
Laboratory 2:0 Microarray Data Processing for cDNA arrays

Day 2: Day August 17, 2004: 08:30 – 10:00

Ben Bolstad, Biostatistics, University of California, Berkeley

Key Concepts

- Normalization for cDNA arrays

What you will be able to do at end of this section

- Perform different normalization methods to cDNA microarray data and compare the results

Introduction

This lab introduces you to some basic preprocessing tools for cDNA microarray data. We will make use of the BioConductor software package *marray*.

First steps

Start R. At the command prompt type

```
library(marray)
```

This will load the *marray* library and any of its dependencies. Next we will load the dataset we shall work on. Normally you'd read the data files which are outputted by your image analysis program, but for the purposes of this lab we shall use a dataset which has been previously read into the software. Type

```
data(swirl)
```

Inspecting the data

Next we will carry out a brief inspection of some of the data. In particular we will look at the attributes for one particular array out of the *swirl* data set. First lets examine the Red background signals. To display these as a pseudo microarray image type

```
image(swirl[,3],xvar="maRb")
```

Similarly the green background signal can be examined by typing

```
image(swirl[,3],xvar="maGb")
```

Next we look at the red and green channel intensities

```
image(swirl[,3],xvar="maRf")
```

```
image(swirl[,3],xvar="maGf")
```

Next lets define a color palette that goes from green to red. We use green and red because these are traditionally used colors and also because they correspond to the coloring of the labelings. In principle we could use any coloring we wish. Type

```
RGcol <- maPalette(low="green",high="red",k=50)
```

Next we create an pseudo-array image of the log fold-change. Type

```
image(swirl[,3],xvar="maM",col=RGcol)
```

It might be more interesting to view only the spots with the most extreme fold-changes. We can do this by typing

```
image(swirl[,3],xvar="maM",col=RGcol,subset=maTop(maM(swirl[,3]),h=.1,  
l=.1))
```

You should experiment with varying the h and l parameters to see the effect on the image.

Next create an MA-plot of the data for this array. Do this by typing

```
plot(swirl[,3])
```

What do you observe?

Next lets boxplot the log fold change values by print-tip group. Type

```
boxplot(swirl[,3])
```

What do you observe?

Normalizing the data within a slide

Now we are going to experiment with different normalization techniques and compare their effects on the data. First we will just try the median normalization which tries to give the same median to the red channel intensities and the green channel intensities. Do this by typing

```
swirl3.mednorm <- maNorm(swirl[,3],norm="median")
```

Next repeat the same images as before

```
image(swirl3.mednorm,xvar="maM",col=RGcol)
image(swirl3.mednorm,xvar="maM",col=RGcol,subset=maTop(maM(swirl[,3]),
h=.1,l=.1))
plot(swirl3.mednorm)
boxplot(swirl3.mednorm)
```

How do they compare with the unnormalized data?

Next consider a global loess normalization. Do this by typing

```
swirl3.loessnorm <-
maNorm(swirl[,3],norm="loess",subset=sample(1:8443,1000))
```

Note that we are using a subset of points to speed up processing.

Repeat the same images as before on the loess normalized data

```
image(swirl3.loessnorm,xvar="maM",col=RGcol)
image(swirl3.loessnorm,xvar="maM",col=RGcol,subset=maTop(maM(swirl[,3])
),h=.1,l=.1))
plot(swirl3.loessnorm)
boxplot(swirl3.loessnorm)
```

What do you observe? How does it compare to the unnormalized data? How about the median normalized data?

Now consider using an loess corresponding to each print tip and performing the normalization.

Type

```
swirl3.printloessnorm <- maNorm(swirl[,3],norm="printTipLoess")
```

Repeat the standard set of plots

```
image(swirl3.printloessnorm,xvar="maM",col=RGcol)
```

```
image(swirl3.printloessnorm,xvar="maM",col=RGcol,subset=maTop(maM(swirl[,3]),h=.1,l=.1))
```

```
plot(swirl3.printloessnorm)
```

```
boxplot(swirl3.printloessnorm)
```

What do you observe? How does it compare to the previous attempts at normalization?

Now consider using the print tip loess normalization followed by a scaling across print tip groups. Type

```
swirl3.scaleprintloessnorm <-  
maNorm(swirl[,3],norm="scalePrintTipMAD")
```

Repeat for the last time the standard set of plots

```
image(swirl3.scaleprintloessnorm,xvar="maM",col=RGcol)
```

```
image(swirl3.scaleprintloessnorm,xvar="maM",col=RGcol,subset=maTop(maM(swirl[,3]),h=.1,l=.1))
```

```
plot(swirl3.scaleprintloessnorm)
```

```
boxplot(swirl3.scaleprintloessnorm)
```

what do you observe and how does it compare with what you saw earlier?

Lets compare the effect of all the normalizations on to the log ratios (ie the M's) all on the same plot

```
plot(density(maM(swirl[,3])), lwd=2, col="red",main="Density plots of
log-ratios M",ylim=c(0,1.2))
lines(density(maM(swirl3.mednorm)), lwd=2, col="blue")
lines(density(maM(swirl3.loessnorm)), lwd=2, col="green")
lines(density(maM(swirl3.printloessnorm)), lwd=2, col="pink")
lines(density(maM(swirl3.scaleprintloessnorm)), lwd=2, col="orange")
legend(1, 0.9,c("None","Median","Loess","print-tip", "print-tip with
loess"),col=c("red","blue","green","pink","orange"),lwd=2)
```

What do you observe?

Normalizing multiple arrays

Next we will normalize across arrays. First lets compare the log FC across all 4 slides in this dataset. Type

```
boxplot(swirl)
```

what do you observe?

Now apply printtip loess normalization to each array

```
swirl.norm <- maNorm(swirl)
```

```
boxplot(swirl.norm)
```

what did you observe from the boxplot?

Now lets apply a between chip scaling to the printtip loess normalized data

```
swirl.norm2 <- maNormScale(swirl.norm)
```

```
boxplot(swirl.norm2)
```

what did you observe from the boxplot?

If you have time

At the command prompt type

```
swirl
```

This will show you the innards of the *marrayRaw* object that stores the raw data. Try to figure out what each of the different slots that constitute the swirl object store.

Final words

In general practice you want to be careful about which normalizations you apply and how you apply them. You should not blindly apply normalization routines without inspecting the data and having some knowledge of the experiment in general. Assumptions are important and it is possible you might make things worse rather than better by normalizing.

Appendix

1. Resources

i) Original Papers

- Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed (2002).
Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, Vol. 30, No. 4, e15.

2. Demonstration Overheads

© 2004 Canadian Genetic Diseases Network