

# Chapter 14: Online Appendices

## Table of contents

---

Online Appendix 14.A: Proof of Equivalence theorem for wAFROC
Online Appendix 14.B: Understanding the AFROC and wAFROC
Online Appendix 14.C: Summary of FROC FOMs
Online Appendix 14.D: Numerical demonstrations of FOM vs. AUC equivalences

---

## Online Appendix 14.A: Proof of equivalence theorem for wAFROC

---

We prove the equivalence between the area under wAFROC plot and the definition of the wAFROC FOM.

With reference to Fig. 14.1, the operating point labeled  $i$  has coordinates  $(FPF_i, wLLF_i)$  given by Eqn. (13.11) and Eqn. (13.17), respectively, reproduced here for convenience:

$$\begin{aligned}
 FPF_i &\equiv FPF(\zeta_i) = \frac{1}{K_1} \sum_{k_1=1}^{K_1} I\left(\max_{l_1} (z_{k_1 l_1}) \geq \zeta_i\right) \\
 wLLF_i &\equiv wLLF(\zeta_i) = \frac{1}{K_2} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_{k_2}} W_{k_2 l_2} I\left(z_{k_2 2 l_2} \geq \zeta_i\right)
 \end{aligned} \quad . \quad (14.99.1)$$

The area  $A_i$  of the leftmost shaded trapezoid in Fig. 14.1 is:

$$A_i = \frac{(FPF_i - FPF_{i+1})(wLLF_i + wLLF_{i+1})}{2} \quad . \quad (14.99.2)$$

Then

$$A_i = \frac{(FPF_i - FPF_{i+1})}{K_2} \frac{1}{2} \left[ \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_{k_2}} W_{k_2 l_2} I\left(z_{k_2 2 l_2} \geq \zeta_i\right) + \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_{k_2}} W_{k_2 l_2} I\left(z_{k_2 2 l_2} \geq \zeta_{i+1}\right) \right] \quad . \quad (14.99.3)$$

Using the probabilistic relation:

$$I\left(z_{k_2 2 l_2} \geq \zeta_i\right) = I\left(z_{k_2 2 l_2} \geq \zeta_{i+1}\right) + I\left(\zeta_i \leq z_{k_2 2 l_2} < \zeta_{i+1}\right) \quad . \quad (14.99.4)$$

we can expand the 1<sup>st</sup> term inside the square bracket:

$$A_i = \frac{(FPF_i - FPF_{i+1})}{K_2} \frac{1}{2} \left[ \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_{k_2}} W_{k_2 l_2} I\left(\zeta_i \leq z_{k_2 2 l_2} < \zeta_{i+1}\right) + \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_{k_2}} W_{k_2 l_2} I\left(z_{k_2 2 l_2} \geq \zeta_{i+1}\right) + \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_{k_2}} W_{k_2 l_2} I\left(z_{k_2 2 l_2} \geq \zeta_i\right) \right] \quad . \quad (14.99.5)$$

The last two terms are equal, therefore:

$$A_i = \frac{(FPF_i - FPF_{i+1})}{K_2} \left[ \frac{1}{2} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_{k_2}} W_{k_2 l_2} I(\zeta_i \leq z_{k_2 2 l_2} < \zeta_{i+1}) + \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_{k_2}} W_{k_2 l_2} I(z_{k_2 2 l_2} \geq \zeta_{i+1}) \right] \quad (14.99.6)$$

The final steps of the proof require that the z-samples be converted to integer ratings, which can be done without loss of ordering information if the number of bins is sufficiently large. Let  $r_{k_i l_i s}$  denote the integer rating of mark  $k_i l_i s$ , which implies that marks with z-samples satisfying  $\zeta_i \leq z_{k_i l_i s} < \zeta_{i+1}$ ,  $i = 0, 1, \dots, R$ , are rated  $r_{k_i l_i s} = i$  (binning rule). The zero rating corresponds to unmarked regions. The dummy thresholds  $\zeta_0$  and  $\zeta_{R+1}$  are defined as  $-\infty$  and  $+\infty$ , respectively. From Eqn. (15) it follows that

$$\begin{aligned} FPF_i - FPF_{i+1} &= \frac{1}{K_1} \sum_{k_1=1}^{K_1} \left[ I\left(\max_{l_1} (z_{k_1 1 l_1}) \geq \zeta_i\right) - I\left(\max_{l_1} (z_{k_1 1 l_1}) \geq \zeta_{i+1}\right) \right] \\ &= \frac{1}{K_1} \sum_{k_1=1}^{K_1} I\left(\zeta_i \leq \max_{l_1} (z_{k_1 1 l_1}) < \zeta_{i+1}\right) \end{aligned} \quad (14.99.7)$$

Because of the binning rule,  $I(\zeta_i \leq \max_{l_1} (z_{k_1 1 l_1}) < \zeta_{i+1})$  can be replaced by  $I(\max_{l_1} (r_{k_1 1 l_1}) = i)$ ,  $I(\zeta_i \leq z_{k_2 2 l_2} < \zeta_{i+1})$  can be replaced by  $I(r_{k_2 2 l_2} = i)$  and  $I(z_{k_2 2 l_2} \geq \zeta_{i+1})$  can be replaced by  $I(r_{k_2 2 l_2} > i)$ . Then Eqn. (16) can then be re-written as:

$$A_i = \frac{1}{K_1 K_2} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_{k_2}} W_{k_2 l_2} \left[ \frac{1}{2} I\left(\max_{l_1} (r_{k_1 1 l_1}) = i\right) I(r_{k_2 2 l_2} = i) + I\left(\max_{l_1} (r_{k_1 1 l_1}) = i\right) I(r_{k_2 2 l_2} > i) \right] \quad (14.99.8)$$

Summing over all values of  $i$ , one gets for the total area under the empirical wAFROC plot:

$$A_{wAFROC} = \frac{1}{K_1 K_2} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_{k_2}} W_{k_2 l_2} \left[ \frac{1}{2} I\left(r_{k_2 2 l_2} = \max_{l_1} (r_{k_1 1 l_1})\right) + I\left(r_{k_2 2 l_2} > \max_{l_1} (r_{k_1 1 l_1})\right) \right] \quad (14.99.9)$$

Explanation: Eqn. (18) follows from the property of the indicator function, which constrains  $i$  in the indicator functions inside the square bracket in Eqn. (17) to  $\max_{l_1} (r_{k_1 1 l_1})$ , where the functions are unity and otherwise they are zero. Using the definition of the Wilcoxon kernel function, Eqn. (3), it follows that:

$$A_{wAFROC} = \frac{1}{K_1 K_2} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_{k_2}} W_{k_2 l_2} \psi\left(\max_{l_1} (r_{k_1 1 l_1}), r_{k_2 2 l_2}\right) \quad (14.99.10)$$

■

### Online Appendix 14.A.1: Contribution of $i = 0$ term:

We prove a theorem that the contribution of the  $i = 0$  term in Eqn. TBA is identical to the area under the extension of the wAFROC from the uppermost non-trivial point to (1,1). According to Eqn. TBA,

$$A_0 = \frac{1}{K_1 K_2} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{N_{k_2}} W_{k_2 l_2} \left[ \frac{1}{2} I\left(\max_{l_1} (r_{k_1 l_1 1}) = 0\right) I\left(r_{k_2 2 l_2 2} = 0\right) + I\left(\max_{l_1} (r_{k_1 l_1 1}) = 0\right) I\left(r_{k_2 2 l_2 2} > 0\right) \right] \quad . \quad (14.99.11)$$

Rearranging the summations:

$$A_0 = \left[ \frac{1}{2} \frac{1}{K_1} \sum_{k_1=1}^{K_1} I\left(\max_{l_1} (r_{k_1 l_1 1}) = 0\right) \frac{1}{K_2} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{N_{k_2}} W_{k_2 l_2} I\left(r_{k_2 2 l_2 2} = 0\right) \right. \\ \left. + \frac{1}{K_1} \sum_{k_1=1}^{K_1} I\left(\max_{l_1} (r_{k_1 l_1 1}) = 0\right) \frac{1}{K_2} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{N_{k_2}} W_{k_2 l_2} I\left(r_{k_2 2 l_2 2} > 0\right) \right] \quad . \quad (14.99.12)$$

The following relation is easily shown:

$$\frac{1}{K_1} \sum_{k_1=1}^{K_1} I\left(\max_{l_1} (r_{k_1 l_1 1}) = 0\right) = 1 - FPF_1 \quad . \quad (14.99.13)$$

This is because the indicator function and the summation symbol counts the numbers of unmarked (zero rated) non-diseased cases and the division by  $K_1$  yields the corresponding contribution to  $FPF$ , which is the complement of the largest FPF value,  $FPF_1$ , obtained by cumulating all ratings 1 and above. Similarly,

$$\frac{1}{K_2} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{N_{k_2}} W_{k_2 l_2} I\left(r_{k_2 2 l_2 2} = 0\right) = 1 - wLLF_1 \quad . \quad (14.99.14)$$

Also,

$$\frac{1}{K_2} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{N_{k_2}} W_{k_2 l_2} I\left(r_{k_2 2 l_2 2} > 0\right) = wLLF_1 \quad . \quad (14.99.15)$$

Using these expressions, Eqn. TBA reduces to:

$$A_0 = \frac{1}{2} (1 - FPF_1) (1 - wLLF_1) + (1 - FPF_1) wLLF_1 \quad . \quad (14.99.16)$$

This expression can be simplified to:

$$A_0 = \frac{(1 - FPF_1)(1 + wLLF_1)}{2} \quad . \quad (14.99.17)$$

This is seen to be the area of a rectangle of base  $(1 - FPF_1)$  and height  $wLLF_1$  plus the area of a triangle of base  $(1 - FPF_1)$  and height  $(1 - wLLF_1)$ , in other words it is the area under the extension of the wAFROC from the uppermost non-trivial point  $(FPF_1, wLLF_1)$  to  $(1, 1)$ .

■

## Online Appendix 14.B: Understanding the AFROC and wAFROC

The code in file **mainExamples.R** simulates a dataset consisting of  $K_1 = 4$  non-diseased cases and  $K_2 = 4$  diseased cases. The numbers of lesions per diseased case **Lk2** are determined at line 5. The simulation parameters defined at line 6 are **mu** = 2, **lambda** = 1, **nu** = 1 and **zeta1** = -1. The first two diseased cases have one lesion each, and each of the remaining two have two lesions. It yields the plots shown in Fig. 14.3 and Fig. 14.5, and the values of the AFROC and wAFROC FOMs.

### Online Appendix 14.B.1: Code Listing

```
rm(list = ls()) # mainExamples.R
require(RJAfroc)
seed <- 1;set.seed(seed)
K1 <- 4;K2 <- 4
maxLL <- 2;Lk2 <- ceiling(runif(K2) * maxLL)
mu <- 2;lambda <- 1;nu <- 1;zeta1 <- -1
frocData <- FROCSimulator(mu = mu, lambda = lambda, nu = nu,
                          K1 = K1, K2 = K2, Lk2 = Lk2, zeta1 = zeta1)

FP <- apply(frocData$NL, 3, max);FP <- FP[1:K1]
frocData$lesionWeight[3, ] <- c(0.6, 0.4)
frocData$lesionWeight[4, ] <- c(0.4, 0.6)

plotAfroc <- EmpiricalOpCharac(frocData,trts = 1, rdrs = 1, opChType = "AFROC", lgdPos = "NULL")
print(plotAfroc$AFROCPlot)

plotwAfroc <- EmpiricalOpCharac(frocData,trts = 1, rdrs = 1, opChType = "wAFROC", lgdPos = "NULL")
print(plotwAfroc$wAFROCPlot)

cat("AFROC AUC = ", FigureOfMerit(frocData, fom = "AFROC"),"\n")
cat("wAFROC AUC = ", FigureOfMerit(frocData, fom = "wAFROC"),"\n")
```

Line 11 - 12 assigns the non-default values of the lesion weights (the default value for two lesions per case would be 0.5 to each lesion). The remaining lines plot the relevant curves. The reader should try different lesion weights, and perhaps increase the number of cases and compare the curves and AUC values.

## Online Appendix 14.C: Summary of FROC FOMs

Summarized in Table 14.A.1 are formulae for all AFROC-based empirical FOMs that have appeared in the literature. Note that the AFROC1 and wAFROC1 FOMs include comparisons between LL and NL z-samples on diseased cases, which are not allowed in the AFROC and wAFROC FOMs. Excepting for the ROC FOM, all AFROC-based FOMs are contained in the interval [0,1]. The ROC FOM is contained in the interval [0.5,1].

Table 14.A.1: Definitions of all currently implemented AFROC-based FOMs in JAFROC and **RJAfroc** software. Note that we do not use the prefix J with these FOMs, e.g., JAFROC  $\rightarrow$  AFROC, wJAFROC  $\rightarrow$  wAFROC, etc. A FOM name ending in 1 implies that *all* highest-rated NLs are used in computing the relevant statistic, including those on diseased cases.

FOM name	Definition	Comments
AFROC	$A_{AFROC} = \theta_{AFROC} = \frac{1}{K_1 \sum_{k_2=1}^{K_2} N_{k_2}} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{N_{k_2}} \psi \left( \max_{l_1} \left( z_{k_1 1 l_1 1} \right), z_{k_2 2 l_2 2} \right)$	
wAFROC	$A_{wAFROC} = \frac{1}{K_1 K_2} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{N_{k_2}} w_{k_2 l_2} \psi \left( \max_{l_1} \left( z_{k_1 1 l_1 1} \right), z_{k_2 2 l_2 2} \right)$	Recommended when one has both diseased and non-diseased cases
AFROC1	$A_{AFROC1} = \frac{1}{(K_1 + K_2) \sum_{k_2=1}^{K_2} N_{k_2}} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{N_{k_2}} \left[ \sum_{k_1=1}^{K_1} \psi \left( \max_{l_1} \left( z_{k_1 1 l_1 1} \right), z_{k_2 2 l_2 2} \right) + \sum_{k_2'=1}^{K_2} \psi \left( \max_{l_1} \left( z_{k_2' 2 l_1 1} \right), z_{k_2 2 l_2 2} \right) \right]$	
wAFROC1	$A_{wAFROC1} = \frac{1}{(K_1 + K_2) K_2} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{N_{k_2}} w_{k_2 l_2} \left[ \sum_{k_1=1}^{K_1} \psi \left( \max_{l_1} \left( z_{k_1 1 l_1 1} \right), z_{k_2 2 l_2 2} \right) + \sum_{k_2'=1}^{K_2} \psi \left( \max_{l_1} \left( z_{k_2' 2 l_1 1} \right), z_{k_2 2 l_2 2} \right) \right]$	Recommended when one does not have any non-diseased cases, i.e., $K_1 = 0$
Inferred ROC	$A_{iROC} = \frac{1}{K_1 K_2} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \psi \left( \max_{l_1} \left( z_{k_1 1 l_1 1} \right), \max_{l_1 l_2} \left( z_{k_2 2 l_1 1}, z_{k_2 2 l_2 2} \right) \right)$	
FROC	$A_{FROC} = \theta_{FROC} = \frac{1}{(K_1 + K_2) \sum_{k_2=1}^{K_2} L_{k_2}} \sum_{k_2=1}^{K_2} \sum_{l_2=1}^{L_{k_2}} \left[ \sum_{k_1=1}^{K_1} \sum_{l_1=1}^{N_{k_1 1}} \psi \left( z_{k_1 1 l_1 1}, z_{k_2 2 l_2 2} \right) + \sum_{k_2'=1}^{K_2} \sum_{l_2'=1}^{N_{k_2' 2}} \psi \left( z_{k_2' 2 l_2' 1}, z_{k_2 2 l_2 2} \right) \right]$	Not-recommended

## Online Appendix 14.D: Numerical demonstrations of FOM vs. AUC equivalences

The code in **mainFOMvsAUC.R** is shown below:

### Online Appendix 14.D.1: Code Listing

```
rm(list = ls()) # mainFOMvsAUC.R; demonstration of equivalences between FOMs and AUCs
library(RJAfroc)
library(caTools)

seed <- 1;set.seed(seed)
Lmax <- 2
K1 <- 7;K2 <- 8; Lk2 <- ceiling(runif(K2, 0, Lmax)) # this generates 1 to Lmax lesions per dis.
case
cat("K1 = ", K1, ", K2 = ", K2, "\n")
mu <- 1.5;lambda <- 0.8;nu <- 0.8 ;zeta1 <- -1
frocDataRaw <- FROCSimulator(
  mu = mu, lambda = lambda, nu = nu,
  K1 = K1, K2 = K2, Lk2 = Lk2, zeta1 = zeta1
)

# compare AfrociAUC vs. AfrociFOM
AfrocPlot <- EmpiricalOpCharac(frocDataRaw,1,1,opChType = "AFROC")
AfrocFOM <- signif(as.numeric(FigureOfMerit(frocDataRaw, fom = "AFROC")), digits = 8)
# trapz(x, y) function returns the numerical trapezoid integral defined by x and y,
# which can be used to calculate the trapezoidal area in our example. Use signif function
# to round the results to 8 significant digits
AfrocAUC <- signif(trapz(AfrocPlot$AFROCPoints$FPF, AfrocPlot$AFROCPoints$LLF), digits = 8)
print(AfrocAUC == AfrocFOM)
cat("AfrocFOM, AfrociAUC = ", AfrociAUC, "\n")

# compare wAfrocAUC vs. wAfrocFOM
wAfrocPlot <- EmpiricalOpCharac(frocDataRaw,1,1,opChType = "wAFROC")
wAfrocFOM <- signif(as.numeric(FigureOfMerit(frocDataRaw, fom = "wAFROC")), digits = 8)
wAfrocAUC <- signif(trapz(wAfrocPlot$wAFROCPoints$FPF, wAfrocPlot$wAFROCPoints$wLLF), digits = 8)
print(wAfrocAUC == wAfrocFOM)
cat("wAfrocFOM, wAfrocAUC = ", wAfrocAUC, "\n")

# compare Afroci1AUC vs. Afroci1FOM
NL <- frocDataRaw$NL
FP <- apply(NL, 3, max)
LL <- frocDataRaw$LL
maxNumLL <- max(frocDataRaw$lesionNum)
sumNumLL <- sum(frocDataRaw$lesionNum)
dim(LL) <- c(K2, maxNumLL)
LL <- LL[is.finite(frocDataRaw$lesionID)] # extract the ratings of available LLs (lesionID is not
-Inf).
uniqueRatings <- unique(c(FP, LL))
uniqueRatings <- uniqueRatings[order(uniqueRatings)] # rearrange the unique ratings with a
ascending order
# cut function bins the data by all neighbouring unique ratings (without losing information).
# table function calculates the number of cases in each bin (1 or 0).
# cumsum calculates the cumulative sum of cases at each unique ratings.
# Then divided by the number of cases or lesions. 0 is the origin.
FPF1 <- c(0, as.vector(cumsum(rev(table(cut(FP, c(uniqueRatings, Inf), right = FALSE)))) / (K1 +
K2)))
LLF1 <- c(0, as.vector(cumsum(rev(table(cut(LL, c(uniqueRatings, Inf), right = FALSE)))) /
sumNumLL))
Afroc1AUC <- signif(trapz(FPF1, LLF1), digits = 8)
Afroc1FOM <- signif(as.numeric(FigureOfMerit(frocDataRaw, fom = "AFROC1")), digits = 8)
print(Afroc1AUC == Afroc1FOM)
cat("Afroc1FOM, Afroci1AUC = ", Afroci1AUC, "\n")

# compare wAfroc1AUC vs. wAfroc1FOM
lesionWeights <- frocDataRaw$lesionWeight[is.finite(frocDataRaw$lesionID)] # extract the weights
of available LLs (lesionID is not -Inf).
wLLF1 <- rep(0, length(uniqueRatings))
for (l in 1:length(LL)){
  # calculate the weights of LLs in each bin (without losing information)
  wLLF1[(which(LL[l] == uniqueRatings))] <- wLLF1[(which(LL[l] == uniqueRatings))] +
lesionWeights[l]
}
# accumulate the weights and divide it by the number of cases to get wLLF1.
wLLF1 <- c(0, cumsum(rev(wLLF1))/K2)
wAfroc1AUC <- signif(trapz(FPF1, wLLF1), digits = 8)
wAfroc1FOM <- signif(as.numeric(FigureOfMerit(frocDataRaw, fom = "wAFROC1")), digits = 8)
print(wAfroc1AUC == wAfroc1FOM)
cat("wAfroc1FOM, wAfroc1AUC = ", wAfroc1AUC, "\n")
```

```
# compare frocAUC vs. frocFOM
NL <- frocDataRaw$NL
NL <- as.vector(NL[NL != -Inf])
frocFOM <- 0
for (l in 1:length(NL)){
  frocFOM <- frocFOM + sum(NL[l] < LL) + 0.5 * sum(NL[l] == LL)
}
frocFOM <- signif(frocFOM / (K1 + K2) / sumNumLL, digits = 8)
frocPlot <- EmpiricalOpCharac(frocDataRaw, 1, 1, opChType = "FROC")
NLF <- frocPlot$FROCPoints$NLF
LLF <- frocPlot$FROCPoints$LLF
frocAUC <- signif(trapz(NLF, LLF), digits = 8)
print(frocFOM == frocAUC)
cat("frocFOM, frocAUC = ", frocAUC, "\n")
```

**Source** the code, yielding the following output.

---

#### Online Appendix 14.D.2: Code Output 1

```
K1 = 7 , K2 = 8
[1] TRUE
AfrocFOM, AfrocAUC = 0.8021978
[1] TRUE
wAfrocFOM, wAfrocAUC = 0.7991071
[1] TRUE
Afroc1FOM, Afroc1AUC = 0.774359
[1] TRUE
wAfroc1FOM, wAfroc1AUC = 0.7729167
[1] TRUE
frocFOM, frocAUC = 0.4
```

The code simulates a small dataset with 7 non-diseased and 8 diseased cases. Each **TRUE** is the result of a logical comparison between a FOM and the corresponding AUC; the accuracy of the comparison is 8 significant digits. The common value is then printed out. For example,  $\theta_{AFROC} = A_{AFROC} = 0.8021978$ . The reader should experiment with the code to confirm that it always yields **TRUE** for the five listed comparisons.