Homework Assignment # 1

Group 4: Ziheng Cheng, Ziqiang Chen, Krithika Krishnan
STA 525: Statistics For Bioinformatics

February 19, 2016

Problem 1

Details about the feature corresponding to row, 47526 :

assayData: 1 features, 42 samples.
phenoData:
sampleNames: GSM484448 GSM484449 ... GSM484489 (42 total).
elementNames: exprs.
featureData:
featureNames: ILMN 2373062.
fvarLabels: ID nuID ... GB ACC (30 total).

It corresponds to the gene, RHBDF2. RHBDF2 (Rhomboid 5 Homolog 2 (Drosophila)) is a Protein Coding gene.It is known to be a TNF ALPHA signalling.

TNF-alpha is an essential component of the innate defence mechanism of the host against pathogenic challenge. Unfortunately, it can also play a major role in the pathology of certain diseases, such as tuberculosis.

The gene, RHBDF2 shows a gene-phenotype relationship - Tylosis with esophageal cancer.

Both tylotic and sporadic squamous esophageal tumor cells show strongly cytoplasmic localization of RHBDF2 compared to control esophageal cells with esophagitis. Blaydon et al. (2012) suggested that the altered RHBDF2 represents a gain-of-function allele that results in sustained EGFR signaling within cells, which in turn leads to a hyperproliferative phenotype.

Problem 2

Assigned row data as a function of the factor for CON, LTB, PTB.

1. Boxplot
2. Violin plot

Problem 3

We created a heatmap of the array data for the best 20 student features using the actual feature values feature values. That graph didn't indicate a defined discrimination between the three samples due to the difference in scaling (0 to 6000) of the sample data. By the use of ranking applied to the array data across all genes, we obtained a second heat map that had a scaling from 1 to 42. This figure (second heat map) suggests about the ability for features to be able to discriminate the CON, LTB and PTB samples. The densely packed yellow regions towards the right of the second figure indicates a clear distinction of the Positive Tuberculosis (PTB) from the other two samples, which is Latent Tuberculosis and the Control group. This trend was observed even in box and violin plot. So this result is consistent.

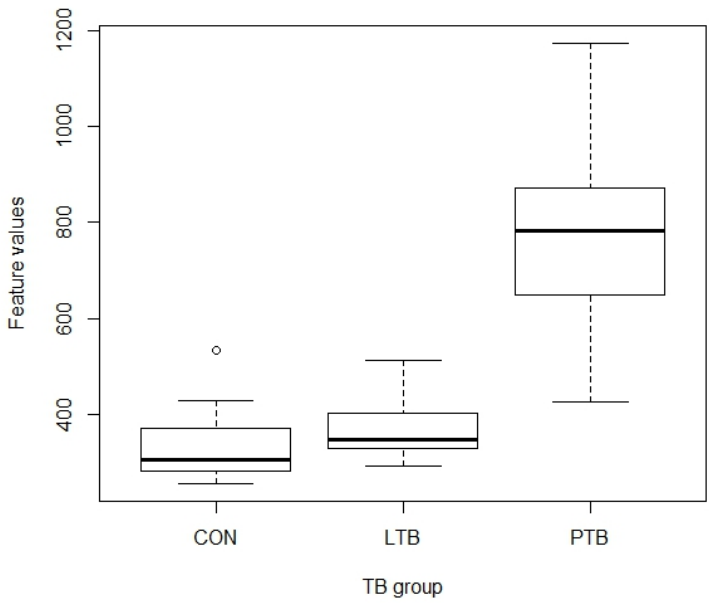

Figure 1: Boxplot

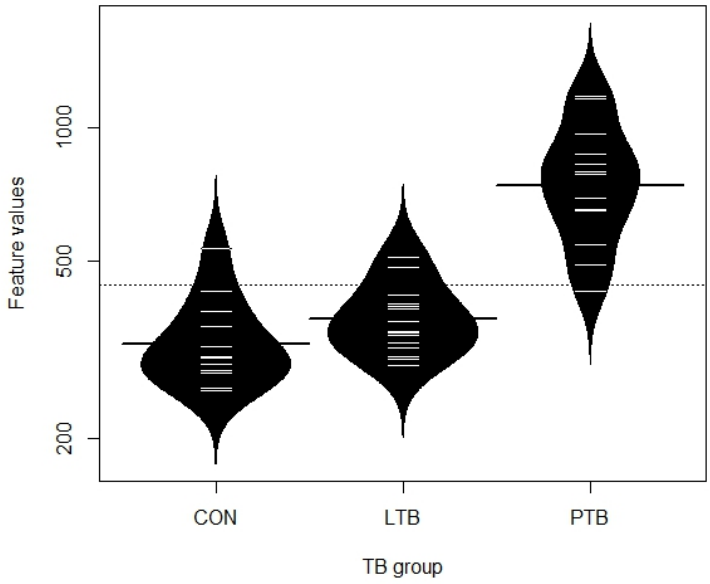

Figure 2: Violinplot

Problem 4

We used 8 different routes for the analysis in comparing the Platinum Spike Data The table indicates a summary of the Area Under the Curve values. The ROC curves for comparing the abilities of the different analysis pathways was plotted to discriminate probesets that were differentially spiked-in from those that were non-differentially spiked-in. mas for Background correction, invariant set for Probe normalization, pmonly for PM correction and medianpolish for Summarization, and no Probe set Normalization provided the best AUC.

The third route (the green line) seemed to perform better than the other routes in terms of AUC value (0.919) when compared with other 7 routes, but no an clear superiority over the others.

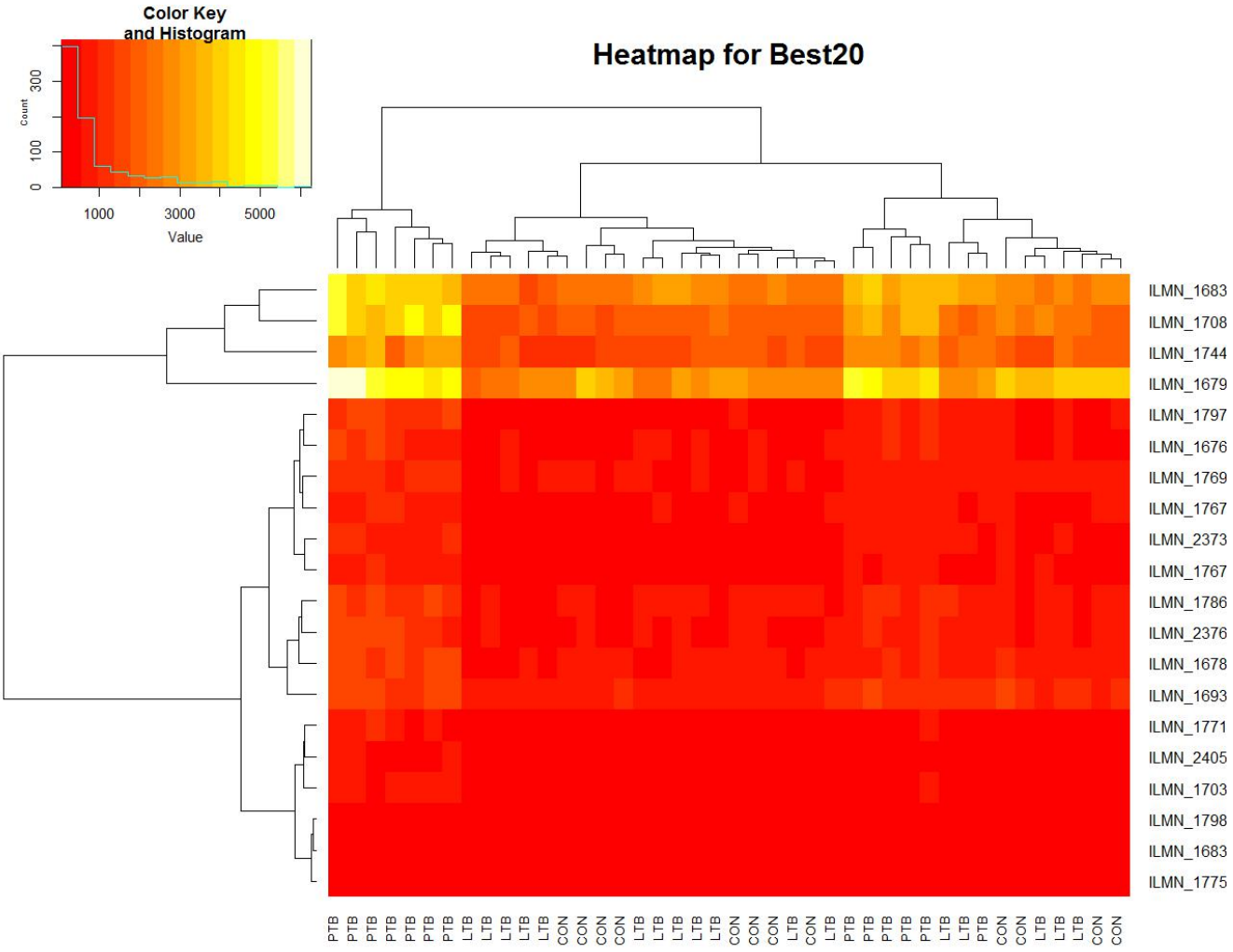

Figure 3: Heat map for best 20

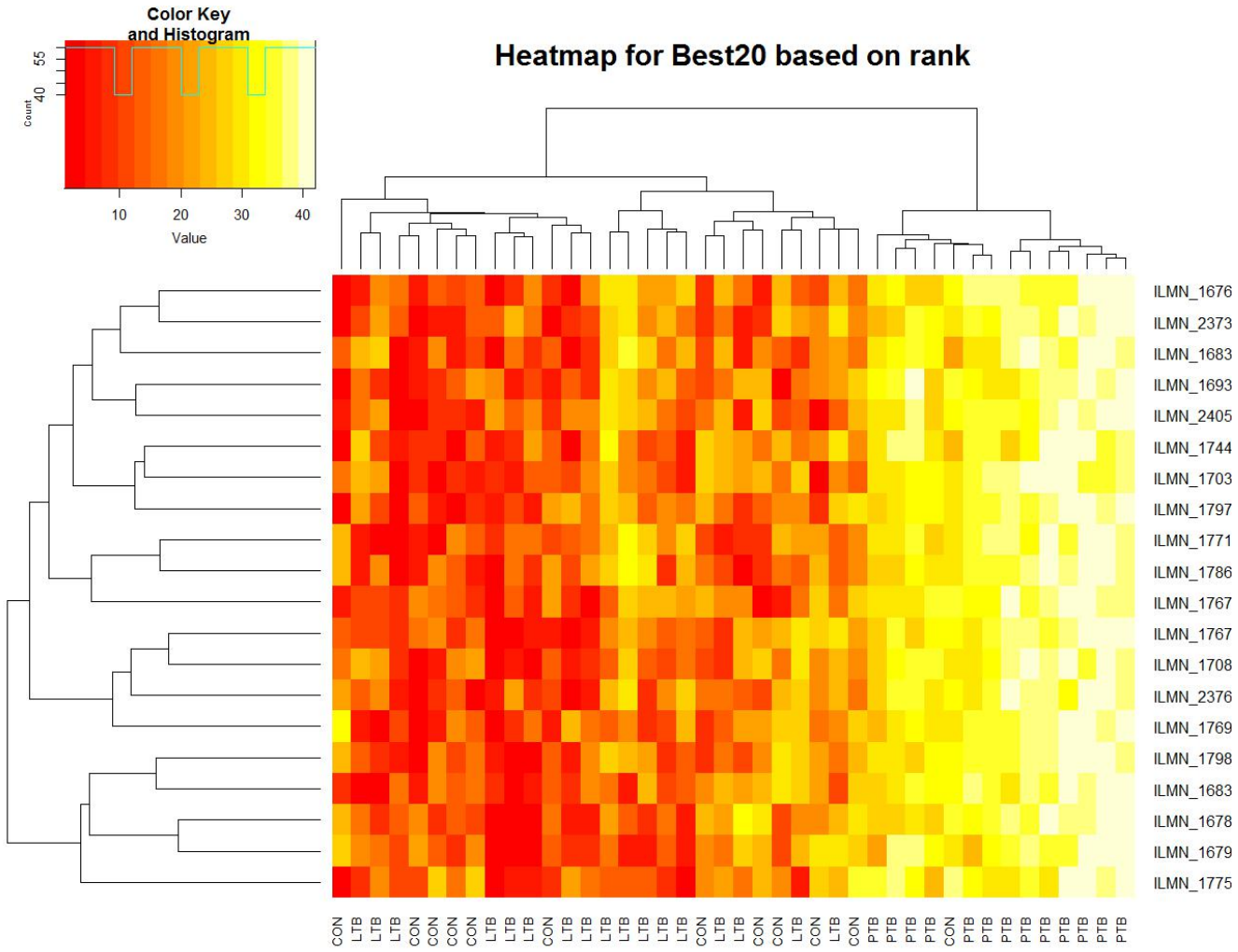

Figure 4: Heat map for best 20 based on rank

ANALYSIS	ROUTE 1	ROUTE 2	ROUTE 3	ROUTE 4	ROUTE 5	ROUTE 6	ROUTE 7	ROUTE 8
Background correction	mas	mas	mas	rma	rma	rma	none	none
Probe Normalization	loess	constant	invariant set	loess	constant	invariant set	loess	constant
PM correction	pmonly	mas	pmonly	pmonly	mas	pmonly	pmonly	mas
summarization	avgdiff	avgdiff	medianpolish	avgdiff	medianpolish	avgdiff	avgdiff	medianpolish

Figure 5: Analysis Table

Table 1: The AUC values for 8 routes

	AUC
Route 1	0.899657
Route 2	0.899804
Route 3	0.919651
Route 4	0.894383
Route 5	0.91328
Route 6	0.902061
Route 7	0.89746
Route 8	0.91259

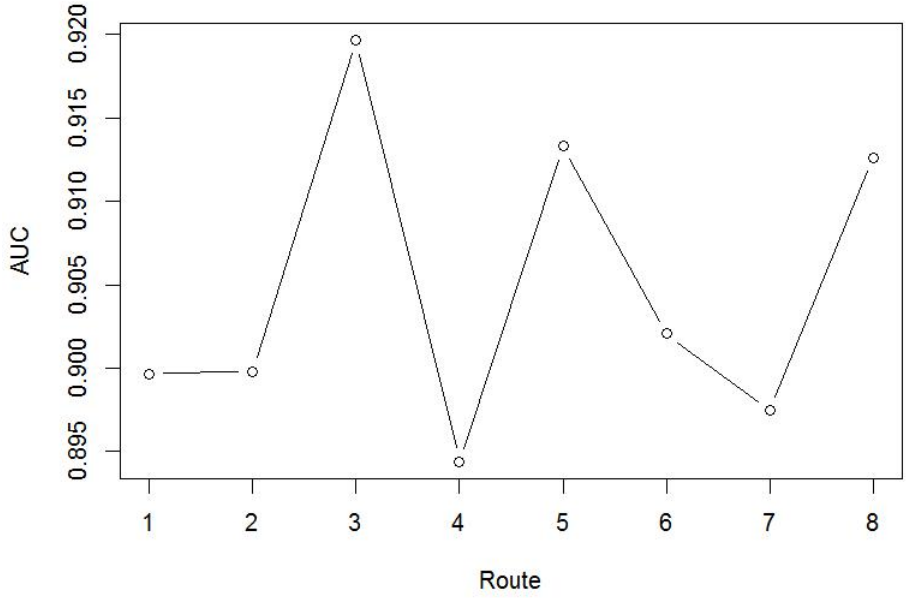

Figure 6: AUC Curve

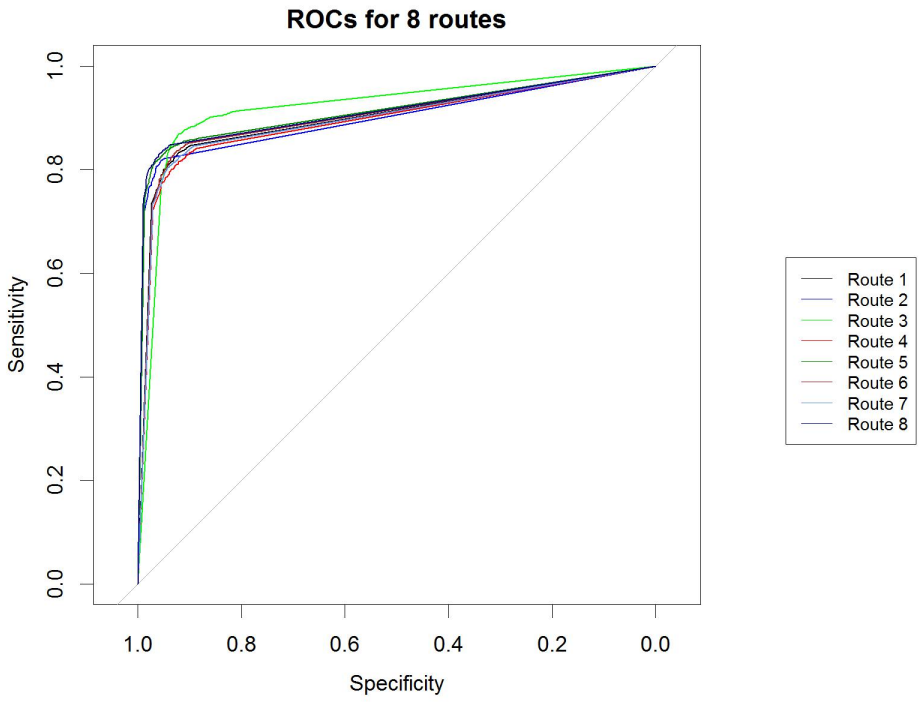

Figure 7: ROC Curve

Appendix

ALL statistical computing code for this homework.

```
##################################################
# STA 525: Statistics For Bioinfomatics
#      Homework Assignment #1
#
#------------------------------------------------
# Group 4 members:
#        Ziheng Cheng
#        Krithika Krishnan
#        Ziqiang Chen
##################################################

#-------------------
# Problem 1
#-------------------

# Legacy code from Dr. Gaile
rm(list=ls())
library(knitr)
library(GEOquery)

# originally, I ran this in the HW1 directory:
# gse19439=getGEO("GSE19439",GSEMatrix=T)
# gse19439=gse19439[[1]]
# save(file="HW1/gse19439.RData",gse19439)
# so that now I can save time and just load:
load( file ="HW1/gse19439.RData")

# 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"))
# get our expression set
X=exprs(gse19439)


# do a quick kruskal-wallis scan
myKrusk=function(i){
  cat(i,"...",fill=F)
  kruskal.test(x=X[i,],g=FTB)$p.value
}
# originally, I ran this in the HW1 directory:
# myPvals=mapply(myKrusk,1:(dim(X)[1])) ;save(file="HW1/myPvals.RData",myPvals)
# so that now I can save time and just load:
load("HW1/myPvals.RData")

# populate vector with last names of the groups in the class.
# note: the code is written this way, becasue it used to be student nameas and not Group names
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]
# print out list of best 4
print(best4)
```

```
for(i in 1:length(GroupLabels))
  cat("Group Label:",GroupLabels[i],
      "\t\t row assignment:",best4[i],fill=T)

## Group Label: Group I row assignment: 6874
## Group Label: Group II row assignment: 10685
## Group Label: Group III row assignment: 26058
## Group Label: Group IV row assignment: 47526

#-------------------------------------
# Problem 1 (our own part)
#-------------------------------------
?AnnotatedDataFrame
featureNames(gse19439[47526,])
myfeat <- featureData(gse19439[47526,])
varLabels(myfeat)
myvarmdat=varMetadata(myfeat)
mypdat=pData(myfeat)
mypdat
summary(mypdat$Probe_Type)

summary(mypdat$ILMN_Gene)


# Preparing for visualization in Problem 2
featVal.g4<- exprs(gse19439[47526,])


#-------------------------------------
# Problem 2
#-------------------------------------

# Library calls
wants <- c("beanplot","gplots","PresenceAbsence","plotROC","pROC","ROCR")
has   <- wants %in% rownames(installed.packages())
if(any(!has)) install.packages(wants[!has])
library("beanplot")

# Create target dataset
dat.g4 <- data.frame(featVal=as.numeric(featVal.g4), FTB=FTB)

# boxplot
boxplot(featVal~FTB, data=dat.g4, xlab="TB group", ylab="Feature values")

# violin plot
beanplot(featVal~FTB, data=dat.g4, xlab="TB group", ylab="Feature values")

save.image(file="HW1/RData/AllDataNeeded.RData")

#-------------------------------------
# Problem 3
#-------------------------------------

# Library calls
library(gplots)
best20 <- order(myPvals)[1:20]
data.matrix <- X[best20,]

# Labeled each column to the 3 group
colnames(data.matrix) <- FTB
```

```
match(data.matrix[4,],featVal.g4) # check if the cols order matched

# Heatmap using actual feature values

jpeg( file="HW1/Figures/Figure1.jpeg",width=1280,height=1024,pointsize=20)
heatmap.2(data.matrix,trace="none",main="Heatmap for Best20",key=T)
dev.off()

# The above heatmap seems less informative
# Try rank them across all the genes

data.matrix.rk <- data.matrix
for (i in 1:nrow(data.matrix)) {
  data.matrix.rk[i,] = as.numeric(rank(data.matrix[i,]))
}

jpeg( file="HW1/Figures/Figure1_rank.jpeg",width=1280,height=1024,pointsize=20)
heatmap.2(data.matrix.rk,trace="none",main="Heatmap for Best20 based on rank",key=T)
dev.off()

#-------------------------------------
# Problem 4
#-------------------------------------
# Library calls
source("http://bioconductor.org/biocLite.R")
biocLite("multtest")
require(multtest)
require(pROC)
# Dataset preparison from Dr. Gaile's lagecy code

# this provides an AffyBatch object
load("HW1/PSpikeData/PSpikeAffyBatch.RData")

spikeDF=read.table( file="HW1/PSpikeData/AffyProbeSpikeValues.csv",sep="\t")
 levels(spikeDF[,2])
summary(spikeDF[,2])
SpikeFC=as.numeric(levels(spikeDF[,2])[spikeDF[,2]])
names(SpikeFC)=spikeDF[,1]
nonZeroDX=which((SpikeFC!=0)&(!is.na(SpikeFC)))3422

nonDEdx = which(SpikeFC[nonZeroDX] == 1)
DEdx = which(SpikeFC[nonZeroDX] != 1)

require(affy)
require(affyPLM)

#-----------------------------------------------
# Make our own function to generate the ROC
#   Note:
#   Here we can set 4 different route-parameter
#   and can choose using the maximum statistisc
#   or the minimum p-value for testing
#-----------------------------------------------
roc.gen <- function(bgcorr,norm,pmcorr,sum,minP=F) {
  exprVal <- expresso(affydata, bgcorrect.method = bgcorr, normalize.method = norm,
                  pmcorrect.method = pmcorr, summary.method = sum)

  featVal <- exprs(exprVal)[nonZeroDX, ]
  nfeat <- dim(featVal)[1]
  group <- factor(c(rep("A", 9), rep("B", 9)))
```

```
  myresponse <- rep(NA, nfeat)
  myresponse[nonDEdx] <- 0
  myresponse[DEdx] <- 1

  if (!minP) {
    resT <- mt.maxT(featVal, group, B = 10000)
    testStat <- resT$teststat[order(resT$index)]
    roc = roc(response = myresponse, predictor = abs(testStat))
  }
  if(minP) {
    resP <- mt.minP(featVal, group, B = 10000)
    adjpVal <- resP$adjp[order(resP$index)]
    roc = roc(response = myresponse, predictor = adjpVal)
  }
  return(roc)
}

# You can change parameters below, I'll recommend using minP method to test
roc.1 <- roc.gen("mas","loess","pmonly","avgdiff",minP=T)
roc.2 <- roc.gen("mas","constant","mas","avgdiff",minP=T)
roc.3 <- roc.gen("mas","invariantset","pmonly","medianpolish",minP=T)
roc.4 <- roc.gen("rma","loess","pmonly","avgdiff",minP=T)
roc.5 <- roc.gen("rma","constant","mas","medianpolish",minP=T)
roc.6 <- roc.gen("rma","invariantset","pmonly","avgdiff",minP=T)
roc.7 <- roc.gen("none","loess","pmonly","avgdiff",minP=T)
roc.8 <- roc.gen("none","constant","mas","medianpolish",minP=T)

# save(roc.1,roc.2,roc.3,roc.4,roc.5,roc.6,roc.7,roc.8,
#      file="HW1/RData/LoadData.RData")

load("HW1/RData/LoadData.RData")

# Plot 8 ROC curves into one figure
jpeg( file="HW1/Figures/Figure2.jpeg",width=1600,height=1200,pointsize=36)

layout(cbind(1, 2), widths=c(4, 1))
plot(roc.1, main = "ROCs␣for␣8␣routes", xlim=c(1,0), ylim=c(0,1))
lines(roc.2, col = "blue")
lines(roc.3, col = "green")
lines(roc.4, col = "red")
lines(roc.5, col = "green4")
lines(roc.6, col = "firebrick")
lines(roc.7, col = "cornflowerblue")
lines(roc.8, col = "darkblue")

par(mar=c(0,0,0,0))
plot.new()
legend("center", lty=c(1,1,1,1,1,1,1,1),lwd=c(1.5,1.5,1.5,1.5,1.5,1.5,1.5,1.5),
       c("Route␣1","Route␣2","Route␣3","Route␣4","Route␣5","Route␣6","Route␣7","Route␣8"),
       col=c("black","blue","green","red","green4","firebrick","cornflowerblue","darkblue"),cex=0.8)

dev.off()

# Extract the AUC info from ROC curves
roc <- list(roc.1,roc.2,roc.3,roc.4,roc.5,roc.6,roc.7,roc.8)
auc <- c()
for (i in 1:8) {
  auc[i] <- as.numeric(roc[[i]]$auc)
}
```

```
# Make a table and figure to visualize them
auc.tbl <- as.matrix(auc)

rownames(auc.tbl) <- c("Route 1","Route 2","Route 3","Route 4","Route 5","Route 6","Route 7","Route 8")
colnames(auc.tbl) <- "AUC"

print(auc.tbl)

write.csv(auc.tbl, file="HW1/output/auc.csv")

jpeg(file="HW1/Figures/Figure3.jpeg",width=800,height=600,pointsize=20)
plot(auc.tbl, type="b", ylab="AUC", xlab="Route")
dev.off()
```