

```
#####
#
# FpAUC index: FITTED PARTIAL AREA UNDER THE ROC CURVE
#   FOR HIGH SENSITIVITY INTERVAL (TPR0, 1)
#   The FpaucHS function computes the FpAUC index
#
# Notation:
#   xsample   = Gold standard
#   ysample   = test scores
#   lower.sen = tpr0 (the minimum sensitivity required)#
#####

#####
# CHECK THE GOLD STANDARD
#   The 1st is the negative group
#   The 2nd is the positive group
CheckGoldStandard <- function(xsample, positive=NULL) {
  xs=NULL; xs.levels=NULL;
  xs <- as.factor(xsample)
  xs.levels <- levels(xs)
  if (length(xs.levels) !=2 ) {
    stop("Error in the Gold Standard: it must be binary")
  }
  if (!is.null(positive)) {
    if (!any(xs.levels == positive)) {
      stop("Error in Gold Standard: positive case must be one of the levels")
    }
  } else {
    positive <- xs.levels[2]
  }
  if (xs.levels[2] != positive) {
    xs <- relevel(xs, ref=positive)
  }
  xs <- (xs == positive)*1
  return(xs)}
#####
# POINTS OF THE ROC CURVE
#   ROCpoints function calculates the coordinates (FPR, TPR)
ROCpoints <- function(xsample, ysample) {
  fpr.p=NULL; sen.p=NULL; xy.roc=NULL; x.p=NULL; y.p=NULL; pts=NULL;
  x.p <- xsample[which(is.na(xsample)==FALSE & is.na(ysample)==FALSE)]
  y.p <- ysample[which(is.na(xsample)==FALSE & is.na(ysample)==FALSE)]
  x.p <- CheckGoldStandard(x.p)
  pts <- sort(y.p)
  pts <- append(pts[-length(pts)]+diff(pts)/2, min(y.p)-1, 0)
  pts <- append(pts, max(y.p)+1, length(pts))
  for (i.pt in 1:length(pts)) {
    pre.p <- NULL
    pt=NULL
    pt <- pts[i.pt]
    pre.p <- (y.p > pt)*1
    fpr.p[i.pt] <- sum((pre.p == 1)*(x.p == 0))/sum(x.p == 0)
    sen.p[i.pt] <- sum((pre.p == 1)*(x.p == 1))/sum(x.p == 1)}
  if (is.unsorted(sen.p)) {
    sen.p <- rev(sen.p)
    fpr.p <- rev(fpr.p)}
  xy.roc <- cbind(fpr=fpr.p, tpr=sen.p)
  return(xy.roc)}
#####
# POINTS OF THE PARTIAL ROC CURVE FOR HIGH SENSITIVITY
#   pHSpoints function calculates the points in (tpr0, 1)

```

```

pHSpoints <- function(xsample, ysample, lower.sen) {
  pts.roc=NULL; fpr.roc=NULL; sen.roc=NULL; fpr.p=NULL; sen.p=NULL;
  ppoints=NULL;
  i.low=NULL; j.low=NULL; lscale=NULL;
  if ((lower.sen >= 1) || (lower.sen<0)) {stop("Error in the prefixed TPR
range")}
  pts.roc <- ROCpoints(xsample, ysample)
  fpr.roc <- pts.roc[,1]
  sen.roc <- pts.roc[,2]
  i.low <- min(which(sen.roc >= lower.sen))
  j.low <- max(i.low -1, 1)
  fpr.p <- fpr.roc[i.low:length(fpr.roc)]
  sen.p <- sen.roc[i.low:length(sen.roc)]
  if ((sen.roc[i.low] > lower.sen) && (i.low>1)) {
    sen.p <- append(sen.p, lower.sen, 0)
    lscale <- (sen.p[1]-sen.roc[j.low])/(sen.roc[i.low]-sen.roc[j.low])
    fpr.p <- append(fpr.p, fpr.roc[j.low]+(fpr.roc[i.low]-
fpr.roc[j.low])*lscale, 0)}
  ppoints=cbind(fprp=fpr.p, tprp=sen.p)
  return(ppoints)}
#####
# PARTIAL AREA UNDER THE ROC CURVE FOR HIGH SENSITIVITY
# pauchS function calculates the pAUC between (tpr0, 1)
pauchS <- function(xsample, ysample, lower.sen) {
  pts.proc=NULL; fpr.p=NULL; sen.p=NULL; esp.p=NULL; auc.p=NULL;
  pts.proc <- pHSpoints(xsample, ysample, lower.sen)
  fpr.p <- pts.proc[,1]
  sen.p <- pts.proc[,2]
  esp.p <- 1-fpr.p
  auc.p <- sum(diff(sen.p)*apply(cbind(esp.p[-1],esp.p[-length(esp.p)]), 1,
mean))
  return(auc.p)}
#####
# PARTIAL NEGATIVE LIKELIHOOD RATIO FOR HIGH SENSITIVITY
# pNLR_HS function calculates the NLR between (tpr0, 1)
pNLR_HS <- function(xsample, ysample, lower.sen) {
  pts.proc=NULL; fpr.p=NULL; sen.p=NULL; pnlr=NULL;
  pts.proc <- pHSpoints(xsample, ysample, lower.sen)
  fpr.p <- pts.proc[,1]
  sen.p <- pts.proc[,2]
  fpr.p <- fpr.p[-length(fpr.p)]
  sen.p <- sen.p[-length(sen.p)]
  pnlr <- (1-sen.p)/(1-fpr.p)
  return(pnlr)}
#####
# PARTIAL ROC CURVE SHAPE FOR HIGH SENSITIVITY
# shapepROC function classifies the ROC curve by its pNLR
shapepROC <- function(xsample, ysample, lower.sen) {
  nlr.p=NULL; nlr0=NULL; sproc=NULL;
  nlr.p <- pNLR_HS(xsample, ysample, lower.sen)
  nlr0 <- nlr.p[1]
  if ((all(nlr.p <= nlr0)) & (is.finite(nlr0))) {
    sproc <- "BpNLR"
  } else {
    if (all(nlr.p <= 1)) {
      sproc <- "pProp"
    } else {
      sproc <- "other"
    }
  }
  return(sproc)}

```

```
#####
# FITTED PARTIAL AREA INDEX FOR HIGH SENSITIVITY
# FpaucHS function calculates the FpAUC index in (tpr0, 1)
FpaucHS <- function(xsample, ysample, lower.sen) {
  fpr.p=NULL; sen.p=NULL; pauc.p=NULL; sproc.p=NULL; fpauc.p=NULL;
  fpr.p <- pHSpoints(xsample, ysample, lower.sen)[,1]
  sen.p <- pHSpoints(xsample, ysample, lower.sen)[,2]
  pauc.p <- paucHS(xsample, ysample, lower.sen)
  sproc.p <- shapepROC(xsample, ysample, lower.sen)
  fpauc.p <- switch(sproc.p,
                    BpNLR = pauc.p/((1-fpr.p[1])*(1-sen.p[1])),
                    pProp = (pauc.p+(1-sen.p[1])*(sen.p[1]-fpr.p[1]))/
((1+sen.p[1]-2*fpr.p[1])*(1-sen.p[1])),
                    other = (pauc.p+(1-sen.p[1])*(1-fpr.p[1]))/(2*(1-
fpr.p[1])*(1-sen.p[1]))
                    )
  return(fpauc.p)}
#####
```