Appendix III

S-PLUS functions for the parametric approach

The parametric approach introduced in Chapter 4 is based in two functions: `plike`, that computes the contribution to the log likelihood of a subject and `tloglik`, that computes the total loglikelihood taken into account the different missing data patterns in the data. $-2\log L$ will be minimized by the S-PLUS function `ms` that minimizes a sum of nonlinear functions over parameters in a data frame.

```
"plike"<-function(obs, beta, sigma, alpha, alph1, alph2, alph3, clogodds,
cc4 = NULL, cpp = NULL, model = 1)
{
################################################################
#
# INICIALITZACIO DE VARIABLES
#
################################################################
     len <- dim(obs)[1]
     r1 <- as.numeric(is.na(C4))
     r2 <- as.numeric(is.na(PP))
     RA1 <- as.numeric(RA == 1)
     RA2 <- as.numeric(RA == 2)
     BA1 <- as.numeric(BA == 1)
     BA2 <- as.numeric(BA == 2)
```

```
    C4 <- as.numeric(C4 > 14)
    if(!is.null(cc4)) {
        C4 <- cc4
    }
    if(!is.null(cpp)) {
        PP <- cpp
    }
#####################################################################
#
# CALCUL DE LES PROBABILITATS CONDICIONADES DE V A Y I X
#
#####################################################################
    vbeta <- beta[1] + beta[2] * C4 + beta[3] * PP
    vsigma <- sigma[1] + sigma[2] * C4 + sigma[3] * PP #
#####################################################################
#
# CALCUL DE LES PROBABILITATS DE RESPOSTA CONDICIONADES Y (NO),
# V I X
#
#####################################################################
    valpha1 <- alpha[1, 1] + I(model > 1) * (alpha[1, 2] * TR + alpha[1,
3] * RA1 + alpha[1, 4] * RA2 + alpha[1, 5] * BA1 + alpha[1, 6] * BA2) +
I((model == 3) || (model == 5)) * alpha[1, 7] * C4 + I((model == 4) ||
(model == 5)) * alpha[1, 8] * PP
    pi1 <- exp(valpha1)/(1 + exp(valpha1))
    valpha2 <- alpha[2, 1] + I(model > 1) * (alpha[2, 2] * TR + alpha[2,
3] * RA1 + alpha[2, 4] * RA2 + alpha[2, 5] * BA1 + alpha[2, 6] * BA2) +
I((model == 3) || (model == 5)) * alpha[2, 7] * C4 + I((model == 4) ||
(model == 5)) * alpha[2, 8] * PP
    pi2 <- exp(valpha2)/(1 + exp(valpha2))
    valpha3 <- alpha[3, 1] + I(model > 1) * (alpha[3, 2] * TR + alpha[3,
3] * RA1 + alpha[3, 4] * RA2 + alpha[3, 5] * BA1 + alpha[3, 6] * BA2) +
I((model == 3) || (model == 5)) * alpha[3, 7] * C4 + I((model == 4) ||
(model == 5)) * alpha[3, 8] * PP
```

```
    pi3 <- exp(valpha3)/(1 + exp(valpha3))
    pir <- pi1^r1 * (1 - pi1)^(1 - r1) * (pi2^r2 * (1 - pi2)^(1 - r2))^r1
* (pi3^r2 * (1 - pi3)^(1 - r2))^(1 - r1) #
###################################################################
#
# CALCUL DE LES PROBABILITATS DE LES COVARIANTS X
#
###################################################################
    pix1 <- exp(alph1)/(1 + exp(alph1))
    pix2 <- exp(alph2)/(1 + exp(alph2))
    pix3 <- exp(alph3)/(1 + exp(alph3))
    pix <- pix1^C4 * (1 - pix1)^(1 - C4) * (pix3^PP * (1 - pix3)^(1 -
PP))^C4 * (pix2^PP * (1 - pix2)^(1 - PP))^(1 - C4) #
###################################################################
#
# CALCUL DE LES PROBABILITATS DE Y CONDICIONADA A X
#
###################################################################
    piycx <- (1/(vsigma * SV) * (exp( - vbeta) * SV)^(1/vsigma))^MO *
exp( - (exp( - vbeta) * SV)^(1/vsigma)) #
###################################################################
#
# CALCUL DE LES PROBABILITATS CONDICIONADES DE V A Y I X
#
###################################################################
    logodd <- matrix(rep(0, 17 * len), ncol = 17)
    odd <- matrix(rep(1, 17 * len), ncol = 17)
    prob <- matrix(rep(1/3, 17 * len), ncol = 17)
    for(j in 1:17) {
        logodd[, j] <- clogodds[j, 1] + clogodds[j, 2] * log(SV) + clogodds[j,
3] * MO + clogodds[j, 4] * log(SV) * MO + clogodds[j, 5] * C4 + clogodds[j,
6] * PP
        odd[, j] <- exp(logodd[, j])
    }
```

```
     prob[, 1] <- odd[, 1]/(1 + odd[, 1])
     for(j in 1:8) {
          prob[, 2 * j] <- odd[, 2 * j]/(1 + odd[, 2 * j] + odd[, 2 * j
+ 1])
          prob[, 2 * j + 1] <- odd[, 2 * j + 1]/(1 + odd[, 2 * j] + odd[,
2 * j + 1])
     }
     pivtr <- (1 - prob[, 1])^(1 - TR) * prob[, 1]^TR
     pivra <- ((1 - prob[, 2] - prob[, 3])^I(RA == 0) * prob[, 2]^I(RA
== 1) * prob[, 3]^I(RA == 2))^(1 - TR) * ((1 - prob[, 4] - prob[, 5])^I(RA
== 0) * prob[, 4]^I(RA == 1) * prob[, 5]^I(RA == 2) )^(TR)
     pivba <- ((1 - prob[, 6] - prob[, 7])^I(BA == 0) * prob[, 6]^I(BA
== 1) * prob[, 7]^I(BA == 2))^(I(TR == 0) * I(RA == 0)) * ((1 - prob[,
8] - prob[, 9])^I(BA == 0) * prob[, 8]^I(BA == 1) * prob[, 9]^I(BA == 2))^(I(TR
== 0) * I(RA == 1)) * ((1 - prob[, 10] - prob[, 11])^I(BA == 0) * prob[,
10]^I(BA == 1) * prob[, 11]^I(BA == 2))^(I(TR == 0) * I(RA == 2)) * ((1
- prob[, 12] - prob[, 13])^I(BA == 0) * prob[, 12]^I(BA == 1) * prob[,
13]^I(BA == 2))^(I(TR == 1) * I(RA == 0)) * ((1 - prob[, 14] - prob[, 15])^I(BA
== 0) * prob[, 14]^I(BA == 1) * prob[, 15]^I(BA == 2))^(I(TR == 1) * I(RA
== 1)) * ((1 - prob[, 16] - prob[, 17])^I(BA == 0) * prob[, 16]^I(BA ==
1) * prob[, 17]^I(BA == 2))^(I(TR == 1) * I(RA == 2))
     pivcxy <- pivtr * pivra * pivba #
##################################################################
#
# CALCUL DE LES CONTRIBUCIONS A LA VERSEMBLANCA
#
##################################################################
     value <- pix * piycx * pivcxy * pir
     value
}


"tloglik"<-function(beta, sigma, alpha, alph1, alph2, alph3, clogodds,
model)
```

```
{
#############################################################################
#
# CALCUL DEL LOG DE LA VERSEMBLANCA PER A LES DADES COMPLETAMENT OBSERVADES
#
#############################################################################
     attach(data11)
     obs <- data11
     value <- log(plike(obs, beta, sigma, alpha, alph1, alph2, alph3, clogodds,
, , model))
     detach("data11") #
#############################################################################
#
# CALCUL DEL LOG DE LA VERSEMBLANCA PER A LES DADES AMB NOMES CD4
#
#############################################################################
     attach(data10)
     obs <- data10
     value <- c(value, log(plike(obs, beta, sigma, alpha, alph1, alph2,
alph3, clogodds, , 0, model) + plike(obs, beta, sigma, alpha, alph1, alph2,
alph3, clogodds, , 1, model)))
     detach("data10") #
#############################################################################
#
# CALCUL DEL LOG DE LA VERSEMBLANCA PER A LES DADES AMB NOMES PPD
#
#############################################################################
     attach(data01)
     obs <- data01
     value <- c(value, log(plike(obs, beta, sigma, alpha, alph1, alph2,
alph3, clogodds, 0, , model) + plike(obs, beta, sigma, alpha, alph1, alph2,
alph3, clogodds, 1, , model)))
     detach("data01") #
#############################################################################
```

```
#
# CALCUL DEL LOG DE LA VERSEMBLANCA PER A LES DADES AMB CD4 I PPD NO OBSERV.
#
################################################################################
     attach(data00)
     obs <- data00
     value <- c(value, log(plike(obs, beta, sigma, alpha, alph1, alph2,
alph3, clogodds, 0, 0, model) + plike(obs, beta, sigma, alpha, alph1, alph2,
alph3, clogodds, 0, 1, model) + plike(obs, beta, sigma, alpha, alph1, alph2,
alph3, clogodds, 1, 0, model) + plike(obs, beta, sigma,
alpha, alph1, alph2, alph3, clogodds, 1, 1, model)))
     detach("data00")
     value
}
```