Supplementary material for “A prediction model for colon cancer surveillance data”

In this supplemental document, we provide sample R code for implementing the proposed model using function ‘gnm’ in the R package ‘gnm’.

We begin by assuming that you are able to partition your data into matrices where each row corresponds to an interval between colonoscopies and each month corresponds to a one-month interval.

For simplicity, we present the code assuming only one Z matrix is used (specifically, the one corresponding to an Adenoma).

Suppose one individual’s patient data is as given in Table 1.

Table 1: Patient historical record for a sample patient.

Age	**Test**	**Test Result**	**Reason for Test**
67.53	COL	Adenoma	Surveillance
72.79	COL	Normal	Surveillance
73.18	COL	Adenoma	Symptoms

Matrices corresponding to age, indicator of interval, and Adenoma are then created, which will be of the form given in Table 2 and Table 3. Matrices must have the same number of columns to use the R code that is presented here. The number of columns of the matrices were chosen to accommodate the longest interval between age at first COL and age at last COL. The age in the first column of the “age.mat” matrix corresponds to the age at the first COL for each individual (rounded to the nearest 1/12th of a year) and is not the same for each individual.

Table 2: Matrices corresponding to the sample patient’s age (age.mat) and an indicator of which of the patient’s data will be used in the interval (ind.mat). Note that since the individual has three colonoscopies there will be two rows corresponding to this individual in each matrix.

age.mat

1	**2**	**3**	$\cdots$	**62**	**63**	**64**	**65**	$\cdots$	**68**
67.50	67.58	67.67	$\cdots$	72.58	72.67	0	0	$\cdots$	0
0	0	0	$\cdots$	0	0	72.75	72.58	$\cdots$	73.08

ind.mat

1	**2**	**3**	$\cdots$	**62**	**63**	**64**	**65**	$\cdots$	**68**
1	1	1	$\cdots$	1	1	0	0	$\cdots$	0
0	0	0	$\cdots$	0	0	1	1	$\cdots$	1

Table 3: Matrix corresponding to the sample patient's Adenoma status (i.e., the Z matrix). Note that if the patient had another Adenoma within 12 months of the patient's first Adenoma, the second row would have 1s starting from that point and the first row would revert back to 0s.

Ad.mat

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	⋯	**67**	**68**
1	1	1	1	1	1	1	1	1	1	1	1	0	0	⋯	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	⋯	0	0

The following R code is commented to describe how matrices of this form can then be used to implement the model described in the paper.

```
library(gnm)                #load package
N<-ncol(age.mat)            #all matrices should have the same number of columns

#This step was done purely for memory space purposes (with many matrices/columns,
#the length of the string used in the code exceeds the memory capacity of R)
let.mat<-expand.grid(letters,letters)
let.mat<-let.mat[1:N,]

#Assign columns of age matrix  (age.mat) to character ‘a’
as<-NULL
for(i in 1:N)
{as<-c(as,paste0("a",let.mat[i,1],let.mat[i,2]))}
age.mat<-data.frame(age.mat)
names(age.mat)<-as

#Assign columns of indicator matrix (ind.mat) to character ‘X’
Xs<-NULL
for(i in 1:N)
{Xs<-c(bs,paste0("X",let.mat[i,1],let.mat[i,2]))}
names(ind.mat)<-Xs

#Assign columns of Adenoma matrix (Ad.mat) to character ‘b’
bs<-NULL
for(i in 1:N)
{bs<-c(bs,paste0("b",let.mat[i,1],let.mat[i,2]))}
names(Ad.mat)<-bs

subs<-paste("list(substitute(",sep="",paste0(Xs,collapse=",substitute(",")"),",substitute(",
                          paste0(as,collapse=",substitute(",")"),",substitute(",
                          paste0(bs,collapse=",substitute(",")"),")")

#Creating the function of the form in the integral in Equation (4) using discretized sums
termfun<-function(predLabels,varLabels)
{
  expr<-"log("
  for(i in 1:(N-1))
  {expr<-paste0(expr,varLabels[i],"*1/12*(",varLabels[i+N],"35)^",predLabels[1],
                          "*exp(",predLabels[2],"*",varLabels[i+2*N],")+")}
  expr<-paste0(expr,varLabels[N],"*1/12*(",varLabels[N+N],"-35)^",predLabels[1],
                          "*exp(",predLabels[2],"*",varLabels[N+2*N],"))")
}
```

```
#If using 2 covariate matrices (e.g., Adenoma within 1 yr and Advanced Adenoma within 1 yr)
#the following code is used, and can be adapted to include additional covariates

#termfun<-function(predLabels,varLabels)
#{
#  expr<-"log("
#  for(i in 1:(N-1))
#  {expr<-paste0(expr,varLabels[i],"*1/12*(",varLabels[i+N],"-#35)^",predLabels[1],
#                       "*exp(",predLabels[2],"*", varLabels[i+2*N],"+",
#                               predLabels[3],"*",varLabels[i+3*N],")+")}
#  expr<-paste0(expr,varLabels[N],"*1/12*(",varLabels[N+N],"-#35)^",predLabels[1],
#                       "*exp(",predLabels[2],"*",varLabels[N+2*N],"+",
#                               predLabels[3],"*",varLabels[N+3*N],"))")
#}

for(i in 1:N)
{
  assign(as[i],age.mat[,i])
  assign(Xs[i],ind.mat[,i])
  assign(bs[i],b.mat[,i])
}

zlist<-""
for(i in 1:N)
{zlist<-paste0(zlist,"X",i,",")}
for(i in 1:N)
{zlist<-paste0(zlist,as[i],",")}
for(i in 1:(N-1))
{zlist<-paste0(zlist,bs[i],",")}
zlist<-paste0(zlist,bs[N])

#If additional matrices are used then those covariates can be included after "Ad1yr=1"
#The names alpha1 and Ad1yr correspond to the names wanted for the coefficients
#in the model summary table and can be changed.
#The 1s correspond to the starting values and can be changed

eval(parse(text=paste0("tformu<-function(",zlist,"){
                       list(predictors=list(alpha1=1,Ad1yr=1),
                       variables=eval(parse(text=subs)),
                       term=termfun)}")))
class(tformu)<-"nonlin"

#Include subject-specific covariates (e.g., location, gender, famrisk) and COL-specific
#covariates (e.g., indicator that reason for COL is symptoms) as vectors in this step
#E.g, An indicator for whether the reason for the test is symptoms would be given as
#c(...,0,1,...) where the 0 and 1 correspond to the entries for the sample patient
#y is the response vector of the form c(...,0,0,...) where the 0s correpsond to the entries
#for the sample patient (since the patient did not have AAC)

fit.expr<-paste0("cbind(y,1-y)~as.factor(location)+as.factor(gender)+
       as.factor(symptoms)+as.factor(famrisk)+tformu(",zlist,")")

#Final model
fit <-gnm(as.formula(fit.expr),family=binomial(link="cloglog"))
```