Homework #1

Suruchi Ahuja, Abbas Rizvi, Hoi Lam Tai, Jingchen Zhang

February 15, 2016

Problem 1

Dr. Gaile provided R code to explore 'potentially interesting genes'. Important portions of Dr. Gaile's code will be pasted in R chunks throughout this submission.

The dataset gse19439.RData from Lecture 04 (2016) was loaded.

```
load("/Users/aarizvi/Dropbox/Group2/HW1/abbas_hw1/gse19439.RData")
gse19439
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 48791 features, 42 samples
##   element names: exprs
## protocolData: none
## phenoData
##   sampleNames: GSM484448 GSM484449 ... GSM484489 (42 total)
##   varLabels: title geo_accession ... data_row_count (46 total)
##   varMetadata: labelDescription
## featureData
##   featureNames: ILMN_1343291 ILMN_1343295 ... ILMN_2416019 (48791
##     total)
##   fvarLabels: ID nuID ... GB_ACC (30 total)
##   fvarMetadata: Column Description labelDescription
## experimentData: use 'experimentData(object)'
## Annotation: GPL6947
```

gse19439 is an ExpressionSet object. An ExpressionSet contains gene expression data from a microarray experiment. The ExpressionSet also contains 'meta-data', which can be accessed using functions such as phenoData, which describes the samples in the experiment, or featureData, which contains probe IDs and other descriptive information. For more information on ExpressionSet objects please see the Bioconductor vignette ExpressionSetIntroduction.

To access the expression dataset that is contained within the ExpressionSet, the function exprs() can be utilized. We assigned X as our expression data.

```
X <- exprs(gse19439)
```

A Kruskal Wallis scan was conducted in order to test whether the samples originated from the same distribution.

```
myKrusk <- function(i){
        cat(i,"...",fill=F)
        kruskal.test(x=X[i,], g=FTB)$p.value
}
```

The p-values that were calculated from the Kruskal-Wallis test were assigned as the numeric vector `myPvals`. The order of `myPvals` was subsequently rearranged in ascending order and the row position index was subsetted into `best4`.

```
load("/Users/aarizvi/Dropbox/Group2/HW1/abbas_hw1/myPvals.RData")
GroupLabels <- c("Group I","Group II","Group III","Group IV")
# pick the best 4 p-values and assign them to the students.
best4 <- order(myPvals)[1:4]
best4
```

```
## [1]  6874 10685 26058 47526
```

Dr. Gaile assigned the order ranking (1 through 4) to STA 525 groups that were assigned the same value (e.g. `best4[2]` would be assigned to Group 2.) Since we are group 2, our row assignment was determined to be `10685` (as shown above). We then subsetted the `gse19439` `ExpressionSet` to just contain information from our row using the variable `myrow`.

```
myrow <- gse19439[10685,]
```

Our row corresponds to the probe ID ILMN_1703335.

```
featureNames(myrow)
```

```
## [1] "ILMN_1703335"
```

We accessed `featureData` of `myrow` to obtain additional information regarding the gene. The `featureData` revealed that the probe ID ILMN_1703335 corresponds to the ‘LACTB’ GeneSymbol. LACTB is an enzyme known as serine beta-lactamase-like protein. More information on LACTB is summarized in Table 1.

```
gene.info <- pData(featureData(myrow))
gene.info <- t(gene.info)
library(xtable)
tab <- xtable(gene.info, caption="Group 2 Gene of Interest Summary")
print.xtable(tab,
             size = "footnotesize",
             include.rownames=TRUE,
             include.colnames=FALSE,
             comment=FALSE)
```

Problem 2

The experimental conditions of the samples in `gse19439` belong to one of 3 different tuberculosis phenotype groups. The groups are: 1. control (CON), 2. latent TB (LTB), and 3. positive TB (PTB). The actual group that individual samples correspond to was stored in `phenoData` of `gse19439`. The groupings were accessed and trimmed into 3 letter abbreviations using `substring` for readability purposes. The groupings were assigned to the variable `FTB` which is of the class `factor`. This is very useful because it allows us to use `FTB` as an index when subsetting a `data.frame`.

ID	ILMN_1703335
nuID	iMeiqS6uUsu15619eA
Species	Homo sapiens
Source	RefSeq
Search_Key	ILMN_24565
Transcript	ILMN_24565
ILMN_Gene	LACTB
Source_Reference_ID	NM_032857.2
RefSeq_ID	NM_032857.2
Unigene_ID	
Entrez_Gene_ID	114294
GI	26051230
Accession	NM_032857.2
Symbol	LACTB
Protein_Product	NP_116246.2
Array_Address_Id	1570669
Probe_Type	I
Probe_Start	1493
SEQUENCE	ATACTGGAGGGGCAGTGGGTGCCAGTAGTGTCCTGCTGGTCCTTCCTGAA
Chromosome	15
Probe_Chr_Orientation	+
Probe_Coordinates	40256913-40256962
Cytoband	15q22.2b
Definition	Homo sapiens lactamase, beta (LACTB), nuclear gene encoding mitochondrial protein, transcript variant 1, mRNA.
Ontology_Component	membrane [goid 16020] [evidence IEA]; integral to membrane [goid 16021] [evidence IEA]
Ontology_Process	beta-lactam antibiotic catabolism [goid 30655] [evidence IEA]; response to antibiotic [goid 46677] [evidence IEA]
Ontology_Function	beta-lactamase activity [goid 8800] [evidence IEA]; hydrolase activity [goid 16787] [evidence IEA]
Synonyms	FLJ14902; G24; MRPL56
Obsolete_Probe_Id	FLJ14902; G24; MRPL56
GB_ACC	NM_032857.2

Table 1: Group 2 Gene of Interest Summary

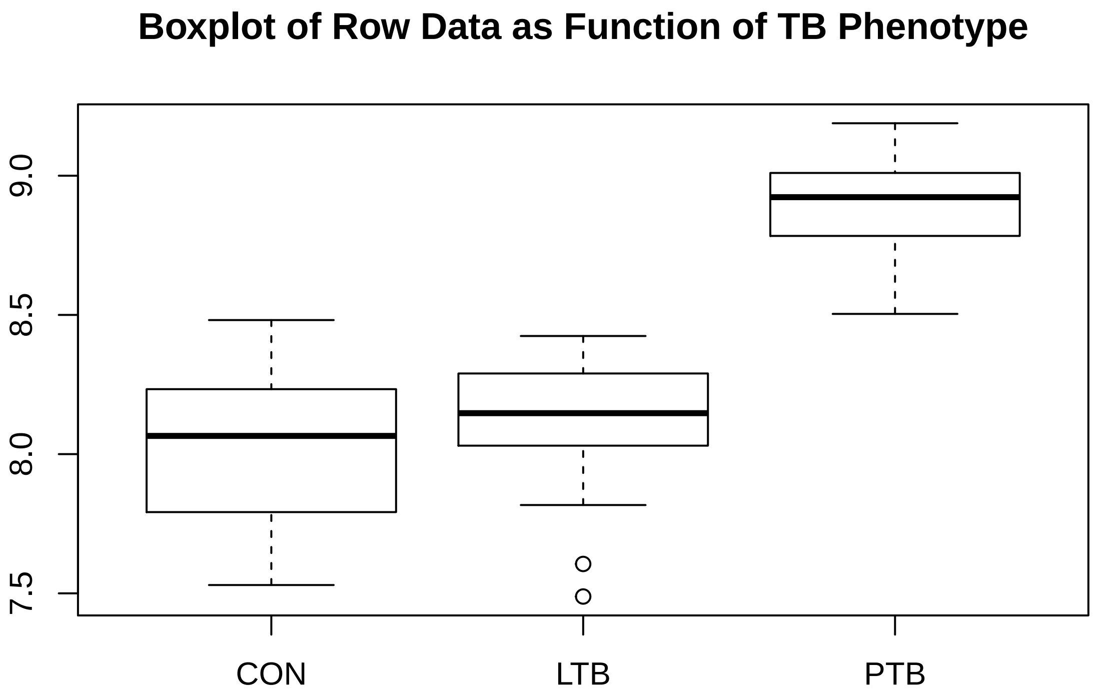
```
# make a factor object for Control, latent TB and Positive TB
tmp <- as.character(pData(phenoData(gse19439))[,1])
J <- length(tmp) # J=number of samples
TBgroup <- rep("",J)
for(j in 1:J) TBgroup[j]=substring(tmp[j],1,3)
# make a factor for TBgroup
FTB <- factor(TBgroup,levels=c("CON","LTB","PTB"))
```

2.1 Evaluating LACTB Gene Expression Distribution per Phenotype Grouping

A boxplot, violin plot, and bean plot were produced in order to visualize the distribution of gene expression for LACTB. Each of the plots show the same general trend in regards to the groups. The mean of expression values for the LACTB gene is higher in PTB than LTB, and LTB than CON. The positive TB has the highest mean of expression values for LACTB, and the control group has the highest variances, which could have many different biological implications. In order to investigate these implications, a much more thorough study must be conducted with this gene and tuberculosis. An example biological implication could be that increased LACTB expression is a result from a positive TB infection and may potentially be a useful biomarker to differentiate from individuals whom possess the active disease from TB latent and TB negative individuals.

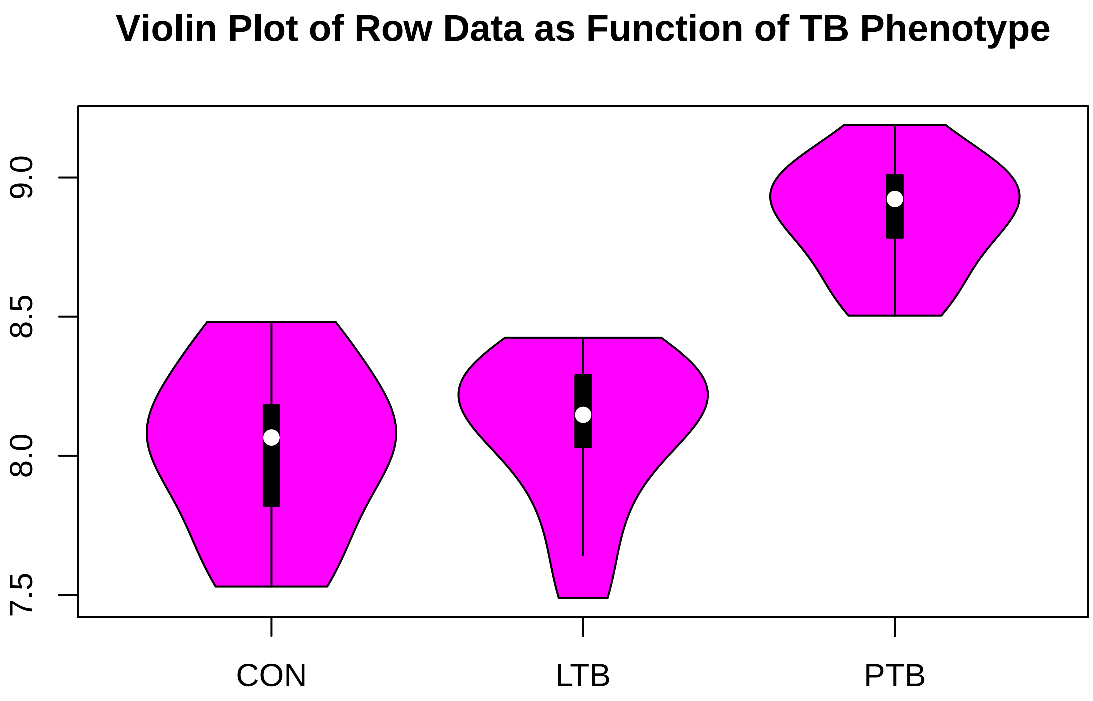
2.1.1 Boxplot

```
boxplot(log2(X[10685,sampleNames(myrow)[FTB == 'CON']]),
        log2(X[10685,sampleNames(myrow)[FTB == 'LTB']]),
        log2(X[10685,sampleNames(myrow)[FTB == 'PTB']]),
        main = "Boxplot of Row Data as Function of TB Phenotype",
        names = c("CON", "LTB", "PTB"))
```



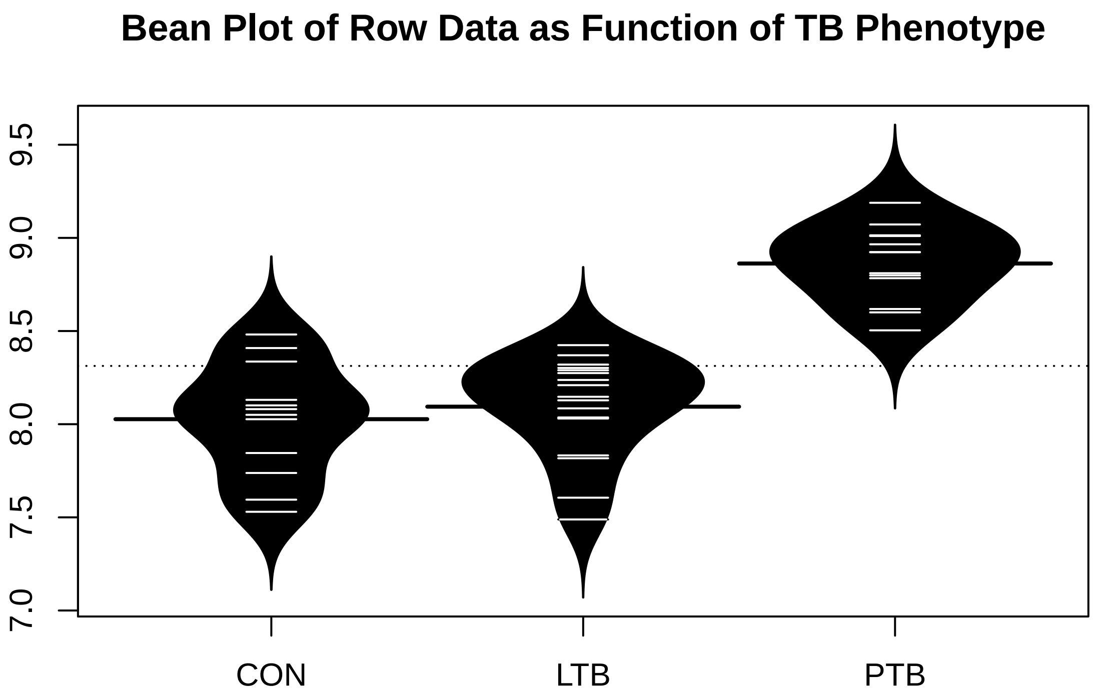
2.1.2 Violin Plot

```
require(vioplot)
vioplot(log2(X[10685,sampleNames(myrow)[FTB == 'CON']]),
        log2(X[10685,sampleNames(myrow)[FTB == 'LTB']]),
        log2(X[10685,sampleNames(myrow)[FTB == 'PTB']]),
        names = c("CON", "LTB", "PTB"))
title("Violin Plot of Row Data as Function of TB Phenotype")
```



2.1.3 Bean Plot

```
require(beanplot)
beanplot(log2(X[10685,sampleNames(myrow)[FTB == 'CON']]),
        log2(X[10685,sampleNames(myrow)[FTB == 'LTB']]),
        log2(X[10685,sampleNames(myrow)[FTB == 'PTB']]),
        names = c("CON", "LTB", "PTB"))
title("Bean Plot of Row Data as Function of TB Phenotype")
```

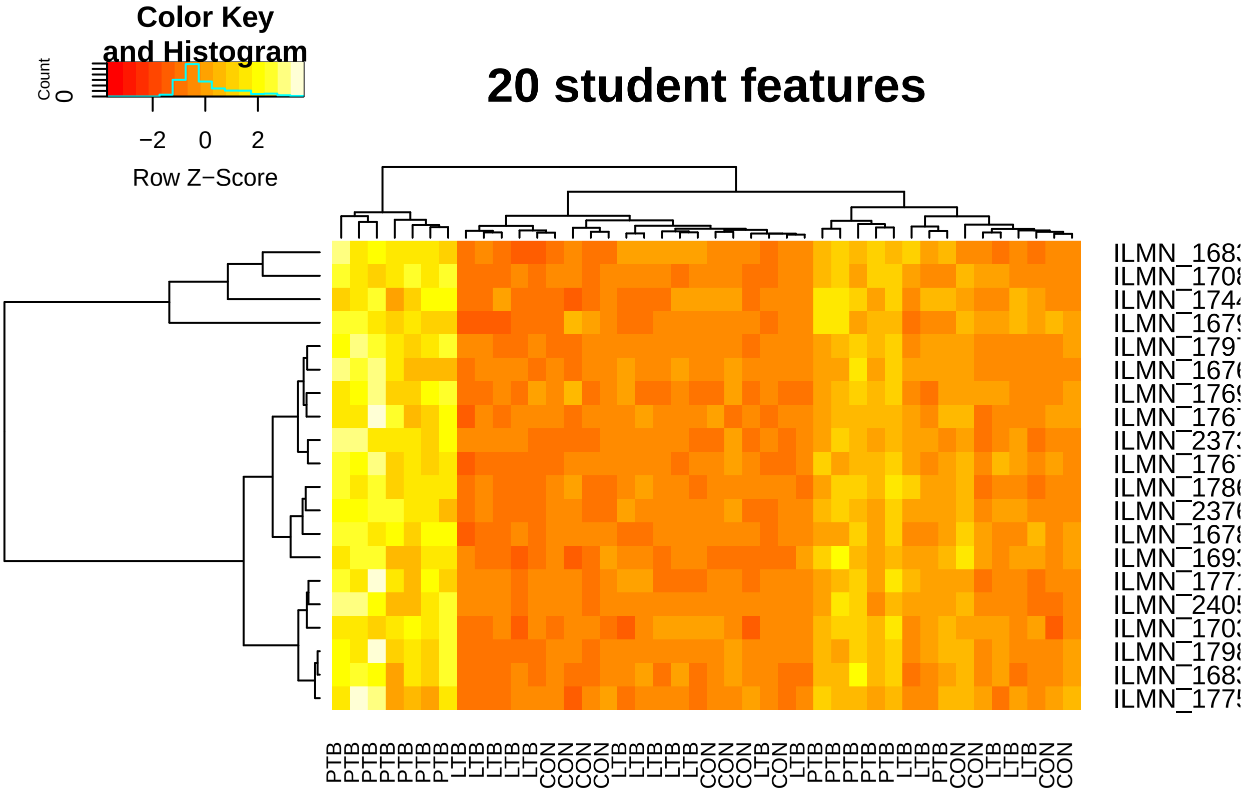


Problem 3

3.1 Heatmaps

A heatmap is a false color image with a dendrogram added to the left side and to the top. The best 20 student features were clustered and plotted in a heatmap using `heatmap.2`. The clustering method worked reasonably well such that members of the same group were closer in distance than members of other groups (see column labels of heatmap).

```
require(gplots)
best20 <- order(myPvals)[1:20]
x.best <- X[best20,]
heatmap.2(X[best20,], main="20 student features",
          scale = "row", trace="none", labCol = FTB)
```



As shown in the heatmap, the first 7 samples and the samples from 28 to 32 with label "PTB" are quite brighter than the others. This shows that the "PTB" group seems has a quite different distribution with the other two groups. The first 20 genes have higher expression frequency in "PTB" people than in other groups. However, the differences between group "CON" and "LTB" can not be that siginificant.

Problem 4

The Affymetrix portion of the Platinum Spike dataset was considered. The `AffyBatch` object was stored as the variable `affydata`. The Affy Probe Spike Values were loaded into the data frame `spikeDF`. A comparative analysis of different ways for background correction, probe normalization, PM correction, summarization methods was conducted. We wanted to compare the effectiveness of eight different analysis routes. The final endpoint of our analysis was to generate ROC curves to be able to discriminate between the different analyses. The different routes are summarized in Table 2. The following subsections will describe our analysis workflow.

4.1 Pre-processing

We first needed to pre-process the data, removing any NAs or 0s that are within the dataset.

```r
load("/Users/aarizvi/Dropbox/Group2/HW1/abbas_hw1/PSpikeAffyBatch.RData")
spikeDF <- read.table(file="/Users/aarizvi/Dropbox/Group2/HW1/abbas_hw1/AffyProbeSpikeValues.csv",sep="
levels(spikeDF[,2])
```

```
##  [1] "0"                 "0.25"              "0.285714285714286"
##  [4] "0.4"               "0.666666666666667" "0.833333333333333"
##  [7] "1"                 "1.5"               "1.7"
## [10] "2"                 "3"                 "3.5"
## [13] "MC"                "MF"
```

```r
summary(spikeDF[,2])
```

```
##                 0              0.25 0.285714285714286               0.4
##             13337               192               174               163
## 0.666666666666667 0.833333333333333                 1               1.5
##               189               166              3426               167
##               1.7                 2                 3               3.5
##               166               184                98               445
##                MC                MF
##               231                14
```

```r
#grab Spike fold changes for all entries as numeric ...
#...so anything that was not a number is now an NA
SpikeFC <- as.numeric(levels(spikeDF[,2])[spikeDF[,2]])
```

```
## Warning: NAs introduced by coercion
```

```r
#grab IDs
names(SpikeFC) <- spikeDF$V1

#remove NAs
nonZeroDX <- which(!is.na(SpikeFC) & (SpikeFC != 0))
spikeFC.clean <- SpikeFC[nonZeroDX]

#check how many genes are left in dataset
length(spikeFC.clean)
```

```
## [1] 5370
```

After the pre-processing, the dataset went from 18952 observations to 5370 observations.

4.2 Normalization of 8 routes with expresso() and subset into ExpressionSet

The algorithms that were chosen were based off of their popularity and usage in the Halfon Spike Data. The algorithm of choice per method was added manually to a vector. A for loop was written to assign algorithms of choice in a vector and input them as arguments to the `expresso()` function from the Bioconductor `affy` package in an automated manner.

```
bgcorrect.mtd <- c("rma", "mas", "rma", "mas",
                   "rma", "none", "mas", "mas")
normalized.mtd <- c("constant", "quantiles", "quantiles", "loess",
                    "loess", "constant", "qspline", "qspline")
pmcorrect.mtd <- c("pmonly", "pmonly", "subtractmm", "mas",
                   "mas", "pmonly", "subtractmm", "mas")
summary.mtd <- c("mas", "mas", "avgdiff", "medianpolish",
                 "medianpolish", "avgdiff", "mas", "mas")
route.df <- cbind(bgcorrect.mtd, normalized.mtd, pmcorrect.mtd, summary.mtd)
rownames(route.df) <- paste("Route", 1:8, sep = " ")
route.table <- xtable(route.df, caption = "Routes chosen for analysis")
print.xtable(route.table, size= "footnotesize", comment=FALSE)
```

	bgcorrect.mtd	normalized.mtd	pmcorrect.mtd	summary.mtd
Route 1	rma	constant	pmonly	mas
Route 2	mas	quantiles	pmonly	mas
Route 3	rma	quantiles	subtractmm	avgdiff
Route 4	mas	loess	mas	medianpolish
Route 5	rma	loess	mas	medianpolish
Route 6	none	constant	pmonly	avgdiff
Route 7	mas	qspline	subtractmm	mas
Route 8	mas	qspline	mas	mas

Table 2: Routes chosen for analysis

```
require(affy)
route.expsets <- list()
for (i in 1:length(bgcorrect.mtd)){
        routes <- expresso(affydata,
                           bgcorrect.method = bgcorrect.mtd[i],
                           normalize.method = normalized.mtd[i],
                           pmcorrect.method = pmcorrect.mtd[i],
                           summary.method = summary.mtd[i])
        route.expsets[[i]] <- exprs(routes)[nonZeroDX,]
}
```

4.3 Multiple hypothesis testing

`mt.maxT` is a function used from the `multtest` library. This test is done to reduce the Type 1 and Type 2 errors. It computes permutation adjusted p-values for step-down maxP and minP multiple testing procedures.

```
#create labels for multitesting class labels -- 18 samples, 9 control, 9 experimental
labels <- factor(c(rep(0, 9), rep(1, 9))) #0 is control, 1 is experimental

require(multtest)
stats <- list()
```

```
for (i in 1:length(route.expsets)){
        #conduct multiple t test for 10000 permutations
        testing.routes <- mt.maxT(route.expsets[[i]],
                                  classlabel = labels, B = 10000)
        stats[[i]] <- testing.routes$adjp[order(testing.routes$index)]
}

nrow <- nrow(route.expsets[[1]])
myresponse <- rep(NA, nrow)
#if the spike value is 1 ... assign as 1s in matrix ...
myresponse[which(spikeFC.clean == 1)] = 1
#if spike value is not 1 ... assign as 0s in the matrix
myresponse[which(spikeFC.clean != 1)] = 0
```

4.4 ROC Curves

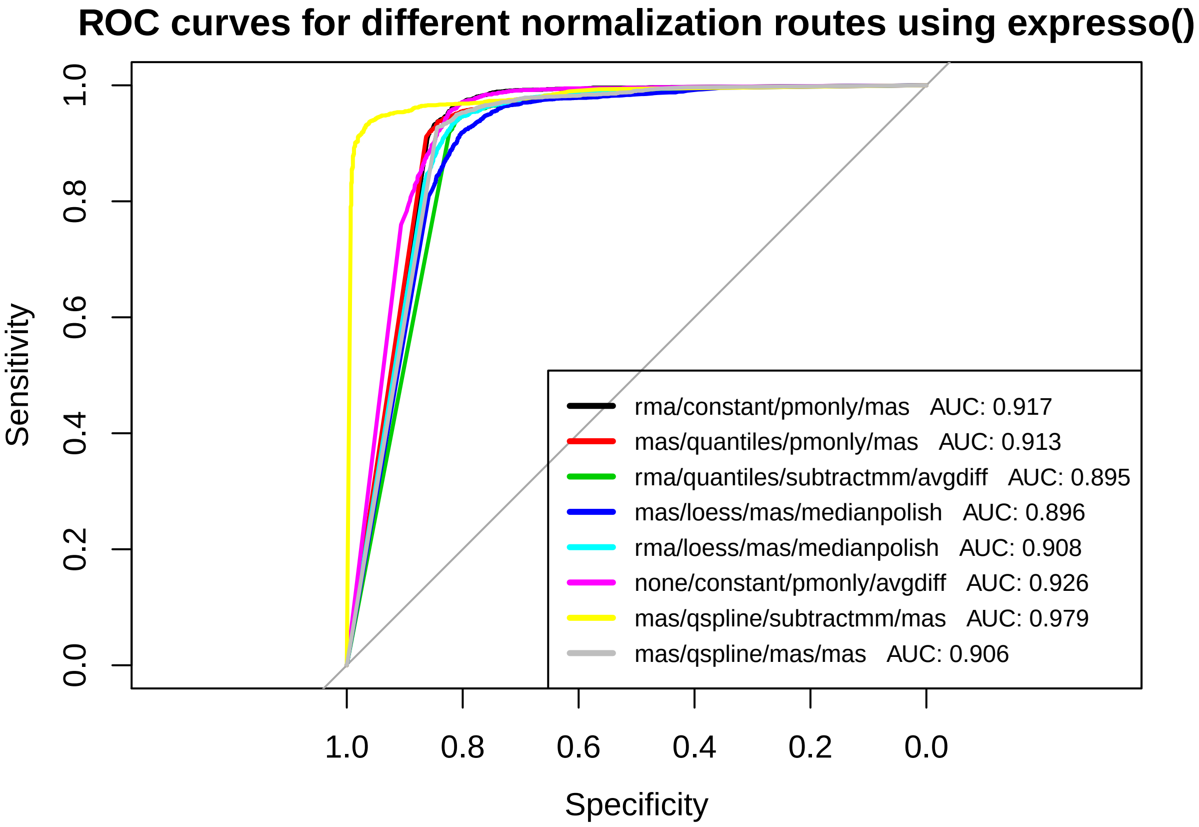
Our ROC (receiver operating characteristic) curves are created by plotting the sensitivity against the specificity. So a plot with both higher sensitivity and higher specificity will be better. In our plot, the more a curve closes to the point in the upper left corner, the better the classifier will be. So among the 8 different methods we chose, the method with “mas/qspline/subtractmm/mas” (which AUC is 0.979) can be regarded as the best one.

```
#apply roc function on input
roc.fnct <- function(x){roc(response = myresponse,
                            predictor = abs(x), plot=TRUE, print.auc=TRUE)}
#apply roc function on all the ordered test statistics in the list stats
roc <- lapply(stats, roc.fnct)
```

```
load("/Users/aarizvi/Dropbox/Group2/HW1/abbas_hw1/question4.RData")
rainbow <-  palette(rainbow(length(route.expsets)))
plot(roc[[1]], main = "ROC curves for different normalization routes using expresso()",
     col=rainbow[1])
```

```
##
## Call:
## roc.default(response = myresponse, predictor = abs(x), plot = TRUE,     print.auc = TRUE)
##
## Data: abs(x) in 1944 controls (myresponse 0) < 3426 cases (myresponse 1).
## Area under the curve: 0.9172
```

```
legendText <- c()
for(i in 1:length(route.expsets)){
        plot(roc[[i]], add=TRUE, col=rainbow[i])
        legendText[i] <- paste(bgcorrect.mtd[i],
                               "/",normalized.mtd[i],"/",
                               pmcorrect.mtd[i],
                               "/",summary.mtd[i],
                               "   AUC: ",
                               round(as.numeric(roc[[i]]$auc),3), sep="")
}
legend("bottomright",
       legendText,
       lty=c(rep(1,length(route.expsets))),
       lwd=c(rep(3,length(route.expsets))),
       col=rainbow,
       pt.cex=1,
       cex=0.75)
```



Attribution of Work

Suruchi Ahuja - Involved with producing heatmaps and tables in R. Wrote section 4.3.

Abbas Rizvi - Wrote R scripts and sections 1-4 of report.

Hoi Lam Tai - Involved with using `xtable` and learned from others R code

Jingchen Zhang - Wrote R scripts and sections 1-4 report.