**Supplemental Appendix**

Outcomes after Multivessel or Culprit-Vessel Intervention for

ST-Elevation Myocardial Infarction in Patients with Multivessel Coronary Disease:

A Bayesian Cross-Design Meta-Analysis

John A. Bittl, MD

Jacqueline E. Tamis-Holland, MD

Christopher D. Lang, MD

Yulei He, PhD

**Bayesian hierarchical meta-analysis**

A 3-level hierarchical structure (Fig. 1) modeled the variation in study outcomes as a function of treatment (multivessel vs. culprit-vessel PCI) and of study type (i.e., RCT vs. matched cohort vs. unmatched cohort):

(1)

where *θi*(*k*) are the study-level treatment effect effects, the sample variances that in many meta-analytic models can replace the variances , *θk* is the study-type level average effect, is the between-study variance for each design, *θ* is the global treatment effect that can be viewed as an average across all possible studies (nested within all possible designs), and *σ2*is the between-study type variance. In this case, prior distributions are required for *θ*, *σ2*, and (1).

Using Bayesian methods, when we have uncertainty about a parameter such as *θ*, we make inferences about it. When other uncertain parameters such as *θi* are dependent upon an uncertain parameter such as *θ*, we have a chain of uncertainty, formalized by a hierarchical model (2). The ﬁrst 2 equations deﬁned the random-effects meta-analysis models for studies separately within each design. In the 3rd equation, the study-type averages were treated as random effects from a normal distribution centered at the global average. The last equation imposes prior distributions for model parameters. This prior states that the *θ*s are like draws from the population of all possible *θ*s (the means of all possible study types). The precision parameter reflects how different the means from each study type are, and the parameter *θ* represents the overall mean of all possible study-type means, both of which need to be estimated. The third stage provides the priors and the unknown parameters from the second stage, as well as from the likelihood (3). Note that the model assumes that *θk*’s are exchangeable conditional on *θ* and *σ2*, while the separate meta-analysis strategy assumes that they are ﬁxed and independent parameters.

We used a 3-level model (Fig. 1) to account for the variation between studies as well as between types of studies (i.e., between study types). If we let *i* index the study and *k* index the type of study (*i* is nested within *k*): *k* = 1 for randomized studies, *k* = 2 for matched cohort studies, and *k* = 3 for other types of cohort studies. If we denote the within-study precision as, the likelihood is:

where *θi(k)* is the result of the *i*th study of the *k*th type and *θk* is the log-transformed mean of the *OR*s ofstudy type *k* (3). The unknown parameters in the likelihood are the 3 *θ*s and the . The prior for the set of *θ*s comprise the second stage. Although other choices are possible, the most common choice of parametric family for the second-stage prior on a set of normal means is the semi-conjugate normal prior (3):

*θ* as it is fairly straightforward to determine that

where k = 0, 1, 2, or 3. This shows that the estimator for *θ* can be expressed as a weighted average of *θk*’s, with the weight being proportional to the inverse of the corresponding

Following the guidance of prior reports (4,5), we chose some informative prior distributions for the model. Speciﬁcally, *θ ~ N*(0*,* 10) implied that the global summary odds ratio was unlikely to exceed 500 in favor of either MVI or CVI. More precisely, *τ*k ~Half-Normal (0*,* 0*.*362) implied that we were 95% sure that the true odds ratio for a study of a particular trial-design type would be within a range from 4 times to a 1/4 the overall odds ratio for that type; and *σ* ~Half-Normal (0*,* 0*.*182) implied that we were 95% sure that the true odds ratio for a trial-design type would be within a range from 2 times to a 1/2 the global odds ratio. Despite the subjectiveness of these priors, they represented reasonable guesses for the magnitudes and ranges of the parameters. On the other hand, because the number of study designs was small (only 3), both the mean and variance estimates for *θ* would be imprecise if they had been largely inferred from the data under vague priors for *θ* and *σ2*.

**Sensitivity Analyses**

Because we assumed that *θk ~N(θ, σ2)* in the hierarchical model (Fig. 1), this could be generalized to to imply that an RCT associated with *θ1* might have a smaller variance , and thus greater precision, than the corresponding variances of the observational studies, despite that they were all centered at the global *θ*. If we imposed a prior for drawn from the a half-normal distribution HN[0.362] and a prior distribution for and from HN[0,1],[[1]](#footnote-1) and thus attached 3-fold greater weight to the RCTs than to the observational studies (1).

A more flexible approach involves defining a new parameters known as the “evidence ratio” (ie, relative weight) to attach a smaller and thus a larger weight study effect *θ1*from the RCTs than to the study effects *θ2* or *θ3* from the observational studies. If we define the weighting factor for RCTs compared with matched cohort studies as and the weighting factor for RCTs compared with unmatched observational studies as , we can impose prior distributions on and to artiﬁcially affect the evidence ratio across different study types. For example, if Half-Normal[2, 0.52], it implies that evidence from randomized studies is likely to be valued 2 times as highly as that of matched cohort studies, but they could be valued as much as 3 (2 + 0*.*5*\**2) times more or just equally valued at 1 (2 *−* 2*\**0*.*5). Similarly if ~ Half-Normal[3, 1], it implies that evidence from RCTs is likely to be valued 3 times as highly as that of unmatched cohort studies, but they could be as much as 5 (3 + 1*\** 2) times more or just equally valued at 1 (5 *−* 2*\** 2). Under these considerations, the prior distributions for *θ*, *σ2*, and :

The posterior inferences for and were essentially determined from their prior distributions, because there was little information from the data to determine the evidence ratio across study types (1).

**[R] codes**

**Cross-design global summary**

#Specify the model in BUGS language, but save it as a string in [R]

modelString="

model

{

# K1 is the number of trials;

for (k in 1:18)

{

# calculate odds ratios;

or[k] <- ((r.multi[k]+0.5)/(n.multi[k]-r.multi[k]+0.5))/((r.culprit[k]+0.5)/(n.culprit[k]-r.culprit[k]+0.5))

logor[k] <- log(or[k]);

varlogor[k] <- (1/(r.multi[k]+0.5))+(1/(n.multi[k]-r.multi[k]+0.5))+(1/(r.culprit[k]+0.5))+(1/(n.culprit[k]-r.culprit[k]+0.5));

invlogor[k] <- 1/varlogor[k];

logor[k] ~ dnorm(theta[k], invlogor[k]);

or.est[k] <- exp(theta[k]);

# study-type level random-effects distributions

theta[k] ~ dnorm(mu.theta.study[study[k]], prec.theta.study[study[k]]);

}

# K2 is the number of study types

for (l in 1:3)

{

mu.theta.study[l] ~ dnorm(mu.theta, prec.theta);

or.theta.study[l] <- exp(mu.theta.study[l]);

prec.theta.study[l] <- 1/(tau.theta.study[l]\*tau.theta.study[l]);

# prior distribution for tau.theta.study based on HN[0.36^2], giving precision 7.72

tau.theta.study[l] ~ dnorm(0, 7.72)I(0,);

}

# prior distribution for mu.theta based on log(500)/1.96 = 3.17 for N[0,10], giving precision 0.1

mu.theta ~ dnorm(0, 0.1);

# prior distribution for tau.theta based on HN[0.18^2], giving precision 30.86

tau.theta ~ dnorm(0, 30.86)I(0,);

prec.theta <- 1/(tau.theta\*tau.theta);

# global summary odds ratio;

or.theta <- exp(mu.theta);

# K1 is the number of trials;

# DATA list(K1=21, K2=3);

# INITIAL VALUES list(mu.theta=0, tau.theta = 1);

# BUGS model specification ends

} .

"

# Write the modelString to a file

writeLines (modelString,con="model.txt")

# Use BRugs to check model

modelCheck ("model.txt")

#load data

dataList = list(n.multi=c(52, 65, 234, 150, 79, 503, 403, 26, 95, 147, 3134, 70, 217, 419, 1108, 442, 77, 367),

n.culprit=c(17, 84, 231, 146, 79, 503, 2418, 354, 25, 156, 25802, 707, 1984, 2118, 3833, 1467, 180, 706),

r.multi=c(1, 6, 12, 4, 19, 59, 41, 5, 9, 12, 246, 11, 27, 6, 81, 12, 2, 26),

r.culprit=c(0, 13, 16, 10, 13, 54, 164, 42, 2, 8, 1321, 57, 111, 72, 168, 40, 14, 127),

study=c(1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3)

)

#Use BRugs commands to put the data into a file and ship the file to BUGS

modelData(bugsData(dataList))

#Initialize the chains

nChain=1

modelCompile(numChains = nChain) #Compile the model

initsList = list(mu.theta=0, tau.theta=1)

modelInits(bugsData(initsList))

modelGenInits()

#R defines a new variable to specify an arbitrary chain length

chainLength1 = 5000

#BRugs tells BUGS to generate a MCMC chain

modelUpdate (chainLength1)

#BRugs keeps a record of parameters

samplesSet(c("mu.theta","prec.theta","or.theta","tau.theta"))

#BRugs asks BUGS for summary statistics

chainLength2 = 10000

thinStep = 2

modelUpdate (chainLength2)

thetaSummary = samplesStats (c("mu.theta","prec.theta","or.theta","tau.theta"));

print(thetaSummary)

**output**

> source("Z:\\Users\\jabittl\\Dropbox\\BayesCulpritCCI\\BRugs18StudiesCrossDesign.R")

model is syntactically correct

data loaded

model compiled

Initializing chain 1:

initial values loaded but chain contain uninitialized variables

initial values generated, model initialized

5000 updates took 0 s

monitor set for variable 'mu.theta'

monitor set for variable 'prec.theta'

monitor set for variable 'or.theta'

monitor set for variable 'tau.theta'

10000 updates took 0 s

mean sd MC\_error val2.5pc median val97.5pc start sample

mu.theta **8.358e-02 1.803e-01** 6.970e-03 -0.301500 0.09143 4.089e-01 5001 10000

prec.theta 1.080e+05 2.844e+06 8.695e+04 6.367000 47.98000 2.273e+04 5001 10000

or.theta 1.105e+00 1.959e-01 7.564e-03 **0.739700 1.09600 1.505e+00** 5001 10000

tau.theta 1.579e-01 1.063e-01 3.704e-03 0.006652 0.14440 3.964e-01 5001 10000

>

>

**4 RCTs meta**

#Export data from Excel in tab-delimited, semicolon- or comma-separated

#form ? file ending in “csv” (see manual “R Data Import/Export”)

bdat<-read.csv("Z:/Users/jabittl/Dropbox/BayesCulpritCCI/4RCT.csv",as.is=TRUE, header=T)

str(bdat)

study\_name<-c(bdat$study\_name)

r.multi<-c(bdat$r.multi)

n.multi<-c(bdat$n.multi)

r.culprit<-c(bdat$r.culprit)

n.culprit<-c(bdat$n.culprit)

mdat<-data.frame(study\_name,n.multi,n.culprit,r.multi,r.culprit)

mdat

m1 = metabin(r.multi, n.multi, r.culprit, n.culprit, sm = "OR", data = mdat, studlab = study\_name)

str(m1)

class(m1)

m1

summary(m1)

forest(m1,col.square="grey",col.diamond="grey",col.i.inside.square="black",rightcols=c("effect", "ci"),lab.e="Multivessel",lab.c="Culprit",xlim=c(0.1,10),xlab="Multivessel PCI better Culprit-vessel PCI better")

dev.copy2eps(file="Meta4RCT.eps")

dev.copy2pdf(file="Meta4RCT.pdf")

**4 RCTs BRugs**

#get csv file

rctdat<-read.csv("Z:/Users/jabittl/Dropbox/BayesCulpritCCI/4RCT.csv",as.is=TRUE, header=T)

str(rctdat)

study\_name<-c(rctdat$study\_name)

r.multi<-c(rctdat$r.multi)

n.multi<-c(rctdat$n.multi)

r.culprit<-c(rctdat$r.culprit)

n.culprit<-c(rctdat$n.culprit)

#Specify the model in BUGS language, but save it as a string in [R]

modelString="

model

{

# K1 is the number of trials;

for (k in 1:4)

{

# calculate odds ratios;

or[k] <- ((r.multi[k]+0.5)/(n.multi[k]-r.multi[k]+0.5))/((r.culprit[k]+0.5)/(n.culprit[k]-r.culprit[k]+0.5))

logor[k] <- log(or[k]);

varlogor[k] <- (1/(r.multi[k]+0.5))+(1/(n.multi[k]-r.multi[k]+0.5))+(1/(r.culprit[k]+0.5))+(1/(n.culprit[k]-r.culprit[k]+0.5));

invlogor[k] <- 1/varlogor[k]; #variance;

logor[k] ~ dnorm(theta[k], invlogor[k]);

or.est[k] <- exp(theta[k]);

theta[k] ~ dnorm(mu.theta, prec.theta); # random effects distribution;

}

mu.theta ~ dnorm(0, 0.001); # uninformative prior distribution

prec.theta ~ dgamma(0.001, 0.001); # uninformative prior distribution;

or.theta <- exp(mu.theta);

# probability of mean effect greater than zero;

pmu0 <- equals(min(mu.theta,0),0);

theta.new ~ dnorm(mu.theta, prec.theta); # predicted theta for a new study;

or.new <- exp(theta.new); # calculate the new OR;

# BUGS model specification ends

}

"

# Write the modelString to a file

writeLines (modelString,con="model.txt")

# Use BRugs to check model

modelCheck ("model.txt")

#load data

dataList = list(n.multi=c(n.multi),

n.culprit=c(n.culprit),

r.multi=c(r.multi),

r.culprit=c(r.culprit)

)

#Use BRugs commands to put the data into a file and ship the file to BUGS

modelData(bugsData(dataList))

#Initialize the chains

nChain=1

modelCompile(numChains = nChain) #Compile the model

initsList = list(mu.theta=(-0.32), prec.theta=1)

modelInits(bugsData(initsList))

modelGenInits()

#R defines a new variable to specify an arbitrary chain length

chainLength1 = 5000

#BRugs tells BUGS to generate a MCMC chain

modelUpdate (chainLength1)

#BRugs keeps a record of parameters

samplesSet(c("mu.theta","or.new","prec.theta","or.theta","theta.new"))

#BRugs asks BUGS for summary statistics

chainLength2 = 10000

thinStep = 2

modelUpdate (chainLength2)

thetaSummary = samplesStats (c("mu.theta","or.new","prec.theta","or.theta","theta.new")); thetaSummary

print(thetaSummary)

output

> source("Z:\\Users\\jabittl\\Dropbox\\BayesCulpritCCI\\BRugs4RCT.R")

'data.frame': 4 obs. of 7 variables:

$ study\_name : chr "Di Mario (2004)" "Politi (2010)" "Wald (2013)" "Gershlick (2014)"

$ Study.yr : int 2004 2010 2013 2014

$ F.U..days. : int 365 900 700 365

$ n.multi.. : int 52 65 234 150

$ n.culprit..: int 17 84 231 146

$ r.multi.. : int 1 6 12 4

$ r.culprit..: int 0 13 16 10

model is syntactically correct

data loaded

model compiled

Initializing chain 1:

initial values loaded but chain contain uninitialized variables

initial values generated, model initialized

5000 updates took 0 s

monitor set for variable 'mu.theta'

monitor set for variable 'or.new'

monitor set for variable 'prec.theta'

monitor set for variable 'or.theta'

monitor set for variable 'theta.new'

10000 updates took 0 s

mean sd MC\_error val2.5pc median val97.5pc start sample

mu.theta **-0.5158 0.3738** 0.01154 -1.1810 -0.5127 0.1789 5001 10000

or.new 1.4460 56.7000 0.57080 0.2030 0.6032 1.7810 5001 10000

prec.theta 175.8000 383.6000 9.06000 0.5712 38.0600 1227.0000 5001 10000

or.theta 0.6481 0.6747 0.01009 0**.3069 0.5989 1.1960** 5001 10000

theta.new -0.5103 0.6167 0.01226 -1.5940 -0.5055 0.5770 5001 10000

>

>

**11 unmatched cohort meta**

#Export data from Excel in tab-delimited, semicolon- or comma-separated

#form ? file ending in “csv” (see manual “R Data Import/Export”) into file called bdat

bdat<-read.csv("Z:/Users/jabittl/Dropbox/BayesCulpritCCI/11Obs.csv",as.is=TRUE, header=T)

str(bdat)

study\_name<-c(bdat$study\_name)

r.multi<-c(bdat$r.multi)

n.multi<-c(bdat$n.multi)

r.culprit<-c(bdat$r.culprit)

n.culprit<-c(bdat$n.culprit)

mdat<-data.frame(study\_name,n.multi,n.culprit,r.multi,r.culprit)

mdat

m1 = metabin(r.multi, n.multi, r.culprit, n.culprit, sm = "OR", data = mdat, studlab = study\_name)

str(m1)

class(m1)

m1

summary(m1)

forest(m1,col.square="grey",col.diamond="grey",rightcols=c("effect", "ci"),lab.e="MULTIVESSEL",lab.c="CULPRIT",xlim=c(0.1,10),xlab="Multivessel better Culprit-vessel better")

dev.copy2eps(file="Meta11Obs.eps")

dev.copy2pdf(file="Meta11Obs.pdf")

**11 unmatched cohort Brugs**

#read csv file:

unmatcheddat<-read.csv("Z:/Users/jabittl/Dropbox/BayesCulpritCCI/11Obs.csv",as.is=TRUE, header=T)

str(unmatcheddat)

study\_name<-c(unmatcheddat$study\_name)

r.multi<-c(unmatcheddat$r.multi)

n.multi<-c(unmatcheddat$n.multi)

r.culprit<-c(unmatcheddat$r.culprit)

n.culprit<-c(unmatcheddat$n.culprit)

#Specify the model in BUGS language, but save it as a string in [R]

modelString="

model

{

# K1 is the number of trials;

for (k in 1:11)

{

# calculate odds ratios;

or[k] <- ((r.multi[k]+0.5)/(n.multi[k]-r.multi[k]+0.5))/((r.culprit[k]+0.5)/(n.culprit[k]-r.culprit[k]+0.5))

logor[k] <- log(or[k]);

varlogor[k] <- (1/(r.multi[k]+0.5))+(1/(n.multi[k]-r.multi[k]+0.5))+(1/(r.culprit[k]+0.5))+(1/(n.culprit[k]-r.culprit[k]+0.5));

invlogor[k] <- 1/varlogor[k]; #variance;

logor[k] ~ dnorm(theta[k], invlogor[k]);

or.est[k] <- exp(theta[k]);

theta[k] ~ dnorm(mu.theta, prec.theta); # random effects distribution;

}

mu.theta ~ dnorm(0, 0.001); # uninformative prior distribution

prec.theta ~ dgamma(0.001, 0.001); # uninformative prior distribution;

or.theta <- exp(mu.theta);

# probability of mean effect greater than zero;

pmu0 <- equals(min(mu.theta,0),0);

theta.new ~ dnorm(mu.theta, prec.theta); # predicted theta for a new study;

or.new <- exp(theta.new); # calculate the new OR;

# BUGS model specification ends

}

"

# Write the modelString to a file

writeLines (modelString,con="model.txt")

# Use BRugs to check model

modelCheck ("model.txt")

#load data

dataList = list(n.multi=c(n.multi),

n.culprit=c(n.culprit),

r.multi=c(r.multi),

r.culprit=c(r.culprit)

)

#Use BRugs commands to put the data into a file and ship the file to BUGS

modelData(bugsData(dataList))

#Initialize the chains

nChain=1

modelCompile(numChains = nChain) #Compile the model

initsList = list(mu.theta=0, prec.theta=1)

modelInits(bugsData(initsList))

modelGenInits()

#R defines a new variable to specify an arbitrary chain length

chainLength1 = 5000

#BRugs tells BUGS to generate a MCMC chain

modelUpdate (chainLength1)

#BRugs keeps a record of parameters

samplesSet(c("mu.theta","or.new","prec.theta","or.theta","theta.new"))

#BRugs asks BUGS for summary statistics

chainLength2 = 10000

thinStep = 2

modelUpdate (chainLength2)

thetaSummary = samplesStats (c("mu.theta","or.new","prec.theta","or.theta","theta.new")); thetaSummary

print(thetaSummary)

output

model is syntactically correct

data loaded

model compiled

Initializing chain 1:

initial values loaded but chain contain uninitialized variables

initial values generated, model initialized

5000 updates took 0 s

monitor set for variable 'mu.theta'

monitor set for variable 'or.new'

monitor set for variable 'prec.theta'

monitor set for variable 'or.theta'

monitor set for variable 'theta.new'

10000 updates took 0 s

mean sd MC\_error val2.5pc median val97.5pc start sample

mu.theta **0.1399 0.2456** 0.003603 -0.3621 0.1440 0.6177 5001 10000

or.new 1.5340 1.7310 0.015980 0.2436 1.1580 5.0760 5001 10000

prec.theta 2.9000 1.9010 0.037330 0.6848 2.4470 7.8490 5001 10000

or.theta 1.1850 0.2993 0.004221 **0.6962 1.1550 1.8550** 5001 10000

theta.new 0.1356 0.7475 0.007919 -1.4120 0.1467 1.6240 5001 10000

3 **matched meta**

#Export data from Excel in tab-delimited, semicolon- or comma-separated

#form ? file ending in “csv” (see manual “R Data Import/Export”) into file called bdat

bdat<-read.csv("Z:/Users/jabittl/Dropbox/BayesCulpritCCI/3Matched.csv",as.is=TRUE, header=T)

str(bdat)

study\_name<-c(bdat$study\_name)

r.multi<-c(bdat$r.multi)

n.multi<-c(bdat$n.multi)

r.culprit<-c(bdat$r.culprit)

n.culprit<-c(bdat$n.culprit)

mdat<-data.frame(study\_name,n.multi,n.culprit,r.multi,r.culprit)

mdat

m1 = metabin(r.multi, n.multi, r.culprit, n.culprit, sm = "OR", data = mdat, studlab = study\_name)

str(m1)

class(m1)

m1

summary(m1)

forest(m1,col.square="grey",col.diamond="grey",rightcols=c("effect", "ci"),lab.e="MULTIVESSEL",lab.c="CULPRIT",xlim=c(0.1,10),xlab="Multivessel better Culprit-vessel better")

dev.copy2eps(file="Meta3Matched.eps")

dev.copy2pdf(file="Meta3Matched.pdf")

**3 matched cohort BRugs**

#Get csv data:

matcheddat<-read.csv("Z:/Users/jabittl/Dropbox/BayesCulpritCCI/3Matched.csv",as.is=TRUE, header=T)

str(matcheddat)

study\_name<-c(matcheddat$study\_name)

r.multi<-c(matcheddat$r.multi)

n.multi<-c(matcheddat$n.multi)

r.culprit<-c(matcheddat$r.culprit)

n.culprit<-c(matcheddat$n.culprit)

#Specify the model in BUGS language, but save it as a string in [R]

modelString="

model

{

# K1 is the number of trials;

for (k in 1:3)

{

# calculate odds ratios;

or[k] <- ((r.multi[k]+0.5)/(n.multi[k]-r.multi[k]+0.5))/((r.culprit[k]+0.5)/(n.culprit[k]-r.culprit[k]+0.5))

logor[k] <- log(or[k]);

varlogor[k] <- (1/(r.multi[k]+0.5))+(1/(n.multi[k]-r.multi[k]+0.5))+(1/(r.culprit[k]+0.5))+(1/(n.culprit[k]-r.culprit[k]+0.5));

invlogor[k] <- 1/varlogor[k]; #variance;

logor[k] ~ dnorm(theta[k], invlogor[k]);

or.est[k] <- exp(theta[k]);

theta[k] ~ dnorm(mu.theta, prec.theta); # random effects distribution;

}

mu.theta ~ dnorm(0, 0.001); # uninformative prior distribution

prec.theta ~ dgamma(0.001, 0.001); # uninformative prior distribution;

or.theta <- exp(mu.theta);

# probability of mean effect greater than zero;

pmu0 <- equals(min(mu.theta,0),0);

theta.new ~ dnorm(mu.theta, prec.theta); # predicted theta for a new study;

or.new <- exp(theta.new); # calculate the new OR;

# BUGS model specification ends

}

"

# Write the modelString to a file

writeLines (modelString,con="model.txt")

# Use BRugs to check model

modelCheck ("model.txt")

#load data

dataList = list(n.multi=c(n.multi),

n.culprit=c(n.culprit),

r.multi=c(r.multi),

r.culprit=c(r.culprit)

)

#Use BRugs commands to put the data into a file and ship the file to BUGS

modelData(bugsData(dataList))

#Initialize the chains

nChain=1

modelCompile(numChains = nChain) #Compile the model

initsList = list(mu.theta=0, prec.theta=1)

modelInits(bugsData(initsList))

modelGenInits()

#R defines a new variable to specify an arbitrary chain length

chainLength1 = 5000

#BRugs tells BUGS to generate a MCMC chain

modelUpdate (chainLength1)

#BRugs keeps a record of parameters

samplesSet(c("mu.theta","or.new","prec.theta","or.theta","theta.new"))

#BRugs asks BUGS for summary statistics

chainLength2 = 10000

thinStep = 2

modelUpdate (chainLength2)

thetaSummary = samplesStats (c("mu.theta","or.new","prec.theta","or.theta","theta.new")); thetaSummary

print(thetaSummary)

**output**

> source("Z:\\Users\\jabittl\\Dropbox\\BayesCulpritCCI\\BRugs3Matched.R")

'data.frame': 3 obs. of 17 variables:

$ study\_name : chr "Roe (2001)" "Hannan (2010)" "Iqbal (2014)"

$ Study.yr : int 2001 2010 2014

$ F.U..days. : int 180 1278 365

$ n.multi.. : int 79 503 403

$ n.culprit.. : int 79 503 2418

$ r.multi.. : int 19 59 41

$ r.culprit.. : int 13 54 164

$ OR : num 1.59 1.1 1.57

$ loge.OR. : num 0.46 0.1 0.45

$ V.theta. : num 0.15 0.04 0.03

$ m..s.2. : num 26.6 102.2 121

$ study.. : num 2 2 2

$ X0 : num 12.3 10.1 54.5

$ Lower : num 0.81 0.81 1.27

$ Upper : num 3.45 1.63 2.24

$ multi.rate : num 24.1 11.7 10.2

$ culprit.rate: num 16.46 10.74 6.78

model is syntactically correct

data loaded

model compiled

Initializing chain 1:

initial values loaded but chain contain uninitialized variables

initial values generated, model initialized

5000 updates took 0 s

monitor set for variable 'mu.theta'

monitor set for variable 'or.new'

monitor set for variable 'prec.theta'

monitor set for variable 'or.theta'

monitor set for variable 'theta.new'

10000 updates took 0 s

mean sd MC\_error val2.5pc median val97.5pc start sample

mu.theta **0.3172 0.3372** 0.004715 -0.1455 0.3156 0.8054 5001 10000

or.new 11.6200 787.2000 7.852000 0.5766 1.3670 3.1940 5001 10000

prec.theta 192.1000 384.5000 7.793000 0.7671 49.4500 1313.0000 5001 10000

or.theta 2.3420 74.6700 0.745400 **0.8646 1.3710 2.2380** 5001 10000

theta.new 0.3105 0.5391 0.006881 -0.5506 0.3125 1.1610 5001 10000

>

Alpha model

#Specify the model in BUGS language, but save it as a string in [R]

modelString="

model

{

# K1 is the number of trials;

for (k in 1:18)

{

# calculate odds ratios;

or[k] <- ((r.multi[k]+0.5)/(n.multi[k]-r.multi[k]+0.5))/((r.culprit[k]+0.5)/(n.culprit[k]-r.culprit[k]+0.5))

logor[k] <- log(or[k]);

varlogor[k] <- (1/(r.multi[k]+0.5))+(1/(n.multi[k]-r.multi[k]+0.5))+(1/(r.culprit[k]+0.5))+(1/(n.culprit[k]-r.culprit[k]+0.5));

invlogor[k] <- 1/varlogor[k];

logor[k] ~ dnorm(theta[k], invlogor[k]);

or.est[k] <- exp(theta[k]);

# study-type level random-effects distributions

theta[k] ~ dnorm(mu.theta.study[study[k]], prec.theta.study[study[k]]);

}

# K2 is the number of study types

for (l in 1:3)

{

mu.theta.study[l] ~ dnorm(mu.theta, prec.theta[l]);

or.theta.study[l] <- exp(mu.theta.study[l]);

# mu.theta.study[l] ~ dnorm (0, 0.001);

prec.theta[l] <- 1/(tau.theta[l]\*tau.theta[l]);

prec.theta.study[l] <- 1/(tau.theta.study[l]\*tau.theta.study[l]);

# prior distribution for tau.theta.study based on HN[0.36^2], giving precision 7.72

tau.theta.study[l] ~ dnorm(0, 7.72)I(0,);

}

# ratio of the RCT odds ratio vs. cohort odds ratio;

rr\_rct\_match<-exp(mu.theta.study[1]-mu.theta.study[2]);

rr\_rct\_cohort<-exp(mu.theta.study[1]-mu.theta.study[3]);

#prior distributions

# prior distribution for mu.theta based on log(500)/1.96 = 3.17 for N[0,10], giving precision 0.1

mu.theta ~ dnorm(0, 0.1);

# prior distribution for tau.theta based on HN[0.18^2], giving precision 30.86

tau.theta[1] ~ dnorm(0, 30.86)I(0,);

tau.theta[2] <- pow(a2\*pow(tau.theta[1],2),1/2);

tau.theta[3] <- pow(a3\*pow(tau.theta[1],2),1/2);

a2 ~ dnorm(2,4)I(0,);

a3 ~ dnorm(3,1)I(0,);

#prec.theta <- 1/(tau.theta\*tau.theta);

# global summary odds ratio;

or.theta <- exp(mu.theta);

# K1 is the number of trials;

# DATA list(K1=21, K2=3);

# INITIAL VALUES list(mu.theta=0, tau.theta = 1);

# BUGS model specification ends

} .

"

# Write the modelString to a file

writeLines (modelString,con="model.txt")

# Use BRugs to check model

modelCheck ("model.txt")

#load data

dataList = list(n.multi=c(52, 65, 234, 150, 79, 503, 403, 26, 95, 147, 3134, 70, 217, 419, 1108, 442, 77, 367),

n.culprit=c(17, 84, 231, 146, 79, 503, 2418, 354, 25, 156, 25802, 707, 1984, 2118, 3833, 1467, 180, 706),

r.multi=c(1, 6, 12, 4, 19, 59, 41, 5, 9, 12, 246, 11, 27, 6, 81, 12, 2, 26),

r.culprit=c(0, 13, 16, 10, 13, 54, 164, 42, 2, 8, 1321, 57, 111, 72, 168, 40, 14, 127),

study=c(1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3)

)

#Use BRugs commands to put the data into a file and ship the file to BUGS

modelData(bugsData(dataList))

#Initialize the chains

nChain=1

modelCompile(numChains = nChain) #Compile the model

initsList = list(mu.theta=0, tau.theta=1)

modelInits(bugsData(initsList))

modelGenInits()

#R defines a new variable to specify an arbitrary chain length

chainLength1 = 5000

#BRugs tells BUGS to generate a MCMC chain

modelUpdate (chainLength1)

#BRugs keeps a record of parameters

samplesSet(c("mu.theta","prec.theta","or.theta","tau.theta"))

#BRugs asks BUGS for summary statistics

chainLength2 = 10000

thinStep = 2

modelUpdate (chainLength2)

thetaSummary = samplesStats (c("mu.theta","prec.theta","or.theta","tau.theta"));

print(thetaSummary)

**output**

> source("Z:\\Users\\jabittl\\Dropbox\\BayesCulpritCCI\\BRugs18StudiesCulpritCrossDesignWeightedAlpha.R")

model is syntactically correct

data loaded

model compiled

Initializing chain 1:

expected the collection operator c error pos 37 (error on line 1)

initial values generated, model initialized

5000 updates took 0 s

monitor set for variable 'mu.theta'

monitor set for variable 'prec.theta'

monitor set for variable 'or.theta'

monitor set for variable 'tau.theta'

10000 updates took 0 s

mean sd MC\_error val2.5pc median val97.5pc start sample

mu.theta **2.931e-02 2.121e-01** 8.317e-03 -0.443000 0.04987 3.922e-01 5001 10000

prec.theta[1] 4.459e+03 1.556e+05 2.606e+03 6.438000 54.53000 1.002e+04 5001 10000

prec.theta[2] 2.140e+03 7.179e+04 1.190e+03 3.094000 28.78000 5.500e+03 5001 10000

prec.theta[3] 1.318e+03 3.482e+04 6.184e+02 2.064000 19.98000 3.849e+03 5001 10000

or.theta 1.052e+00 2.153e-01 8.453e-03 **0.642100 1.05100 1.480e+00** 5001 10000

tau.theta[1] 1.519e-01 1.051e-01 4.150e-03 0.009999 0.13540 3.944e-01 5001 10000

tau.theta[2] 2.134e-01 1.510e-01 5.883e-03 0.013490 0.18660 5.685e-01 5001 10000

tau.theta[3] 2.561e-01 1.838e-01 6.958e-03 0.016190 0.22380 6.962e-01 5001 10000

>

>

Key

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Parameter | Definition | Bittl DM | He text | He code | Lunn | Spieg… |
| Historical data | Individual  trial results | yH | ORi |  |  |  |
| Pooled historical data | Pooled trial results |  | θi |  |  |  |
| Underlying treatment effect for prior |  | θ | θ | mu.theta | µ |  |
| Variance for θi or µ |  | σ2h | s2i | prec.theta | ω2 |  |
| Effective sample size for prior |  | m0 | i |  |  |  |
| Effective sample size for posterior |  | η0 |  |  |  |  |
| OR for new trial | exp(theta.new) |  |  | or.new |  |  |
| θ for new trial |  |  |  | theta.new |  |  |
| Variance on θ |  |  | τ2 | tau.theta |  |  |
| Variance on θk |  |  | σk | tau.theta.study |  |  |

1. He Y, Bittl JA, Wouhib A, Normand S-LT. 2014. Case study in cardiovascular medicine: unprotected left main coronary artery disease. In: Biondi-Zoccai G, editor. Network Meta-Analysis: Evidence Synthesis with Mixed Treatment Comparison. New York: Nova Science Publishers, Inc.

2. Kruschke JK. 2011. Doing Bayesian Data Analysis: A Tutorial with R and BUGS. Oxford, England: Elsevier.

3. Cowles MK. 2013. Applied Bayesian Statistics. Casella G, Fienberg SE, Olkin I, editors. New York: Springer.

4. Peters JL, Rushton L, Sutton AJ, Jones DR, Abrams KR, Mugglestone MA. Bayesian methods for the cross-design synthesis of epidemiological and toxocological evidence. Appl Stat 2005;54:159-172.

5. Prevost TC, Abrams KR, Jones DR. Hierarchical models in generalized synthesis of evidence: an example based on studies of breast cancer screening. Stat in Med 2000;19:3359-3376.

6. Spiegelhalter DJ, Abrams KR, Myles JP. 2004. Bayesian Approaches to Clinical Trials and Health Care Evaluations. Chichester, England: Wiley.

1. The half-normal distribution arises from folding a normal distribution around 0. If is drawn from a normal distribution Normal[2, 0.52], then Half-Normal[0.52] 6. Spiegelhalter DJ, Abrams KR, Myles JP. 2004. Bayesian Approaches to Clinical Trials and Health Care Evaluations. Chichester, England: Wiley.. [↑](#footnote-ref-1)