Microarray Data Analysis using R

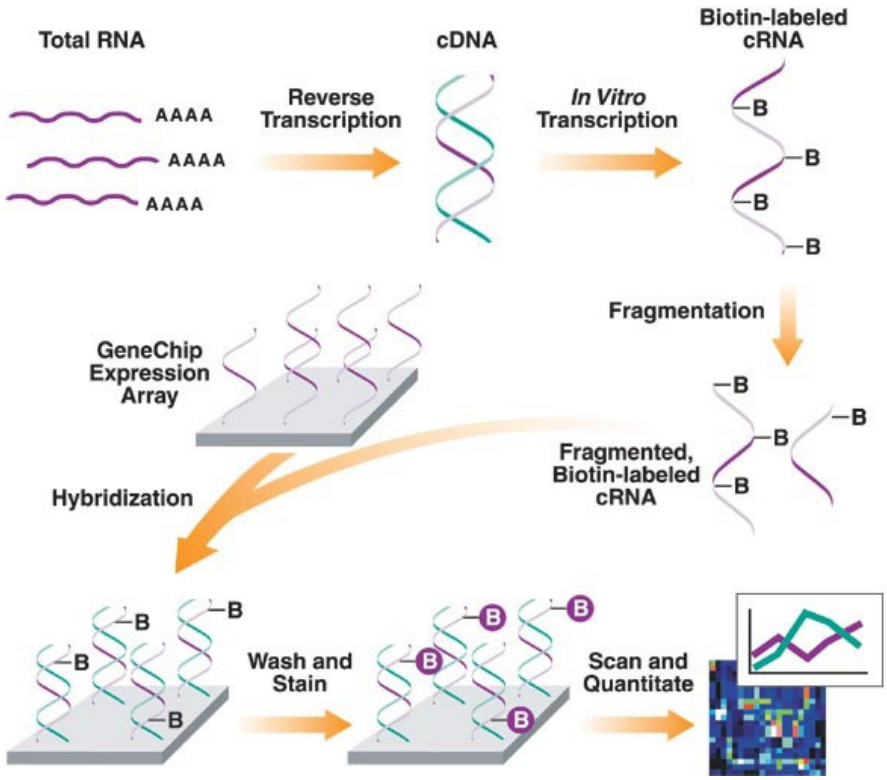
Microarray data analysis is becoming an increasingly integral part of biological research. Analysis of cell expression that would have previously taken months to perform can now be carried out in a matter of hours with the use of these miraculous chips. The analysis of gene expression values is of key importance in bioinformatics. The technique makes it possible to give an initial answer to many important genetic type of questions.

Microarrays

A microarray is device that allows for fast and precise analysis of messenger ribonucleic acid (mRNA) directly from a cell. It consists of two parts: the chip and the optical reader. The chip is constructed from a plate of glass to which tens of thousands of cDNA genes are chemically attached in specific locations called *spots*. An optical reader is used to analyze which of the spots have sample mRNA bound to them and return the results in a format that researchers can use for further analysis.

An experiment is carried out by tagging all mRNA prepared from a sample (cell, tissue, or other biological source) with a fluorescent tag. These tags can be a single color (used in the *Affymetrix GeneChip*) or more colors (such as the complimentary Cy3 red and Cy5 green dies). The sample is then allowed to hybridize with the genes on the microarray chip and then washed to remove all of the unbound mRNA's. The chip is then run through the optical reader which records the location and intensities of the fluorescent tags.

There are three major steps involved in microarray analysis: *experimental design*, *preprocessing*, and *data analysis*



1. Experimental Design

The first step of conducting a microarray analysis is the experimental design. This may well be the most important step since all decisions made here will drastically effect the results of all subsequent steps. It is important to have a clear understanding of exactly what the experiment is attempting to analyze from the start. If a clear plan is not made, it is likely that unanticipated technological and biological confounding factors arise and drastically alter the outcome of the experiment. Examples of such confounding factors include:

1. Hybridization variation between different transcripts
2. Unanticipated cell-cycle or developmental differences between cell lines
3. Variation in exposure levels to chemicals among cell lines
4. Amplification differences between samples

2. Preprocessing

Preprocessing is concerned with connecting the chip with final analysis. Preprocessing includes the following tasks:

1. *Data import* is involved in incorporating various file formats into a desired data object. This can be a challenge because different vendors sometimes utilize different data representations.
2. *Background adjustment* all comes down to one word – noise. The noise can be introduced from various sources such as optical distortion, non-specific hybridization, or equipment damage.
3. *Normalization* between samples needs to be established for a variety of reasons. Some of these reasons may include different reverse transcription efficiency levels or hybridization inequalities between samples. In order to properly summarize the results between samples, reasonable normalization algorithms must be applied to the data. Furthermore, normalization can accommodate for variations in spacial localities between chips (especially those that are done in-house).
4. *Summarization of data* is the process of reducing the various samples into an analysis which can infer biological properties. This is often considered the crux of microarray analysis by many. It involves applying various linear and nonlinear models to a variety of learning techniques including (but by no means limited to): support vector machines (SVM), neural network, and Empirical Bayes algorithms.

3. Quality Control

Quality control is concerned with accuracy and reproducibility.

Program Implementation of R

We consider a case study where two RNA sources are compared through a common reference RNA. The analysis of the log-ratios involves a two-sample comparison of means for each gene. The data is available as an RGList object in the saved R data file **ApoAI.RData**.

Source: *http://www.bioconductor.org/help/course-materials/2005/BioC2005/labs/lab01/Data/apoai.zip*

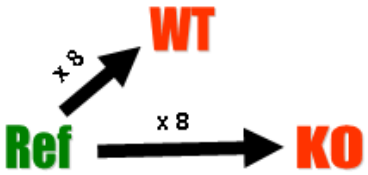
Background. The data is from a study of lipid metabolism by Callow et al (2000). The apolipoprotein AI (ApoAI) gene is known to play a pivotal role in high density lipoprotein (HDL) metabolism. Mice which have the ApoAI gene knocked out have very low HDL cholesterol levels. The purpose of this experiment is to determine how ApoAI deficiency affects the action of other genes in the liver, with the idea that this will help determine the molecular pathways through which ApoAI operates.

Hybridizations. The experiment compared 8 ApoAI knockout mice with 8 wild type (normal) C57BL/6 ("black six") mice, the control mice. For each of these 16 mice, target mRNA was obtained from liver

tissue and labeled using a Cy5 dye. The RNA from each mouse was hybridized to a separate microarray. Common reference RNA was labeled with Cy3 dye and used for all the arrays. The reference RNA was obtained by pooling RNA extracted from the 8 control mice.

Number of arrays	Red (Cy5)	Green (Cy3)
8	Wild Type "black six" mice (**WT**)	Pooled Reference (**Ref**)
8	ApoAI Knockout (**KO**)	Pooled Reference (**Ref**)

Diagrammatically, the experimental design is:



This is an example of a single comparison experiment using a common reference. The fact that the comparison is made by way of a common reference rather than directly as for the swirl experiment makes this, for each gene, a two-sample rather than a single-sample setup.

Source Code:

```
source("http://www.bioconductor.org/biocLite.R")
biocLite("limma")
setwd("C:/Users/Ashok\ Kumar/Desktop")
library(limma)
load("ApoAI.RData")
MA<-normalizeWithinArrays(RG)
design <- cbind("WT-Ref"=1, "KO-WT"=rep(0:1,c(8,8)))
fit<-lmFit(MA,design=design)
fit<-eBayes(fit)
plotMA(fit)
topTable(fit,coef="KO-WT",adjust="fdr")
isGene<-RG$genes$TYPE=="cDNA"
MA2<-MA[isGene,]
fit<-lmFit(MA,design=design)
fit<-eBayes(fit)
plotMA(fit,2)
top10<-order(fit$lods[,"KO-WT"],decreasing=TRUE)[1:10]
A<-fit$Amean
M<-fit$coef[,2]
shortlabels<-substring(fit$genes[,"NAME"],1,5)
text(A[top10],M[top10],labels=shortlabels[top10],cex=0.8,col="blue")
```

Explanation for source code:

Step 1 & 2: Installing **limma** package under R environment
Step 3: Defining our working directory (C:\Users\Ashok Kumar\Desktop**ApoAI.RData**)
Step 4: Loading **limma** package into our R environment
Step 5: Reads in the microarray data
Step 6: Normalizes the samples
Step 7: Creates the linear model we will use to run our experiment
Step 8 & 14: Fits a linear model
Step 9 & 15: Creates a best fit using an Empirical Bayes algorithm

Step 10 & 16: The MA-plot is a plot of the distribution of the red/green intensity ratio ('M') plotted by the average intensity ('A')

Step 11: This is an important table since it shows you the results of the analysis. Important to note here are the p-values (indicate which results are significant) and the M/A values. Large M values indicate large differentials and hence differential gene expression between populations.

Step 12: Remove the control probes from the data

Step 17, 18, 19, 20 & 21: These commands will assign labels to the data points in the graph that correspond to the ten most differentially expressed genes in our experiment

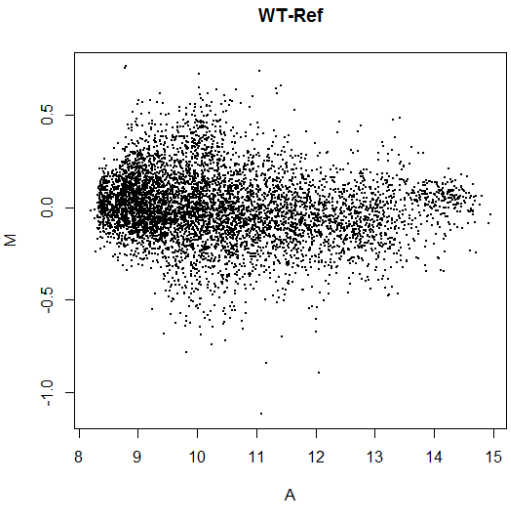


Figure generated at step: 10

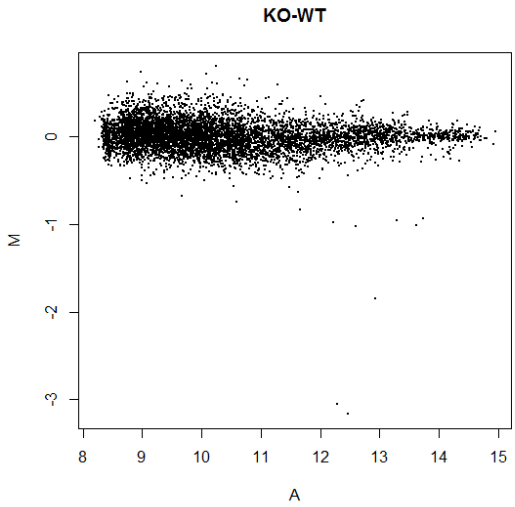


Figure generated at step: 16

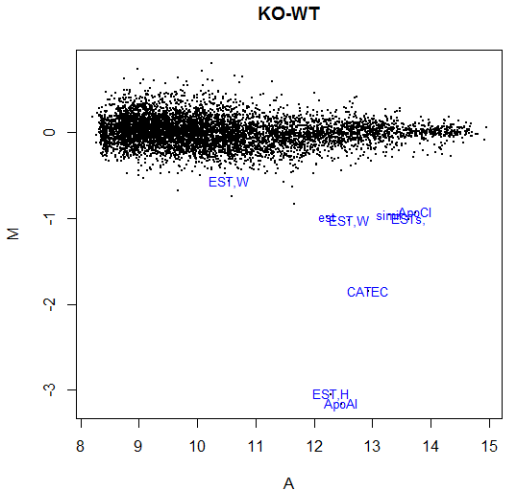


Figure generated at end of step: 21

By observing the resulting data, it is possible to derive a couple of conclusions. First, it is important to note that ApoAI is very distant from the M axis (large differential expression between the two sample groups). This serves as a sanity check since it was ApoAI that we knocked out originally. Additionally, we note that ESTH is also expressed in a largely differential manner. After evaluating the p-values, we verify that this is a significant data point.