Dynamic predictive probabilities to monitor rapid cystic fibrosis disease progression

Rhonda D. Szczesniak, Weiji Su, Cole Brokamp, Ruth H. Keogh, John P. Pestian, Michael Seid, Peter J. Diggle and John P. Clancy

Supplemental R code

The R script below is based on the analysis of the Cystic Fibrosis Foundation Patient Registry (CFFPR). Variables from the dataset are defined in the Introduction and in Table 1 of the article. The database is structured as one record per patient per clinical encounter and includes all clinical/demographic characteristics used in the analyses.

Libraries and data

```
#### R script for CFFPR application and thresholds ####


# load for models ---
library(lmenssp)
# load for bootstrapping ---
library(mvtnorm)
# load data, which was split based on development (data.develop)
# and validation (data.validation) ---
load("model_dataset.RData")

# define model formula ---
# X1 - X5 are spline basis functions
formula00_full <- fev ~  t+age+age2+age3+X1+X2+X3+X4+X5+
  Gender+MRSA+pa+SESlow+cfrd+bcohr+ F508_c+numPEyr+numVisityr+basefev+
  age*Gender+age*bcohr+age*MRSA+age*pa+age*cfrd+age*F508_c+age*SESlow
# fit model ---
develop.ibm <- lmenssp(formula = formula00_full, data = data.develop,
                       id = id, process = "ibm",
                       timeVar = t, silent = FALSE)
# filter random components based on fitted model---
# subject_develop: id’s for whom to carry out filtering
filter_3basefev <- filtered(formula0_full, data = data.develop, id=id,
   process = "ibm", timeVar=t,
   estimate=develop.ibm$estimates,
                            subj.id=subject_develop)

# define covariates from model structure for predictions ---
spn <-c("X1","X2","X3","X4","X5")
derv.pwli <- as.matrix(data.develop[, c("X1.1","X2.1","X3.1","X4.1","X5.1")])
interaction1 <-c("age.Gender","age.MRSA","age.pa", "age.cfrd", "age.SESlow",
"age.bcohr1","age.bcohr2","age.bcohr3","age.bcohr4",
"age.bcohr5","age.F508_c2","age.F508_c3",
"age.isOnEnzymes")
interaction2<-c("Gender","MRSA","pa","cfrd","SESlow","bcohr1","bcohr2",
"bcohr3","bcohr4","bcohr5","F508_c2","F508_c3","isOnEnzymes")

# calculate derivative ---
deriv <- develop.ibm$estimate["t", 1] + develop.ibm$estimate["age", 1] +
  2*data.develop$age*develop.ibm$estimate["age2", 1] +
  3*data.develop$age2*develop.ibm$estimate["age3", 1] +
  as.matrix(derv.pwli) %*%as.matrix(develop.ibm$estimate[spn, 1]) +
  as.matrix(data.develop[,interaction2])%*%
  as.matrix( develop.ibm$estimates[interaction1 ,1]) +
  filter_3basefev$b[, "mean"]
```

```
# define target function (example here with delta_c = -1.5% pred/yr) ---
threshold <- -1.5 -develop.ibm$estimate["t", 1] -
  develop.ibm$estimate["age", 1] -
  2*data.develop$age*develop.ibm$estimate["age2", 1] -
  3*data.develop$age2*develop.ibm$estimate["age3", 1] -
  as.matrix(derv.pwli) %*%as.matrix(develop.ibm$estimate[spn, 1]) -
  as.matrix(data.develop[,interaction2])%*%
  as.matrix(develop.ibm$estimates[interaction1 ,1])

# evaluate predictive probabilities ---
prob <- pnorm(threshold , filter_3basefev$b[, "mean"],
       sqrt(filter_3basefev$b[, "variance"]))

# acquire fitted values ---
varbname <- c( "t", "age", "age2", "age3", "X1","X2", "X3" , "X4","X5",
               "Gender", "MRSA", "pa","SESlow","cfrd","bcohr1","bcohr2",
               "bcohr3","bcohr4", "bcohr5","F508_c2","F508_c3","numPEyr",
               "numVisityr","basefev","age.Gender", "age.bcohr1",
               "age.bcohr2","age.bcohr3","age.bcohr4","age.bcohr5",
               "age.MRSA", "age.pa", "age.cfrd","age.F508_c2",
               "age.F508_c3","age.SESlow", "age.isOnEnzymes")
fitted.mean <- as.matrix(cbind(1,data.develop[,varbname]))%*%
  as.matrix( develop.ibm$estimate[c("Intercept",varbname), 1])

# calculate raw residuals ---
data.develop$raw.resid<-data.develop$fev - fitted.mean

# calculate standardized residuals ---
sd.residual <- NULL
sd.time<-
  #use MLEs
  omega2 <-
  sigma2 <-
  tau2 <- (develop.ibm$estimates["tausq",1])
id <- data.develop[!duplicated(data.develop$id),"id"]
for(i in 1:length(id)) {
  sub <- data.develop$id[data.develop$id==id[i]]
  time <- data.develop[data.develop$id==id[i], "age"]
  raw<- data.develop$raw.resid[data.develop$id==id[i]] m<-length(sub)
  J <- matrix(1,m,m)
  I <- diag(m)
  R <- matrix(nrow=m, ncol=m)
  for(j in 1:m){
    for(k in 1:m){
      R[j,k]<- 0.5*((min(time[j],time[k]))**2)*(max(time[j],time[k])-
                                            (min(time[j],time[k])/3))
      }
  }
  V<- omega2*J+sigma2*R+tau2*I
  S <- t(chol(V))
  dm.factor<- solve(S)%*%raw
  time.lag<- solve(S)%*%time
  sd.residual<-c(sd.residual ,dm.factor)
  sd.time <- c(sd.time ,time.lag)
}
```

```
# Figure 1: FEV1 against age, select subjects
# sub_ID_made includes selected subjects
plot(data.develop$age, data.develop$fev, col="lightgray",
     xlab="Age (years)", ylab="FEV1% predicted",
     ylim=range(data.develop$fev), xlim=range(data.develop$age))
for(i in 1:length(sub_ID_made)){
  lines(data.develop$age[data.develop$id%in%sub_ID_made[i]],
        data.develop$fev[data.develop$id%in%sub_ID_made[i]])
}

# Figure 2: diagnostics plots, standardized empirical residuals ---
par(mfrow=c(2,2))
# (a) residuals vs. fits ---
plot(fitted.mean , sd.residual , col="grey",xlab="Fitted values",
       ylab="Std residuals",type="p",pch=".")
abline(0,0,col="black")
lines(lowess(fitted.mean, sd.residual),col="black",lty=3, lwd = 2)
# (b) residuals vs. age ---
plot(data.develop$age, sd.residual, col="grey",xlab="Age (years)",
       ylab="Std residuals", type="p",pch=".")
abline(0,0,col="black")
lines(lowess(data.develop$age, sd.residual),lty=3, lwd = 2)
# (c) kernel density plot ---
hist(sd.residual, freq=F,xlab="Std residuals",ylab="Density",main=" ")
lines(density(rnorm(30000)))
#(d) Q-Q plot ---
qqnorm(sd.residual,main=" "); qqline(sd.residual)

#Figure 3: variograms
raw_variog_weekly <- variogram(resid=data.develop$raw.resid,
                        timeVar=data.develop$t,
                        id=data.develop$id, binwidth=0.019,numElems=20,
                        inc.var=T,irregular = TRUE)
par(mfrow=c(1,2))
# (a) raw empirical variogram
plot(raw_variog_weekly$bin.mids[1:669],raw_variog_weekly$bin.means[1:669],
        type="l",xlim=c(0,13),
     col="gray",xlab="Lag (years)",ylab="Variogram function (raw)",
         ylim=c(50,330))
lines(lowess(raw_variog_weekly$bin.mids[1:669],
raw_variog_weekly$bin.means[1:669]))
abline(h=raw_variog_weekly$process.var,lty=2)
abline(h=raw_variog_weekly$bin.means[669]-20,lty=2)
abline(h=raw_variog_weekly$bin.means[1]+10,lty=2)
# process variance: 320.61
error_var=(raw_variog_weekly$bin.means[1]+10)/raw_variog_weekly$process.var
within_var = ((raw_variog_weekly$bin.means[669]-20)/
raw_variog_weekly$process.var)-error_var
between_var= 1- within_var - error_var
# (b) std variogram
raw_variog_weekly.std <- variogram(resid=sd.residual,timeVar=sd.time,
id=data.develop$id, binwidth=binsize.CF.weekly,
numElems=30,inc.var=T,irregular = TRUE)
plot(raw_variog_weekly.std$bin.mids,raw_variog_weekly.std$bin.means,type="l",
     xlab="Lag (std time scale)",ylab="Variogram function (std)",
     xlim=c(0,0.65))
abline(h=1,lty=2)
```

```
# Figure 5: individual patient predictions for
# delta_c = -1.5% pred/yr (2 patients)
par(mfrow=c(2,3))
# subject_plot: vector of subject id’s for filtering and plotting
# predictions
for(i in 1:length(subject_plot)){
  subj_id<- subject_plot[i]
  data1 <- data.frame(data.develop)
  data1_sub <- data1[data1$id%in%subj_id,]
  data2 <- data.frame(data.develop[data.develop$id%in%subj_id &
  data.develop$year02_select=="TRUE",])
  filter_data<- rbind(data1[!data1$id%in%subj_id,],data2)
  filter_data <- filter_data[order(filter_data$id,filter_data$age),]
  gender <- ifelse(data1_sub$Gender[1]==0,"Female","Male")
  f508 <- ifelse( data1_sub$F508_c[1]==1, "None",
                  ifelse(data1_sub$F508_c[1]==2,"Homozygous", "Heterozygous"))
  # apply filtering
  filter1 <- filtered(formula00_full, data = data1,
                      id=data1$id, process = "ibm",
                      timeVar=data1$t, estimate=develop.ibm$estimates,
                      subj.id=subj_id)
  #
  filter2 <- filtered(formula00_full, data = filter_data,
                      id=filter_data$id, process = "ibm",
                      timeVar=filter_data$t, estimate=develop.ibm$estimates,
                      subj.id=subj_id)
  # get model structure
  spn <-c("X1","X2","X3","X4","X5")
  derv.pwli <- as.matrix(data.develop[, c("X1.1","X2.1","X3.1","X4.1",
   "X5.1")])
  interaction1 <-c("age.Gender","age.MRSA","age.pa", "age.cfrd",
   "age.SESlow","age.bcohr1","age.bcohr2","age.bcohr3","age.bcohr4",
"age.bcohr5","age.F508_c2","age.F508_c3")
  interaction2<-c("Gender","MRSA","pa","cfrd","SESlow","bcohr1","bcohr2",
  "bcohr3","bcohr4","bcohr5","F508_c2","F508_c3")
  # calculate derivative
  deriv2 <- develop.ibm$estimate["t", 1] +
    develop.ibm$estimate["age", 1] +
    2*data2$age*develop.ibm$estimate["age2", 1]+
    3*data2$age2*develop.ibm$estimate["age3", 1]+
    as.matrix(data2[,c("X1.1","X2.1","X3.1","X4.1","X5.1")]) %*%
    as.matrix(develop.ibm$estimate[c("X1","X2","X3","X4","X5"), 1]) +
    as.matrix(data2[,interaction2])%*%
    as.matrix(develop.ibm$estimates[interaction1,1])+filter2$b[, "mean"]
  deriv_up2  <-deriv2 + 1.96*sqrt(filter2$b[, "variance"])
  deriv_low2 <-deriv2 - 1.96*sqrt(filter2$b[, "variance"])
  #
  deriv <- develop.ibm$estimate["t", 1] +
    develop.ibm$estimate["age", 1] +
    2*data1_sub$age*develop.ibm$estimate["age2", 1] +
    3*data1_sub$age2*develop.ibm$estimate["age3", 1] +
    as.matrix(data1_sub[,c("X1.1","X2.1","X3.1","X4.1","X5.1")])%*%
    as.matrix(develop.ibm$estimate[c("X1","X2","X3","X4","X5"), 1]) +
    as.matrix(data1_sub[,interaction2])%*%
    as.matrix(develop.ibm$estimates[interaction1,1]) + filter1$b[, "mean"]
  deriv_up  <-deriv + 1.96*sqrt(filter1$b[, "variance"])
```

```
deriv_low <-deriv - 1.96*sqrt(filter1$b[, "variance"])
# define target function
threshold2 <- -1.5 -
  develop.ibm$estimate["t", 1]-
  develop.ibm$estimate["age", 1] -
  2*data2$age*develop.ibm$estimate["age2", 1]-
  3*data2$age2*develop.ibm$estimate["age3", 1]-
  as.matrix(data2[,c("X1.1","X2.1","X3.1","X4.1","X5.1")]) %*%
  as.matrix(develop.ibm$estimate[c("X1","X2","X3","X4","X5"), 1])-
  as.matrix(data2[,interaction2])%*%
   as.matrix(develop.ibm$estimates[interaction1,1])
  # compute pred probs
  prob2 <- pnorm(threshold2, filter2$b[, "mean"],
   sqrt(filter2$b[, "variance"]))

filter_result <- filter2
threshold <- threshold2
prob <- as.vector(prob2)
# for CIs
set.seed(1234)
#
B<-100
#step1
bm_data <-filter_result$b
mu_hat <-bm_data[,"mean"]
sigma_hat <-bm_data[,"variance"]
n<-length(mu_hat)
Ynb<-t(rmvnorm(B, rep(0, n) , diag(sigma_hat)))
mu_star <- mu_hat + as.matrix(Ynb*(n^(-0.5)))
#step2
prob_star <- pnorm(threshold, mu_star, rep(sqrt(sigma_hat),B))
MSE2 <- rowMeans((prob_star- prob)**2)
#
prob_up2<-prob2+1.96*sqrt(MSE2)
prob_low2<-prob2-1.96*sqrt(MSE2)

threshold <- -1.5 -
  develop.ibm$estimate["t", 1]-
  develop.ibm$estimate["age", 1] -
  2*data1_sub$age*develop.ibm$estimate["age2", 1]-
  3*data1_sub$age2*develop.ibm$estimate["age3", 1]-
  as.matrix(data1_sub[,c("X1.1","X2.1","X3.1","X4.1","X5.1")]) %*%
  as.matrix(develop.ibm$estimate[c("X1","X2","X3","X4","X5"), 1]) -
  as.matrix(data1_sub[,interaction2])%*%
   as.matrix(develop.ibm$estimates[interaction1,1])

prob1 <- pnorm(threshold, filter1$b[, "mean"],
 sqrt(filter1$b[, "variance"]))
#
filter_result <- filter1
threshold <- threshold
prob <- as.vector(prob1)
# for CIs
set.seed(1234)
#
B<-100
#step1
```

```
  bm_data <-filter_result$b
  mu_hat <-bm_data[,"mean"]
  sigma_hat <-bm_data[,"variance"]
  n<-length(mu_hat)
  Ynb<-t(rmvnorm(B, rep(0, n) , diag(sigma_hat)))
  mu_star <- mu_hat + as.matrix(Ynb*(n^(-0.5)))
  #step2
  prob_star <- pnorm(threshold, mu_star, rep(sqrt(sigma_hat),B))
  MSE1 <- rowMeans((prob_star- prob)**2)
  #
  prob_up<-prob1+1.96*sqrt(MSE1)
  prob_low<-prob1-1.96*sqrt(MSE1)
  # fitted values
  varbname <- c( "t", "age", "age2", "age3", "X1","X2", "X3" , "X4","X5",
   "Gender", "MRSA",   "pa","SESlow","cfrd","bcohr1","bcohr2",
"bcohr3","bcohr4","bcohr5",
"F508_c2","F508_c3","numPEyr","numVisityr",
                 "basefev","age.Gender","age.bcohr1","age.bcohr2",
                 "age.bcohr3","age.bcohr4",
                 "age.bcohr5","age.MRSA","age.pa", "age.cfrd",
                 "age.F508_c2","age.F508_c3","age.SESlow")
  # compute marginal fitted values
  fitted.mean2 <- as.matrix(cbind(1,data2[,varbname]))%*%
  as.matrix( develop.ibm$estimate[c("Intercept",varbname), 1])+
    rep(filter2$u[,2],length(filter2$b[,1])) + filter2$w[, "mean"]
  fit_up2=fitted.mean2+1.96*sqrt(develop.ibm$estimates["tausq",1])
  fit_low2=fitted.mean2-1.96*sqrt(develop.ibm$estimates["tausq",1])

  fitted.mean <- as.matrix(cbind(1,data1_sub[,varbname]))%*%
  as.matrix( develop.ibm$estimate[c("Intercept",varbname), 1])+
    rep(filter1$u[,2],length(filter1$b[,1])) + filter1$w[, "mean"]
  fit_up=fitted.mean+1.96*sqrt(develop.ibm$estimates["tausq",1])
  fit_low=fitted.mean-1.96*sqrt(develop.ibm$estimates["tausq",1])

  # set up plot ---
  time <-data1_sub$age
  time2 <-data2$age
  # fitted fev1
  yrange=range(c(fit_up2,fit_low2,fit_up, fit_low))
  plot(time, data1_sub$fev, ylim=yrange,xlab="Age (years)",
   ylab="FEV1(% predicted)",pch = 19,
       col = "darkgrey",main="Trajectory")

  polygon(c(time,rev(time)),c(fit_low,rev(fit_up)),
   col=rgb(169,169,169,150, maxColorValue=255),border=NA)
  points(time, fitted.mean, col = "black", type = "l",lwd=2)
  points(time, fit_up, col = "black", type = "l", lty = 2)
  points(time, fit_low, col = "black", type = "l", lty = 2)

  polygon(c(time2,rev(time2)),c(fit_low2, rev(fit_up2)),
   col=rgb(1, 0, 0,0.5),border=NA)
  points(time2, fitted.mean2, col = "red", type = "l",lwd=2)
  points(time2, fit_up2, col = "red", type = "l", lty = 2)
  points(time2, fit_low2, col = "red", type = "l", lty = 2)

  # B process
  B_up2=filter2$b[,"mean"]+1.96*sqrt(filter2$b[,"variance"])
  B_low2=filter2$b[,"mean"]-1.96*sqrt(filter2$b[,"variance"])

  B_up=filter1$b[,"mean"]+1.96*sqrt(filter1$b[,"variance"])
  B_low=filter1$b[,"mean"]-1.96*sqrt(filter1$b[,"variance"])
  yrange_B<-range(B_up2,B_up,B_low2,B_low)
  #
  plot(time, filter1$b[,"mean"],
       type = "l",
       ylab=expression(tilde(B)), xlab = "Age (years)",
       ylim=yrange_B, lwd=2,main="Rate of Change")

  polygon(c(time,rev(time)),c( B_low,rev( B_up)),
   col=rgb(169,169,169,150, maxColorValue=255),border=NA)
  points(time, B_up, col = "black", type = "l", lty = 2)
  points(time, B_low, col = "black", type = "l", lty = 2)

  polygon(c(time2,rev(time2)),c(B_low2, rev(B_up2)),
   col=rgb(1, 0, 0,0.5),border=NA)
  points(time2, filter2$b[,"mean"], col = "red", type = "l",lwd=2)
  points(time2, B_up2, col = "red", type = "l", lty = 2)
  points(time2, B_low2, col = "red", type = "l", lty = 2)

  # risk prob (-1.5)
  plot(time, prob1,
       type = "l",
       ylab="Probability of Rapid Decline", xlab = "Age (years)",
       ylim=c(0,1), lwd=2,main=substitute(paste(delta,"= -1.5")))

  polygon(c(time,rev(time)),c(prob_low,rev(prob_up)),
          col=rgb(169,169,169,150, maxColorValue=255),border=NA)

  points(time, prob_up, col = "black", type = "l", lty = 2)
  points(time, prob_low, col = "black", type = "l", lty = 2)

  polygon(c(time2,rev(time2)),c(prob_low2, rev(prob_up2)),
   col=rgb(1, 0, 0,0.5),border=NA)
  points(time2, prob2, col = "red", type = "l",lwd=2)
  points(time2, prob_up2, col = "red", type = "l", lty = 2)
  points(time2, prob_low2, col = "red", type = "l", lty = 2)
}

# Figure 6: individual patient predictions for delta_c = -3% pred/yr
# and delta(t*) (same two patients as Figure 5)
par(mfrow=c(2,2))
# apply filtering and calculate fitted values as before
for(i in 1:length(subject_plot)){
  subj_id<- subject_plot[i]
  # define target function
  threshold2 <- -3 -
    develop.ibm$estimate["t", 1]-
    develop.ibm$estimate["age", 1] -
    2*data2$age*develop.ibm$estimate["age2", 1]-
    3*data2$age2*develop.ibm$estimate["age3", 1]-
    as.matrix(data2[,c("X1.1","X2.1","X3.1","X4.1","X5.1")]) %*%
    as.matrix(develop.ibm$estimate[c("X1","X2","X3","X4","X5"), 1])-
    as.matrix(data2[,interaction2])%*%
     as.matrix(develop.ibm$estimates[interaction1,1])
```

```
prob2 <- pnorm(threshold2, filter2$b[, "mean"],
 sqrt(filter2$b[, "variance"]))

#########
filter_result <- filter2
threshold <- threshold2
prob <- as.vector(prob2)
#CIs
set.seed(1234)
#
B<-100
#step1
bm_data <-filter_result$b
mu_hat <-bm_data[,"mean"]
sigma_hat <-bm_data[,"variance"]
n<-length(mu_hat)
Ynb<-t(rmvnorm(B, rep(0, n) , diag(sigma_hat)))
mu_star <- mu_hat + as.matrix(Ynb*(n^(-0.5)))
#step2
prob_star <- pnorm(threshold, mu_star, rep(sqrt(sigma_hat),B))
MSE2 <- rowMeans((prob_star- prob)**2)
#
prob_up2<-prob2+1.96*sqrt(MSE2)
prob_low2<-prob2-1.96*sqrt(MSE2)
#
threshold1 <- -3 -
  develop.ibm$estimate["t", 1]-
  develop.ibm$estimate["age", 1] -
  2*data1_sub$age*develop.ibm$estimate["age2", 1]-
  3*data1_sub$age2*develop.ibm$estimate["age3", 1]-
  as.matrix(data1_sub[,c("X1.1","X2.1","X3.1","X4.1","X5.1")])
   %*%as.matrix(develop.ibm$estimate[c("X1","X2","X3","X4","X5"), 1])-
  as.matrix(data1_sub[,interaction2])%*%
  as.matrix(develop.ibm$estimates[interaction1,1])
prob1 <- pnorm(threshold1, filter1$b[, "mean"], sqrt(filter1$b[, "variance"]))
##
#########
filter_result <- filter1
threshold <- threshold1
prob <- as.vector(prob1)
# CIs
set.seed(1234)
#
B<-100
#step1
bm_data <-filter_result$b
mu_hat <-bm_data[,"mean"]
sigma_hat <-bm_data[,"variance"]
n<-length(mu_hat)
Ynb<-t(rmvnorm(B, rep(0, n) , diag(sigma_hat)))
mu_star <- mu_hat + as.matrix(Ynb*(n^(-0.5)))
#step2
prob_star <- pnorm(threshold, mu_star, rep(sqrt(sigma_hat),B))
MSE1 <- rowMeans((prob_star- prob)**2)
#
prob_up<-prob1+1.96*sqrt(MSE1)
```

```
prob_low<-prob1-1.96*sqrt(MSE1)

# personalized threshold
threshold_uni0 <- -1*(threshold1+3)
threshold_uni0 <-min(threshold_uni0)
#
threshold_uni2 <- threshold_uni0 -
  develop.ibm$estimate["t", 1]-
  develop.ibm$estimate["age", 1] -
  2*data2$age*develop.ibm$estimate["age2", 1]-
  3*data2$age2*develop.ibm$estimate["age3", 1]-
  as.matrix(data2[,c("X1.1","X2.1","X3.1","X4.1","X5.1")])
  %*%as.matrix(develop.ibm$estimate[c("X1","X2","X3","X4","X5"), 1]) -
  as.matrix(data2[,interaction2])%*%as.matrix(develop.ibm$estimates[interaction1,1])
prob2_p <- pnorm(threshold_uni2, filter2$b[, "mean"], sqrt(filter2$b[, "variance"]))
filter_result <- filter2
threshold <- threshold_uni2
prob <- as.vector(prob2_p)
#CIs
set.seed(1234)
#
B<-100
#step1
bm_data <-filter_result$b
mu_hat <-bm_data[,"mean"]
sigma_hat <-bm_data[,"variance"]
n<-length(mu_hat)
Ynb<-t(rmvnorm(B, rep(0, n) , diag(sigma_hat)))
mu_star <- mu_hat + as.matrix(Ynb*(n^(-0.5)))
#step2
prob_star <- pnorm(threshold, mu_star, rep(sqrt(sigma_hat),B))
MSE2_p <- rowMeans((prob_star- prob)**2)
#
prob_up2_p<-prob2_p+1.96*sqrt(MSE2_p)
prob_low2_p<-prob2_p-1.96*sqrt(MSE2_p)
#
threshold_uni <- threshold_uni0 -
  develop.ibm$estimate["t", 1]-
  develop.ibm$estimate["age", 1] -
  2*data1_sub$age*develop.ibm$estimate["age2", 1]-
  3*data1_sub$age2*develop.ibm$estimate["age3", 1]-
  as.matrix(data1_sub[,c("X1.1","X2.1","X3.1","X4.1","X5.1")])
  %*%as.matrix(develop.ibm$estimate[c("X1","X2","X3","X4","X5"), 1])-
  as.matrix(data1_sub[,interaction2])%*%as.matrix(develop.ibm$estimates[interaction1,1])
prob_p <- pnorm(threshold_uni, filter1$b[, "mean"], sqrt(filter1$b[, "variance"]))
#
filter_result <- filter1
threshold <- threshold_uni
prob <- as.vector(prob_p)
#CIs
set.seed(1234)
#
B<-100
#step1
bm_data <-filter_result$b
mu_hat <-bm_data[,"mean"]
sigma_hat <-bm_data[,"variance"]
```

```
n<-length(mu_hat)
Ynb<-t(rmvnorm(B, rep(0, n) , diag(sigma_hat)))
mu_star <- mu_hat + as.matrix(Ynb*(n^(-0.5)))
#step2
prob_star <- pnorm(threshold, mu_star, rep(sqrt(sigma_hat),B))
MSE1_p <- rowMeans((prob_star- prob)**2)
#
prob_up_p<-prob_p+1.96*sqrt(MSE1_p)
prob_low_p<-prob_p-1.96*sqrt(MSE1_p)

time <-data1_sub$age
time2 <-data2$age

# set up plot
plot(time, prob1,
     type = "l",
     ylab="Probability of Rapid Decline", xlab = "Age (years)",
     ylim=c(0,1), lwd=2,
     main=substitute(paste(delta,"= -3")))

polygon(c(time,rev(time)),c(prob_low,rev(prob_up)),
        col=rgb(169,169,169,150, maxColorValue=255),border=NA)

points(time, prob_up, col = "black", type = "l", lty = 2)
points(time, prob_low, col = "black", type = "l", lty = 2)

polygon(c(time2,rev(time2)),c(prob_low2, rev(prob_up2)),col=rgb(1, 0, 0,0.5),border=NA)
points(time2, prob2, col = "red", type = "l",lwd=2)
points(time2, prob_up2, col = "red", type = "l", lty = 2)
points(time2, prob_low2, col = "red", type = "l", lty = 2)

# plotting personalized threshold
thres <- round(threshold_uni0,2)
if(example.plots==T){
  if(i==1){
    plot(time, prob_p,
         type = "l",
         ylab="Probability of Rapid Decline", xlab = "Age (years)",
         ylim=c(0,1), lwd=2,main=expression(paste(delta,"= -2.51")))
  }
  if(i==2){
    plot(time, prob_p,
         type = "l",
         ylab="Probability of Rapid Decline", xlab = "Age (years)",
         ylim=c(0,1), lwd=2,main=expression(paste(delta,"= -2.53")))
  }
}
else{
  plot(time, prob_p,
       type = "l",
       ylab="Probability of Rapid Decline", xlab = "Age (years)",
       ylim=c(0,1), lwd=2,
       main=bquote(paste(delta,’=’,.(thres))))
}
```

```
  polygon(c(time,rev(time)),c(prob_low_p,rev(prob_up_p)),
          col=rgb(169,169,169,150, maxColorValue=255),border=NA)
  points(time, prob_up_p, col = "black", type = "l", lty = 2)
  points(time, prob_low_p, col = "black", type = "l", lty = 2)

  polygon(c(time2,rev(time2)),c(prob_low2_p, rev(prob_up2_p)),
   col=rgb(1, 0, 0,0.5),border=NA)
  points(time2, prob2_p, col = "red", type = "l",lwd=2)
  points(time2, prob_up2_p, col = "red", type = "l", lty = 2)
  points(time2, prob_low2_p, col = "red", type = "l", lty = 2)
}
```