Lab #4
Normalization and Bioconductor

In this lab, we will be working with a few data sets, each run on a different platform. The first data set is an R object generated from a 2-channel cDNA array that is called `swirl`. This data set is an experiment that was run on a zebrafish to study the early development. The data is named such because "swirl is a point mutant in the BMP2 gene that affects the dorsal/ventral body axis." The objective of the experiment was to evaluate the transcript differences between wildtype zebrafish and those with this mutation. As I mentioned above, `swirl` is an R object, so the format and structure of this binary file has to be accessed through various R functions. If you type "swirl", you will immediately see that there are attributes that make up this file (e.g. @maInfo) beyond the typical channel information. Included is metadata information that makes up the experimental parameters, in addition to the raw intensity data.

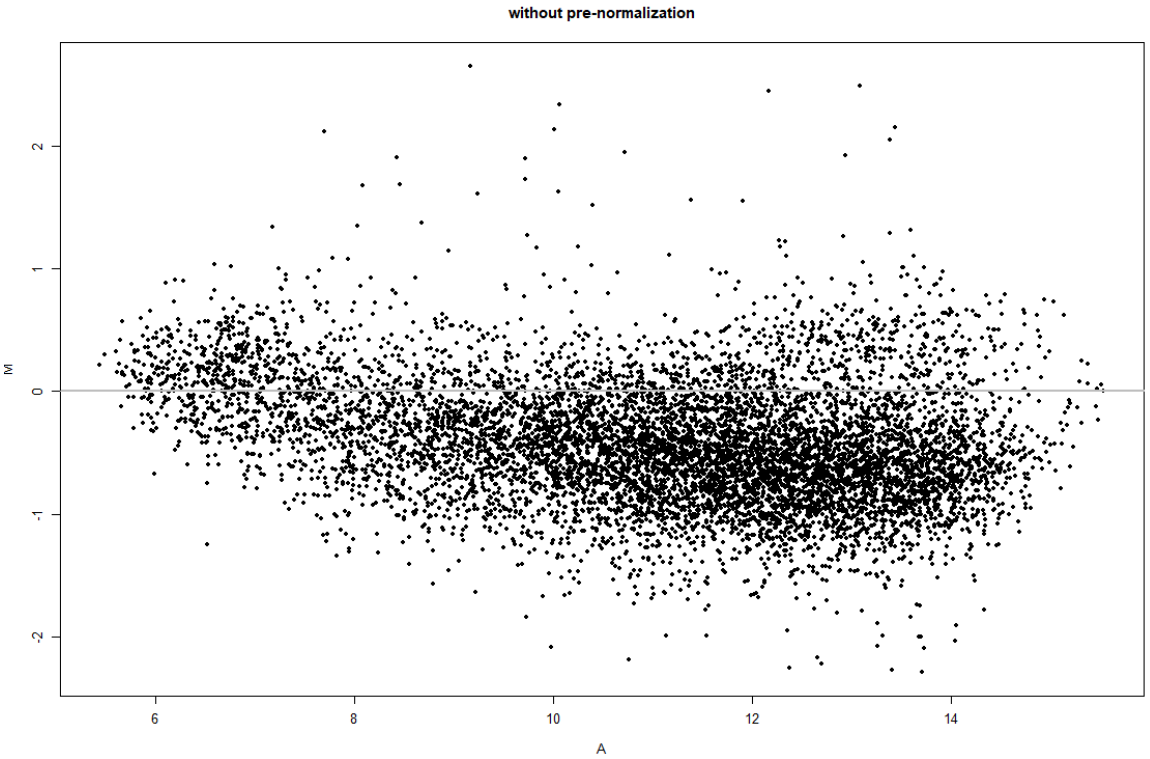
The second 2 data sets are raw intensity files – one from an Agilent platform and the other from an Affymetrix platform. Both of these files are on the course website. These are not R objects, rather the Agilent files are raw text files generated from the Agilent software and the files are raw binary files generated from the Affymetrix software.

Since both R objects and raw data files are typically what an analyst is given when asked to analyze an experiment, this lab will give you experience processing raw intensity files and normalizing them appropriately. This is typically the first step in conducting any microarray analysis, so it is important to make sure that the data is normalized appropriately before beginning any subsequent steps.

1.) Load the marray library and the swirl data set. This data set is an R metadata object, so there are multiple pieces of information (e.g., red/green background and foreground intensities, chip layout design, etc.) that are stored in this R data object.

2.) Plot an MvA plot of array 3 without any stratified lines.

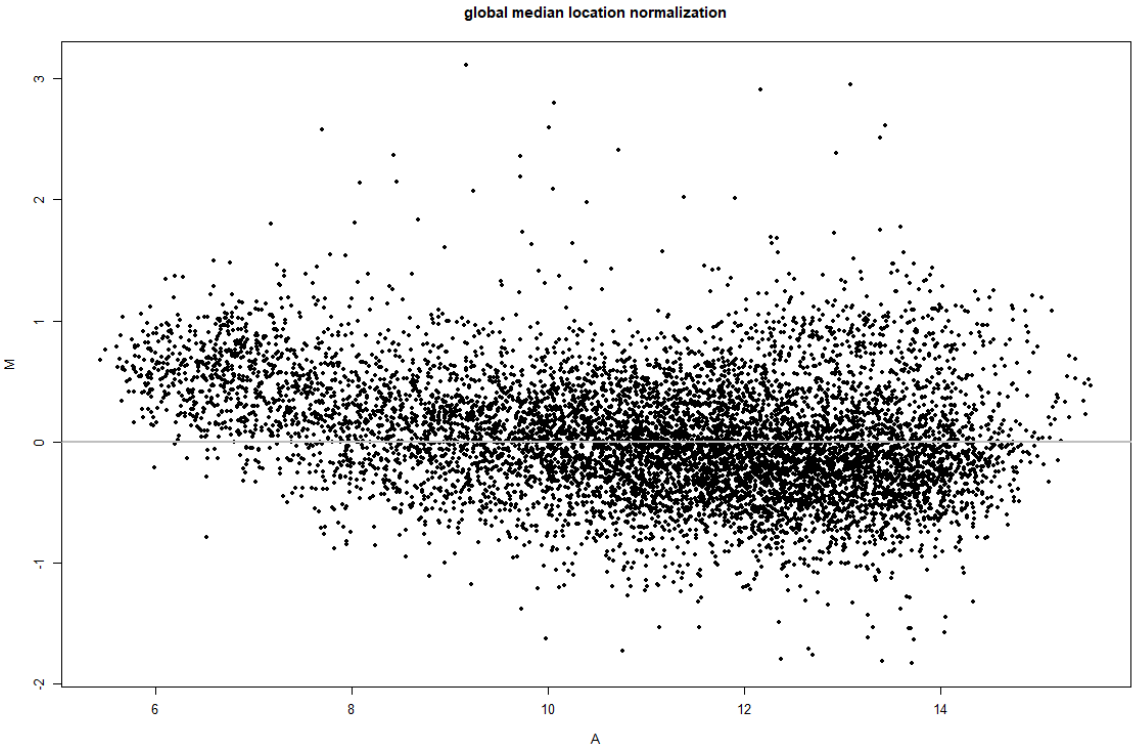
```
maPlot(swirl[,3],main='without pre-normalization',lines.func = NULL,legend.func=NULL)
```



3.) Normalize array 3 by global median location normalization.

```
mnorm<-maNorm(swirl[,3], norm="median", span=0.45)
maPlot(mnorm,main='global median location normalization',lines.func = NULL,legend.func=NULL)
```

4.) Plot an MvA plot of the normalized array without the stratified lines or legend.

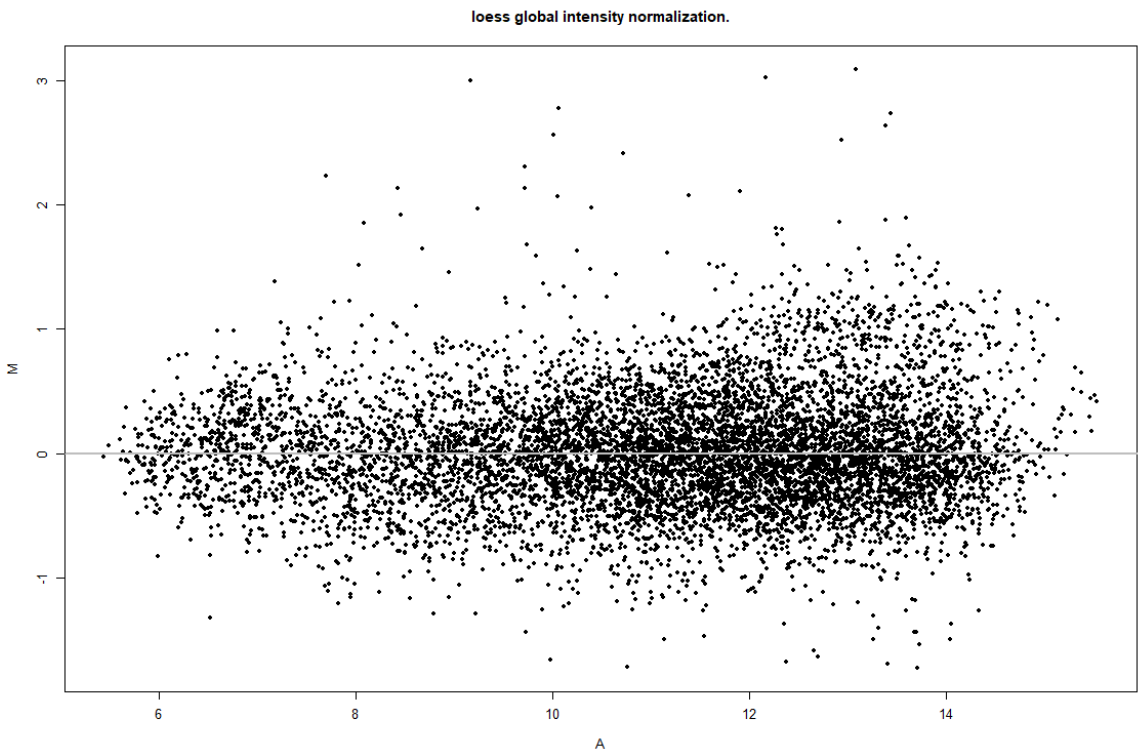


5.) What is different between the normalized data and the non-normalized data?

The M -axis range is larger in the plot of without any normalized data, which means the points in that plot actually are more loosely distributed.

6.) Repeat #3 and #4 applying loess global intensity normalization.

```
mnorm<-maNorm(swirl[,3], norm="loess", span=0.45)
maPlot(mnorm,main='loess global intensity normalization.',lines.func = NULL,legend.func=NULL)
```

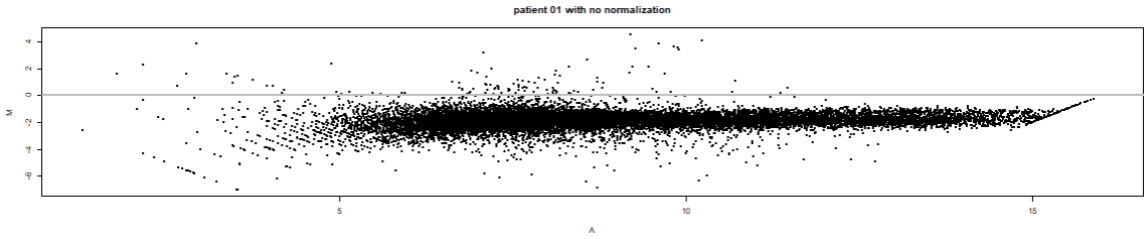


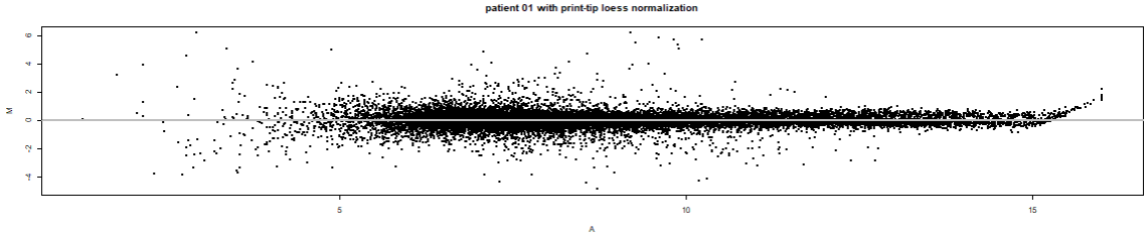
7.) Which of the two normalizations appears to be better for this array?
The losses global intensity normalization appears to be more regression to the 0-axis. So the losses global intensity normalization seems more suitable for this array

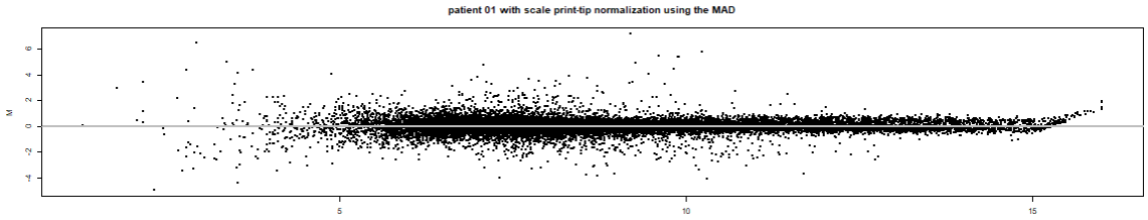
8.) Now we would like to read in raw GenePix files for 2 cDNA arrays that represent 2 patient samples. Go to the course website and retrieve the compressed file called 'GenePix files'. Open it up and put the contents in a directory. Now using the sample code below, read in the 2 array files.

```
> dir.path <- "C:\\Documents and Settings\\higgsb\\Desktop\\"
> a.cdna <- read.GenePix(path=dir.path,name.Gf = "F532 Median",name.Gb ="B532
Median", name.Rf = "F635 Median", name.Rb = "B635 Median",name.W ="Flags")
```

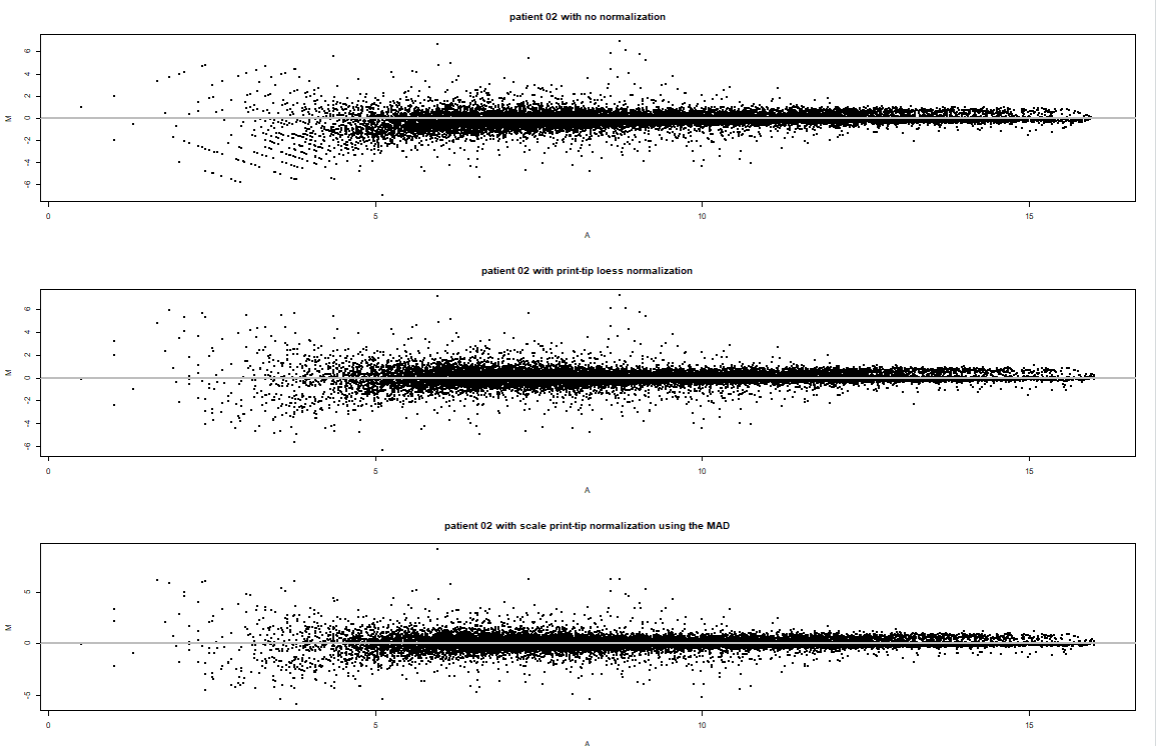
9.) Using the a.cdna object, which is analogous to the swirl metadata object, normalize both arrays and provide MvA plots for each array normalized by the following 3 methods: no normalization, print-tip loess normalization, and scale print-tip normalization using the MAD. Hint: use the par(mfrow=c(3,1)) function to put the 3 plots for a single patient array on the same page.








```
par(mfrow=c(3,1))
mnorm1<-maNorm(a.cdna[,1], norm="none", span=0.45)
maPlot(mnorm1,main="patient 01 with no normalization",lines.func = NULL,legend.func=NULL)

mnorm2<-maNorm(a.cdna[,1], norm="printTipLoess", span=0.45)
maPlot(mnorm2,main="patient 01 with print-tip loess normalization",lines.func = NULL,legend.func=NULL)

mnorm3<-maNorm(a.cdna[,1], norm="scalePrintTipMAD", span=0.45)
maPlot(mnorm3,main="patient 01 with scale print-tip normalization using the MAD",lines.func = NULL,legend.func=NULL)

par(mfrow=c(3,1))
mnorm1<-maNorm(a.cdna[,2], norm="none", span=0.45)
maPlot(mnorm1,main="patient 02 with no normalization",lines.func = NULL,legend.func=NULL)

mnorm2<-maNorm(a.cdna[,2], norm="printTipLoess", span=0.45)
maPlot(mnorm2,main="patient 02 with print-tip loess normalization",lines.func = NULL,legend.func=NULL)

mnorm3<-maNorm(a.cdna[,2], norm="scalePrintTipMAD", span=0.45)
maPlot(mnorm3,main="patient 02 with scale print-tip normalization using the MAD",lines.func = NULL,legend.func=NULL)
```



10.) Finally, we would like to create a data matrix that can be written out to a file with 19,200 rows and 2 columns (i.e. each patient array). Using the functions maM(), maGnames(), and maLabels(), figure out how to create the data matrix, get the probe IDs, and assign the probe IDs to the row names. Do this for the 2 normalized metadata objects that you created in #9 above (don't worry about the un-normalized data matrix).

```
mnorm1<-maNorm(a.cdna[,1], norm="printTipLoess", span=0.45)
mnorm2<-maNorm(a.cdna[,2], norm="printTipLoess", span=0.45)

p1 <- data.frame(maGnames(mnorm1)@maLabels, mnorm1@maM, mnorm2@maM)
```

11.) Load the following libraries: affy, limma, affydata, affyPLM, and fpc.

12.) Now we would like to read in 3 raw Affymetrix .CEL files and normalize them with 2 different algorithms. These 3 arrays represent 3 normal healthy subjects that should have similar expression profiles. They are on the course website in the compressed file called Affymetrix .CEL files. Use the following code below to read in a metadata object for the 3 arrays (dir.path should be the same as above).

```
> fns <- sort(list.celfiles(path=dir.path,full.names=TRUE))
> data.affy <- ReadAffy(filenames=fns,phenoData=NULL)
```

```
fns <- sort(list.celfiles(path=dir.path,full.names=TRUE))
data.affy <- ReadAffy(filenames=fns,phenoData=NULL)
normalize.methods(data.affy[1])
data.affy.normalized01_q <- normalize(data.affy[1],,method="quantiles")
data.affy.normalized01_l <- normalize(data.affy[1],,method="loess")
data.affy.normalized02_q <- normalize(data.affy[2],,method="quantiles")
data.affy.normalized02_l <- normalize(data.affy[2],,method="loess")
data.affy.normalized03_q <- normalize(data.affy[3],,method="quantiles")
data.affy.normalized03_l <- normalize(data.affy[4],,method="loess")
```

13.) Using the function: expresso in addition to exprs(), create the normalized data matrices with 54,675 rows and 3 columns for the 2 different normalization algorithms. Be sure to use normalize.method="quantiles", summary.method="medianpolish", and for RMA: pmcorrect.method="pmonly"
MAS: pmcorrect.method="mas"

```
> exprs.norn.rma <- expresso(data.affy,bgcorrect.method = "rma",normalize.method="quantiles", summary.method="medianpolish", pmcorrect.method="pmonly")
background correction: rma
normalization: quantiles
PM/MM correction : pmonly
expression values: medianpolish
background correcting...done.
normalizing...done.
54675 ids to be processed
|                    |
|####################|
> exprs.norn.mas <- expresso(data.affy,bgcorrect.method = "mas",normalize.method="quantiles", summary.method="medianpolish", pmcorrect.method="mas")
background correction: mas
normalization: quantiles
PM/MM correction : mas
expression values: medianpolish
background correcting...done.
normalizing...done.
54675 ids to be processed
|                    |
|####################|
```

14.) Now use the cor() function to calculate the correlation between the 3 arrays for both normalized data matrices. Since these 3 subjects are all healthy normal individuals, we would expect to see somewhat good correlation structure between them all when looking across all genes on the array. Which normalization method has a higher overall correlation structure for these 3 normal healthy subjects? Show how you arrived at this answer.

```
> cor(as.matrix(exprs.norn.mas[,1]),as.matrix(exprs.norn.mas[,2])
             normal2.CEL
normal1.CEL   0.8959518
> cor(as.matrix(exprs.norn.mas[,1]),as.matrix(exprs.norn.mas[,3])
             normal3.CEL
normal1.CEL   0.9003087
> cor(as.matrix(exprs.norn.mas[,2]),as.matrix(exprs.norn.mas[,3])
             normal3.CEL
normal2.CEL   0.9486851
> cor(as.matrix(exprs.norn.rma[,1]),as.matrix(exprs.norn.mas[,2])
             normal2.CEL
normal1.CEL    0.865617
> cor(as.matrix(exprs.norn.rma[,1]),as.matrix(exprs.norn.mas[,3])
             normal3.CEL
normal1.CEL   0.8724149
> cor(as.matrix(exprs.norn.rma[,2 ]),as.matrix(exprs.norn.mas[,3]
             normal3.CEL
normal2.CEL   0.8924374
```

For each comparison between the 3 arrays, the PCCs of method MAS are more closer to +1 than method RMA. So MAS has a higher overall correlation structure among the 3 subjects