```
ROCreg <-
function(method = c("NM", "SM","PROCGLM", "SROCGLM"),
   model,
   group,
   tag.healthy,
   marker = NULL,
   data,
   se.fit = FALSE,
   na.action = na.omit,
   control = controlROCreg())
{
### Bootstrap samples
bootstrap.sample <- function(data, group, method) {
if(method=="coutcome")
    res<-do.call("rbind", lapply(split(data,data[,group]), function(x)x[sample(nrow(x),
replace=TRUE),]))
  else
    res<-data[sample(nrow(data), replace=TRUE),]
   res
}
### Empirical survival function
emp.surv.function <- function(sequence, values)
sapply(sequence, function(s, values) mean(values >= s), values = values)

### Normal model
ROC.NM <- function(model, group, tag.healthy, marker, data, se.fit = FALSE, ROC.fit = NULL) {
model.h <- as.formula(paste(marker, paste(model[[1]], collapse= "")))
model.d <- as.formula(paste(marker, paste(model[[2]], collapse= "")))
data.h <- data[data[,group] == tag.healthy,]
data.d <- data[data[,group] != tag.healthy,]
```

1

2

2

3

```
fit.h <- lm(model.h, data = data.h)
fit.d <- lm(model.d, data = data.d)
```

ls fit to healthy and disease groups

```
sd.h <- sqrt(sum((fit.h$residuals)^2/fit.h$df.residual))
sd.d <- sqrt(sum((fit.d$residuals)^2/fit.d$df.residual))
```

} — s.e of the residuals

```
coeffs <- c(names(coefficients(fit.d)), names(coefficients(fit.h))
[is.na(match(names(coefficients(fit.h)),names(coefficients(fit.d))))])
beta.h <- beta.d <- rep(0, length(coeffs))
   names(beta.h) <- names(beta.d) <- coeffs
beta.h[match(names(coefficients(fit.h)), coeffs)] <- coefficients(fit.h)
beta.d[match(names(coefficients(fit.d)), coeffs)] <- coefficients(fit.d)

beta.ROC <- c((beta.d - beta.h)/sd.d, "h" = sd.h/sd.d)
```

← standardized data

```
if(se.fit) {
cov.h <- cov.d <- rep(0, length(coeffs))
  names(cov.h) <- names(cov.d) <- coeffs
cov.h[match(names(diag(vcov(fit.h))), coeffs)] <- diag(vcov(fit.h))
cov.d[match(names(diag(vcov(fit.d))), coeffs)] <- diag(vcov(fit.d))
M.hat<-(sd.d)/sqrt(cov.h + cov.d)

m.h<-model.matrix(fit.h)
m.d<-model.matrix(fit.d)
m.h<-ginv(t(m.h)%*%m.h)
m.d<-ginv(t(m.d)%*%m.d)
}
ROC.fit <- if(!se.fit)
list(terms = list(h = terms(fit.h), d = terms(fit.d)), beta=list(h = coefficients(fit.h), d =
coefficients(fit.d)), sd = list(h=sd.h, d=sd.d), beta.ROC = beta.ROC)
else
list(terms = list(h = terms(fit.h), d = terms(fit.d)), beta=list(h = coefficients(fit.h), d =
coefficients(fit.d)), sd = list(h=sd.h, d=sd.d), beta.ROC = beta.ROC, M=M.hat, AUCic =
list(h=m.h,d=m.d),df=list(h=fit.h$df.residual,d=fit.d$df.residual))

}
```

3

```
#### Semipar model
ROC.SM <- function(model, group, tag.healthy, marker, data, se.fit = FALSE, ROC.fit = NULL) {
```

4

```
model.h <- as.formula(paste(marker, paste(model[[1]], collapse= "")))
model.d <- as.formula(paste(marker, paste(model[[2]], collapse= "")))
data.h <- data[data[,group] == tag.healthy,]
data.d <- data[data[,group] != tag.healthy,]

if(!se.fit) {
```

optional ls fit

```
      fit.h <- lm(model.h, data = data.h)
   fit.d <- lm(model.d, data = data.d)
}  else {
      data.h.mf <- as.data.frame(cbind(marker = data.h[, marker], model.matrix(ROC.fit$terms$h,
model.frame(ROC.fit$terms$h, data.h))))
      data.d.mf <- as.data.frame(cbind(marker = data.d[, marker], model.matrix(ROC.fit$terms$d,
model.frame(ROC.fit$terms$d, data.d))))
     names(data.h.mf)[1] <- names(data.d.mf)[1] <- marker
   fit.h <- lm(as.formula(paste(marker, "~.-1", sep = "")), data = data.h.mf)
   fit.d <- lm(as.formula(paste(marker, "~.-1", sep = "")), data = data.d.mf)
}
sd.h <- sqrt(sum((fit.h$residuals)^2/fit.h$df.residual))
sd.d <- sqrt(sum((fit.d$residuals)^2/fit.d$df.residual))
coeffs <- c(names(coefficients(fit.d)), names(coefficients(fit.h))
[is.na(match(names(coefficients(fit.h)),names(coefficients(fit.d))))])
beta.h <- beta.d <- rep(0, length(coeffs))
   names(beta.h) <- names(beta.d) <- coeffs
beta.h[match(names(coefficients(fit.h)), coeffs)] <- coefficients(fit.h)
beta.d[match(names(coefficients(fit.d)), coeffs)] <- coefficients(fit.d)

beta.ROC <- c((beta.h - beta.d)/sd.d, "h" = sd.h/sd.d)
```

← standardized as before!

```
ROC.fit <- if(!se.fit)
```

```
list(terms = list(h = terms(fit.h), d = terms(fit.d)), beta=list(h = coefficients(fit.h), d =
coefficients(fit.d)), sd = list(h=sd.h, d=sd.d), residuals=list(h=fit.h$residuals,d=fit.d$residuals),
beta.ROC = beta.ROC)
else
list(beta.ROC = beta.ROC)

}
```

4

```
### Parametric ROC-GLM
ROC.PROCGLM <- function(model, group, tag.healthy, marker, data, names.cov, card.T,
ROC.model, FPFint, est.surv, se.fit = FALSE, ROC.fit = NULL) {
```

5

```
  data.h<- data[data[,group] == tag.healthy,]
data.d <-data[data[,group] != tag.healthy,]
n.d<-dim(data.d)[1]

model.h <- as.formula(paste(marker, paste(model[[1]], collapse= "")))
fit.h <- lm(model.h, data = data.h)
```

fitting a model?

```
sd.h <- sqrt(sum((fit.h$residuals)^2/fit.h$df.residual))
   pre.placement.values<-(data.d[, marker]-predict(fit.h,newdata=data.d))/sd.h
```

The healthy data has been standardized

```
   if(est.surv =="empirical") {
placement.values<-emp.surv.function(pre.placement.values, values=(data.h[, marker]-fit.h$fit)/
sd.h)
```

estimated model

```
} else if(est.surv =="normal") {
   placement.values<-(1-pnorm(pre.placement.values))
```

binormal model

```
}
n.col <- length(names.cov) + 2
 data.d.wide<-data.frame(cbind(data.d[, c(group, marker,
names.cov)],t(matrix(rep(rbind(1:card.T/(card.T
+1)),n.d),nrow=card.T,ncol=n.d)),matrix(rep(placement.values,card.T),nrow=n.d,ncol=card.T)))
   names(data.d.wide)<-c(names(data.d.wide)[1:n.col], paste("fpr",
1:card.T,sep=""),paste("inv.f.h",1:card.T,sep=""))
   data.d.long<-reshape(data.d.wide,drop=group, varying=list(names(data.d.wide)[(1:card.T)+
n.col], names(data.d.wide)[((card.T+1):(2*card.T))+ n.col]),
v.names=c("fpr","inv.f.h"),timevar="fpr",direction="long")
   q.f <- switch(ROC.model, "binormal" = "qnorm", "logistic" = "qlogis")
data.d.long$h <- eval(parse(text = q.f))(data.d.long$fpr)
```

finding $\Phi^{-1}(t)$

```
data.d.long$uit <- as.numeric(data.d.long$fpr >= data.d.long$inv.f.h)
data.d.long <- data.d.long[order(data.d.long$id), ]

link.f <- switch(ROC.model, "binormal" = "probit", "logistic" = "logit" )
```

either logit or probit model!

```
if(!se.fit) {
  model.glm <- if(!is.null(FPFint)) as.formula(paste(paste("uit",paste(model[[2]], collapse =
""),"+"), paste(attributes(terms(as.formula(FPFint)))$term.labels,"h", sep="*")))
  else as.formula(paste("uit", paste(model[[2]], collapse = ""), "+ h"))
      fit.glm <- glm(model.glm, family = binomial(link = link.f), data = data.d.long)
      ROC.fit <- list(terms = terms(fit.glm), beta.ROC = coefficients(fit.glm), fit.h = fit.h)
    } else {
      model.glm <- as.formula("uit ~ .-1")
      fit.glm <- glm(model.glm, family = binomial(link = link.f), data = as.data.frame(cbind("uit" =
data.d.long$uit, model.matrix(ROC.fit$terms, model.frame(ROC.fit$terms, data.d.long)))))
```

```
      ROC.fit <- list(beta.ROC = coefficients(fit.glm))
    }
ROC.fit
}   5
### Semiparametric ROC-GLM
ROC.SROCGLM <- function(model, group, tag.healthy, marker, data, names.cov, set.h,
ROC.model, est.surv, se.fit = FALSE, ROC.fit = NULL) {   6
data.h <- data[data[,group] == tag.healthy,]
data.d <- data[data[,group] != tag.healthy,]
n.d <- dim(data.d)[1]

model.response <- as.formula(paste(marker, paste(model[[1]], collapse= "")))
   fit.h <- lm(model.response, data = data.h)
   sd.h <- sqrt(sum((fit.h$residuals)^2/fit.h$df.residual))
pre.placement.values <- (data.d[, marker] - predict(fit.h, newdata = data.d))/sd.h

if(est.surv =="empirical") {
placement.values<-emp.surv.function(pre.placement.values, values=(data.h[, marker]-fit.h$fit)/
sd.h)
```

some Issue as before !

```
} else if(est.surv =="normal") {
   placement.values<-(1-pnorm(pre.placement.values))
}

design.matrix.d.pre <- data.d[, names.cov, drop = FALSE]
names(design.matrix.d.pre) <- names.cov
if(!se.fit)
      design.matrix.d <- model.matrix(model[[2]], design.matrix.d.pre)
else
      design.matrix.d <- model.matrix(ROC.fit$terms,model.frame(ROC.fit$terms,
design.matrix.d.pre))
pairwise <- matrix(NA, nrow = n.d * (n.d - 1)/2, ncol = ncol(design.matrix.d))
for (i in 1:(n.d-1))
  for (j in (i+1):n.d)  {
    pairwise[(i-1)*n.d - (i-1)*i/2 + j - i, 1] <- as.numeric(pre.placement.values[i] >=
pre.placement.values[j])
    pairwise[(i-1)*n.d - (i-1)*i/2 + j - i, -1] <- design.matrix.d[i, -1] - design.matrix.d[j, -1]
    }
```

Comparison of placement values

difference in covariates

```
dimnames(pairwise)[[2]] <- c("u", dimnames(design.matrix.d)[[2]][-1])
link.f <- switch(ROC.model, "binormal" = "probit", "logistic" = "logit" )
if (ROC.model != "binormal")
    stop("Links different to probit are not implemented for Semiparametric ROC-GLM method")
else {
    fit <- glm(u ~ .-1, data = as.data.frame(pairwise), family = binomial(link = link.f))
  beta.est <- fit$coefficients*sqrt(2)
    }
if(!se.fit) {
  h <- sapply(1:length(set.h),function(s, a, data, offsetData, beta, link.f) {
  df <- data.frame(u = as.numeric(data <= a[s]))
  glm(u ~ offset(offsetData[, -1, drop = FALSE] %*% beta), data = df, family = binomial(link =
link.f), control = glm.control(maxit = 100))$coefficients[1]
  }, data = placement.values, offsetData = design.matrix.d, a = set.h, beta=beta.est, link.f =
link.f)
  ROC.fit <- list(terms = attr(eval(model.frame(model[[2]], design.matrix.d.pre)), "terms"), p.h =
set.h, h = h, beta.ROC = beta.est, fit.h=fit.h)
} else
      ROC.fit <- list(beta.ROC = beta.est)
ROC.fit
}

if(inherits(model, "formula")) model <- c(model)
if(length(model) == 1) model[[2]] <- model[[1]]
if(inherits(model, "character")) {
  m <- list()
    m[[1]] <- as.formula(model[[1]])
     m[[2]] <- as.formula(model[[2]])
     model = m
  }
  step.h <- control$step.h
  ROC.model <- control$ROC.model
  card.T <- if (method == "PROCGLM") control$card.T
 if (se.fit) nboot <- control$nboot
  set.h <- seq(0, 1, by = step.h)
  names.cov.m1 <- all.vars(model[[1]])
  names.cov.m2 <- all.vars(model[[2]])
  names.cov <- c(names.cov.m2, names.cov.m1[is.na(match(names.cov.m1,names.cov.m2))])
  ind.healthy <- data[,group] == tag.healthy
   data <- data[order(!ind.healthy),]
   data <- data[!is.na(data[,marker]), ]
   method <- match.arg(method)

  ROC.fit <- switch(method,
    "NM" = ROC.NM(model, group, tag.healthy, marker, data, se.fit=se.fit),
    "SM" = ROC.SM(model, group, tag.healthy, marker, data),
"PROCGLM" = ROC.PROCGLM(model, group, tag.healthy, marker, data, names.cov, card.T,
ROC.model, FPFint = control$FPFint, control$est.surv),
"SROCGLM" = ROC.SROCGLM(model, group, tag.healthy, marker, data, names.cov, set.h,
ROC.model, control$est.surv))

  if(se.fit & method !="NM") {
 beta.boot <- matrix(nrow = length(ROC.fit$beta.ROC), ncol = nboot, dimnames =
list(names(ROC.fit$beta.ROC), 1: nboot))
 for(i in 1: nboot) {
 data.boot <- bootstrap.sample(data, group, control$resample.m)
   ROC.boot.fit <- switch(method,
   "SM" = ROC.SM(model, group, tag.healthy, marker, data.boot, se.fit=se.fit, ROC.fit =
ROC.fit),
    "PROCGLM" = ROC.PROCGLM(model, group, tag.healthy, marker, data.boot, names.cov,
card.T, ROC.model, FPFint = control$FPFint, est.surv=control$est.surv, se.fit = se.fit, ROC.fit =
ROC.fit),
    "SROCGLM" = ROC.SROCGLM(model, group, tag.healthy, marker, data.boot, names.cov,
set.h, ROC.model, est.surv=control$est.surv, se.fit = se.fit, ROC.fit = ROC.fit))
    beta.boot[ , i]<- ROC.boot.fit$beta.ROC
}
  se.beta <- sqrt(apply(beta.boot, 1, var))
  ROC.fit$se.beta <- se.beta
  names(ROC.fit$se.beta) <- names(ROC.fit$beta.ROC)
}
   ROCreg <- list(call = match.call(), group = group, method = method, marker = marker, control
= control, names.cov = list(g = names.cov, m1 = names.cov.m1, m2 = names.cov.m2), data =
data, ROC.fit = ROC.fit, se.fit = se.fit)
   class(ROCreg) <- "ROCreg"
   ROCreg
}
```