

# decetaaxel-analysis

Zhaoliang Zhou

4/17/2022

```
library(glmnet)

## Loading required package: Matrix
## Loaded glmnet 4.1-3
library(coefplot)

## Loading required package: ggplot2
library(ROCR)

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

## 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

load(file="C:/Users/leonZ/Desktop/UMN/Spring 2022/PubH 7475 - Statistical Learnign and Data mining/final
load(file="C:/Users/leonZ/Desktop/UMN/Spring 2022/PubH 7475 - Statistical Learnign and Data mining/final

#####
### code chunk number 5: docetaxelBreastCancer.Snw:42-46
#####
sensDoce <- read.csv("C:/Users/leonZ/Desktop/UMN/Spring 2022/PubH 7475 - Statistical Learnign and Data m
as.is=TRUE)
doceic50s <- sensDoce$"IC.50"
names(doceic50s) <- sensDoce$"Cell.Line.Name"

#####
### code chunk number 6: docetaxelBreastCancer.Snw:50-58
#####
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(doceic50s)]
```

```
pDataUniqueOrd <- pDataUnique[commonCellLines, ]
doceic50sOrd <- doceic50s[commonCellLines]
trainDataOrd <- gdsc_brainarray_syms[, pDataUniqueOrd$"Array.Data.File"]
```

```
dat1 <- t(trainDataOrd)
dim(dat1)
```

```
## [1] 482 12092
```

```
length(pDataUniqueOrd$Characteristics.CellLine.)
```

```
## [1] 482
```

## ridge

```
homdata <- homogenizeData(doceVivoNorm_syms, trainDataOrd, selection = 2)
```

```
##
## 8239 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
```

```
ridge_cv <- cv.glmnet(t(homdata$train), doceic50sOrd, alpha = 0)
# Best lambda value
best_lambda <- ridge_cv$lambda.min
best_lambda
```

```
## [1] 72.69454
```

```
best_ridge <- glmnet(t(homdata$train), doceic50sOrd, alpha = 0, lambda = best_lambda)
```

```
preds <- predict(best_ridge, newx = t(homdata$test) )
preds
```

```
## s0
## GSM4903.CEL -5.652490
## GSM4907.CEL -5.832562
## GSM4908.CEL -5.444328
## GSM4914.CEL -4.943075
## GSM4915.CEL -5.461420
## GSM4917.CEL -5.154739
## GSM4919.CEL -5.422557
## GSM4920.CEL -5.280706
## GSM4921.CEL -5.376496
## GSM4923.CEL -5.182137
## GSM4901.CEL -4.842042
## GSM4902.CEL -5.273612
## GSM4904.CEL -5.125376
```

```
## GSM4905.CEL -4.462948
## GSM4906.CEL -5.617433
## GSM4909.CEL -4.078978
## GSM4910.CEL -4.709038
## GSM4911.CEL -5.054165
## GSM4912.CEL -5.177950
## GSM4913.CEL -5.246358
## GSM4916.CEL -5.310969
## GSM4918.CEL -4.946460
## GSM4922.CEL -5.149333
## GSM4924.CEL -4.880405
```

```
length(preds)
```

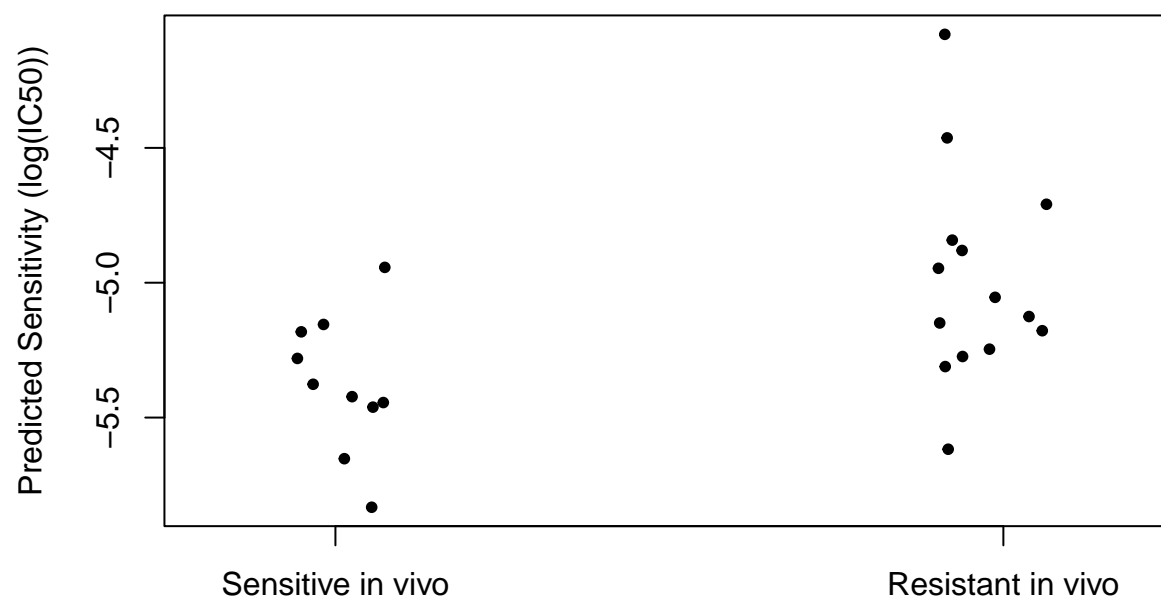
```
## [1] 24
```

```
t.test(preds[1:10], preds[11:24], alternative="less")
```

```
##
## Welch Two Sample t-test
##
## data: preds[1:10] and preds[11:24]
## t = -2.9301, df = 21.904, p-value = 0.003887
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -0.1589055
## sample estimates:
## mean of x mean of y
## -5.375051 -4.991076
```

```
stripchart(list("Sensitive in vivo"=preds[1:10],
"Resistant in vivo"=preds[11:24]), vertical=TRUE,
method="jitter", pch=20, ylab="Predicted Sensitivity (log(IC50))", main="(a)")
```

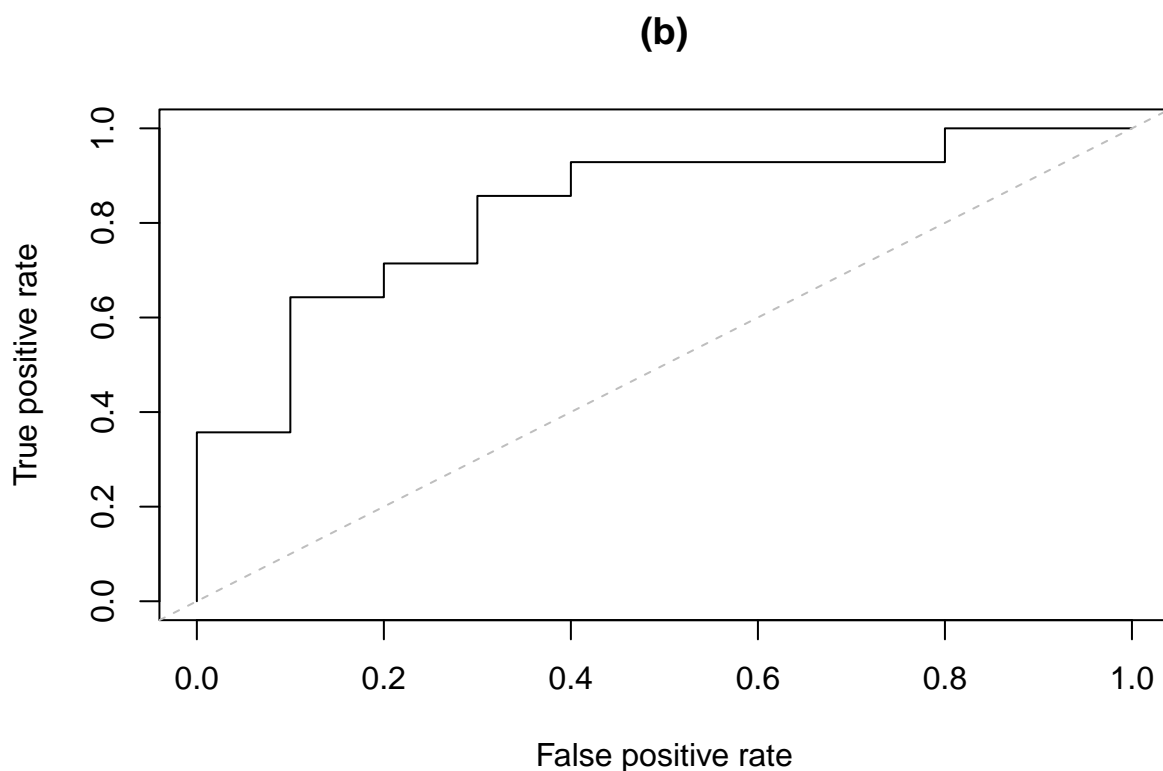
(a)



```
pred <- prediction(preds, c(rep("sens", 10), rep("res", 14)),
  label.ordering=c("sens", "res"))
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
print(paste("AUC:", performance(pred, measure = "auc")@"y.values"[[1]]))
```

```
## [1] "AUC: 0.828571428571429"
```

```
plot(perf, main="(b)")
abline(0, 1, col="grey", lty=2)
```



## LASSO

```
lasso_cv <- cv.glmnet(t(homdata$train), doceic50s0rd, alpha = 1)
# Best lambda value
best_lambda <- lasso_cv$lambda.min
best_lambda
```

```
## [1] 0.1394216
```

```
best_lasso <- glmnet(t(homdata$train), doceic50s0rd, alpha = 1, lambda = best_lambda)
```

```
preds_lasso <- predict(best_lasso, newx = t(homdata$test))
preds_lasso
```

```
##          s0
## GSM4903.CEL -6.188587
## GSM4907.CEL -5.919239
## GSM4908.CEL -5.906113
## GSM4914.CEL -4.444241
## GSM4915.CEL -5.416213
## GSM4917.CEL -5.314470
## GSM4919.CEL -6.014612
## GSM4920.CEL -5.556180
## GSM4921.CEL -6.020851
## GSM4923.CEL -5.354803
## GSM4901.CEL -4.381878
```

```

## GSM4902.CEL -5.009069
## GSM4904.CEL -5.205035
## GSM4905.CEL -3.797694
## GSM4906.CEL -6.114775
## GSM4909.CEL -3.274140
## GSM4910.CEL -4.483572
## GSM4911.CEL -4.861305
## GSM4912.CEL -4.990360
## GSM4913.CEL -5.415999
## GSM4916.CEL -5.275348
## GSM4918.CEL -4.916336
## GSM4922.CEL -5.139295
## GSM4924.CEL -4.736201

length(preds_lasso)

## [1] 24

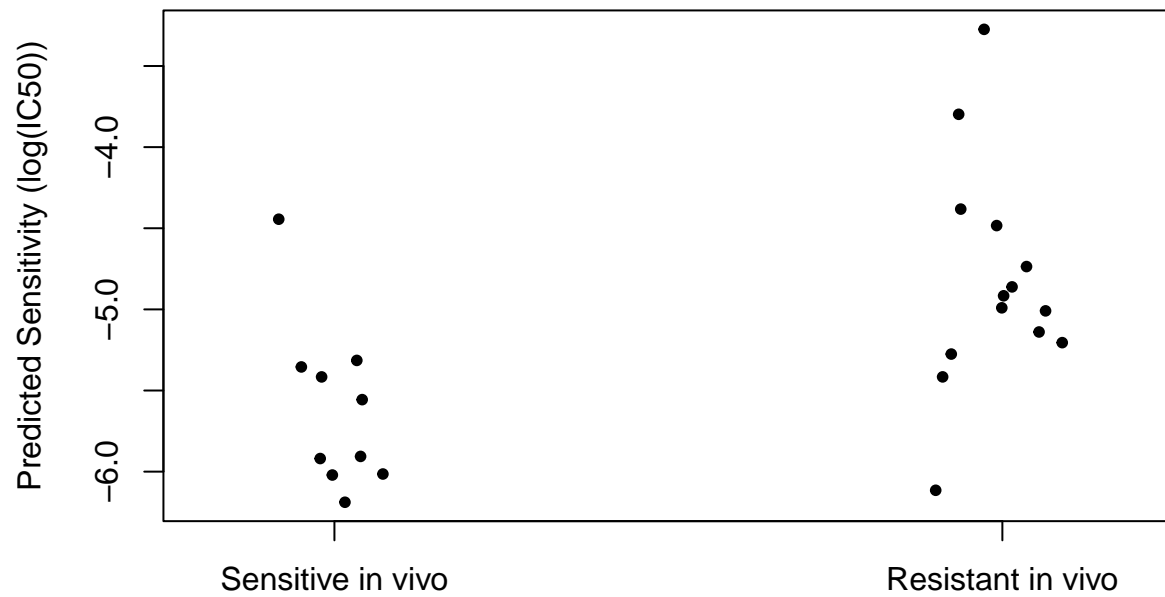
t.test(preds_lasso[1:10], preds_lasso[11:24], alternative="less")

##
## Welch Two Sample t-test
##
## data: preds_lasso[1:10] and preds_lasso[11:24]
## t = -3.1684, df = 21.939, p-value = 0.002231
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -0.3594551
## sample estimates:
## mean of x mean of y
## -5.613531 -4.828643

stripchart(list("Sensitive in vivo"=preds_lasso[1:10],
"Resistant in vivo"=preds_lasso[11:24]), vertical=TRUE,
method="jitter", pch=20, ylab="Predicted Sensitivity (log(IC50))", main="(a)")

```

(a)

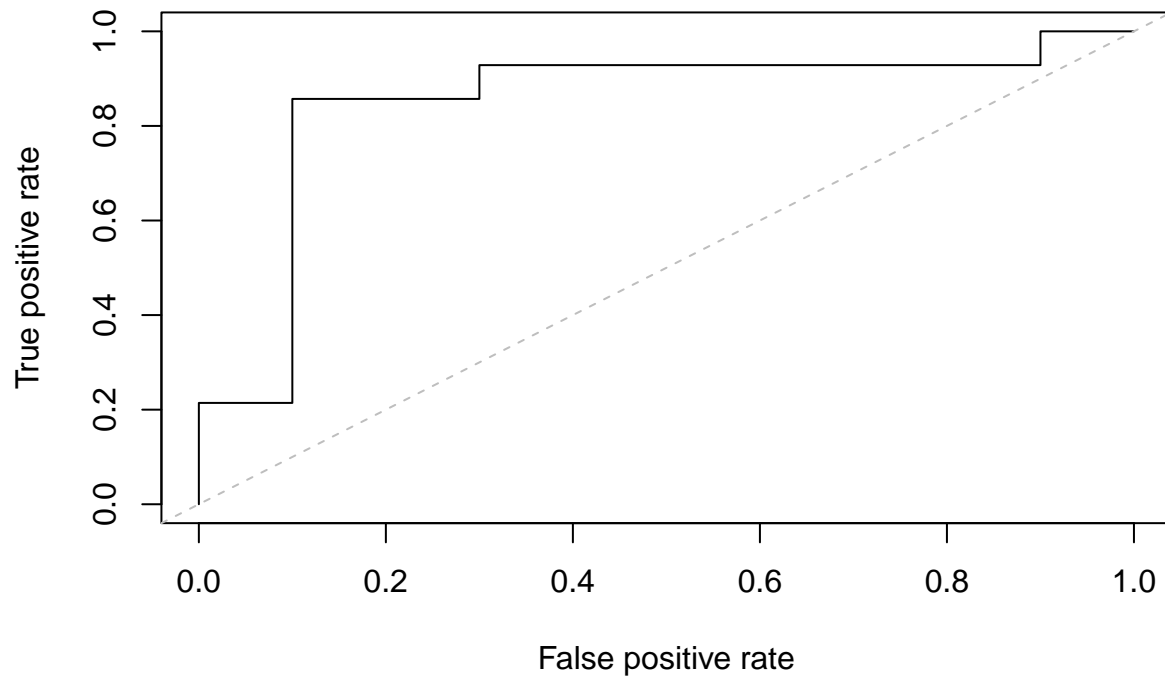


```
pred <- prediction(preds_lasso, c(rep("sens", 10), rep("res", 14)),  
  label.ordering=c("sens", "res"))  
perf <- performance(pred, measure = "tpr", x.measure = "fpr")  
print(paste("AUC:", performance(pred, measure = "auc")@"y.values"[[1]]))
```

```
## [1] "AUC: 0.85"
```

```
plot(perf, main="(b)")  
abline(0, 1, col="grey", lty=2)
```

(b)



```
## Adapted from @Mehrad Mahmoudian:
myCoefs <- coef(best_lasso, s="lambda.min");

## Asseble into a data.frame
myResults <- data.frame(
  features = myCoefs@Dimnames[[1]][ which(myCoefs != 0 ) ], #intercept included
  coefs    = myCoefs          [ which(myCoefs != 0 ) ] #intercept included
)
head(myResults)
```

```
##      features      coefs
## 1 (Intercept) -11.766110134
## 2      ABCB1    0.358698209
## 3      ACTN4    0.004925228
## 4      AKAP6    0.165840382
## 5      ARID4A   0.146997534
## 6      BCAT2   -0.103480203
```

## Check section

```
length(doceic50s0rd)
```

```
## [1] 482
```

```
dim(t(homdata$train))
```



## [1] 482 8173