SUPPLEMENTAL WEB MATERIAL

A Bayesian Sensitivity Analysis to Partition BMI into Components of Body Composition: an Application to Head and Neck Cancer Survival

Patrick T. Bradshaw, Jose P. Zevallos, Kathy Wisniewski, and Andrew F. Olshan

Correspondence to Dr. Patrick T. Bradshaw, Division of Epidemiology and Biostatistics, School of Public Health, University of California, Berkeley, Berkeley, CA 94720. (tel: (510) 664-7299; fax: (510) 643-5163; email: pbradshaw@berkeley.edu)

Author affiliations: School of Public Health, Division of Epidemiology and Biostatistics, University of California Berkeley, Berkeley, CA (Patrick T. Bradshaw); Department of Otolaryngology/Head and Neck Surgery, Washington University School of Medicine, St. Louis, MO (Jose P. Zevallos); Department of Epidemiology, Gillings School of Global Public Health, The University of North Carolina, Chapel Hill, NC (Kathy Wisniewski and Andrew F. Olshan).

CONTENTS

1 Web Appendix 1 **2**
1.1 Prepare data for piecewise exponential model 2
1.2 Specify piecewise exponential model for Nimble 3
1.3 Define constants, data, and parameters for Nimble 4
1.4 Obtain samples from the posterior 4
1.5 Assess convergence and obtain summary statistics 5

2 Web Appendix 2 **6**

List of Tables

1 Geweke z-statistics and Gelman-Rubin scale reduction factors. 6

1 WEB APPENDIX 1

Here we present the R and Nimble[1] code for the main model presented in the paper.

1.1 Prepare data for piecewise exponential model

For the piecewise exponential model[2] we expand the single-observation per subject dataset (`all.data`) into three, 4-year long intervals per subject with corresponding start-stop times for each interval using the `survSplit` command in the `survival` package. Relevant variables in the dataset are `followup` (length of time under observation) and `dead` [indicator of death (=1) or censoring (=0)]:

```
library(survival)

### Create multi-observation dataset for PE model:
# Intervals of 4-years each (over the ~12 years follow-up)
all.data$subject <- seq(1,nrow(all.data)) # Create faux subject id

# Split data into individual records for each interval:
chance.recoded.multi <- survSplit(Surv(followup, dead) ~ ., all.data,
                                  cut=seq(4*365.25,12*365.25,by=4*365.25),
                                  episode="int",start="start")
```

Next we calculate the length of time under observation within each interval (will serve as offset in piecewise exponential model):

```
chance.recoded.multi$dtime <- chance.recoded.multi$followup -
                                                chance.recoded.multi$start
```

Because the covariates do not change across intervals, we use the single-observation dataset to create model matrix for **w**, containing the confounders. We also specify the matrix for the predictors of percent body fat, excluding BMI, and we use the expanded data to create interval indicators for the baseline hazards:

```
# Design matrix for PH regression (without BMI)
w <- model.matrix(~ agerefdate.c + factor(sex) + factor(race) + factor(educ) +
                        factor(total_alc.cat) + factor(cigdurcat2) +
                        total_number_of_vegetable_servin.c +
                        total_number_of_fruit_servings.c +
                        stage.2cat + comorbid.c,
                    data=all.data)[,-1] # remove intercept,
                                        # which is accounted in baseline hazard

# Indicators for intervals:
int <- model.matrix(~ factor(int), data=chance.recoded.multi)

# Design matrix for percent body fat model
# (variable coding matches that from NHANES regression):
Z <- model.matrix(~  I(bmi/10) + I(agerefdate/10) + male + race_aa + race_other +
                     evercig, data=all.data)
```

1.2 Specify piecewise exponential model for Nimble

We then specify the Nimble code for the piecewise exponential model with a prior for unmeasured percent body fat, as described in the text:

```
library(nimble)
library(coda)

pe.model.Code <- nimbleCode({
  # PIECEWISE EXPONENTIAL MODEL FOR OUTCOME:
  # Over all person-time observations (individuals * intervals) (Note nested
   index)
  for (i in 1:N) {
    log.h[i] <- xb.wg[subject[i]] + inprod(interval[i,1:N.j], b.j[1:N.j]);
    dead[i] ~ dpois(exp(log.h[i])*dtime[i]);
  }

  for (j in 1:M) { # Over individuals only--covariates do not change over
   interval
    # Sample percent fat mass (logit transform)
    mu.logit.pct.fm[j] <- inprod(Z[j,1:N.a], a[1:N.a]);
    logit.pct.fm[j] ~ dnorm(mu.logit.pct.fm[j], sd=sd.logit.pct.fm);

    pct.fm[j] <- expit(logit.pct.fm[j]);  # Transform back to proportion

    # Transform BMI into FMI and LMI (multiply BMI by % BF and % lean)
    #     Create linear predictor as function of FMI and LMI
    #     and add covariate portion to linear predictor (w*g)
    xb.wg[j] <- (bmi[j]*pct.fm[j])*b[1] + (bmi[j]*(1-pct.fm[j]))*b[2] +
                  inprod(w[j,1:N.g], g[1:N.g]);
  }

  # PRIORS ON ln-HRs
  # multivariate normal prior
  b[1:N.b] ~ dmnorm(mu.b[1:N.b], prec=tau.b[1:N.b, 1:N.b]);
  g[1:N.g] ~ dmnorm(mu.g[1:N.g], prec=tau.g[1:N.g, 1:N.g]);
  b.j[1:N.j] ~ dmnorm(mu.j[1:N.j], prec=tau.j[1:N.j, 1:N.j]);

  # PRIORS ON ALPHA (LMI-BMI ASSOCIATION)
  a[1:N.a] ~ dmnorm(mu.a[1:N.a], prec=tau.a[1:N.a, 1:N.a]);

  # Calculate HRs from proportional hazards model:
  for (j in 1:N.b) HR[j] <- exp(b[j]);

})
```

1.3 Define constants, data, and parameters for Nimble

Specify constants and data objects for Nimble (details omitted for space):

```
# Constants (parameters on priors, and indexing variables):
pe.model.consts.model1 <- list(N=N,
                               N.b=N.b,
                               N.g=N.g,
                               N.a=N.a,
                               N.j = N.j,
                               mu.b=mu.b, tau.b=tau.b,
                               mu.g=mu.g, tau.g=tau.g,
                               mu.j=mu.j, tau.j=tau.j,
                               M = M,
                               subject=chance.recoded.multi$subject,
                               mu.a=mu.a, tau.a=tau.a,
                               sd.logit.pct.fm = sd.logit.pct.fm)

# Data for model:
pe.model.data <- list(dead=chance.recoded.multi$dead,
                      dtime=chance.recoded.multi$dtime,
                      w=w, interval=int,
                      bmi=all.data$bmi,
                      Z=Z)

n.iter <- 300000
n.thin <- 25
n.burn <- n.iter/2
n.chains <- 3
```

1.4 Obtain samples from the posterior

Create and compile Nimble model, and obtain samples from posterior:

```
# Create the Nimble Model
pe.model.model1 <- nimbleModel(code=pe.model.Code,
                               constants=pe.model.consts.model1,
                               data=pe.model.data)

# Compile the Nimble Model
C.pe.model.model1 <- compileNimble(pe.model.model1)

# Configure the MCMC Algorithm
pe.model.conf.model1 <- configureMCMC(pe.model.model1, print=FALSE, thin=n.thin,
                                      monitors=c("a","b","HR","b.i","sd.ffmi"))

```

```
# Build the MCMC Function
pe.model.MCMC.model1 <- buildMCMC(pe.model.conf.model1)

# Compile the MCMC function and the Nimble Model
C.pe.model.MCMC.model1 <- compileNimble(pe.model.MCMC.model1)

# Sample:
pe.MCMCsamples.model1 <- runMCMC(C.pe.model.MCMC.model1,
                          niter=n.iter, nchains=n.chains,
                          nburnin = (n.iter/(2*n.thin)),
                          returnCodaMCMC = T)
```

1.5 Assess convergence and obtain summary statistics

Use the commands in the CODA package[3] to generate convergence diagnostics.

```
# Assess convergence:
plot(pe.MCMCsamples.model1, ask=FALSE)       # Traceplots and posterior density
   plots
geweke.diag(pe.MCMCsamples.model1)           # Geweke diagnostic
gelman.diag(pe.MCMCsamples.model1)      # Gelman-Rubin diagnostic

# Summarize samples over all chains:
summ.pe.model1 <- summary(pe.MCMCsamples.model1)
```

REFERENCES

1. NIMBLE Development Team. NIMBLE: An R Package for Programming with BUGS models, Version 0.6-6. http://r-nimble.org, 2017.

2. Ibrahim JG, Chen MH, Sinha D. *Bayesian Survival Analysis*. Wiley Online Library, 2005. https://onlinelibrary.wiley.com/doi/full/10.1002/0470011815.b2a11006. Accessed 9 August 2019.

3. Plummer M, Best N, Cowles K, et al. CODA: Convergence Diagnosis and Output Analysis for MCMC. *R News* 2006;6(1):7–11.

2 WEB APPENDIX 2

As a representative example, we present convergence diagnostics (Geweke z-statistics and Gelman-Rubin scale reduction factor) for sensitivity analysis Model 1 for all-cause mortality. The a[] parameters correspond to the coefficients of the logit(p_{BF}) model, b[1] and b[2] are the coefficients (log-hazard ratios) on the latent FMI and LMI variables, respectively, the b.i[] are the log of the baseline hazard terms, and the g[] parameters are the coefficients (log-hazard ratios) for the confounders.

Web Table 1: Geweke z-statistics and Gelman-Rubin scale reduction factors for sensitivity analysis model 1 for all-cause mortality.

	Geweke z-statistic			Gelman-Rubin Scale Reduction Factor	
Parameter	Chain 1	Chain 2	Chain 3	Point Estimate	Upper CI
a[1]	0.68	1.08	-1.54	1.00	1.00
a[2]	-0.88	-1.09	1.98	1.00	1.00
a[3]	0.10	-0.93	0.19	1.00	1.00
a[4]	-1.20	0.10	0.09	1.00	1.00
a[5]	0.79	0.56	0.29	1.00	1.00
a[6]	-0.13	1.26	-0.23	1.00	1.00
a[7]	0.75	-0.58	-0.54	1.00	1.00
b[1]	2.74	-0.07	1.19	1.00	1.00
b[2]	-1.98	0.46	-0.97	1.00	1.01
b.i[1]	1.31	-0.58	1.02	1.01	1.02
b.i[2]	1.71	0.39	0.32	1.00	1.00
b.i[3]	0.57	0.96	1.94	1.00	1.00
g[1]	-1.49	-0.02	0.94	1.00	1.00
g[2]	-1.87	0.31	-1.04	1.00	1.01
g[3]	-0.95	0.60	-0.60	1.00	1.00
g[4]	-0.02	2.43	-0.62	1.00	1.01
g[5]	-0.72	-0.09	-0.15	1.00	1.00
g[6]	-0.34	-1.09	0.18	1.00	1.01
g[7]	1.40	1.05	-0.63	1.00	1.01
g[8]	1.03	1.66	-0.99	1.00	1.01
g[9]	1.02	1.38	-0.16	1.00	1.01
g[10]	0.85	0.30	0.31	1.00	1.00
g[11]	1.53	0.33	-0.59	1.00	1.01
g[12]	1.68	0.59	-1.51	1.00	1.01
g[13]	0.36	0.49	-1.32	1.00	1.01
g[14]	0.18	0.33	-1.85	1.00	1.00
g[15]	-0.27	0.59	-0.11	1.00	1.00
g[16]	-0.53	-1.16	0.18	1.00	1.00
g[17]	-1.51	0.59	1.22	1.00	1.00