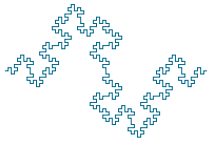
High-Throughput Sequencing Course

Receiver Operator Characteristic

Biostatistics and Bioinformatics



Summer 2018

Duke University
School of Medicine

Receiver Operator Characteristic

- A graphical tool that can be used to evaluate the performance of a a binary classifier of the form

$$g(m) = \begin{cases} 1 & M \geq m \\ 0 & M < m, \end{cases}$$

  where M is a quantitative marker
- Example: M could denote the the predicted probability of a "positive" response from your genomic classifier
- The ROC curve is a representation of sensitivity as a function of the specificity of the classifier
- The ROC contrasts sensivity and specificity of the classifier as the treshold is varied

Confusion Matrix: Terms

Among n samples

- P: The number of *positive* samples
- N: The number of *negative* samples
- TP: Number of *true-positives*
- TN: Number of *true-negatives*

Confusion Matrix: Relationship among terms

- $n = P + N$
- $P = TP + FN$
- $N = TN + FP$
- $n = TP + FP + TN + FN$

Confusion Matrix: Layout

		Prediction	
		Positive	Negative
Truth	Positive	TP	FN
	Negative	FN	TN

Sensitivity and Specificity

- Sensitivity is defined as the proportion of "positive" samples correctly predicted to be "positive"

$$\text{Sensitivity} = \frac{TP}{P}$$

- Specificity is defined as the proportion of "negative" samples correctly predicted to be "negative"

$$\text{Specificity} = \frac{TN}{N}$$

Receiver Operator Characteristics (ROC)

- Let M be the value of the marker
- In our case, M could be the predicted probability of a positive event from our classifier
- For each of the n samples we have (Y, M)
- For a given "cutoff" m let

$$\hat{Y} = g(m) = \begin{cases} 0 & M < m \\ 1 & M \geq m. \end{cases}$$

- In other words, "plug in" the observed value of the marker M into $g(m)$ to get either a positive or negative prediction
- ROC : A plot of Sensitivity versus 1-Specificity at a given cutoff m

Example

Table: Toy Example

		m=1		m=2	
M	Y	Yhat	Res	Yhat	Res
1.1	0	1	FP	0	TN
2.3	1	1	TP	0	FN
0.9	0	0	TN	0	TN
3.1	0	1	FP	1	FP
2.1	1	1	TP	0	FN
2.5	1	1	TP	1	TP
0.1	1	0	FN	0	FN

Note: the prediction $\hat{Y}$ does *not* require that the true state (Y) is known. The result (TP, TN, FP, FN) *does*.

Simulate data from logistic model

This function will simulate n samples from the following logistic model

$$\log \frac{P(Y=1)}{1-P(Y=1)} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

```
sim.binomial <- function(n, b0, b1, b2, b3) {
    x1 <- rnorm(n)
    x2 <- rnorm(n)
    x3 <- rnorm(n)
    lx <- b0 + b1 * x1 + b2 * x2 + b3 * x3
    px <- exp(lx)/(1 + exp(lx))
    y <- rbinom(n, 1, px)
    data.frame(y, px, x1, x2, x3)
}
# apply(replicate(1000,glm(y~x1+x2+x3,family='binomial',data=sim.binomial(100,0.1,0,0,0))$coef),1,mean)
```

Simulate Logistic Model

- The model has three features X_1, X_2 and X_3
- For each sample in the training set we have (Y, X_1, X_2, X_2) where $Y = 1$ (positive) or $Y = 0$ (negative)
- The model is trained on the basis of data from n samples
- The "trained" model is of the form

$$\log \frac{\hat{P}(Y=1)}{1-\hat{P}(Y=1)} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3$$

- For a "new" sample, the marker M on the basis of this trained model is

$$M = \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3)}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3)}$$

LOOCV function for Logistic Regression

```
logistic.loocv <- function(simdat) {
    preds <- foreach(i = 1:nrow(simdat), .combine = rbind) %do% {
        ## remove the i-th sample from the training data set
        traindati <- simdat[-i, ]
        ## Get the data for the left out sample
        testi <- simdat[i, , drop = FALSE]
        ## Fit a logistic model based on this data set
        trainmodi <- glm(y ~ x1 + x2 + x3, family = binomial, data = traindati)
        ## Predicted probability the left out sample against the trained model
        probhat <- predict(trainmodi, newdata = testi, type = "response")
        ## Predicted outcome: If the predicted probability is < 0.5 predict as 0 or 1
        ## otherwise
        yhat <- ifelse(probhat < 0.5, 0, 1)
        ## Output data.frame
        data.frame(y = testi$y, yhat = yhat, probhat = probhat)
    }
    return(preds)
}
```

Functions to calculate ROC and AUC using the ROCR package

```
### Helper function to produce ROC and AUC using the ROCR package The
### arguments are m: A positive marker y: The outcome (0 or 1)

ROC <- function(preddat, mlab = "probhat", ylab = "y") {
    m <- preddat[[mlab]]
    y <- preddat[[ylab]]
    performance(prediction(m, y), measure = "tpr", x.measure = "fpr")
}

AUC <- function(preddat, mlab = "probhat", ylab = "y") {
    m <- preddat[[mlab]]
    y <- preddat[[ylab]]
    performance(prediction(m, y), "auc")@y.values[[1]]
}
```

Function to calculate confusion matrix for a given cutoff

```
confusion <- function(cutoff, preddat, mlab = "probhat", ylab = "y") {
    m <- preddat[[mlab]]
    y <- preddat[[ylab]]
    yhat <- ifelse(m < cutoff, 0, 1)
    ### Calculate TN, FN, FP and TP
    TN <- sum(y == 0 & yhat == 0)
    FN <- sum(y == 1 & yhat == 0)
    FP <- sum(y == 0 & yhat == 1)
    TP <- sum(y == 1 & yhat == 1)
    ### Get P and N
    P <- TP + FN
    N <- TN + FP
    ### get sensitivity, specificity, and 1-spec
    sens <- TP/P
    spec <- TN/N
    FNR <- 1 - spec
    data.frame(cutoff, n = P + N, P, N, sens, spec, FNR)
}
```

ROC: Analysis

- Simulate data

```
set.seed(31219)
mydat <- sim.binomial(100, 0.1, 0, 1, 0)
```

- Perform LOOCV

```
mypreds <- logistic.loocv(mydat)
```

- Calculate ROC

```
myroc <- ROC(mypreds)
```

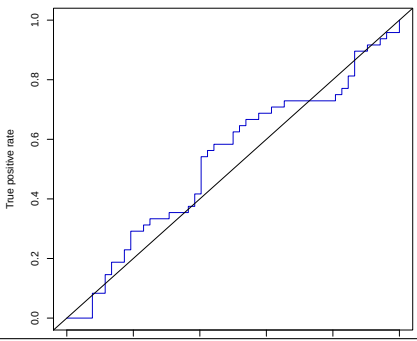
ROC: Plot

Get AUC

```
AUC(mypreds)
```

```
## [1] 0.5300481
```

```
plot(myroc, col = "blue3")
abline(0, 1)
```

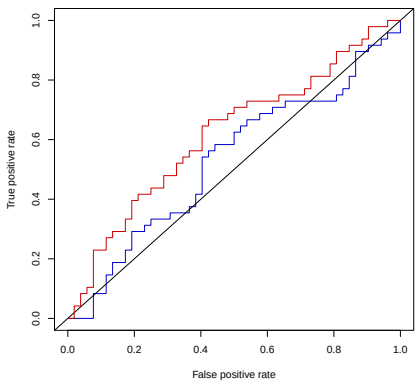


Add resubstition ROC

```
### Fit model using ALL n samples
trainmod <- glm(y ~ x1 + x2 + x3, family = binomial, data = mydat)
### Get the predicted probabilities
phat <- predict(trainmod, type = "response")
### Compare AUCs
AUC(mypreds)


## [1] 0.5300481


AUC(data.frame(probhat = phat, y = mydat$y))


## [1] 0.6169872
```

Show resubstition ROC

```
### Plot LOOCV ROC
plot(myroc, col = "blue3")
abline(0, 1)
### Add resub ROC curve
plot(ROC(data.frame(probhat = phat, y = mydat$y)), col = "red3", add = TRUE)
```



Look at the performance object

```
lookatperf <- function(perfobj, k) {
    data.frame(cutoff = perfobj@alpha.values[[1]][k], tpr = perfobj@y.values[[1]][k],
        fpr = perfobj@x.values[[1]][k])
}
### Look at some of the tpr and fpr
perf3 <- lookatperf(myroc, 8:10)
perf3


##      cutoff        tpr        fpr
## 1 0.6586603 0.06250000 0.07692308
## 2 0.6516634 0.08333333 0.07692308
## 3 0.6512297 0.08333333 0.09615385
```

Calculate using our own confusion function

```
perf3
```

```
##      cutoff        tpr        fpr
## 1 0.6586603 0.06250000 0.07692308
## 2 0.6516634 0.08333333 0.07692308
## 3 0.6512297 0.08333333 0.09615385
```

```
confusion(perf3$cutoff[1], mypreds)
```

```
##      cutoff   n  P  N   sens      spec        FNR
## 1 0.6586603 100 48 52 0.0625 0.9230769 0.07692308
```

```
confusion(perf3$cutoff[2], mypreds)
```

```
##      cutoff   n  P  N       sens      spec        FNR
## 1 0.6516634 100 48 52 0.08333333 0.9230769 0.07692308
```

```
confusion(perf3$cutoff[3], mypreds)
```

```
##      cutoff   n  P  N       sens      spec        FNR
## 1 0.6512297 100 48 52 0.08333333 0.9038462 0.09615385
```

ROC: Analysis with larger effect size

- Simulate data

```
set.seed(31219)
mydat <- sim.binomial(100, 0.1, 0, 2, 0)
```

- Perform LOOCV

```
mypreds <- logistic.loocv(mydat)
```

- Calculate ROC

```
myroc <- ROC(mypreds)
```

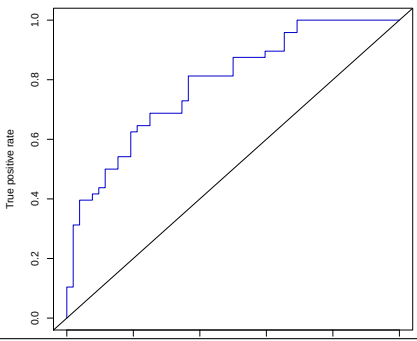
ROC: Plot

Get AUC

```
AUC(mypreds)
```

```
## [1] 0.7864583
```

```
plot(myroc, col = "blue3")
abline(0, 1)
```



Add resubstition ROC

```
### Fit model using ALL n samples
trainmod <- glm(y ~ x1 + x2 + x3, family = binomial, data = mydat)
### Get the predicted probabilities
phat <- predict(trainmod, type = "response")
### Compare AUCs
AUC(mypreds)


## [1] 0.7864583


AUC(data.frame(probhat = phat, y = mydat$y))


## [1] 0.8165064
```

Show resubstition ROC

```
### Plot LOOCV ROC
plot(myroc, col = "blue3")
abline(0, 1)
### Add resub ROC curve
plot(ROC(data.frame(probhat = phat, y = mydat$y)), col = "red3", add = TRUE)
```