Lab: cDNA Microarray Preprocessing

Nolwenn Le Meur

October 10^{th} 2006

This lab will guide you through the first steps of a spotted cDNA microarray analysis. These steps comprise loading data into R/Bioconductor, quality assessment of the measurements, and preprocessing of the raw data via background correction and normalization. We based the exercices on **Chapter 4** of *Bioinformatics and Computational Biology Solutions Using R and Bioconductor* by R. Gentleman, V. J. Carey, W. Huber, R. A. Irizarry and S. Dudoit.

1 Preliminaries

1.1 libraries

You will need the packages *marray*, *arrayQuality*, *vsn*, *limma*, *beta7* and *convert*.

```
> library("marray")
> library("arrayQuality")
> library("limma")
> library("vsn")
> library("beta7")
> library("convert")
```

1.2 beta7 dataset

In this case study, we make use of an experiment conducted by the Erle Lab in UC San Francisco. This experiment aims to study the cell adhesion molecule integrin alpha4/beta7 which assists in directing the migration of blood lymphocytes to the intestine and associated lymphoid tissues. The goal of the study is to identify differentially expressed genes between the alpha4/beta7+ and alpha4/beta7- memory T helper cells. The study hypothesizes that differentially expressed genes may play a role in the adhesion or migration of T cells. Further details and results of the experiments can be found in Rodriguez (2004).

The data set given here is a subset from the original dataset consisting of 6 replicated slides from different subjects. Complete information about the array platform and data from each

of the individual arrays is available from GEO (accession number GSE 1039). Each hybridization involved beta 7+ cell RNA from a single subject (labeled with one dye) and beta7- cell RNA from the same subject (labeled with the other dye). Target RNA was hybridized to microarrays containing 23,184 probes including the Operon Human version 2 set of 70-mer oligonucleotide probes and 1760 controls spots (e.g., negative, positive and normalization controls spots). Microarrays were printed using 12x4 print-tips and are thus partitioned into a 12x4 grid matrix. Each grid consists of a 21x23 spot matrix that was printed with a single print-tip.

Each of the arrays were scanned using an Axon GenePix 4000B scanner and images were processed using GenePix 5.0 image processing software. The data comprises 6 GenePix gpr output files. Each gpr file contains 23,184 rows and 56 columns; rows correspond to probes (spots) while columns correspond to different statistics from the image analysis output. The gpr files also contain probe names and IDs.

First we look for the data package and the directory that contains the raw data files. We then read the file in R. Make extensive use of the help(<object>) command to find information about particular objects.

```
> datadir <- system.file("beta7", package="beta7")
> filename <- list.files(path=datadir,pattern="\\.gpr$")
> f <- function(x) as.numeric(x$Flags > -75)
> RG <- read.maimages(file.path(datadir,filename), source="genepix", wt.fun=f)
> TargetInfo <- read.marrayInfo(file.path(datadir, "TargetBeta7.txt"))
> RG$printer <- getLayout(RG$genes)
```

2 Data quality assessment: example of *arrayQuality*

The last paragraph of Chapter 4 "Preprocessing Two-Color Arrays" by Y.H Yang and A.C. Paquet contains a case study for spotted chip preprocessing and quality control using the *arrayQuality* package. Working through exercise 1 to 8 should take you around 20 min, depending on your experience with R.

3 Background correction

This exercise is adapted from G.K Smyth's Bioconductor Short Course, Seattle Aug. 2005. Try different background correction using the *backgroundCorrect* from the *limma* package and compare the resulting the MA-plots.

```
> RGsu <- backgroundCorrect(RG, method="subtract")  # the default
> RGno <- backgroundCorrect(RG, method="none")
> RGne <- backgroundCorrect(RG, method="normexp")
```

Examine closely the MA-plots from the three background correction methods. Notice that subtracting produces a decreasing fan effect with intensity while not background correcting produces an increasing fan effect. The 'normexp' produces a more balanced stabilization of the variances. It also preserves all the data. To examine all the MA-plots efficiently, you may find it helpful to use the following commands, which write all the MA-plots to png disk files in compact format:

```
> plotMA3by2(RGsu, prefix="MAsu")
> plotMA3by2(RGno, prefix="MAno")
> plotMA3by2(RGne, prefix="MAne")
```

4 Normalization

Try different normalization methods using the *normalizeWithinArrays* and *normalizeBetweenArrays* from the *limma* package and *vsn*.

4.1 Within array normalization

limma propose various methods to normalize within array. Try then and compare the results using MA-plots. Note that by default the background is subtracted but any of the described method above can be used.

```
> MAmed <- normalizeWithinArrays(RG,method="median",bc.method="normexp")
> MAlo <- normalizeWithinArrays(RG,method="loess",bc.method="normexp")
> MApt <- normalizeWithinArrays(RG,method="printtiploess",bc.method="normexp")

> oldpar<-par()
> nf <- matrix(c(1,2,3,4), 2, 2, byrow = TRUE)
> layout(nf)

> plotMA(MAmed[,1],main="median normalization")
> abline(a=1,b=0,col="blue")
> abline(a=0,b=0,col="red")
> abline(a=-1,b=0,col="blue")

> plotMA(MAlo[,1],main="loess")
> abline(a=1,b=0,col="blue")
> abline(a=0,b=0,col="red")
> abline(a=-1,b=0,col="blue")

> plotMA(MApt[,1],main="print-tip loess")
> abline(a=1,b=0,col="blue")
> abline(a=0,b=0,col="red")
```

```
> abline(a=-1,b=0,col="blue")

> plotMA(MAvsn[,1],main="vsn")
> abline(a=1,b=0,col="blue")
> abline(a=0,b=0,col="red")
> abline(a=-1,b=0,col="blue")

> par(oldpar)
```

4.2 Separate-channel normalization

One approach is to use the quantile normalization after a normalization within arrays. This normalization (or scaling) addresses the comparability of the distributions of log intensities between array. The other is the vsn-method. Note that vsn expects unnormalized data as input.

```
> MAquantile <- normalizeBetweenArrays(MApt,method="quantile")
> MAvsn <- normalizeBetweenArrays(RGne,method="vsn")

> oldpar<-par()
> layout(matrix(c(1,2),1,2,byrow=TRUE))
> plotDensities(MAquantile)
> plotDensities(MAvsn)
> par(oldpar)
```

5 Convert

marray and *limma* propose equivalent but different classes, objects and methods to read, preprocess and analysis the data. You can convert from one to the other using the *convert*.

```
> class(MAvsn)
> MAvsn <- as(MAvsn,"marrayNorm")
> class(MAvsn)
> MAvsn <- as(MAvsn,"MAList")
> class(MAvsn)
```

You can also convert to an `expressionSet`

```
> vsnExpr <- as(MAvsn,"expressionSet")
```

Session Info
This document was generated using R version 2.4.0 alpha (2006-09-14 r39321) and BioC 1.8 on a x86_64 linux The code was tested using R2.3.1 and BioC 1.8 on i386-pc-mingw32