**Chapter 10: ORH Analysis**

**Online Supplementary Material**

# Online Appendix 10.A: The DeLong method for estimating the covariance matrix

DeLong et al described a non-parametric method1 for estimating the covariance between two empirical AUCs that are based on the same cases (earlier, in book Chapter 7, a simpler result from that paper was used, namely estimating the variance of an empirical AUC). Define *structural components*:

 .

The kernel function  is defined by:

 .

Define the *IJ* x *IJ* matrix such that its  element is given by

 .

Similarly the *IJ* x *IJ* matrix  is defined such that its  element is given by

 .

In matrix notation, the covariance matrix  between  and  can be estimated using:

 .

Specifically,

 .

The R implementation of these formulae is in file VarCovMtrxDLStr.R. The Str is for "*structural components method*".

For completeness, and for possible research use, the implementation of Eqn. 4 ibid, *under the assumption that one can replace expected values with observed values*, is included in file VarCovMtrxDLEqn4.R. For large numbers of cases this gives almost identical results to those given by the other "standard" methods. According to Prof. Elizabeth DeLong (private communication, ca. 2010) one should use the structural components method, which is expected to be almost equivalent to the jackknife.

# Online Appendix 10.B: Estimation of covariance matrix: single-reader multiple-treatment

The following code computes the correlations for a single reader interpreting a common case-set in multiple treatments. The jackknife and bootstrap methods of computing  and  are implemented in mainVarCov1.R, a listing of which follows. The DeLong structural components method is implemented in function VarCovMtrxDLStr(), which is included in a .R file with the same name, sourced at line 9.

### Online Appendix 10.B.1: Code Listing

#mainVarCov1.R

rm(list = ls())

library(RJafroc)

source("VarCov1Bs.R")

source("Wilcoxon.R")

source("VarCov1Bs.R")

source("VarCov1Jk.R")

source("VarCovs.R")

source("VarCovMtrxDLStr.R")

seed <- 1;set.seed(seed)

fileName <- "CXRinvisible3-20mm.xlsx"

frocData <- DfReadDataFile(fileName, format = "JAFROC")

rocData <- DfFroc2Roc(frocData)

jSelect <- 1 # selects the reader to be analyzed

rocData1R <- DfExtractDataset(rocData, rdrs = jSelect)

zik1 <- rocData1R$NL[,1,,1];K <- dim(zik1)[2]

I <- dim(zik1)[1]

zik2 <- rocData1R$LL[,1,,1];K2 <- dim(zik2)[2]

K1 <- K-K2;zik1 <- zik1[,1:K1]

FOM <- array(dim=c(I))

for (i in 1:I) {

FOM[i] <- Wilcoxon(zik1[i,],zik2[i,])

}

FOMik <- array(dim = c(I, K))

for (i in 1:I) {

for (k in 1:K) {

if (k <= K1) {

FOMik[i,k] <- Wilcoxon(zik1[i,-k], zik2[i,])

} else {

FOMik[i,k] <-

Wilcoxon(zik1[i,], zik2[i,-(k-K1)])

}

}

}

ret1 <- VarCov1Jk(FOMik)

cat("data file = ", fileName, "\n")

cat("number of treatments = ", I,

"\nnumber of non-diseased cases = ", K1,

"\nnumber of diseased cases = ", K2, "\n")

cat("reader = ", jSelect, "\n")

cat("OR variance components using jackknife\n")

cat("Variance = ", ret1$Var,

"\nCov1 = ", ret1$Cov1,

"\nrho = ", ret1$Cov1/ret1$Var, "\n")

#to save the bs Auc values

B <- 2000;aucBs <- array(dim = c(I,B))

for (b in 1 : B){

# bs indices for non-diseased

k1b <- ceiling( runif(K1) \* K1 )

# bs indices for diseased

k2b <- ceiling( runif(K2) \* K2 )

for ( i in 1 : I) aucBs[i,b] <-

Wilcoxon(zik1[i,k1b], zik2[i,k2b])

}

ret2 <- VarCov1Bs(aucBs)

cat("OR variance components using bootstrap\n")

cat("Variance = ", ret2$Var, "\nCov1 = ",

ret2$Cov1, "\nrho = ",

ret2$Cov1/ret2$Var, "\n")

mtrxDLStr <- VarCovMtrxDLStr(rocData1R)

VarCovDLStr <- VarCovs(mtrxDLStr)

cat("OR variance components using DeLong method\n")

cat("Variance = ", VarCovDLStr$var,

"\nCov1 = ",

VarCovDLStr$cov1, "\nrho = ",

VarCovDLStr$cov1/VarCovDLStr$var, "\n")

Line 12 sets fileNamne to CXRinvisible3-20mm.xlsx, which is loaded at line 14 and converted to ROC data at line 15 using the highest rating on each case as its inferred ROC rating. Line 16 selects the reader to be analyzed, jSelect, currently set to reader 1, and the next line extracts the ratings for this reader in all treatments and saves it to a dataset named rocData1R. Line 19 extracts the ratings of non-diseased cases and line 21 that of diseased cases.

Insert a breakpoint (red dot) to the left of line number 24 (in the grey area of the window) and click Source. Look at the Environment window to confirm that zik1, representing , is an [1:4,1:52] array and zik2, representing , is an [1:4,1:106] array. Also confirm from the Environment panel that rocData contains 4 treatments and 5 readers (see lengths of 1st and 2nd indices of NL or LL lists) while rocData1R contains a single reader. Each z-sample is indexed by  where  is the treatment index with *I* = 4, and  is the truth index and  indexes cases in truth state *t*. The next 4 lines initialize the FOM[1:4] array using the Wilcoxon() function, i.e., empirical ROC- AUCs. Click on Next twice to enter the for-loop; to get out of the for-loop one could keep clicking on Next or click the green "*step out of function*" arrow; hovering over any icon reveals what it will do if clicked, in this case the hover message is "*Execute the remainder of the current function or loop*" and also shows a keyboard shortcut. Either way, the cursor advances to line 29.

Highlight FOM and click on Run. This yields the following values: 0.5696662, 0.5296626, 0.6358853 and 0.6249093. The first two values are close to chance-level performance as the nodules were invisible on 2-view chest x-rays (CXR) and were only included because they were visible on CT, which served as the gold standard. The treatments are, in order, 1:CXR, 2:CXR+DE, 3:TOMO and 4:TOMO+DE, where DE stands for dual energy and TOMO stands for chest-tomosynthesis2.

Lines 29-40 calculate the *jackknife FOM* values , not to be confused with *jackknife pseudovalues*, see first paragraph in book Section 9.6.2. Line 33 does this for non-diseased cases and line 35 for diseased cases. Notice that by allowing a negative array index R makes it particularly easy to remove a case. Click on Next twice and step out of the for-loops to get to line 40. Examine the structure of FOMik; it is a numeric array [1:4, 1:158]. Line 40 calculates  and  using the jackknife and save the results to ret1. The explanation of this code, contained in file VarCov1Jk.R, is in Appendix 10.B.5. For now simply execute it by clicking on Next. Examine the structure of ret1 and the ratio shown below.

### Online Appendix 10.B.2: Code snippet

Browse[2]> str(ret1)

List of 2

$ Var : num 0.00161

$ Cov1: num 0.000497

Browse[2]> ret1$Cov1/ret1$Var

[1] 0.3078498

The jackknife estimates of  and  are 0.00161 and 0.000497, respectively. As expected, covariance is less than variance and their ratio is the correlation . Lines 42 – 45 print out these values in a relatively "user-friendly" way. Keep clicking on Next until the cursor has advanced to line 52.

Line 52 sets the number of bootstraps B to 2000 and allocates memory for the bootstrap figure-of-merit values. The bootstrap method was explained in Chapter 7. Click on Next and step out of the for-loop to get to line 62. Execute this line (click Next), which calculates the bootstrap estimates of  and , using the code in file VarCov1Bs.R in Appendix 10.B.6, to get to line 63. Examine the structure of ret2.

### Online Appendix 10.B.3: Code snippet

Browse[2]> str(ret2)

List of 2

$ Var : num 0.00158

$ Cov1: num 0.000527

Browse[2]> ret2$Cov1/ret2$Var

[1] 0.3346733

The bootstrap estimates of  and  are 0.00158 and 0.000527, respectively. As expected, the covariance is smaller than the variance and their ratio is the correlation . Lines 64 – 66 print out these values.

The remaining lines (68 - 75) calculate and print the corresponding quantities using the DeLong method of structural components. Click Continue to execute the rest of the code.

Remove any break point, exit debug mode click source to obtain the following code output.

### Online Appendix 10.B.4: Code output

> source(...)

data file = CXRinvisible3-20mm.xlsx

number of treatments = 4

number of non-diseased cases = 52

number of diseased cases = 106

reader = 1

OR variance components using jackknife

Variance = 0.001614554

Cov1 = 0.0004970402

rho = 0.3078498

OR variance components using bootstrap

Variance = 0.001575106

Cov1 = 0.0005271459

rho = 0.3346733

OR variance components using DeLong method

Variance = 0.001600124

Cov1 = 0.0004926574

rho = 0.3078871

Note the close correspondence of the results using the three methods.

### Online Appendix 10.B.4: Calculation of Var and Cov1

Listed below is the code for VarCov1Jk.R for the function VarCov1Jk() which takes one argument, the jackknife FOM matrix JackFoMMatrix, with dimension [1:I,1:K] and returns a list variable containing Var and Cov1.

### Online Appendix 10.B.5: Code Listing VarCov1Jk

VarCov1Jk <- function (JackFoMMatrix)

{

I <- dim(JackFoMMatrix)[1]

K <- dim(JackFoMMatrix)[12

Cov <- array(dim = c(I, I))

for (i in 1:I){

for (ip in 1:I){

Cov[i, ip] <- cov(JackFoMMatrix[i,], JackFoMMatrix[ip,])

}

}

Var <- 0

count <- 0

for (i in 1:I){

Var <- Var + Cov[i, i]

count <- count + 1

}

Var <- Var / count

Var <- Var \* (K-1)^2/K

Cov1 <- 0

count <- 0

for (i in 1:I){

for (ip in 1:I){

if (ip != i){

Cov1 <- Cov1 + Cov[i, ip]

count <- count + 1

}

}

}

Cov1 <- Cov1 / count

Cov1 <- Cov1 \* (K-1)^2/K

return (list (

Var = Var,

Cov1 = Cov1

))

}

Lines 3 and 4 extract the dimensions of the jackknife FOM matrix. The action takes place in lines 7-11 where the variable ip denotes . The cov() function (supplied by R) takes as arguments JackFoMMatrix [i, ] , which stands for the *K*-length array  (*k* = 1, 2, ..., *K*) and JackFoMMatrix [ip, ] , which stands for the *K*-length array , and calculates the covariance, i.e., it implements the term inside the square brackets on the right hand side of book Equation 10.16. Confirm that the values are exactly those predicted by the term inside the square brackets on the right hand side of book Equation 10.16; this will exercise one's ability to write a simple function implementing the formula (for the faint hearted, there is a function in file CovarianceFirstPrinciples.R implementing this equation from scratch). Line 13-19 averages the variances over the different treatments (i.e., it averages over the diagonal terms of the matrix shown in the right hand side of book Equation 10.38, and similarly, line 23-32 averages the co-variances (i.e., it averages over all the off-diagonal terms); after application the variance inflation factor, two numbers Var and Cov1 are returned as a list variable.

Listed next is the code in VarCov1Bs.R for the function VarCov1Bs() which takes one argument, the bootstrap FOM matrix FomBs, with dimension [1:I,1:B] and returns a list variable containing Var and Cov1.

### Online Appendix 10.B.6: Code Listing VarCov1Bs

VarCov1Bs <- function (FomBs)

{

I <- dim(FomBs)[1]

Covariance <- array(dim = c(I, I))

for (i in 1:I){

for (ip in 1:I){

Covariance[i, ip] <- cov(FomBs[i, ], FomBs[ip, ])

}

}

#CovarianceFirstPrinciples(FomBs)

Var <- 0

count <- 0

for (i in 1:I){

Var <- Var + Covariance[i, i]

count <- count + 1

}

Var <- Var / count

Cov1 <- 0

count <- 0

for (i in 1:I){

for (ip in 1:I){

if (ip != i){

Cov1 <- Cov1 + Covariance[i, ip]

count <- count + 1

}

}

}

Cov1 <- Cov1 / count

return (list (

Var = Var,

Cov1 = Cov1

))

}

The action takes place in lines 6-10. The cov() function takes as arguments the *B*-length array FomBs[i,] , which stands for , and the *B*-length array FomBs[ip,] , which stands for , and calculates the covariance. Line 12-18 averages the variances over the different treatments (i.e., it averages over all diagonal terms), and similarly, line 20-30 averages the co-variances (i.e., it averages over all off-diagonal terms); this yields the two numbers Var and Cov1, which are returned as a list variable. Note the absence of the variance inflation term in the bootstrap method.



# Online Appendix 10.C: Comparing DBMH and ORH methods for single-reader multiple-treatment

A listing of the file mainOrDbmh1R follows:

### Online Appendix 10.C.1: Code listing

rm(list = ls()) #mainOrDbmh1R.R

library(RJafroc)

source("Wilcoxon.R")

source("VarCov1Jk.R")

fileName <- "CXRinvisible3-20mm.xlsx"

frocData <- DfReadDataFile(fileName, format = "JAFROC")

rocData <- DfFroc2Roc(frocData)

jSelect <- 1

rocData <- DfExtractDataset(rocData, rdrs = jSelect)

cat("data file = ", fileName, "\n")

cat("selected reader = ", jSelect, "\n")

seed <- 1; set.seed(seed)

ret1 <- StSignificanceTesting(

rocData,

FOM = "Wilcoxon",

method = "DBMH",

option = "FRRC")

cat("DBMH: F-stat = ",

ret1$fFRRC,

"\nddf = ", ret1$ddfFRRC,

"\nP-val = ", ret1$pFRRC,"\n")

ret2 <- StSignificanceTesting(

rocData,

FOM = "Wilcoxon",

method = "ORH",

option = "FRRC")

cat("ORH (Jackknife): F-stat = ",

ret2$fFRRC, "\nddf = ",

ret2$ddfFRRC, "\nP-val = ",

ret2$pFRRC,"\n")

ret3 <- StSignificanceTesting(

rocData,

FOM = "Wilcoxon",

method = "ORH",

option = "FRRC",

covEstMethod = "DeLong")

cat("ORH (DeLong): F-stat = ",

ret3$fFRRC,

"\nddf = ", ret3$ddfFRRC, "\nP-val = ",

ret3$pFRRC,"\n")

ret4 <- StSignificanceTesting(

rocData,

FOM = "Wilcoxon",

method = "ORH",

option = "FRRC",

covEstMethod = "Bootstrap")

cat("ORH (Bootstrap): F-stat = ",

ret4$fFRRC, "\nddf = ",

ret4$ddfFRRC, "\nP-val = ",

ret4$pFRRC,"\n")

Line 5 – 7 reads a FROC dataset and converts it to an ROC dataset rocData. Line 8 selects the first reader for the single-reader multiple treatment analysis; the resulting dataset is called rocData1R. Insert a break point at line 14 and source the code. Examine the structure of rocData and rocData1R in the Environment panel. The following code snippet shows that the single reader data has four treatments, 106 diseased cases and 52 non-diseased cases.

### Online Appendix 10.C.2: Code snippet

Browse[2]> str(rocData$NL)

num [1:4, 1:5, 1:158, 1] 0 4 0 4 5 5 0 5 0 0 ...

Browse[2]> str(rocData$LL)

num [1:4, 1:5, 1:106, 1] 3 3 4 0 0 0 4 4 0 0 ...

Browse[2]> str(rocData1R$NL)

num [1:4, 1, 1:158, 1] 0 4 0 4 0 0 0 0 0 0 ...

Browse[2]> str(rocData1R$LL)

num [1:4, 1, 1:106, 1] 3 3 4 0 4 0 0 0 2 2 ...

Line 14 – 18 performs DBMH analysis using the Wilcoxon figure of merit, saving the results to ret1. Note the options "FRRC" and method = "DBMH": with a single reader one can only perform fixed-reader analysis. The results are printed out in lines 19 – 22.

Line 24 – 32 perform similar operations, saving the results to ret2, except this time the analysis method is "ORH" and the covariance estimation is the jackknife, which being the default, does not need to be explicitly specified.

Line 34 – 43 perform similar operations, saving the results to ret3, except this time the analysis method is "ORH" and the covariance estimation uses the DeLong method, which needs to be explicitly specified: covEstMethod = "DeLong".

Line 45 – 54 perform ORH analysis, saving the results to ret4, using the bootstrap as the covariance estimation method: covEstMethod = "Bootstrap".

Remove any break points, exit debug mode and source the code, yielding the following output.

### Online Appendix 10.C.2: Code output

> source(...)

data file = CXRinvisible3-20mm.xlsx

selected reader = 1

DBMH: F-stat = 2.200775

ddf = 471

P-val = 0.0871945

ORH (Jackknife): F-stat = 2.200775

ddf = Inf

P-val = 0.08571326

ORH (DeLong): F-stat = 2.220742

ddf = Inf

P-val = 0.08347962

ORH (Bootstrap): F-stat = 2.180297

ddf = Inf

P-val = 0.08806371

Note the close correspondence between the four methods of analyzing the data. The author leaves it as an exercise for the reader to calculate and print confidence intervals for individual treatment FOMs and confidence intervals for inter-treatment FOM differences. The values can be confirmed by running the ORH significance testing procedure coded in RJafroc.

# Online Appendix 10.D: Minimal Implementation of ORH method

A minimal version of ORH analysis, but which shows all of the steps, is implemented in file mainORH.R listed below (the RJafroc package has the full implementation with more detailed output):

### Online Appendix 10.D.1: Code Listing

rm(list = ls()) #mainORH.R

library(RJafroc);library(ggplot2)

source("VarCovMtrxJK.R")

source("VarCovs.R")

source("Wilcoxon.R")

alpha <- 0.05

options(digits = 4)

cat("alpha =", alpha,"\n")

ROC <- FALSE

fileName <- "CXRinvisible3-20mm.xlsx"

frocData <- DfReadDataFile(

fileName,

format = "JAFROC",

renumber = "TRUE")

rocData <- DfFroc2Roc(frocData)

zijk1 <- rocData$NL[,,,1]

K <- dim(zijk1)[3];I <- dim(zijk1)[1]

J <- dim(zijk1)[2]

zijk2 <- rocData$LL[,,,1]

K2 <- dim(zijk2)[3]

K1 <- K-K2

zijk1 <- zijk1[,,1:K1]

cat("data file = ", fileName, "\n")

cat("number of treatments = ", I,

"\nnumber of readers = ", J,

"\nnumber of non-diseased cases = ", K1,

"\nnumber of diseased cases = = ", K2, "\n")

FOM <- UtilFigureOfMerit(

rocData, FOM = "Wilcoxon")

mtrxJK <- VarCovMtrxJK(rocData)

VarCovJK <- VarCovs(mtrxJK)

Var <- VarCovJK$var

Cov1 <- VarCovJK$cov1

Cov2 <- VarCovJK$cov2

Cov3 <- VarCovJK$cov3

msT <- 0

for (i in 1:I){

msT <- msT + (mean(FOM[i,]) - mean(FOM))^2

}

msT <- msT \* J / (I - 1)

msTR <- 0

for (i in 1:I){

for (j in 1:J) {

msTR <- msTR +

(FOM[i,j] - mean(FOM[i,]) - mean(FOM[,j]) + mean(FOM))^2

}

}

msTR <- msTR / ((I - 1)\*(J - 1))

cat("\nRandom reader random case analysis\n")

MS\_DEN\_DIFF\_FOM\_RRRC <- (msTR+max(J\*(Cov2-Cov3),0))

F\_ORH <- msT / MS\_DEN\_DIFF\_FOM\_RRRC

ndf <- (I-1)

ddfH <- MS\_DEN\_DIFF\_FOM\_RRRC^2/(msTR^2/((I-1)\*(J-1)))

cat("Hillis ddfH = ", ddfH, "\n")

FCrit <- qf(1 - alpha, ndf, ddfH)

cat("F statistic is ", F\_ORH, "and critical value of F is ", FCrit, "\n")

pValue <- 1 - pf(F\_ORH, ndf, ddfH)

cat("pvalue = ", pValue, "\n")

trtMeans <- array(dim = I)

for (i in 1:I) trtMeans[i] <- mean(FOM[i,])

trtDiff <- array(dim = c(I,I))

trtStr <- array(dim = c(I,I))

for (i1 in 1:(I-1)) {

for (i2 in (i1+1):I) {

trtDiff[i1,i2] <- trtMeans[i1]- trtMeans[i2]

trtStr[i1,i2] <- gsub(", ", "", toString(c(i1,-i2)))

}

}

trtDiff <- trtDiff[!is.na(trtDiff)]

strDiff <- trtStr[!is.na(trtStr)]

std\_DIFF\_FOM\_RRRC <- sqrt(2\*MS\_DEN\_DIFF\_FOM\_RRRC/J)

nDiffs <- I\*(I-1)/2

CI\_DIFF\_FOM\_RRRC <- array(dim = c(nDiffs, 3))

for (i in 1 : nDiffs) {

CI\_DIFF\_FOM\_RRRC[i,1] <- trtDiff[i]

CI\_DIFF\_FOM\_RRRC[i,2] <-

qt(alpha/2,df = ddfH)\*std\_DIFF\_FOM\_RRRC + trtDiff[i]

CI\_DIFF\_FOM\_RRRC[i,3] <-

qt(1-alpha/2,df = ddfH)\*std\_DIFF\_FOM\_RRRC + trtDiff[i]

cat("For pairing", strDiff[i],

"\nmean diff is ", CI\_DIFF\_FOM\_RRRC[i,1],

"\nand 95% CI is ", CI\_DIFF\_FOM\_RRRC[i,2],

CI\_DIFF\_FOM\_RRRC[i,3], "\n")

}

cat("\nFixed reader random case analysis\n")

MS\_DEN\_DIFF\_FOM\_FRRC <- Var-Cov1+(J-1)\*max((Cov2-Cov3),0)

FDbmFR <- msT / MS\_DEN\_DIFF\_FOM\_FRRC

ndf <- (I-1)

ddf <- Inf

cat("ddf = ", ddf, "\n")

FCrit <- qf(1 - alpha, ndf, ddf)

cat("F statistic is ", FDbmFR,

"\nand critical value of F is ", FCrit, "\n")

pValue <- 1 - pf(FDbmFR, ndf, ddf);cat("p-value is ", pValue, "\n")

std\_DIFF\_FOM\_FRRC <- sqrt(2\*MS\_DEN\_DIFF\_FOM\_FRRC/J)

nDiffs <- I\*(I-1)/2

CI\_DIFF\_FOM\_FRRC <- array(dim = c(nDiffs, 3))

for (i in 1 : nDiffs) {

CI\_DIFF\_FOM\_FRRC[i,1] <- trtDiff[i]

CI\_DIFF\_FOM\_FRRC[i,2] <-

qt(alpha/2,df = ddf)\*std\_DIFF\_FOM\_FRRC + trtDiff[i]

CI\_DIFF\_FOM\_FRRC[i,3] <-

qt(1-alpha/2,df = ddf)\*std\_DIFF\_FOM\_FRRC + trtDiff[i]

cat("For pairing", strDiff[i],

"\nmean diff is ", CI\_DIFF\_FOM\_FRRC[i,1],

"\nand 95% CI is ", CI\_DIFF\_FOM\_FRRC[i,2],

CI\_DIFF\_FOM\_FRRC[i,3], "\n")

}

cat("\nRandom reader fixed case analysis\n")

MS\_DEN\_DIFF\_FOM\_RRFC <- msTR

FDbmFC <- msT / MS\_DEN\_DIFF\_FOM\_RRFC

ndf <- (I-1)

ddf <- (I-1)\*(J-1)

cat("ddf = ", ddf, "\n")

FCrit <- qf(1 - alpha, ndf, ddf)

cat("F statistic is ", FDbmFC,

"\nand critical value of F is ", FCrit, "\n")

pValue <- 1 - pf(FDbmFC, ndf, ddf)

cat("p-value is ", pValue, "\n")

std\_DIFF\_FOM\_RRFC <- sqrt(2\*MS\_DEN\_DIFF\_FOM\_RRFC/J)

nDiffs <- I\*(I-1)/2

CI\_DIFF\_FOM\_RRFC <- array(dim = c(nDiffs, 3))

for (i in 1 : nDiffs) {

CI\_DIFF\_FOM\_RRFC[i,1] <- trtDiff[i]

CI\_DIFF\_FOM\_RRFC[i,2] <-

qt(alpha/2,df = ddf)\*std\_DIFF\_FOM\_RRFC + trtDiff[i]

CI\_DIFF\_FOM\_RRFC[i,3] <-

qt(1-alpha/2,df = ddf)\*std\_DIFF\_FOM\_RRFC + trtDiff[i]

cat("For pairing", strDiff[i],

"\nmean diff is ", CI\_DIFF\_FOM\_RRFC[i,1], "\nand 95% CI is ",

CI\_DIFF\_FOM\_RRFC[i,2], CI\_DIFF\_FOM\_RRFC[i,3], "\n")

}

plotM <- list(1, 2, 3, 4)

plotR <- list(c(1:5), c(1:5), c(1:5), c(1:5))

plot <- PlotEmpiricalOperatingCharacteristics(

dataset = rocData, trts = plotM, rdrs = plotR)

p <- plot$Plot +

scale\_colour\_manual(values=c("black","grey", "blue", "darkblue")) +

theme(axis.title.y = element\_text(size = 25,face="bold"),

axis.title.x = element\_text(size = 30,face="bold"),

legend.position = c(0.5,0.05), legend.direction = "horizontal",

legend.text = element\_text(size = 15, face = "bold"),legend.key.size = unit(2, "lines")) +

scale\_x\_continuous(expand = c(0, 0)) +

scale\_y\_continuous(expand = c(0, 0))

p$layers[[1]]$aes\_params$size <- 2 # line

print(p)

Lines 1-15 are similar to the previous examples excepting this time interest is in all readers, not just one reader. The file CXRinvisible3-20mm.xlsx is read and converted to an ROC dataset object rocData. Lines 17 – 23 extract the numbers of treatments, readers, non-diseased and diseased cases and the corresponding ratings (zijk1 and zijk2). Use standard debugging techniques to be convinced that the ratings are in the range 0 to 5. The 0 rating comes from unmarked non-diseased cases and unmarked lesions.

Lines 31– 32 populates the matrix FOM[1:I,1:J] containing the empirical AUCs for all treatment-reader combinations. Insert a break point at line 34 and click Source. Highlight FOM and click Run.

### Online Appendix 10.D.2: Code snippet

Browse[2]> FOM

[,1] [,2] [,3] [,4] [,5]

[1,] 0.5696662 0.4783200 0.5463534 0.5448113 0.5522496

[2,] 0.5296626 0.4960994 0.5599601 0.5244013 0.5714804

[3,] 0.6358853 0.6369739 0.6703556 0.6422351 0.6413280

[4,] 0.6249093 0.6269049 0.6673621 0.5862663 0.6262700

The array has four rows, corresponding to the four treatments, and five columns, corresponding to the five readers. Line 34 uses the included function VarCovMtrxJK() to estimate the covariance matrix, using the jackknife. [The reader is encouraged to step into the function to see how it works – there is a descriptive icon in the debug window that needs to be clicked.] The result is saved to mtrxJK. The function VarCovs() called at line 35 performs the requisite averaging, returning the 4 parameters characterizing the covariance matrix, book Equations 10.38 and 10.39. The result is saved to VarCovJK. Click Next and examine the structure of mtrxJK.

### Online Appendix 10.D.3: Code snippet

Browse[2]> str(mtrxJK)

num [1:4, 1:4, 1:5, 1:5] 0.001442 0.000976 0.000297 0.00047 0.000976 ...

As expected, the covariance matrix is *IJ* x *IJ*, excepting that the indexing has been organized as *II* x *JJ* (there are 4 treatments and 5 readers). Click Next and examine the structure of VarCovJK.

### Online Appendix 10.D.4: Code snippet

Browse[2]> str(VarCovJK)

List of 4

$ var : num 0.00142

$ cov1: num 0.000434

$ cov2: num 0.000285

$ cov3: num 0.000139

The 4 distinct elements of the covariance matrix, after the averaging, are shown above. The ordering shown in book Equation 10.40 is obeyed.

Lines 41- 45 calculate MST and lines 47 – 54 calculate MSTR, implementing book Equation 10.43. Lines 56 – 65 implements random-reader random-case analysis and prints out the value of the F-statistic, the critical value of the F-statistic and the p-value. These result from a straightforward application of the relevant formulae.

Lines 67 – 78 are a little complicated, but the basic idea is to identify all possible treatment pairings and to assign string names to them for clarity of printed output. Exit debug mode, clear any breakpoint and insert a new break point at line 80 and click Source. Highlight trtDiff (for reader-averaged inter-treatment FOM difference) and click Run and repeat for strDiff (for "*helpful string identifying reader-averaged inter-treatment FOM difference*"):

### Online Appendix 10.D.5: Code snippet

Browse[2]> trtDiff

[1] 0.001959361 -0.107075472 -0.109034833 -0.088062409 -0.090021771 0.019013062

Browse[2]> strDiff

[1] "1-2" "1-3" "2-3" "1-4" "2-4" "3-4"

Lines 80 – 93 calculates and prints the reader-averaged differences in FOMs between different treatment pairings, with helpful strings indicating the specific pairing each difference applies to, and also calculates and prints the confidence interval. This completes random-reader random-case analyses.

Lines 95 – 119 repeats the process for fixed reader analysis.

Lines 121 – 145 repeats the process for fixed case analysis.

The remaining code displays the reader-averaged empirical ROC plots for the four treatments. Click Continue to execute the rest of the code. Exit debug mode, clear any existing breakpoints and click source.

### Online Appendix 10.D.6: Code output

> source(...)

alpha = 0.05

data file = CXRinvisible3-20mm.xlsx

number of treatments = 4

number of readers = 5

number of non-diseased cases = 52

number of diseased cases = = 106

Random reader random case analysis

Hillis ddfH = 70.52

F statistic is 13.3 and critical value of F is 2.735

pvalue = 5.645e-07

For pairing 1-2

mean diff is 0.001959

and 95% CI is -0.04244 0.04636

For pairing 1-3

mean diff is -0.1071

and 95% CI is -0.1515 -0.06267

For pairing 2-3

mean diff is -0.109

and 95% CI is -0.1534 -0.06463

For pairing 1-4

mean diff is -0.08806

and 95% CI is -0.1325 -0.04366

For pairing 2-4

mean diff is -0.09002

and 95% CI is -0.1344 -0.04562

For pairing 3-4

mean diff is 0.01901

and 95% CI is -0.02539 0.06342

Fixed reader random case analysis

ddf = Inf

F statistic is 10.54

and critical value of F is 2.605

p-value is 6.276e-07

For pairing 1-2

mean diff is 0.001959

and 95% CI is -0.04707 0.05099

For pairing 1-3

mean diff is -0.1071

and 95% CI is -0.1561 -0.05805

For pairing 2-3

mean diff is -0.109

and 95% CI is -0.1581 -0.06001

For pairing 1-4

mean diff is -0.08806

and 95% CI is -0.1371 -0.03903

For pairing 2-4

mean diff is -0.09002

and 95% CI is -0.139 -0.04099

For pairing 3-4

mean diff is 0.01901

and 95% CI is -0.03001 0.06804

Random reader fixed case analysis

ddf = 12

F statistic is 32.25

and critical value of F is 3.49

p-value is 5.035e-06

For pairing 1-2

mean diff is 0.001959

and 95% CI is -0.0292 0.03312

For pairing 1-3

mean diff is -0.1071

and 95% CI is -0.1382 -0.07592

For pairing 2-3

mean diff is -0.109

and 95% CI is -0.1402 -0.07788

For pairing 1-4

mean diff is -0.08806

and 95% CI is -0.1192 -0.0569

For pairing 2-4

mean diff is -0.09002

and 95% CI is -0.1212 -0.05886

For pairing 3-4

mean diff is 0.01901

and 95% CI is -0.01215 0.05017

All three p-values are highly significant. The significant differences are shown in red font. As an advanced exercise, the reader should read the original manuscript3 and try to correlate the highlighting shown above with the results in the manuscript.

# Online Appendix 10.E: Proof of Eqn. 10.64

The OR model is:

 .

The sample mean for treatment  is:

 .

The variance of the sample mean for treatment  is:

 .

The terms are uncorrelated. Thus, the covariance between pairs of terms is 0, and above equation can be written as:

 .

Since  is a fixed number and  is fixed effect, their variances are 0.  and  are also uncorrelated for  and the covariance between them is 0. Likewise, the covariance between  and  is also 0. The covariance between the error terms for different readers is included as follows:

 .

For a single treatment, the term  can be removed, yielding Eqn. 10.67, reproduced below.

 . .



# Online Appendix 10.F: Single-treatment multiple-reader analysis

This relates to book section 10.7. The problem is to compare the average performance of a group of readers interpreting a common set of cases in a single treatment against a specified value. The file mainSingleTreatment.R, listed below, demonstrates single treatment analysis.

### Online Appendix 10.F.1: Code Listing

rm(list = ls()) # mainSingleTreatment.R

library(RJafroc)

source("Wilcoxon.R")

source("CovJk.R")

source("CovDL.R")

source("SingleTreatmentAnalysis.R")

alpha <- 0.05

ROC <- FALSE

if (ROC) {

#fileName <- "Franken1.lrc"

fileName <- "VanDyke.lrc"

rocData <- DfReadDataFile(

fileName,

format = "MRMC")

} else {

fileName <- "CXRinvisible3-20mm.xlsx"

frocData <- DfReadDataFile(

fileName,

format = "JAFROC")

rocData <- DfFroc2Roc(frocData)

}

cat("data file = ", fileName, "\n")

i <- 1 # select the treatment to be analyzed

# extract the first treatment

rocData1T <-

DfExtractDataset(rocData, trts = i)

fomArray <-

UtilFigureOfMerit(

rocData1T, FOM = "Wilcoxon")

thetaDot <- mean(fomArray[i, ])

mu0 <- 0.583422;mu0 <- 0.583422#mu0 <- 0.6

ret <- SingleTreatmentAnalysis(

rocData1T,

mu0,

covEstMthd = "JK",

alpha = alpha)

cat("The NH is that thetaDot = mu0, where thetaDot= ",

thetaDot, "\nand mu0 = ", mu0,"\n")

cat("The mean FOM for the anal2zed treatment is:", thetaDot,"\n")

cat("The", 100 \* (1 - alpha), "% CI for the preceding value is:", "(", ret$ci[1], ",", ret$ci[2], ")\n")

cat("The t-statistic to test\nH0: (analyzed treatment = standard) is:", ret$tStat,

"\nand the and p-value is ", ret$pVal, "\n")

cat("The difference in rdr.avg minus standard = ",thetaDot - mu0, "\n")

cat("The", 100 \* (1 - alpha), "% CI of the preceding value is", "(", ret$ciDiff[1], ",", ret$ciDiff[2], ")\n")

Line 8 sets the ROC flag to FALSE, causing the FROC data in CXRinvisible3-20mm.xlsx to be loaded at lines 17 – 19. Line 20 converts the FROC data to ROC. The ROC dataset object is named rocData. Line 24 selects the first treatment to be selected, and line 26-27 extracts the data for this treatment. Line 28 – 30 calculates the FOM for each reader. Since there are 5 readers, this is an array of length 5. Insert a break point at line 31 and click source. Highlight fomArray and click Run.

Browse[2]> fomArray

Rdr - 1 Rdr - 2 Rdr - 3 Rdr - 4 Rdr - 5

Trt - 1 0.5696662 0.47832 0.5463534 0.5448113 0.5522496

Click Next three times and the cursor advances to line 33. Click on the "*Step into the current function call*" button. The code pointer advances to line 1 of SingleTreatmentAnalysis.R, listed below.

### Online Appendix 10.F.2: Code Listing

SingleTreatmentAnalysis <- function(rocData1T, mu0, covEstMthd = "JK", alpha = 0.05){

fomArray <- FigureOfMerit(rocData1T, "Wilcoxon")

J <- length(fomArray)

NL <- rocData1T$NL

LL <- rocData1T$LL

K <- dim(NL)[3]

K2 <- dim(LL)[3]

K1 <- K - K2

msR <- 0

thetaDot <- mean(fomArray)

for (j in 1:J){

msR <- msR + (fomArray[j] - thetaDot)^2

}

msR <- msR / (J - 1)

zijk1 <- NL[ , , 1:K1,

]

dim(zijk1) <- c(1, J, K1)

zijk2 <- LL

dim(zijk2) <- c(1, J, K2)

if (covEstMthd == "JK"){

cov2 <- CovJk(zijk1, zijk2)$Cov2

}else if (covEstMthd == "DL"){

cov2 <- CovDL(zijk1, zijk2)$Cov2

}

msSingle <- msR + max(J \* cov2, 0)

dfSingle <- msSingle^2 / (msR^2/(J - 1))

sigmaSingle <- sqrt(msSingle / J)

tStat <- (thetaDot - mu0)/sigmaSingle

pVal <- 2 \* pt(abs(tStat), dfSingle, lower.tail = FALSE)

halfCIWidth <- qt(alpha/2, dfSingle, lower.tail = FALSE) \* sigmaSingle

ci <- c(thetaDot - halfCIWidth, thetaDot + halfCIWidth)

ciDiff <- ci - mu0

return(list(

ci = ci,

tStat = tStat,

pVal = pVal,

ciDiff = ciDiff

))

}

Click Next repeatedly while watching the Environment panel. Line 9 – 14 calculates MSR according to book Equation 10.68. Lines 15 – 18 extracts the ratings and dimensions the arrays for consistency with the functions CovJk and CovDl, corresponding to the jackknife and DeLong methods for estimate the covariance matrix. Currently the jackknife method is selected. Exit debug mode and insert a breakpoint at line 25 and click source. This compiles the file in debug mode. Switch to mainSingleTreatment.R and place the cursor anywhere on line 33 and click Run. Notice that the cursor has advanced to line 25 in SingleTreatmentAnalysis.R. Highlight cov2 and click Run, revealing its value to be 0.0001900035. Line 25 implements the term in parenthesis in book Equation 10.69. The remaining lines implement the formulae in book section 10.7. When satisfied, exit debug mode, clear any breakpoints and source the main code in mainSingleTreatment.

### Online Appendix 10.F.2: Code output

> source(...)

data file = CXRinvisible3-20mm.xlsx

The NH is that thetaDot = mu0, where thetaDot= 0.5382801

and mu0 = 0.583422

The mean FOM for the anal2zed treatment is: 0.5382801

The 95 % CI for the preceding value is: ( 0.4931374 , 0.5834228 )

The t-statistic to test

H0: (analyzed treatment = standard) is: -2.166462

and the and p-value is 0.05000364

The difference in rdr.avg minus standard = -0.04514188

The 95 % CI of the preceding value is ( -0.0902846 , 8.361764e-07 )

The p-value is just over 0.05, so the difference between average reader performance, 0.5382801, and the specified value 0.583422 is not significant. Change mu0 to 0.6 at line 32 and click source.

### Online Appendix 10.F.2: Code output

> source(...)

data file = CXRinvisible3-20mm.xlsx

The NH is that thetaDot = mu0, where thetaDot= 0.5382801

and mu0 = 0.6

The mean FOM for the anal2zed treatment is: 0.5382801

The 95 % CI for the preceding value is: ( 0.4931374 , 0.5834228 )

The t-statistic to test

H0: (analyzed treatment = standard) is: -2.962078

and the and p-value is 0.01129818

The difference in rdr.avg minus standard = -0.06171988

The 95 % CI of the preceding value is ( -0.1068626 , -0.01657716 )

The p-value is 0.01129818, so the difference between average reader performance and the specified value 0.6 is significant.

# References

1. DeLong ER, DeLong DM, Clarke-Pearson DL. Comparing the Areas Under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach. *Biometrics.* 1988;44:837-845.

2. Dobbins JT, McAdams HP, Sabol JM, et al. Multi-Institutional Evaluation of Digital Tomosynthesis, Dual-Energy Radiography, and Conventional Chest Radiography for the Detection and Management of Pulmonary Nodules. Radiology. *Radiology.* 2016;000(000):(in press).

3. Dobbins III JT, McAdams HP, Sabol JM, et al. Multi-Institutional Evaluation of Digital Tomosynthesis, Dual-Energy Radiography, and Conventional Chest Radiography for the Detection and Management of Pulmonary Nodules. *Radiology.* 2016;282(1):236-250.