, but also for data that is not. Let's see!

First of all, we type in the working directory and upload the limma package:

```
# Set the working directory or path where your data is localized.
> setwd("C:/...")
> library(limma)
```

At this point I have to make an interruption to say that the analysis of Agilent data requires a so called **"target file"**, which is just a **tab delimited text file** created by the user, and containing the experimental desing. Have a look at the one for my *A. niger* experiment:

```
SampleNumber  FileName     Condition
1         cbl1.txt     CBL
2         cbl2.txt     CBL
3         cbl3.txt     CBL
4         cbx1.txt     CBX
4         cbx2.txt     CBX
6         cbx3.txt     CBX
7         csl1.txt     CSL
8         csl2.txt     CSL
9         csl3.txt     CSL
10        csx1.txt     CSX
11        csx2.txt     CSX
12        csx3.txt     CSX
```

As you can see, in my experiment, I have 12 samples representing 4 conditions or treatments (i.e., CBL, CBX, CSL and CSX), and each condition has 3 replicates whose intensity signals are store in different files output by the scanner or any other Agilent image analyser. I have simply called my target file "target.txt" and store it in the current working directory. So now we can continue with the R code:

```
> targets = readTargets("targets.txt")

> raw = read.maimages(targets, source="agilent",green.only=TRUE)


# Subtract the background signal.
> raw_BGcorrected = backgroundCorrect(raw, method="normexp", offset=16)
# Then normalize and log-transformed the data.
```

Blog Archive

▼ 2012 (4)
► June (1)
▼ May (2)
Coroutines in Python: an example applied to Bioinf...
Agilent microarray data analysis in R
► April (1)

About Me

Mannheimia

Mannheimia succiniciproducens lives inside the stomachs of cows and helps them digest grasses. It is an isolated capnophilic (grows best in the presence of carbon dioxide) bacterium found in the bovine rumen. M. succiniciproducens is a non pathogenic, non-spore-forming, mesophilic, Gram-negative bacterium of the genus coccobacillus. Mannheimia succiniciproducens efficiently fixes carbon dioxide and produces substantial amounts of succinic acid. Researchers believe it will be possible to use this bacterium for the efficient production of succinic acid, an important industrial chemical that can be used as a green feedstock for the manufacture of biodegradable polymers, synthetic resins and various chemical intermediates and additives. The M. succiniciproducens genome is a single circular chromosome of 2,314,078 base pairs (bp) with no plasmid. Nat. Biotechnol. 22(10):1275-1281(2004)

View my complete profile

```
> raw_BGandNormalized = normalizeBetweenArrays(raw_BGcorrected,method="quantile")
# Finally calculate the average intensity values from the probes of each gene.
> raw_aver = avereps(raw_BGandNormalized,ID=raw_BGandNormalized$genes$ProbeName)
```

We can also asses the quality of the data before and after normalization. For that, we can use boxplots or MA plots:

```
# Before normalization.
> boxplot(log(as.matrix(raw_BGcorrected)),las=2,ylab="Log2(Intensity)")
# After it.
> boxplot(as.matrix(raw_BGandNormalized), las=2, ylab = "Log2(Intensity)")


# Now some MA plots of only one replicate per condition:
> library(affy)
> mva.pairs(as.matrix(raw)[,c(1,4,7,10)]) # Before BG correction
> mva.pairs(as.matrix(raw_BGcorrected)[,c(1,4,7,10)]) # Before normalization.
> mva.pairs(as.matrix(raw_BGandNormalized)[,c(1,4,7,10)]) # After it.
```

After that we can proceed with the differential expression analysis. For that we need to create a design matrix and a contrast matrix. For the former, the user can use the information already store in the target file:

```
> f = factor(targets$Condition)
> design = model.matrix(~f)
```

Or to make this post more didactic erase the above two lines and let's create the design matrix in a different way:

```
> design = cbind(CBL = c(1,1,1,0,0,0,0,0,0,0,0,0),    # First 3 replicates -> CBL
                 CBX = c(0,0,0,1,1,1,0,0,0,0,0,0),    # The following 3 -> CBX
                 CSL = c(0,0,0,0,0,0,1,1,1,0,0,0),    # The following 3 -> CSL
                 CSX = c(0,0,0,0,0,0,0,0,0,1,1,1))   # Last 3 replicates -> CSX

# Now fit the average intensity data to the design matrix:
> fit = lmFit(raw_aver, design)
```

Then we create the constrast matrix for each comparison we are interested in and asses for the differences in expression using a t-test.

```
> contrastMatrix = makeContrasts("CBL-CSL","CBX-CSX","CBL-CBX","CSL-CSX", levels=design)


> fit2 = contrasts.fit(fit, contrastMatrix)
> fit2 = eBayes(fit2)
```

Finally we might want to see the top ten significant hits in each comparison, or observe the highly significant or all the significant hits, or we may want to store the upregulated and downregulated hits in different files. For that we just need to type in the final lines:

```
# Now let's look for the top ten signficants hits in each comparison:
> topTable(fit2, coef = "CBL-CSL")   # The ten top significants in CBL vs CSL.
> topTable(fit2, coef = "CBX-CSX")   # The ten top significants in CBX vs CSX.
> topTable(fit2, coef = "CBL-CBX")   # The ten top significants in CBL vs CBX.
> topTable(fit2, coef = "CSL-CSX")   # The ten top significants in CSL vs CSX.

# The significant hits based on adjusted p-value for the comparison CBL vs CSL:
> sig = length(which(topTable(fit2, coef = "CBL-CSL",number=42370)[,15]<0.05))
> signif = topTable(fit2, coef = "CBL-CSL",number=sig)


> upregulated = signif[which(signif[,11]>0),]   # The upregulated hits in CBL.
> downregulated = signif[which(signif[,11]<0),]  # The downregulated ones.

# Save them in different files for future annotation or functional cluster
# analysis:
> write.table(upregulated, "CBLvsCSL_Upre.txt", sep="\t")
> write.table(downregulated, "CBLvsCSL_Downre.txt", sep="\t")
```

To end this post I just want to make some notes. Firstly, the parameters of the topTable function can be studied by just typing in `> help(topTable)`. For my A. niger data, I have used `number=42370` because the number of probes in my microarrays are less but close to that number. Secondly, the indexes `[,15]` and `[,11]` may change for the user's data depending on the column localization of the adjusted p-value and the fold change, respectively, in the user's own data frame of significant hits. Finally, with some more scripts the user can also make heatmaps and cluster analyses. That might be a topic for another post in this blog. Till next time!

Posted by Mannheimia at 8:17 PM

Labels: Agilent, microarray data analysis, R

2 comments:

Unknown February 23, 2016 at 1:44 PM

Thanks a lot, Working fine for Agilent one color microarray data

Reply

Unknown July 9, 2020 at 2:55 PM

Hey! I am getting bored, please fchat with me ;) ;) ;) ... [redacted]

Reply

Enter your comment...

Comment as: federicoalessaı **Sign out**

Publish **Preview** ☐ **Notify me**

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)