bort-analysis-final

Zhaoliang Zhou

4/24/2022

```r
###################################################
library("ridge")
library("sva")
```

```
## Loading required package: mgcv

## Loading required package: nlme

## This is mgcv 1.8-39. For overview type 'help("mgcv-package")'.

## Loading required package: genefilter

## Loading required package: BiocParallel
```

```r
library("car")
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:genefilter':
##
##     Anova
```

```r
library("preprocessCore")
library("ROCR")
library("GEOquery")
```

```
## Loading required package: Biobase

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
## Setting options('download.file.method.GEOquery'='auto')
```

```
## Setting options('GEOquery.inmemory.gpl'=FALSE)
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-3
```

```
library(coefplot)
```

```
## Loading required package: ggplot2
```

```
library(ROCR)
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
library(e1071)
```

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:coefplot':
##
##     extractPath
```

```
library(nnet)
```

```
##
## Attaching package: 'nnet'
```

```
## The following object is masked from 'package:mgcv':
##
##     multinom
```

```
library(pls)
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:coefplot':
##
##     coefplot
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
###################################################
### code chunk number 4: bortezomib.Snw:31-36
###################################################
scriptsDir <- "C:/Users/leonz/Desktop/UMN/Spring 2022/PubH 7475 - Statistical Learnign and Data mining/
source(file.path(scriptsDir, "compute_phenotype_function.R"))
source(file.path(scriptsDir, "summarizeGenesByMean.R"))
source(file.path(scriptsDir, "homogenize_data.R"))
source(file.path(scriptsDir, "do_variable_selection.R"))

source("C:/Users/leonz/Desktop/UMN/Spring 2022/PubH 7475 - Statistical Learnign and Data mining/final pr
```

```
###################################################
### code chunk number 5: bortezomib.Snw:42-44
###################################################
load("C:/Users/leonz/Desktop/UMN/Spring 2022/PubH 7475 - Statistical Learnign and Data mining/final proj
# bortezomib_mas5 <- getGEO("GSE9782") # uncomment this line to download the data directly from GEO.


###################################################
### code chunk number 6: bortezomib.Snw:49-57
###################################################
exprDataU133a <- cbind(exprs(bortezomib_mas5[[1]]), exprs(bortezomib_mas5[[2]]))
bortIndex <- c(which(pData(phenoData(bortezomib_mas5[[1]]))[,"characteristics_ch1.1"] == "treatment = PS
dexIndex <- c(which(pData(phenoData(bortezomib_mas5[[1]]))[,"characteristics_ch1.1"] == "treatment = Dex
studyIndex <- c(as.character(pData(bortezomib_mas5[[1]])[, "characteristics_ch1"]), as.character(pData(b

# exprDataU133a <- exprs(bortezomib_mas5[[1]])
# bortIndex <- which(pData(phenoData(bortezomib_mas5[[1]]))[,"characteristics_ch1.1"] == "treatment = PS
# dexIndex <- which(pData(phenoData(bortezomib_mas5[[1]]))[,"characteristics_ch1.1"] == "treatment = Dex
```

```
###################################################
### code chunk number 7: bortezomib.Snw:63-71
###################################################
library("hgu133a.db") # version 2.8.0
```

```
## Loading required package: AnnotationDbi

## Loading required package: stats4

## Loading required package: IRanges

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:Matrix':
##
##     expand, unname

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:nlme':
##
##     collapse

## The following object is masked from 'package:grDevices':
##
##     windows
## Loading required package: org.Hs.eg.db

##

##
```

```r
x <- hgu133aSYMBOL
mapped_probes <- mappedkeys(x)
names(mapped_probes) <- as.character(x[mapped_probes])
affy2sym <- as.character(x[mapped_probes])
sym2affy <- affy2sym
names(sym2affy) <- names(affy2sym)
rownames(exprDataU133a) <- sym2affy[rownames(exprDataU133a)]
```

```r
###################################################
### code chunk number 8: bortezomib.Snw:76-77
###################################################
load(file="C:/Users/leonz/Desktop/UMN/Spring 2022/PubH 7475 - Statistical Learnign and Data mining/fina


###################################################
### code chunk number 9: bortezomib.Snw:84-90
###################################################
sensBortezomib <- read.csv("C:/Users/leonz/Desktop/UMN/Spring 2022/PubH 7475 - Statistical Learnign and
as.is=TRUE)
bortic50s <- sensBortezomib$"IC.50"
names(bortic50s) <- sensBortezomib$"Cell.Line.Name"
tissue <- sensBortezomib$"Tissue"
names(tissue) <- sensBortezomib$"Cell.Line.Name"
```

```r
###################################################
### code chunk number 10: bortezomib.Snw:96-104
###################################################
pData <- read.delim("C:/Users/leonz/Desktop/UMN/Spring 2022/PubH 7475 - Statistical Learnign and Data m
pDataUnique <- pData[pData$Source.Name %in% names(which(table(pData$Source.Name) ==
1)), ]
rownames(pDataUnique) <- pDataUnique$Source.Name
commonCellLines <- rownames(pDataUnique)[rownames(pDataUnique) %in% names(bortic50s)]
pDataUniqueOrd <- pDataUnique[commonCellLines, ]
bortic50sOrd <- bortic50s[commonCellLines]
trainDataOrd <- gdsc_brainarray_syms[, pDataUniqueOrd$"Array.Data.File"]


###################################################
### code chunk number 11: bortezomib.Snw:110-111
###################################################
print(sum(grep("myeloma", sensBortezomib$Tissue), ignore.case=TRUE))
```

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

```r
###################################################
### code chunk number 12: bortezomib.Snw:117-119
###################################################
predictedSensitivity133a <- calcPhenotype(exprDataU133a[, bortIndex], trainDataOrd, bortic50sOrd,
selection=1)
```

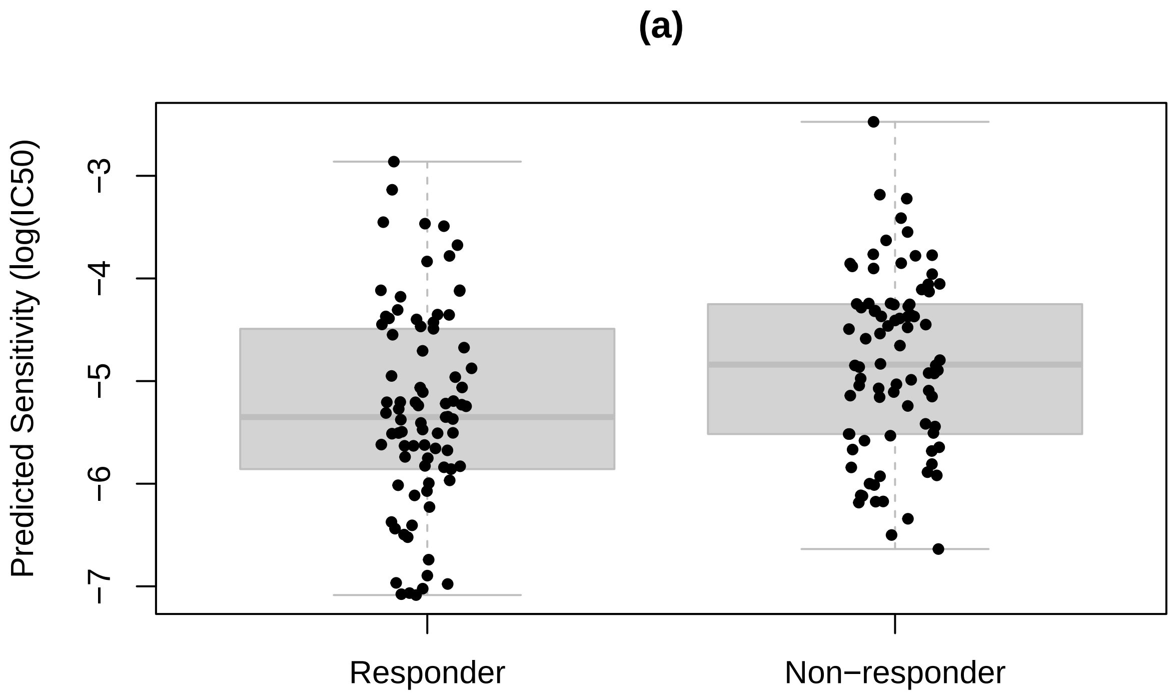
```
## Warning in if (class(testExprData) != "matrix") stop("ERROR: \"testExprData\"
## must be a matrix."): the condition has length > 1 and only the first element
## will be used

## Warning in if (class(trainingExprData) != "matrix") stop("ERROR:
## \"trainingExprData\" must be a matrix."): the condition has length > 1 and only
## the first element will be used

##
##  11198  gene identifiers overlap between the supplied expression matrices...
##

## Found2batches

## Adjusting for0covariate(s) or covariate level(s)

## Standardizing Data across genes

## Fitting L/S model and finding priors

## Finding parametric adjustments

## Adjusting the Data

##
## Fitting Ridge Regression model... Done
##
## Calculating predicted phenotype...

## Warning in if (class(homData$test) == "numeric") {: the condition has length > 1
## and only the first element will be used

## Done
```

```
#predictedSensitivity133a
```

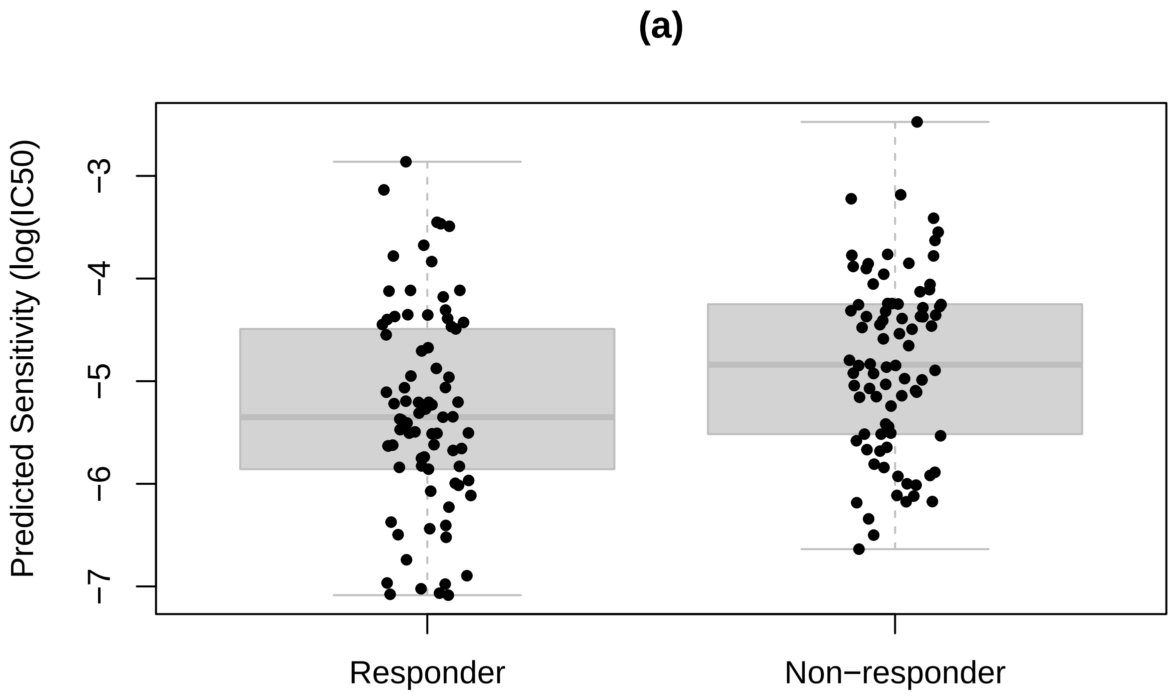
```
###################################################
### code chunk number 13: bortezomib.Snw:125-129
###################################################
resp133a <- c(as.character(pData(bortezomib_mas5[[1]])[, "characteristics_ch1.8"]),
as.character(pData(bortezomib_mas5[[2]])[, "characteristics_ch1.8"]))[bortIndex]
t.test(predictedSensitivity133a[resp133a == "PGx_Responder = NR"],
predictedSensitivity133a[resp133a == "PGx_Responder = R"], alternative="greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  predictedSensitivity133a[resp133a == "PGx_Responder = NR"] and predictedSensitivity133a[resp13
## t = 3.4735, df = 160.9, p-value = 0.0003301
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.2716172       Inf
## sample estimates:
## mean of x mean of y
## -4.821714 -5.340358
```

```
###################################################
### code chunk number 14: fig4aPlot
###################################################
lTwoa <- list("Responder"=predictedSensitivity133a[resp133a == "PGx_Responder = R"],
"Non-responder"=predictedSensitivity133a[resp133a == "PGx_Responder = NR"])
boxplot(lTwoa, outline=FALSE, border="grey", ylab="Predicted Sensitivity (log(IC50)",
main="(a)")
stripchart(lTwoa, vertical=TRUE, pch=20, method="jitter", add=TRUE)
```



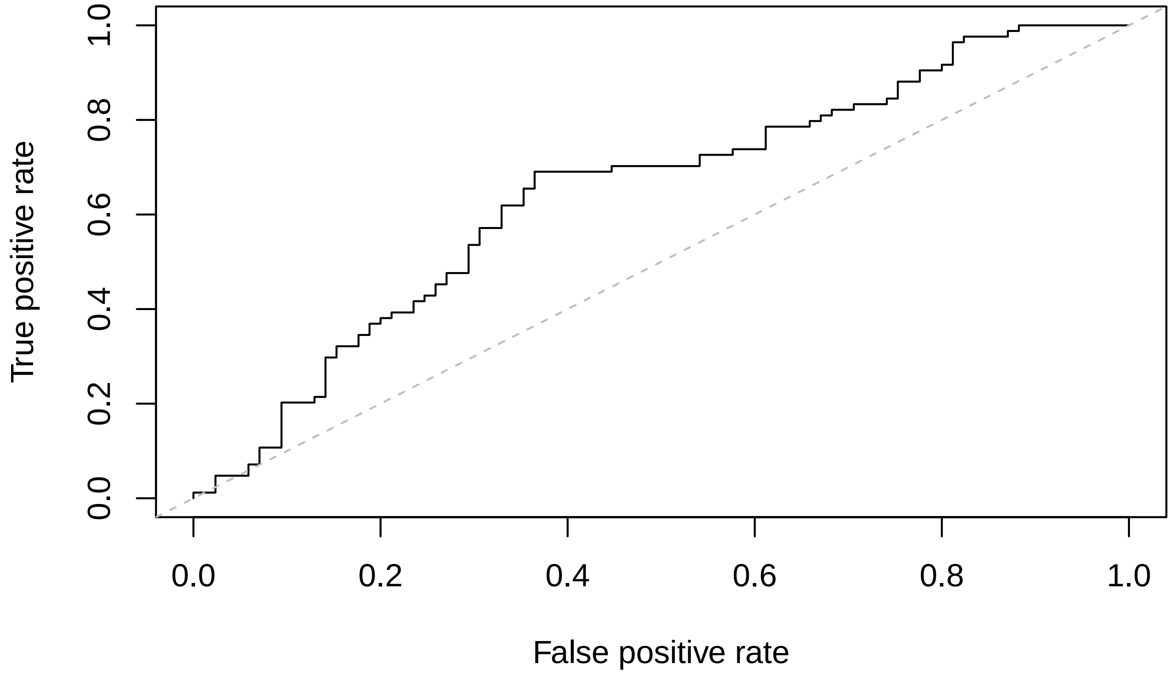
```
########################################################
### code chunk number 15: fig3a
########################################################
lTwoa <- list("Responder"=predictedSensitivity133a[resp133a == "PGx_Responder = R"],
"Non-responder"=predictedSensitivity133a[resp133a == "PGx_Responder = NR"])
boxplot(lTwoa, outline=FALSE, border="grey", ylab="Predicted Sensitivity (log(IC50)",
main="(a)")
stripchart(lTwoa, vertical=TRUE, pch=20, method="jitter", add=TRUE)
```



```
predA <- prediction(c(lTwoa[[1]], lTwoa[[2]]), c(rep("sens", length(lTwoa[[1]])),
rep("res", length(lTwoa[[2]]))), label.ordering=c("sens", "res"))
perfA <- performance(predA, measure = "tpr", x.measure = "fpr")
print(paste("AUC:", performance(predA, measure = "auc")@"y.values"[[1]]))
```

```
## [1] "AUC: 0.645238095238095"
```

```
plot(perfA)
abline(0, 1, col="grey", lty=2)
```



ridge (glmnet)

```
set.seed(99)
#predictedSensitivity_ridge2 <- calcPhenotype_ridge_glmnet(doceVivoNorm_syms, trainDataOrd, doceic50sOr
#selection=1)

predictedSensitivity133a_ridge2 <- calcPhenotype_ridge_glmnet(exprDataU133a[, bortIndex],
                                                              trainDataOrd,
                                                              bortic50sOrd,
                                                              selection=1)
```

```
## Warning in if (class(testExprData) != "matrix") stop("ERROR: \"testExprData\"
## must be a matrix."): the condition has length > 1 and only the first element
## will be used
```

```
## Warning in if (class(trainingExprData) != "matrix") stop("ERROR:
## \"trainingExprData\" must be a matrix."): the condition has length > 1 and only
## the first element will be used
```

```
##
##  11198  gene identifiers overlap between the supplied expression matrices...
##
```

```
## Found2batches
```

```
## Adjusting for0covariate(s) or covariate level(s)
## Standardizing Data across genes
```

```
## Fitting L/S model and finding priors
```

```
## Finding parametric adjustments
```

```
## Adjusting the Data
```

```
##
## Fitting glmnet ridge Regression model... Done
##
## Calculating predicted phenotype...
## best lambda min is: 12.87951
```

```
## Warning in if (class(homData$test) == "numeric") {: the condition has length > 1
## and only the first element will be used
```

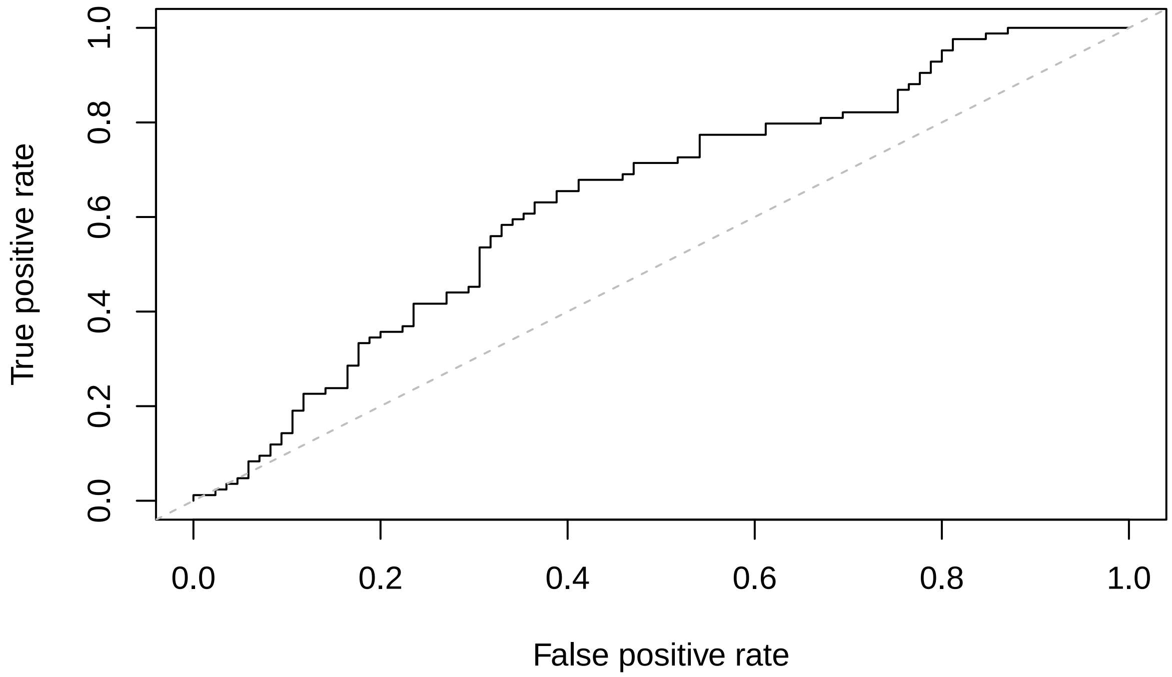
```
## Done
```

```
#predictedSensitivity133a_ridge2
```

```
lTwoa_ridge2 <- list("Responder"=predictedSensitivity133a_ridge2[resp133a == "PGx_Responder = R"],
"Non-responder"=predictedSensitivity133a_ridge2[resp133a == "PGx_Responder = NR"])

predA_ridge2 <- prediction(c(lTwoa_ridge2[[1]], lTwoa_ridge2[[2]]), c(rep("sens", length(lTwoa_ridge2[[
rep("res", length(lTwoa_ridge2[[2]]))), label.ordering=c("sens", "res"))

perfA_ridge2 <- performance(predA_ridge2, measure = "tpr", x.measure = "fpr")

print(paste("AUC:", performance(predA_ridge2, measure = "auc")@"y.values"[[1]]))
```

```
## [1] "AUC: 0.638655462184874"
```

```
plot(perfA_ridge2)
abline(0, 1, col="grey", lty=2)
```



lasso

```
set.seed(99)
#predictedSensitivity_ridge2 <- calcPhenotype_ridge_glmnet(doceVivoNorm_syms, trainDataOrd, doceic50sOr
#selection=1)

predictedSensitivity133a_lasso <- calcPhenotype_lasso(exprDataU133a[, bortIndex],
                                                              trainDataOrd,
                                                              bortic50sOrd,
                                                              selection=1)
```

```
## Warning in if (class(testExprData) != "matrix") stop("ERROR: \"testExprData\"
## must be a matrix."): the condition has length > 1 and only the first element
## will be used
```

```
## Warning in if (class(trainingExprData) != "matrix") stop("ERROR:
## \"trainingExprData\" must be a matrix."): the condition has length > 1 and only
## the first element will be used
```

```
##
##  11198  gene identifiers overlap between the supplied expression matrices...
##
```

```
## Found2batches
```

```
## Adjusting for0covariate(s) or covariate level(s)
## Standardizing Data across genes
```

```
## Fitting L/S model and finding priors
```

```
## Finding parametric adjustments
```

```
## Adjusting the Data
```

```
##
## Fitting glmnet lasso Regression model... Done
##
## Calculating predicted phenotype...
## best lambda min is: 0.05199482
```

```
## Warning in if (class(homData$test) == "numeric") {: the condition has length > 1
## and only the first element will be used
```

```
## Done
```

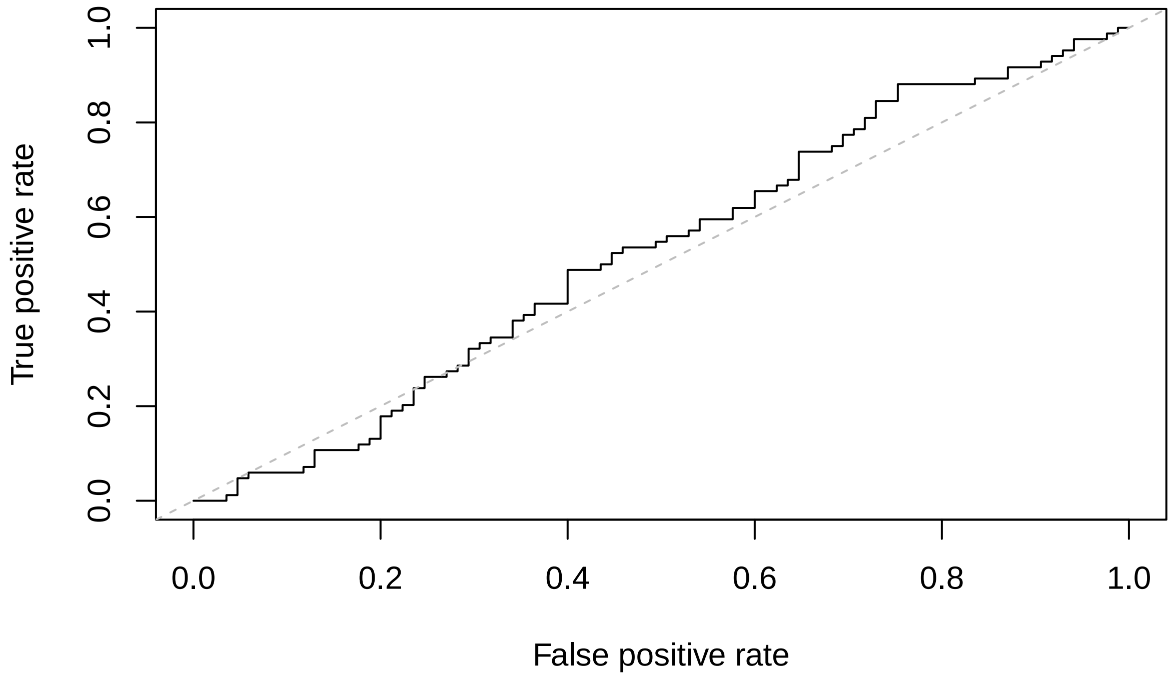
```
predictedSensitivity133a_lasso1 <- predictedSensitivity133a_lasso$preds
head(predictedSensitivity133a_lasso$coeffs)
```

```
##      features         coefs
## 1 (Intercept)  3.644123e+00
## 2        TFRC -1.146568e-05
## 3       CRIP1  4.338738e-05
## 4       ITM2B -1.070420e-04
## 5       ANXA1 -2.910715e-05
## 6        EGR1 -9.985080e-06
```

```
lTwoa_lasso <- list("Responder"=predictedSensitivity133a_lasso1[resp133a == "PGx_Responder = R"],
"Non-responder"=predictedSensitivity133a_lasso1[resp133a == "PGx_Responder = NR"])

predA_lasso <- prediction(c(lTwoa_lasso[[1]], lTwoa_lasso[[2]]), c(rep("sens", length(lTwoa_lasso[[1]]))
rep("res", length(lTwoa_lasso[[2]]))), label.ordering=c("sens", "res"))

perfA_lasso <- performance(predA_lasso, measure = "tpr", x.measure = "fpr")

print(paste("AUC:", performance(predA_lasso, measure = "auc")@"y.values"[[1]]))
```

```
## [1] "AUC: 0.526330532212885"
```

```
plot(perfA_lasso)
abline(0, 1, col="grey", lty=2)
```



elastic net

```
set.seed(99)
#predictedSensitivity_ridge2 <- calcPhenotype_ridge_glmnet(doceVivoNorm_syms, trainDataOrd, doceic50sOr
#selection=1)

predictedSensitivity133a_enet <- calcPhenotype_enet(exprDataU133a[, bortIndex],
                                                              trainDataOrd,
                                                              bortic50sOrd,
                                                              selection=1)
```

```
## Warning in if (class(testExprData) != "matrix") stop("ERROR: \"testExprData\"
## must be a matrix."): the condition has length > 1 and only the first element
## will be used
```

```
## Warning in if (class(trainingExprData) != "matrix") stop("ERROR:
## \"trainingExprData\" must be a matrix."): the condition has length > 1 and only
## the first element will be used
```

```
##
##  11198  gene identifiers overlap between the supplied expression matrices...
##
```

```
## Found2batches
```

```
## Adjusting for0covariate(s) or covariate level(s)
## Standardizing Data across genes
```

```
## Fitting L/S model and finding priors
```

```
## Finding parametric adjustments
```

```
## Adjusting the Data
```

```
##
## Fitting glmnet enet Regression model... Done
##
## Calculating predicted phenotype...
## best lambda min is: 0.1089412
```

```
## Warning in if (class(homData$test) == "numeric") {: the condition has length > 1
## and only the first element will be used
```

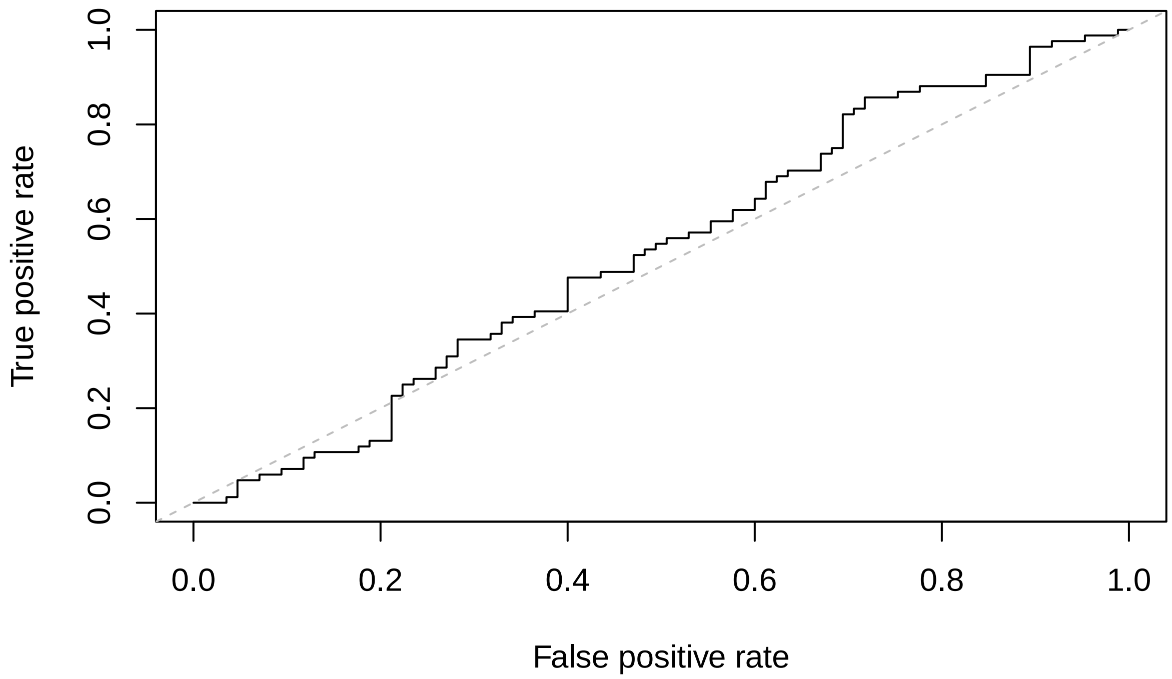
```
## Done
```

```
#predictedSensitivity133a_enet
```

```
lTwoa_enet <- list("Responder"=predictedSensitivity133a_enet[resp133a == "PGx_Responder = R"],
"Non-responder"=predictedSensitivity133a_enet[resp133a == "PGx_Responder = NR"])

predA_enet <- prediction(c(lTwoa_enet[[1]], lTwoa_enet[[2]]), c(rep("sens", length(lTwoa_enet[[1]])),
rep("res", length(lTwoa_enet[[2]]))), label.ordering=c("sens", "res"))

perfA_enet <- performance(predA_enet, measure = "tpr", x.measure = "fpr")

print(paste("AUC:", performance(predA_enet, measure = "auc")@"y.values"[[1]]))
```

```
## [1] "AUC: 0.530952380952381"
```

```
plot(perfA_enet)
abline(0, 1, col="grey", lty=2)
```



svm

```
set.seed(99)
#predictedSensitivity_ridge2 <- calcPhenotype_ridge_glmnet(doceVivoNorm_syms, trainDataOrd, doceic50sOr
#selection=1)

predictedSensitivity133a_svm <- calcPhenotype_svm(exprDataU133a[, bortIndex],
                                                              trainDataOrd,
                                                              bortic50sOrd,
                                                              selection=1)
```

```
## Warning in if (class(testExprData) != "matrix") stop("ERROR: \"testExprData\"
## must be a matrix."): the condition has length > 1 and only the first element
## will be used
```

```
## Warning in if (class(trainingExprData) != "matrix") stop("ERROR:
## \"trainingExprData\" must be a matrix."): the condition has length > 1 and only
## the first element will be used
```

```
##
##  11198  gene identifiers overlap between the supplied expression matrices...
##
```

```
## Found2batches
```

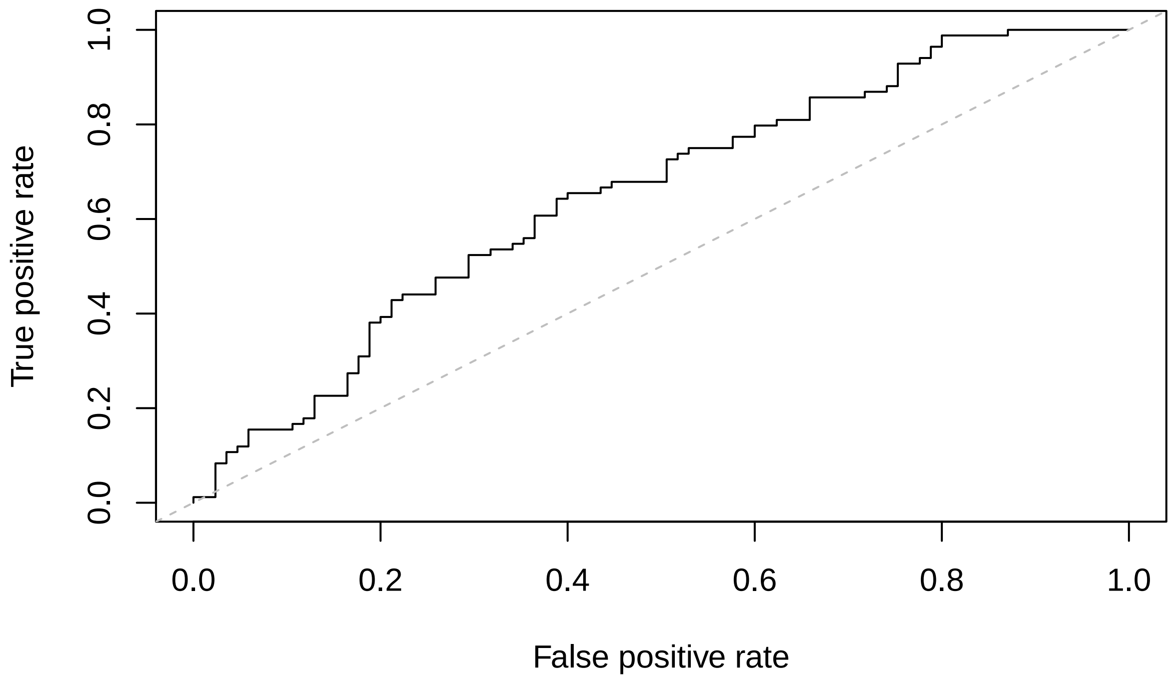
```
## Adjusting for0covariate(s) or covariate level(s)
## Standardizing Data across genes

## Fitting L/S model and finding priors

## Finding parametric adjustments

## Adjusting the Data

##
## Fitting svm Regression model... Done
##
## Calculating predicted phenotype... best gamma 0.0001122965 best cost 1

## Warning in if (class(homData$test) == "numeric") {: the condition has length > 1
## and only the first element will be used

## Done
```

```
#predictedSensitivity133a_svm
```

```
lTwoa_svm <- list("Responder"=predictedSensitivity133a_svm[resp133a == "PGx_Responder = R"],
"Non-responder"=predictedSensitivity133a_svm[resp133a == "PGx_Responder = NR"])

predA_svm <- prediction(c(lTwoa_svm[[1]], lTwoa_svm[[2]]), c(rep("sens", length(lTwoa_svm[[1]])),
rep("res", length(lTwoa_svm[[2]]))), label.ordering=c("sens", "res"))

perfA_svm <- performance(predA_svm, measure = "tpr", x.measure = "fpr")

print(paste("AUC:", performance(predA_svm, measure = "auc")@"y.values"[[1]]))
```

```
## [1] "AUC: 0.65"
```

```
plot(perfA_svm)
abline(0, 1, col="grey", lty=2)
```



ann

```
# lasso selected variables
lasso_select_vars2 <- c(predictedSensitivity133a_lasso$coeffs$features)

# remove intercept
lasso_select_vars2 <-  lasso_select_vars2[-1]
lasso_select_vars2
```

```
##  [1] "TFRC"     "CRIP1"    "ITM2B"    "ANXA1"    "EGR1"     "ARID5B"  
##  [7] "TKT"      "PARP6"    "POLR2G"   "CNN2"     "TDO2"     "MINPP1"  
## [13] "DYNC1H1"  "MRPL16"   "PRKAG1"   "CDC37"    "GNE"      "ATXN1"   
## [19] "NTAN1"    "TOR1B"    "PEPD"     "EDN1"     "VPS4B"    "FN1"     
## [25] "CA14"     "ZNF292"   "DDA1"     "BDH2"     "HTATSF1"  "CEACAM4" 
## [31] "STX10"    "CLP1"     "AKAP10"   "C1orf105" "ALKBH1"   "AAR2"    
## [37] "SUPT3H"   "ANGEL1"   "CER1"     "PTGDS"    "JAG2"     "MMP8"    
## [43] "OLFML2A"  "GTF2A1"   "APOE"     "NEUROD2"  "TRIT1"    "CYSLTR2" 
## [49] "RNF111"   "CXCL14"   "ZNF12"    "CYP46A1"  "SFTPB"    "SOCS7"   
## [55] "AHCTF1"   "SYNGR4"   "TRAF6"    "DLX4"     "MMP14"    "TIMP4"   
## [61] "WDR4"     "PRKAA2"   "ZNF467"   "OR2H2"    "FGFR1"    "KCNB2"   
## [67] "NCAM2"    "CLUL1"    "MDM4"     "TTC38"    "NEUROD1"  "FN3K"    
## [73] "DMC1"
```

```
length(lasso_select_vars2)
```

```
## [1] 73
```

```
trainDataOrd_subset2 <- trainDataOrd[rownames(trainDataOrd) %in% lasso_select_vars2, ]  # Extract rows
dim(trainDataOrd_subset2)                                                        # Print data frame subset
```

```
## [1]  73 280
```

```
set.seed(99)
#predictedSensitivity_ridge2 <- calcPhenotype_ridge_glmnet(doceVivoNorm_syms, trainDataOrd, doceic50sOr
#selection=1)

predictedSensitivity133a_ann <- calcPhenotype_ann(exprDataU133a[, bortIndex],
                                                            trainDataOrd_subset2,
                                                            bortic50sOrd,
                                                            selection=1)
```

```
## Warning in if (class(testExprData) != "matrix") stop("ERROR: \"testExprData\"
## must be a matrix."): the condition has length > 1 and only the first element
## will be used
```

```
## Warning in if (class(trainingExprData) != "matrix") stop("ERROR:
## \"trainingExprData\" must be a matrix."): the condition has length > 1 and only
## the first element will be used
```

```
##
##  73  gene identifiers overlap between the supplied expression matrices...
##
```

```
## Found2batches
```

```
## Adjusting for0covariate(s) or covariate level(s)
```

```
## Standardizing Data across genes
```

```
## Fitting L/S model and finding priors
```

```
## Finding parametric adjustments
```

```
## Adjusting the Data
```

```
##
## Fitting ANN Regression model... # weights:  61
## initial  value 2654.020282
## iter  10 value 68.691011
## iter  20 value 68.426203
## iter  30 value 66.089126
## iter  40 value 65.293233
## iter  50 value 64.281777
## iter  60 value 64.152370
## iter  70 value 64.063063
## iter  80 value 63.466761
## iter  90 value 63.210423
## iter 100 value 63.129204
## iter 110 value 62.846542
## iter 120 value 62.610755
## iter 130 value 62.518028
## iter 140 value 62.486556
## iter 150 value 62.342098
## iter 160 value 62.179621
## iter 170 value 62.164566
## iter 180 value 62.123214
## iter 190 value 62.102294
## iter 200 value 62.093400
## iter 210 value 62.087302
## iter 220 value 62.050310
## iter 230 value 62.028179
## iter 240 value 62.022166
## iter 250 value 61.991343
## iter 260 value 61.152446
## iter 270 value 61.055032
## iter 280 value 60.902580
## iter 290 value 60.866956
## iter 300 value 60.835421
## iter 310 value 60.053994
## iter 320 value 59.962717
## iter 330 value 59.857383
## iter 340 value 59.761111
## iter 350 value 59.722987
## iter 360 value 59.568218
## iter 370 value 59.448353
## iter 380 value 59.404515
## iter 390 value 59.274271
## iter 400 value 59.260613
## iter 410 value 59.243114
## iter 420 value 59.229148
## iter 430 value 59.213798
## iter 440 value 59.210217
## iter 450 value 59.207603
## iter 460 value 59.195705
## iter 470 value 59.129934
## iter 480 value 59.124086
## iter 490 value 59.084843
## iter 500 value 59.053480
## iter 510 value 59.045391
## iter 520 value 59.014850
## iter 530 value 58.247677
## iter 540 value 58.219010
## iter 550 value 58.112925
## iter 560 value 58.063978
## iter 570 value 58.057760
## iter 580 value 58.056834
## final  value 58.056654
## converged
## Done
##
## Calculating predicted phenotype... /nbest size is: 1
```

```
## Warning in if (class(homData$test) == "numeric") {: the condition has length > 1
## and only the first element will be used
```

```
## Done
```

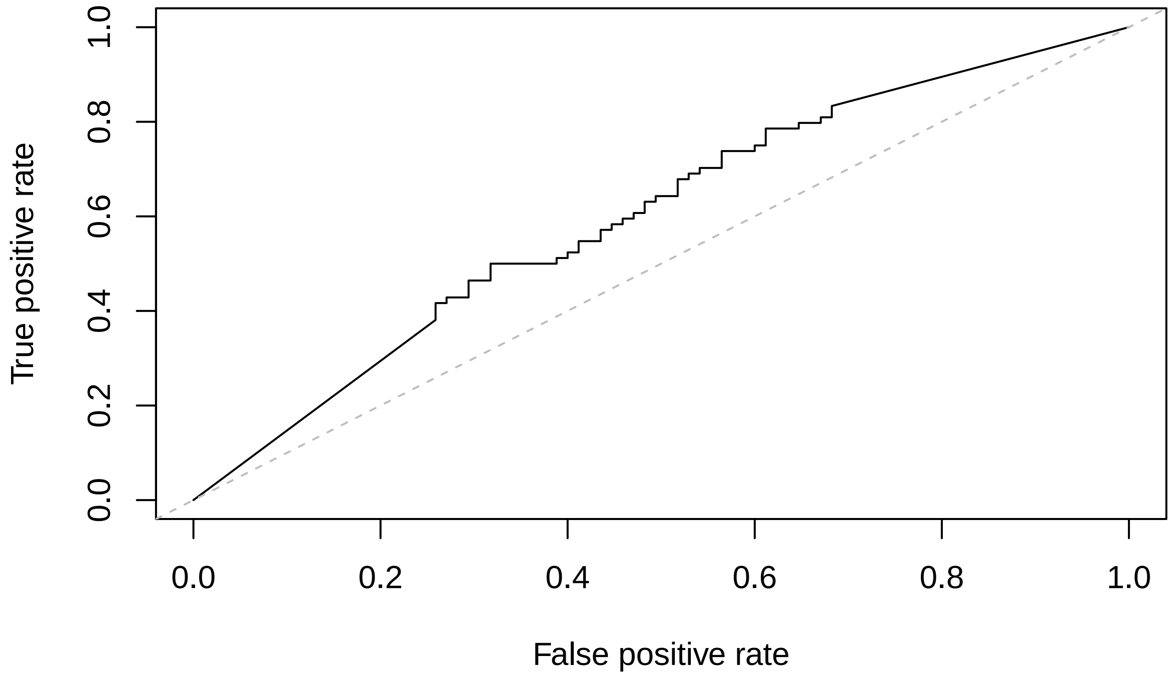
```
#predictedSensitivity133a_svm
```

```
lTwoa_ann <- list("Responder"=predictedSensitivity133a_ann[resp133a == "PGx_Responder = R"],
"Non-responder"=predictedSensitivity133a_ann[resp133a == "PGx_Responder = NR"])
```

```
predA_ann <- prediction(c(lTwoa_ann[[1]], lTwoa_ann[[2]]), c(rep("sens", length(lTwoa_ann[[1]])),
rep("res", length(lTwoa_ann[[2]]))), label.ordering=c("sens", "res"))

perfA_ann <- performance(predA_ann, measure = "tpr", x.measure = "fpr")

print(paste("AUC:", performance(predA_ann, measure = "auc")@"y.values"[[1]]))
```

```
## [1] "AUC: 0.600560224089636"
```

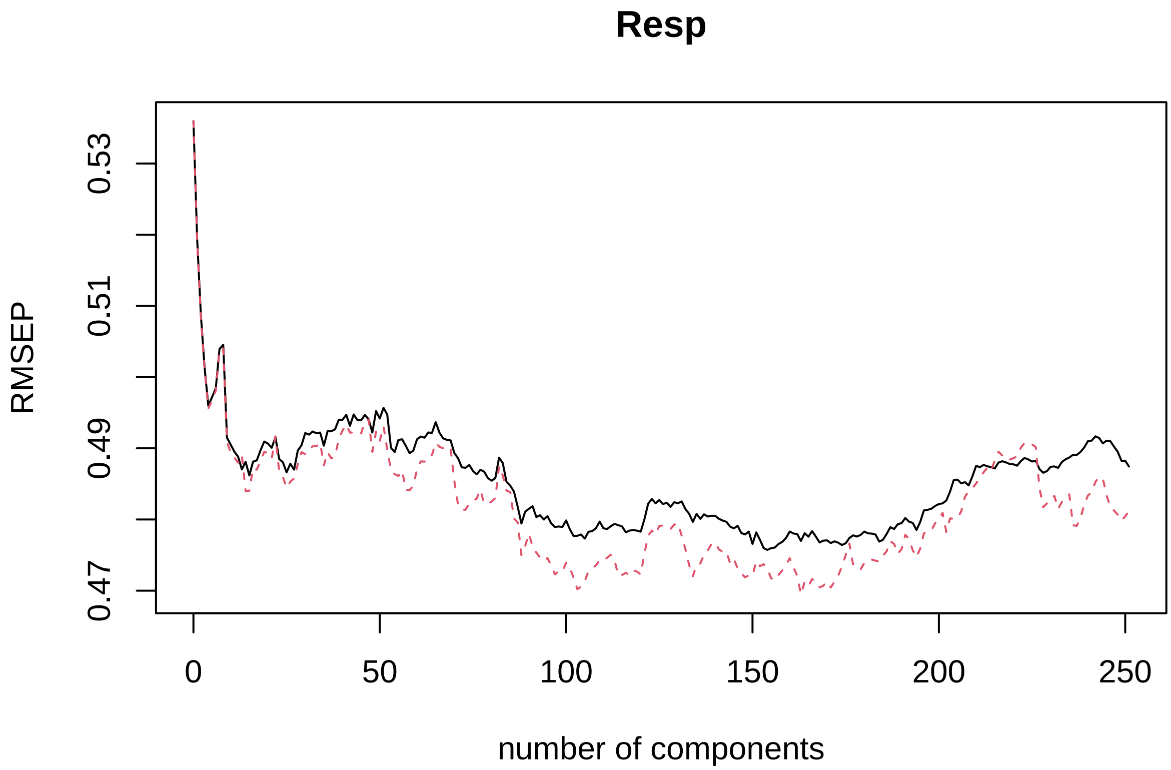
```
plot(perfA_ann)
abline(0, 1, col="grey", lty=2)
```



pcr

```
set.seed(99)
#predictedSensitivity_ridge2 <- calcPhenotype_ridge_glmnet(doceVivoNorm_syms, trainDataOrd, doceic50sOr
#selection=1)

predictedSensitivity133a_pcr <- calcPhenotype_pcr(exprDataU133a[, bortIndex],
                                                  trainDataOrd,
                                                  bortic50sOrd,
                                                  selection=1,
                                                  Ncomp=150)
```

```
## Warning in if (class(testExprData) != "matrix") stop("ERROR: \"testExprData\"
## must be a matrix."): the condition has length > 1 and only the first element
## will be used

## Warning in if (class(trainingExprData) != "matrix") stop("ERROR:
## \"trainingExprData\" must be a matrix."): the condition has length > 1 and only
## the first element will be used

##
##  11198  gene identifiers overlap between the supplied expression matrices...
##

## Found2batches

## Adjusting for0covariate(s) or covariate level(s)

## Standardizing Data across genes

## Fitting L/S model and finding priors

## Finding parametric adjustments

## Adjusting the Data

##
## Fitting PCR Regression model... Done
##
## Calculating predicted phenotype... /n

## Warning in if (class(homData$test) == "numeric") {: the condition has length > 1
## and only the first element will be used
```



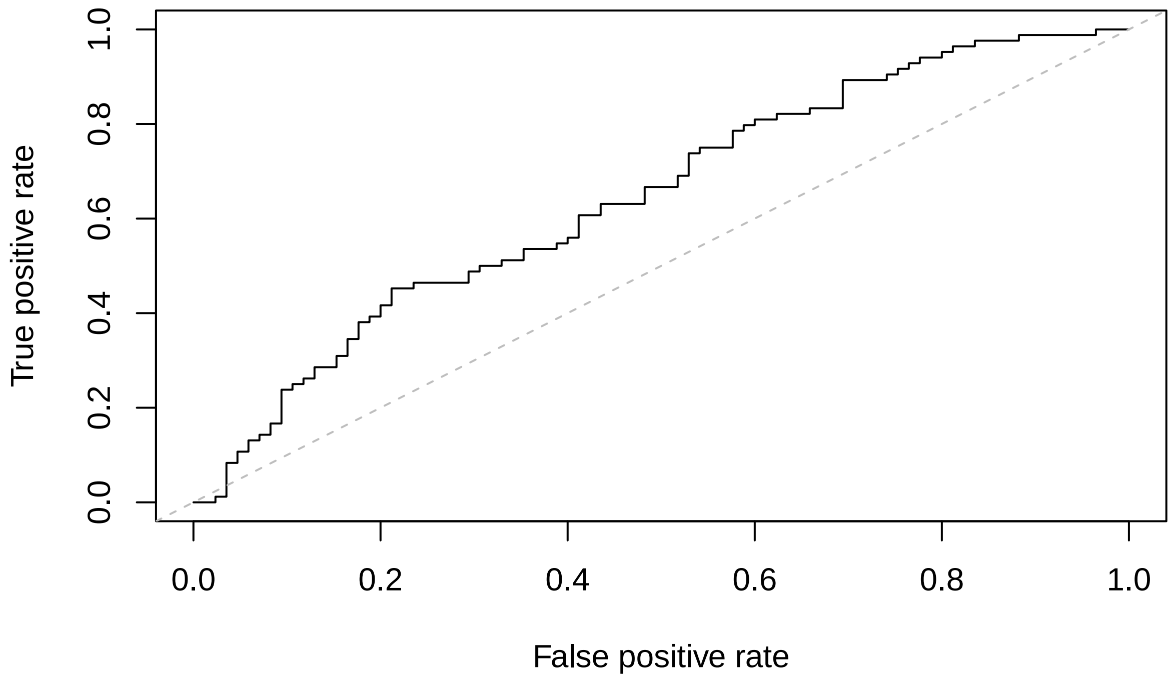
```
## Done
```

```
#predictedSensitivity133a_svm
```

```
lTwoa_pcr <- list("Responder"=predictedSensitivity133a_pcr[resp133a == "PGx_Responder = R"],
"Non-responder"=predictedSensitivity133a_pcr[resp133a == "PGx_Responder = NR"])

predA_pcr <- prediction(c(lTwoa_pcr[[1]], lTwoa_pcr[[2]]), c(rep("sens", length(lTwoa_pcr[[1]])),
rep("res", length(lTwoa_pcr[[2]]))), label.ordering=c("sens", "res"))

perfA_pcr <- performance(predA_pcr, measure = "tpr", x.measure = "fpr")

print(paste("AUC:", performance(predA_pcr, measure = "auc")@"y.values"[[1]]))
```

```
## [1] "AUC: 0.644117647058824"
```

```
plot(perfA_pcr)
abline(0, 1, col="grey", lty=2)
```

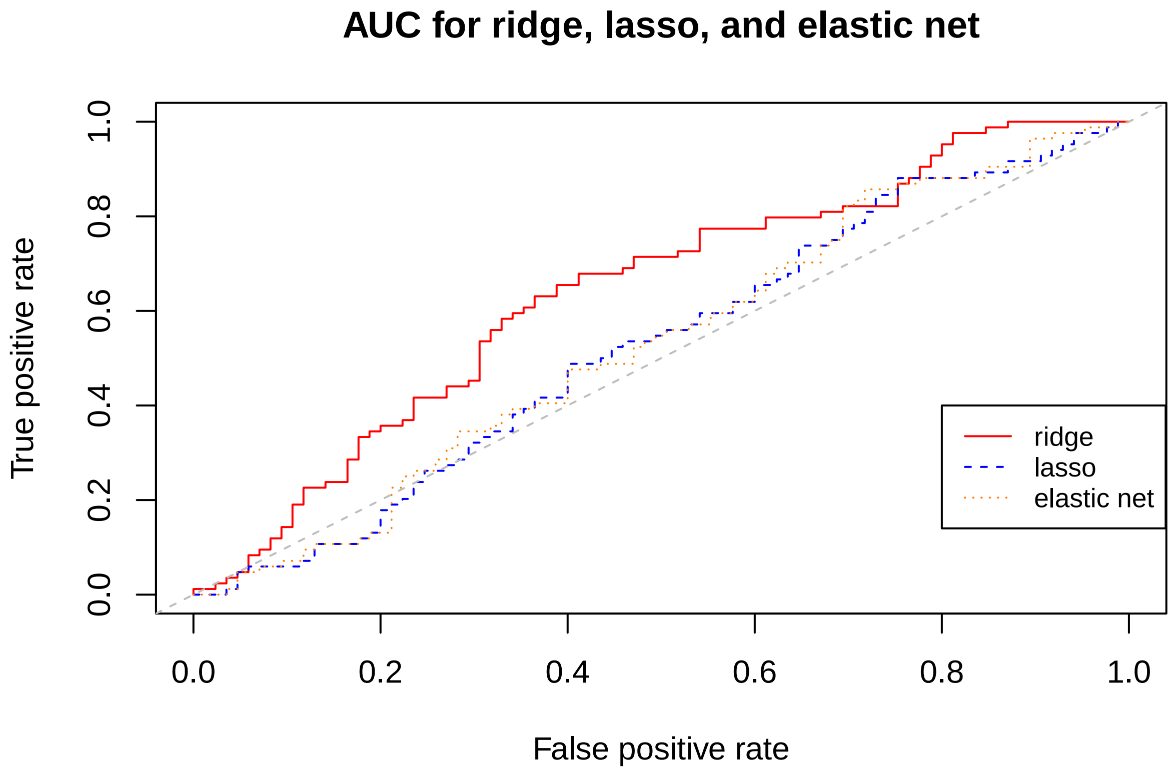


roc curves

1st set of plots: ridge, lasso, elestic net

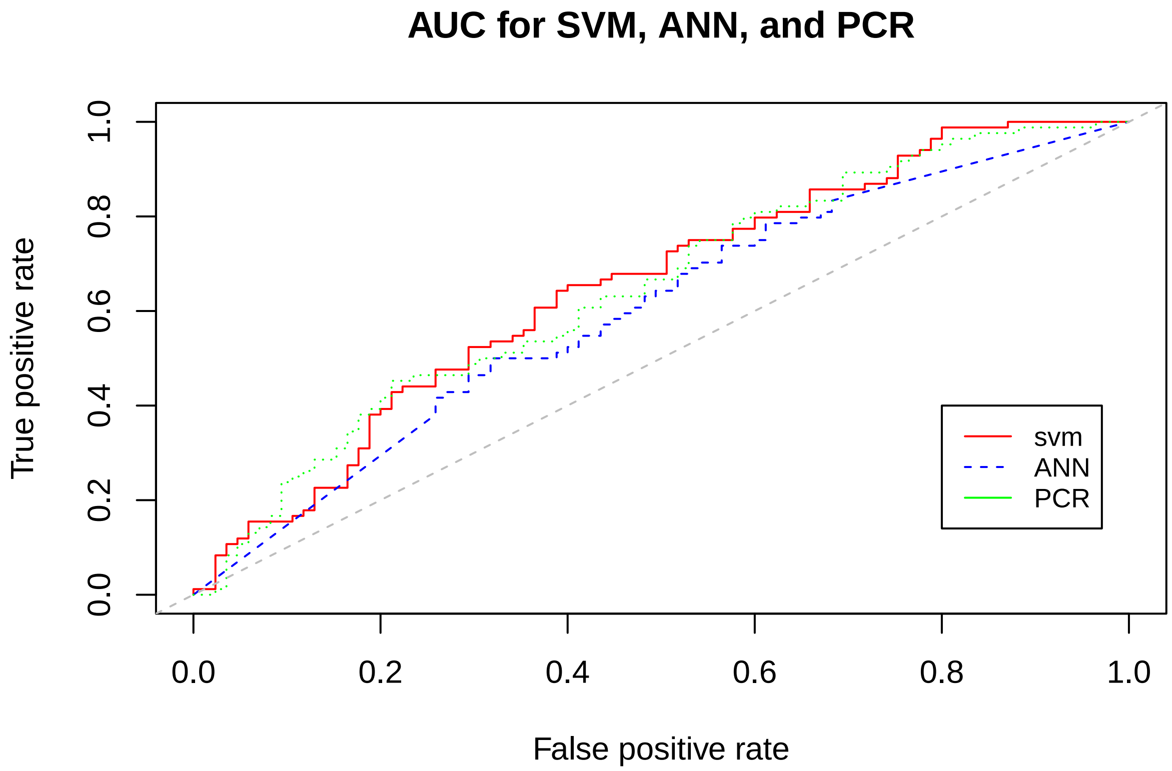
```
plot(perfA_ridge2 ,main="AUC for ridge, lasso, and elastic net", col="red", lty=1)
lines(perfA_lasso@x.values[[1]], perfA_lasso@y.values[[1]], col="blue", lty=2)
lines(perfA_enet@x.values[[1]], perfA_enet@y.values[[1]], col="darkorange1", lty=3)
abline(0, 1, col="grey", lty=2)
```

```
legend(0.8, 0.4, legend=c("ridge", "lasso","elastic net"),
       col=c("red", "blue","darkorange1"), lty=1:3, cex=0.8)
```



part 2: svm, ann, pcr

```
plot(perfA_svm ,main="AUC for SVM, ANN, and PCR", col="red", lty=1)
lines(perfA_ann@x.values[[1]], perfA_ann@y.values[[1]], col="blue", lty=2)
lines(perfA_pcr@x.values[[1]], perfA_pcr@y.values[[1]], col="green", lty=3)
abline(0, 1, col="grey", lty=2)
legend(0.8, 0.4, legend=c("svm", "ANN","PCR"),
       col=c("red", "blue","green"), lty=1:2, cex=0.8)
```


AUC for SVM, ANN, and PCR
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
svm
ANN
PCR