eAPPENDIX 1

McCandless[14] describes sensitivity analysis for unmeasured confounding. Here we describe a similar methodology to explore sensitivity to selection bias in a meta-analysis of observational studies. Our approach combines the bias modeling approach of Welton et al.[37] with the odds ratio decomposition for selection bias described by Rothman et al.[15] Let (y_j, σ_j) for $j \in 1 : k \ldots$ denote the log relative risk estimates and standard errors in a sequence of k independent observational studies. These could be cohort studies, case-control studies with matched or independent samples, or cross-sectional studies (e.g Table 1). Each estimator y_j has already been adjusted for the available covariates in study j using stratification or adjustment. For example, Table 2 summarizes effect measures and adjustment techniques. The incidence of osteoporotic fractures in the elderly is low. The rate of hip fractures in white women over 65 years is roughly 5.0 events per 1000 person-years, and lower among males and persons of non-white race or ethnicity.[20,21] Consequently, following Toh et al.[4] we allow for the possibility that the quantity y_j may be a log odds ratio or log hazard ratio.

To adjust for selection bias in the j^{th} study, we use the Bayesian meta-analysis model of Carlin,[40] which assumes that y_j arises from a normal distribution

$$y_j \sim N(\theta_j^*, \sigma_j^2) \qquad \text{for } j \in 1 : k. \tag{2}$$

with mean θ_j^* and variance σ_j^2. The quantity σ_j is the standard error of y_j, which is assumed to be known, whereas θ_j^* is the conditional log relative risk for the association between the dichotomous exposure and outcome in study j among those who have been selected.

In observational studies that compare prevalent users of statins with non-users, the

log relative risk parameter θ_j^* is vulnerable to selection bias. Let θ_j be the true causal log relative risk for the effect of statin use on fracture risk in study j, in the absence of selection bias, and conditional on covariates in study j. To adjust for selection bias, we use the decomposition of the odds ratio in case-control studies for the setting where the probabilities of selection are known.[15–17] Write

$$\exp(\theta_j) \approx \exp(\theta_j^*) \times \exp(\Psi), \tag{3}$$

or equivalently,

$$\theta_j \approx \theta_j^* + \Psi.$$

The approximation in equation (3) is exact if $\exp(\theta_j)$ and $\exp(\theta_j^*)$ are odds ratios (see Rothman et al.,[15] equation 19.16), and it is approximate if $\exp(\theta_j)$ and $\exp(\theta_j^*)$ are relative risks and the disease is rare.

The quantity $\exp(\Psi)$ is equal to the relative probability of selection in fracture cases versus controls, among exposed participants, divided by the relative probability of selection for fracture cases versus controls among unexposed participants. Following Greenland,[18] we assume that the magnitude of selection bias does not depend on measured covariates. Furthermore, following McCandless[14], we assume that Ψ does not depend on j, which implies no heterogeneity in the magnitude of selection bias across studies.

Equations (2) and (3) imply the following model for the data

$$y_j \sim N(\theta_j - \Psi, \sigma_j^2) \qquad \text{for } j \in 1:k.$$

To complete the model specification and adjust for selection bias in a meta-analysis, we require prior distributions for θ_j and Ψ. Equation (1) assigns a normal distribution to Ψ

to model prior beliefs about the magnitude and uncertainty of selection bias. For θ_j, we follow convention in Bayesian random effects meta-analysis[40] and assign

$$\begin{aligned} \theta_j &\sim N(\mu, \tau^2) \qquad \text{for } j \in 1:k \\ \mu &\sim N(0, 10^3) \\ \tau &\sim \text{Uniform}(0, 10^3). \end{aligned}$$

The pooled log relative risk μ, adjusted for selection bias, is the primary target of inference, whereas τ captures the heterogeneity of effects across studies. Write $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$, $\boldsymbol{y} = (y_1, \ldots, y_k)$ and $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_k)$. To study the posterior distribution, denoted $P(\boldsymbol{\theta}, \Psi, \mu, \tau^2 | \boldsymbol{y}, \boldsymbol{\sigma})$, we use Markov chain Monte Carlo as illustrated in eAppendix 2.

eAPPENDIX 2

```
##############################################################################
##############################################################################
## R code for Bayesian sensitivitiy analysis for unmeasured confounding and
## selection bias in a meta-analysis
##
##
## McCandless (2013) "Statin Use and Fracture Risk: Can We Quantify The Healthy
## User Effect?"
##############################################################################
##############################################################################

##############################################################################
## Bayesian sensitivity analysis for unmeasured confounding in a sensitivity
## analysis
##############################################################################

library(meta)

# Raw Data of McCandless (2012) Statins and Fracture Risk: Can we Quantify
# the Healthy User Effect?
tmp1 <- c(0.48, 0.55, 0.50, 1.01, 0.43, 0.68, 0.87, 0.63, 0.62, 1.10, 0.95, 0.58,
     0.49, 0.76, 0.92, 0.64, 1.02)
tmp2 <- c(0.27, 0.44, 0.33, 0.88, 0.24, 0.50, 0.83, 0.47, 0.45, 1.01, 0.59, 0.34,
     0.15, 0.50, 0.64, 0.58, 0.95)
tmp3 <- c(0.83, 0.69, 0.76, 1.16, 0.78, 0.93, 0.92, 0.84, 0.85, 1.20, 1.52, 0.99,
1.57, 1.16, 1.32, 0.72, 1.10)

ys <- log(tmp1)
sigmas.sq <- (apply(cbind(log(tmp3) - log(tmp1),log(tmp1) -
          log(tmp2))/1.96, 1, mean))^2

##############################################################################
## Random effects analysis using DerSimonian Laird in library(meta)
##############################################################################
tmp <- metagen(ys, sqrt(sigmas.sq), sm="RR")

tmp$TE.random    ## Pooled log relative risk estimate
tmp$seTE.random  ## Standard error of pooled log relative risk estimate
```

```
##############################################################################
## Bayesian sensitivity analysis for unmeasured confounding in a sensitivity
## analysis
##############################################################################
expit <- function(a){exp(a)/(1+exp(a))}

## Helper function that computes the bias-corrected log relative risk as
## a function  of bias parameters (lambda, gamm0, gamm1)
a <- function(ys, lambda, gamm0, gamm1){
     p1 <- expit(gamm0 + gamm1)
     p0 <- expit(gamm0)
     ys - log( (exp(lambda) * p1 + (1-p1))/
          (exp(lambda) * p0 + (1-p0)) )
  }

## The log likelihood function
l.lik <- function(thetas, lambda, gamm0, gamm1, ys, sigmas.sq){
     sum(-(0.5/sigmas.sq)*(thetas - a(ys, lambda, gamm0, gamm1))^2)
     }

nburn <- 1000   ## number of burn-in iterations
nrep  <- 100000 ## number of MCMC iterations
thin <- 10      ## how much to thin the chain

k <- length(ys)

## Storage for MCMC output
res.thetas <- matrix(NA, nrow=nrep/thin, ncol=k)
res.tau.sq <- matrix(NA, nrow=nrep/thin, ncol=1)
res.mu <- matrix(NA, nrow=nrep/thin, ncol=1)
res.lambda <- res.gamm0 <- res.gamm1 <- matrix(NA, nrow=nrep/thin, ncol=1)

## Initialize MCMC chain
thetas.cur <- rnorm(k, ys, 0.1)  ## The bias corrected relative risk parameters
tau.sq.cur <- 0.1                ## The between study variance in the bias
                                 ## corrected relative risk parameters
mu.cur <- rnorm(1, 0, 0.1)       ## The bias corrected pooled relative risk parameter
lambda.cur <- rnorm(1, 0, 0.1)   ## The logarithm of the relative risk for the
                                 ## association between U and the outcome
gamm1.cur <- rnorm(1, 0, 0.1)    ## The log odds ratio for the associaiton
                                 ## between U and the exposure (statin use)
```

```
gamm0.cur <- rnorm(1, 0, 0.1)    ## The logit prevalence of U in people without
                                 ## the exposure (statin non-users)
tune <- 0.5 ## A tuning parameter

## Prior distributions for hyperparameters in the Bayesian random effects
v.0 <- 10^-3     ## v.0, sig.sq.0, tau.sq.0, mu.0 are defined in equation 3.11 of
sig.sq.0 <- 10^3 ## Gelman et al. (2004) Bayesian Data Analysis 2nd edition
tau.sq.0 <- 10^3
mu.0 <- 0

## Hyperparameters in prior distribution for bias parameters, based on adjustment
## for Influenza Vaccination.
tmp <- 7966/20783
m.g0 <- log(tmp/(1-tmp));
sd.g0 <- sqrt((tmp*(1-tmp)*20783)^-1)
m.g1 <- log(1.21);
sd.g1 <- (log(1.31) - log(1.12))/(2*1.96)
sqrt(sd.g0^2 + sd.g1^2)
m.l <- log(0.54);
sd.l <- (log(0.94) - log(0.32))/(2*1.96)

for (mnlp in (-nburn):nrep){
  if((mnlp %% 100)==0){print(mnlp)}

  ## Update mu.cur and tau.sq.cur using the algorithm of equation 3.13 of Gelman
  ## et al. (2004) Bayesian Data Analysis 2nd edition.
  grid <- seq(0.05, 3, by=0.05)
  mu.k.grid <- (mu.0/tau.sq.0 + k*mean(thetas.cur)/grid)/(1/tau.sq.0 + k/grid)
  tau.sq.k.grid <- (1/tau.sq.0 + k/grid)^{-1}

  wts <- sqrt(tau.sq.k.grid) * dnorm(mu.k.grid, mean=mu.0, sd=sqrt(tau.sq.0)) *
       grid^{-k/2} * exp(-(1/(2*grid)) * sum((thetas.cur - mean(thetas.cur))^2))
  tau.sq.cur <- grid[sample(1:length(grid), 1, prob=wts)]

  mu.k <- (mu.0/tau.sq.0 + k*mean(thetas.cur)/tau.sq.cur)/(1/tau.sq.0 + k/tau.sq.cur)
  tau.sq.k <- (1/tau.sq.0 + k/tau.sq.cur)^{-1}

  mu.cur <- rnorm(1, mu.k, sqrt(tau.sq.k))

  ## update thetas.cur
  tmp <- matrix(0, nrow=k, ncol=k)
```

```
diag(tmp) <- (sigmas.sq*tau.sq.cur)/(sigmas.sq + tau.sq.cur)

thetas.cur <- (sigmas.sq*mu.cur+tau.sq.cur*a(ys,lambda.cur,gamm0.cur,gamm1.cur))/
     (sigmas.sq + tau.sq.cur) + t(chol(tmp)) %*% rnorm(k)

## Update bias parameters lambda.cur, gamm0.cur, gamm1.cur

lambda.cnd <- rnorm(1, lambda.cur, sd=tune)

log.num <- l.lik(thetas.cur, lambda.cnd, gamm0.cur, gamm1.cur, ys, sigmas.sq) -
     (0.5/sd.l^2) * sum((lambda.cnd - m.l)^2)
log.acc <- log.num -
     (l.lik(thetas.cur, lambda.cur, gamm0.cur, gamm1.cur, ys, sigmas.sq) -
     (0.5/sd.l^2) * sum((lambda.cur - m.l)^2))

if (log(runif(1)) < log.acc) {lambda.cur <- lambda.cnd}

gamm0.cnd <- rnorm(1, gamm0.cur, sd=tune)

log.num <- l.lik(thetas.cur, lambda.cur, gamm0.cnd, gamm1.cur, ys, sigmas.sq) -
              (0.5/sd.g0^2) * sum((gamm0.cnd - m.g0)^2)
log.acc <- log.num -
     (l.lik(thetas.cur, lambda.cur, gamm0.cur, gamm1.cur, ys, sigmas.sq) -
     (0.5/sd.g0^2) * sum((gamm0.cur - m.g0)^2))

if (log(runif(1)) < log.acc) {gamm0.cur <- gamm0.cnd}

gamm1.cnd <- rnorm(1, gamm1.cur, sd=tune)

log.num <- l.lik(thetas.cur, lambda.cur, gamm0.cur, gamm1.cnd, ys, sigmas.sq) -
     (0.5/sd.g1^2) * sum((gamm1.cnd - m.g1)^2)
log.acc <- log.num -
     (l.lik(thetas.cur, lambda.cur, gamm0.cur, gamm1.cur, ys, sigmas.sq) -
     (0.5/sd.g1^2) * sum((gamm1.cur - m.g1)^2))

if (log(runif(1)) < log.acc) {gamm1.cur <- gamm1.cnd}

## Storage of MCMC output
if (mnlp > 0){
     res.thetas[floor(mnlp/thin),] <- thetas.cur
  res.tau.sq[floor(mnlp/thin)] <- tau.sq.cur
```

```
    res.mu[floor(mnlp/thin)] <- mu.cur
    res.lambda[floor(mnlp/thin)] <- lambda.cur
    res.gamm0[floor(mnlp/thin)] <- gamm0.cur
    res.gamm1[floor(mnlp/thin)] <- gamm1.cur
  }

}

mean(res.mu)  ## Pooled log relative risk estimate adjusted for a binary unmeasured
sd(res.mu)    ## Standard error of pooled log relative risk estimate

##############################################################################
## Bayesian sensitivity analysis for selectoin bias in a meta-analysis
##############################################################################

expit <- function(a){exp(a)/(1+exp(a))}

## Helper function that computes the bias-corrected log relative risk as
## a function of bias  parameters (lambda, gamm0, gamm1)
a <- function(ys, psi){
  tmp <- rep(psi, 17)
  tmp[c(9,16,17)] <- 0 ## To prevent adjustment for selection bias in the
                       ## 3 studies that utilized a new user design
  ys + tmp
  }

## The log likelihood function
l.lik <- function(thetas, psi, ys, sigmas.sq){
  sum(-(0.5/sigmas.sq)*(thetas - a(ys, psi))^2)
    }

nburn <- 1000   ## number of burn-in iterations
nrep  <- 100000 ## number of MCMC iterations
thin <- 10      ## how much to thin the chain

k <- length(ys)

## Storage for MCMC output
res.thetas <- matrix(NA, nrow=nrep/thin, ncol=k)
res.tau.sq <- matrix(NA, nrow=nrep/thin, ncol=1)
res.mu <- matrix(NA, nrow=nrep/thin, ncol=1)
```

```
res.psi <- matrix(NA, nrow=nrep/thin, ncol=1)

## Initialize MCMC chain
thetas.cur <- rnorm(k, ys, 0.1)  ## The bias corrected relative risk parameters
tau.sq.cur <- 0.1                ## The between study variance in the bias
                                 ## corrected  relative risk parameters
mu.cur <- rnorm(1, 0, 0.1)       ## The bias corrected pooled relative risk parameter
psi.cur <- rnorm(1, 0, 0.1)      ## The logarithm of the bias factor exp(psi) described
                                 ## in equation 19.16 of Rothman et al. (2008)
                                 ## Modern Epidemiology
tune <- 0.5 ## A tuning parameter

## Prior distributions for hyperparameters in the Bayesian random effects
v.0 <- 10^-3     ## v.0, sig.sq.0, tau.sq.0, mu.0 are defined in equation 3.11 of
sig.sq.0 <- 10^3 ## Gelman et al. (2004) Bayesian Data Analysis 2nd edition
tau.sq.0 <- 10^3
mu.0 <- 0

## Hyperparameters in prior distribution for bias parameters, based on
## adjustment for selection bias

m.o <- log(0.84)- log(0.54);
sd.o <- sqrt((mean(c(log(0.91)-log(0.84), log(0.84)-log(0.77)))/1.96)^2 +
     (mean(c(log(0.66)-log(0.54), log(0.54)-log(0.45)))/1.96)^2) ;

for (mnlp in (-nburn):nrep){
  if((mnlp %% 100)==0){print(mnlp)}

  ## Update mu.cur and tau.sq.cur using the algorithm of equation 3.13 of Gelman
  ## et al. (2004) Bayesian Data Analysis 2nd edition.
  grid <- seq(0.05, 3, by=0.05)
  mu.k.grid <- (mu.0/tau.sq.0 + k*mean(thetas.cur)/grid)/(1/tau.sq.0 + k/grid)
  tau.sq.k.grid <- (1/tau.sq.0 + k/grid)^{-1}

  wts <- sqrt(tau.sq.k.grid) * dnorm(mu.k.grid, mean=mu.0, sd=sqrt(tau.sq.0)) *
       grid^{-k/2} * exp(-(1/(2*grid)) * sum((thetas.cur - mean(thetas.cur))^2))
  tau.sq.cur <- grid[sample(1:length(grid), 1, prob=wts)]

  mu.k <- (mu.0/tau.sq.0 + k*mean(thetas.cur)/tau.sq.cur)/(1/tau.sq.0 + k/tau.sq.cur)
  tau.sq.k <- (1/tau.sq.0 + k/tau.sq.cur)^{-1}
```

```
  mu.cur <- rnorm(1, mu.k, sqrt(tau.sq.k))

  ## update thetas.cur
  tmp <- matrix(0, nrow=k, ncol=k)
  diag(tmp) <- (sigmas.sq*tau.sq.cur)/(sigmas.sq + tau.sq.cur)

  thetas.cur <- (sigmas.sq*mu.cur + tau.sq.cur*a(ys, psi.cur))/
                (sigmas.sq + tau.sq.cur) + t(chol(tmp)) %*% rnorm(k)

  ## Update bias parameters psi.cur
  psi.cnd <- rnorm(1, psi.cur, sd=tune)

  log.num <- l.lik(thetas.cur, psi.cnd, ys, sigmas.sq) -
       (0.5/sd.o^2) * sum((psi.cnd - m.o)^2)
  log.acc <- log.num -
       (l.lik(thetas.cur, psi.cur, ys, sigmas.sq) -
       (0.5/sd.o^2) * sum((psi.cur - m.o)^2))

  if (log(runif(1)) < log.acc) {psi.cur <- psi.cnd}

  ## Storage of MCMC output
  if (mnlp > 0){
    res.thetas[floor(mnlp/thin),] <- thetas.cur
    res.tau.sq[floor(mnlp/thin)] <- tau.sq.cur
    res.mu[floor(mnlp/thin)] <- mu.cur
    res.psi[floor(mnlp/thin)] <- psi.cur
  }

}

mean(res.mu)  ## Pooled log relative risk estimate adjusted for selection bias
sd(res.mu)    ## Standard error of pooled log relative risk estimate
```