Cross-design and cross-data format synthesis using Network Meta-Analysis (crosnma)

Tasnim Hamza

6/7/2021

library(crosnma)

# Introduction

Network meta-analysis (NMA) is a common tool to compare multiple interventions simultaneously. This requires synthesize the available evidence that might come as different study design or data format. As different study designs, the evidence can be obtained from non-randomized studies (NRS) or randomized controlled trials (RCT). RCT evidence quality is considered the highest and at a lower risk of bias, however, the restricted settings when RCTs are conducted limit the generalizability of their results. Therefore, combining RCT and NRS evidence in NMA may help overcome some of the limitations. Such evidence can be accessible in different data formats as individual participant data (IPD) or aggregate data (AD). We establish a cross-design and cross-data format package (crosnma) to utilize all available and relevant evidence.

We describe the R package crosnma in a worked example about relapsing remitting multiple sclerosis (RRMS). The primary outcome is the occurrence of relapses in two years (binary outcome, 0/1). In the analysis, the outcome will be expressed as odds ratio (OR). The aim is to compare the efficacy of four treatments. Data is available from 6 different studies in different formats and different designs. The studies are either available as study-level data std.data (2 studies) or as individual participant data prt.data (four studies). Three of the prt.data are RCTs and one comes from an observational study (NRS). The two std.data are RCTs.

The prt.data contains 2950 rows, each refers to a participant in the study. We display the first few rows of the data set:

head(prt.data)  
## study outcome trt design age sex bias year  
## 1 1 0 D rct 20 1 low 2002  
## 2 1 0 D rct 29 0 low 2002  
## 3 1 0 D rct 35 0 low 2002  
## 4 1 0 D rct 38 0 low 2002  
## 5 1 0 D rct 37 0 low 2002  
## 6 1 0 D rct 32 0 low 2002

For each participant, we have information for the occurrence of the relapses (0=no, 1=yes) outcome, the treatment label trt, the age (in years) age and sex (0 = Female, 1 = Male) sex of the participant. The other following columns are set on study-level (it is repeated for each participant of the study): the study id study, the design of the study ( needs to be either rct or nrs) design, the risk of bias on each study (set as a low, high or unclear) bias and the year of publication year.

We display the full data that is available on study-level:

head(std.data)  
## study outcome n trt design age sex bias year  
## 1 1 19 25 A rct 34.3 0.2 low 2010  
## 2 1 11 25 C rct 34.3 0.3 low 2010  
## 3 2 97 126 A rct 30.0 0.4 unclear 2015  
## 4 2 89 125 C rct 30.0 0.5 unclear 2015

For all studies, we provide the following information on each arm level: the total number of participants n and the number of those who relapsed outcome, the treatment label trt. At study-level, we have the study id study, the design of the study design, the mean age age, the percentage of males sex, the risk of bias on the study bias and the year of publication year.

# The synthesis model of individual participant data (IPD) and aggregate data (AD)

We introduce the model to synthesis individual participant data (IPD) and aggregate data (AD). Each data format could come from randomized clinical trials (RCT) or non-randomized studies (NRS). These two designs could be combined in one of four different methods. We describe the four synthesis models for network meta-regression. However, we can run the network meta-analysis by decline the covariates terms from the models. Let’s begin with describing the notations that will be used in the description of the models.

|  |  |  |
| --- | --- | --- |
| Notation | Description | Argument in crosnma.model() |
|  | participant id |  |
|  | study id |  |
|  | treatment index |  |
|  | number of studies with index denote the design or format of the study |  |
|  | binary outcome (0/1) | outcome |
|  | probability of the event to occur |  |
|  | the number of events per arm | outcome |
|  | the sample size per arm | n |
|  | the study-specific reference | \* |
|  | The treatment effect of the study-specific reference |  |
|  | log(OR) of treatment k relative to |  |
|  | the covariate | covariate |
|  | the mean covariate for study |  |
|  | the basic parameters, (A is the reference in the network) | use reference to assign the reference treatment |
|  | study characteristics to estimate the bias probability | bias.covariate |
|  | inflation factor of variance for the NRS estimates | the element var.infl in run.nrs |
|  | mean shift of the NRS estimates | the element mean.shift in run.nrs |

\*it is assigned automatically to the reference in the network (), if it is available on the study. If not, it is assigned automatically to the first alphabetically treatment on the study

## Naive synthesis

We synthesis the evidence from RCT and NRS without acknowledging the differences between them. For both of them, we model the IPD data and the AD data as the following.

**model IPD only**

**model AD only**

## Using NRS as a prior

We run the model in Section 2.1 two consequent times. First we estimate the relative treatment effects only using the NRS (use run.nrs in crosnma.model() to control the process). Second, we use the NRS estimates () as a prior information for the basic parameters of RCT data, . To control the NRS influence in the RCT estimates, we can either inflate the NRS variance by (the inflated variance is ) or shift the NRS means by .

## RoB-adjusted model 1

We incorporate the risk of bias (RoB) into the IPD and the AD models by adding the bias term to each.

**model IPD only**

**model AD only**

The bias indicator

## RoB-adjusted model 2

Another way to incorporate the study RoB is by adding the bias adjusted relative treatment effect instead of the unadjusted one . Then to model by a bi-modal normal distribution as in Table 2 below. We estimate the bias probability by either assigning a beta distribution or by using a study characteristics through a logistic transformation.

**model IPD only**

**model AD only**

**Combine IPD and AD**

The table below summarizes the different assumptions implemented in the package about combining the parameters in the model described the previous section.

|  |  |  |
| --- | --- | --- |
| Parameter | Assumptions | Argument in crosnma.model() |
| relative treatment effect () | Random-effects: | trt.effect='random' |
|  | Common-effect: | trt.effect='common' |
| Covariate effect | Independent effects: | reg0.effect='independent' |
|  | Random-effects: | reg0.effect='random' |
| Within-study covariate-treatment interaction () | Random-effects: | regw.effect='random' |
|  | Common-effect: | regw.effect='common' |
| Between-study covariate-treatment interaction () | Random-effects: | regb.effect='random' |
|  | Common-effect: | regb.effect='common' |
| bias-adjusted relative treatment effect () | Random-effects: | trt.effect='random' |
| Bias effect () | Random-effects: | bias.effect='random' |
|  | Common-effect: | bias.effect='common' |
| Bias probability () |  |  |
|  |  |  |

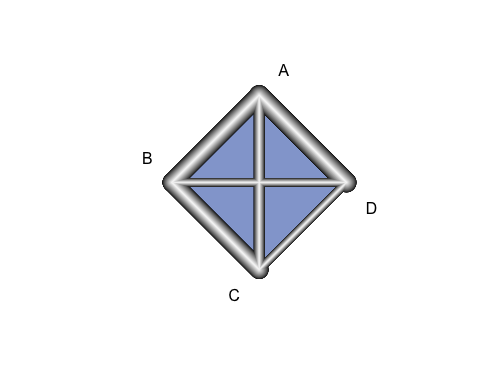
# Explore the network

## Check network connectivity

The network should be checked for its connectivity before running the analysis. This is a vital step as the model will run even if the network is not connected.

**Network plot**

netplot(prt.data,std.data)



## Network characteristics

In the following table, we summarize the number of studies from each design and each data format:

knitr::kable(ns.tab(prt.data,std.data))

|  |  |  |
| --- | --- | --- |
|  | IPD | AD |
| RCT | 3 | 2 |
| NRS | 1 | 0 |

# The R implementation

There are two steps to run the NMA/NMR model. The first step is to create JAGS model using crosnma.model() which involves JAGS code and the reformatted data. In the second step, the output of that function will be used in crosnma.run() to run the analysis through JAGS [@plummer\_jags].

## Naïve synthesis

This is the simplest method where the evidence from the two designs are combined naively without acknowledging the differences between them, see Section 2.2.1 in [@hamza\_2021].

**Set up JAGS model**

We start by indicating the names of the datasets on participant- (prt.data) and study-level (std.data). Then, the name of the variables on each dataset need to be given respectively in prt.data and std.data. Next, the reference treatment need to be assigned ( drug A). By choosing trt.effect=random, we assign a normal distribution to each relative treatment effect across studies, see the table in Section 2.1. Finally, the different designs; RCT and NRS are combined naively; method.bias = 'naive'.

# jags model: code+data  
 mod1 <- crosnma.model(prt.data=prt.data,  
 std.data=std.data,  
 trt=c('trt','trt'),  
 study=c('study','study'),  
 outcome=c('outcome','outcome'),  
 n='n',  
 design=c('design','design'),  
 trt.effect='random',  
 reference='A',  
 method.bias = 'naive'  
 )

**Run JAGS**

Next, we fit the NMA model using crosnma.run()which requires the number of adaptations, iterations, thinning and chains to be set.

# run jags  
jagsfit1 <- crosnma.run(model=mod1,  
 n.adapt = 20,  
 n.iter=50,  
 thin=1,  
 n.chains=3)  
## NOTE: Stopping adaptation

**Output**

*Table of estimates*

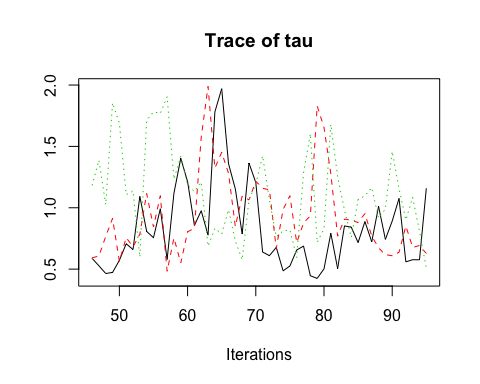
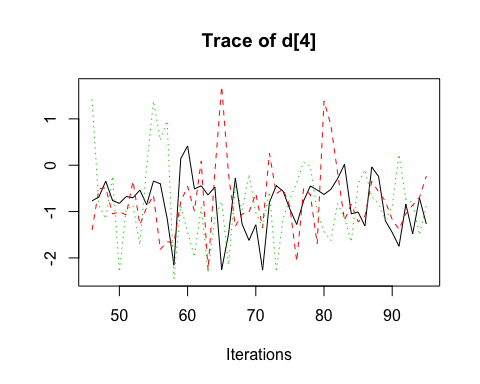
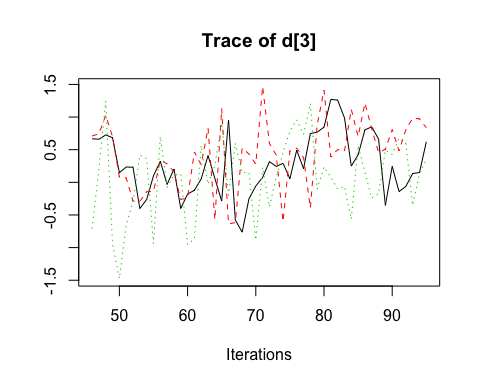
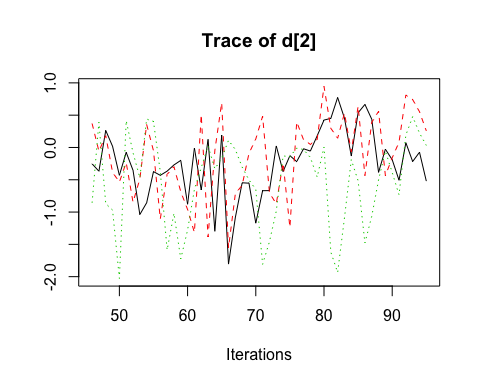
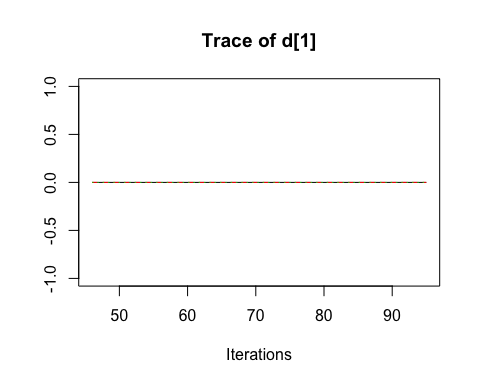
We summarize the estimated parameters by the following table.

knitr::kable(summary(jagsfit1,expo=T,))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Mean | SD | 2.5% | 50% | 97.5% | Rhat | n.eff |
| d.A | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | NaN | 0 |
| d.B | 0.739 | 1.873 | 0.174 | 0.804 | 2.027 | 1.084 | 151 |
| d.C | 1.291 | 1.730 | 0.407 | 1.302 | 3.478 | 1.078 | 95 |
| d.D | 0.442 | 2.090 | 0.103 | 0.437 | 2.921 | 1.014 | 156 |
| tau | 0.964 | 0.363 | 0.480 | 0.877 | 1.837 | 1.122 | 48 |

*Trace plot of estimates*

coda::traceplot(jagsfit1$samples)



## Using NRS as a prior

In this part, we run a generic NMR model using sex and age as covariates. To combine both designs, we use NRS evidence as a prior information for the RCT likelihood, see Section 2.2.2 in [@hamza\_2021].

**Set up JAGS model**

In this case, three additional arguments are needed; the first one is to run network meta-regression and the rest are to control combining RCT and NRS.

1. We set a list of 2 covariates in prt.data and std.data, respectively, covariate=list(c('age','sex'),c('age','sex')).
2. We indicate using NRS as a priormethod.bias='prior'.
3. This means that a NMA is initially run with only NRS data. This requires the following MCMC settings: the number of adaptations, iterations, burn-ins, thinning and chains.

In this method, the prior of the RCT basic parameters is set to a normal distribution that is either a minimally informative prior d~dnorm(0, 1e-4) (for those we don’t have NRS information) or the NRS mean and variance (for those which we observe the NRS evidence). The mean can be shifted by mean.shift to reflect the potential bias from NRS or the variance can be inflated by var.infl to control the influence of NRS on the estimation from RCTs.

# jags model: code+data  
mod2 <- crosnma.model(prt.data=prt.data,  
 std.data=std.data,  
 trt=c('trt','trt'),  
 study=c('study','study'),  
 outcome=c('outcome','outcome'),  
 n='n',  
 design=c('design','design'),  
 reference='A',  
 trt.effect='random',  
 #---------- meta-regression ----------   
 covariate = list(c('age','sex'),c('age','sex')),  
 #---------- bias adjustment ----------   
 method.bias='prior',  
 run.nrs=list(n.adapt = 10,  
 n.iter=10,  
 n.burnin = 5,  
 thin=1,  
 n.chains=1))  
## The data is analyzed assuming the studies has the same design  
## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 0  
## Unobserved stochastic nodes: 258  
## Total graph size: 1078  
##   
## Initializing model

**Run JAGS**

The MCMC is run under the same set up as in the naive model.

# run jags  
jagsfit2 <- crosnma.run(model=mod2,  
 n.adapt = 20,  
 n.iter=50,  
 thin=1,  
 n.chains=3)  
## NOTE: Stopping adaptation

**Output**

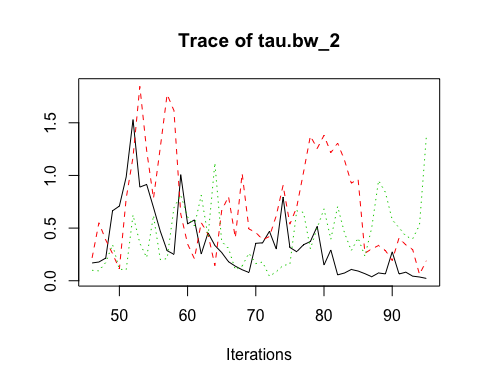
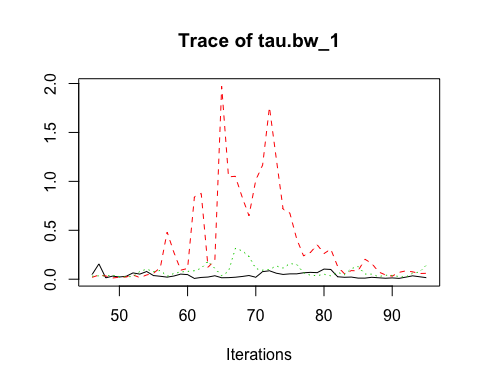
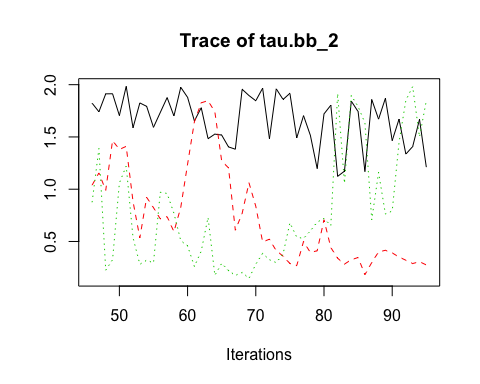
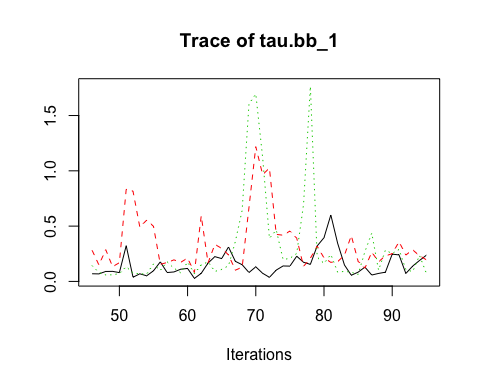
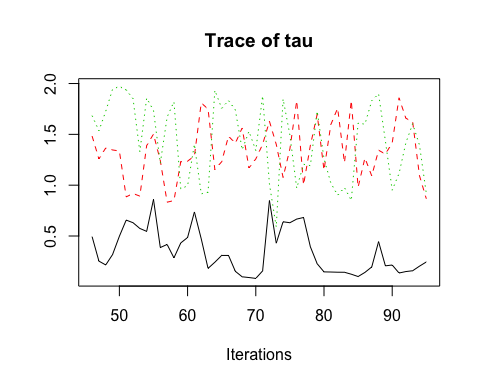
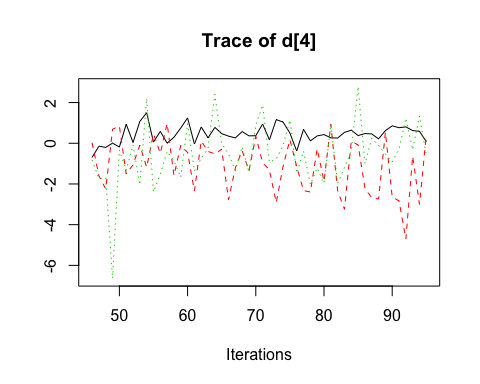
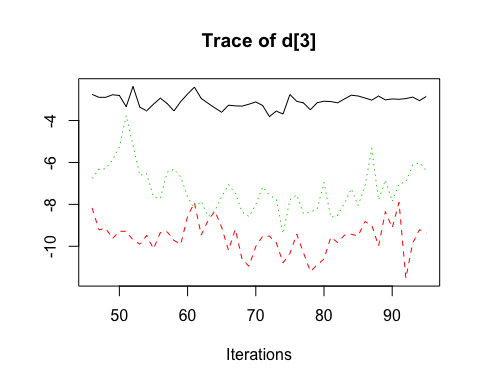
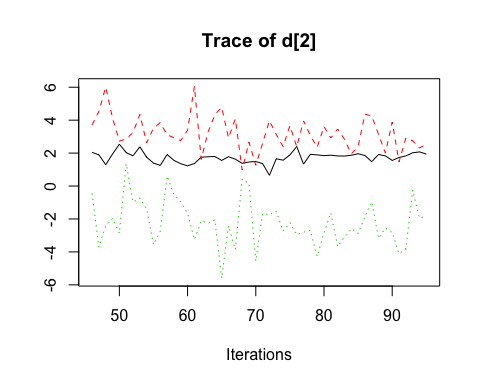
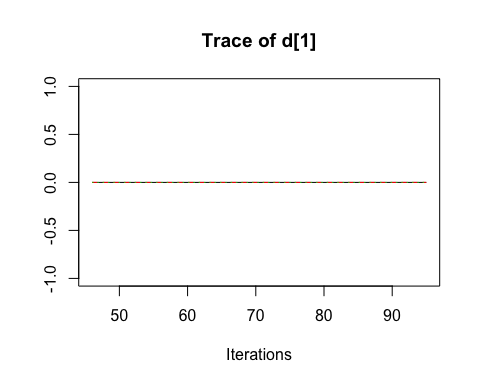
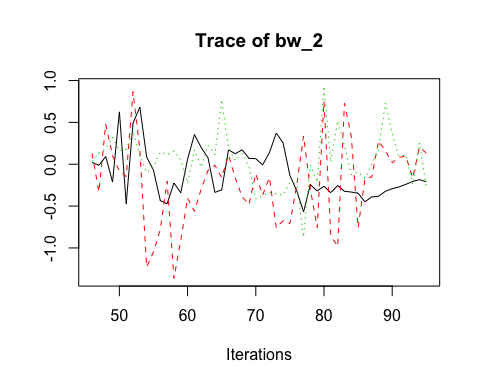
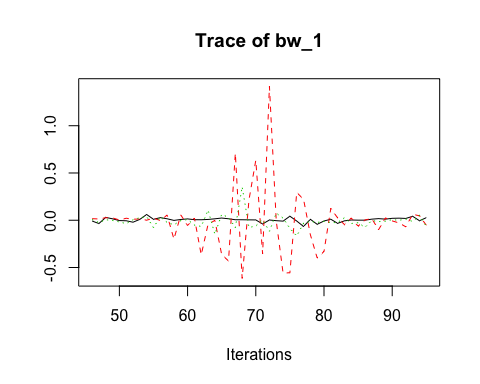
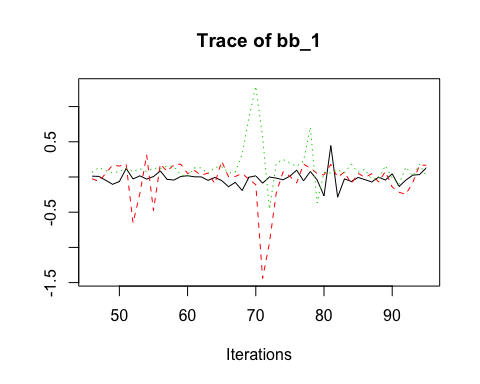
*Table of estimates*

knitr::kable(summary(jagsfit2))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Mean | SD | 2.5% | 50% | 97.5% | Rhat | n.eff |
| bb\_1 | 1.028 | 1.280 | 0.630 | 1.028 | 1.615 | 1.185 | 168 |
| bb\_2 | 0.856 | 2.847 | 0.103 | 0.839 | 6.646 | 1.544 | 57 |
| bw\_1 | 0.993 | 1.209 | 0.665 | 1.002 | 1.363 | 1.251 | 328 |
| bw\_2 | 0.913 | 1.480 | 0.396 | 0.916 | 2.088 | 1.129 | 60 |
| d.A | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | NaN | 0 |
| d.B | 2.482 | 12.238 | 0.020 | 5.753 | 82.393 | 4.182 | 126 |
| d.C | 0.001 | 16.497 | 0.000 | 0.001 | 0.064 | 6.215 | 61 |
| d.D | 0.666 | 3.809 | 0.053 | 0.882 | 4.969 | 1.341 | 150 |
| tau | 1.038 | 0.575 | 0.118 | 1.131 | 1.908 | 3.109 | 86 |
| tau.bb\_1 | 0.263 | 0.292 | 0.054 | 0.174 | 1.149 | 1.164 | 38 |
| tau.bb\_2 | 1.063 | 0.612 | 0.198 | 1.057 | 1.962 | 1.954 | 64 |
| tau.bw\_1 | 0.166 | 0.310 | 0.011 | 0.061 | 1.083 | 1.523 | 34 |
| tau.bw\_2 | 0.493 | 0.393 | 0.043 | 0.389 | 1.422 | 1.227 | 38 |

*Trace plot of estimates*

coda::traceplot(jagsfit2$samples)



## RoB-adjusted model 1

In this part, we run again the NMR model using sex and age as covariates. But now, the overall relative treatment effects are estimated from both NRS and RCT with adjustment to study-specific bias following the method introduced by [@dias\_2010], see also Section 2.3 above.

**Set up JAGS model**

We provide the name of the bias variable bias=c('bias','bias') in prt.data and std.data, respectively. The effect of bias is assumed to be additive to the treatment effect so bias.type='add'. The bias effect is set to be equal across studies bias.effect='common'. It is optional to give a list of the different priors to control the estimates of the corresponding parameters. Here, we set a uniform distribution to the common heterogeneity of the treatment effect across studies, tau.trt='dunif(0,3)'. We also set beta distribution () as priors for the probability of bias on each study. We assumed these four categories: high bias RCT pi.high.rct='dbeta(5,1)', low bias RCT pi.low.rct='dbeta(1,20)',high bias NRS pi.high.nrs='dbeta(30,1)' and low bias NRS pi.low.nrs='dbeta(1,2)'. The ratio controls the skewness of the beta distribution. When the ratio goes further beyond 1, the mean becomes located closer to 1 and the study needs ‘major’ bias adjustment and vice versa.

# jags model: code+data  
mod3 <- crosnma.model(prt.data=prt.data,  
 std.data=std.data,  
 trt=c('trt','trt'),  
 study=c('study','study'),  
 outcome=c('outcome','outcome'),  
 n='n',  
 design=c('design','design'),  
 reference='A',  
 trt.effect='random',  
 #---------- meta-regression ----------   
 covariate = list(c('age','sex'),c('age','sex')),  
 #---------- bias adjustment ----------   
 method.bias='adjust1',  
 bias=c('bias','bias'),   
 bias.type='add',  
 bias.effect='common',  
 #---------- assign a prior ----------   
 prior=list(tau.trt='dunif(0,3)',  
 pi.high.rct='dbeta(5,1)',  
 pi.low.rct='dbeta(1,20)',  
 pi.high.nrs='dbeta(30,1)',  
 pi.low.nrs='dbeta(1,2)'  
 )  
 )

**Run JAGS**

# run jags  
jagsfit3 <- crosnma.run(model=mod3,  
 n.adapt = 20,  
 n.iter=50,  
 thin=1,  
 n.chains=3)  
## NOTE: Stopping adaptation

**Output**

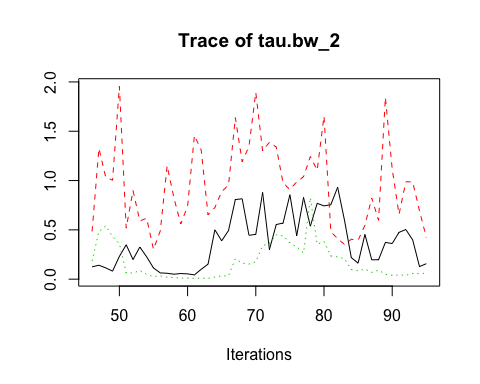
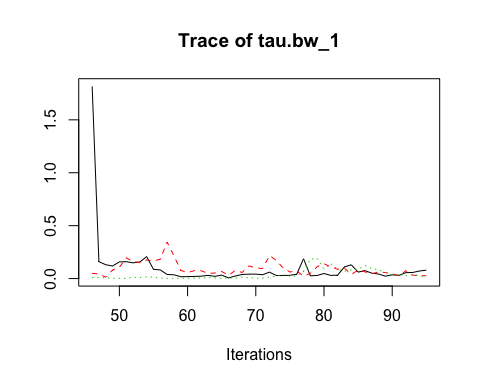
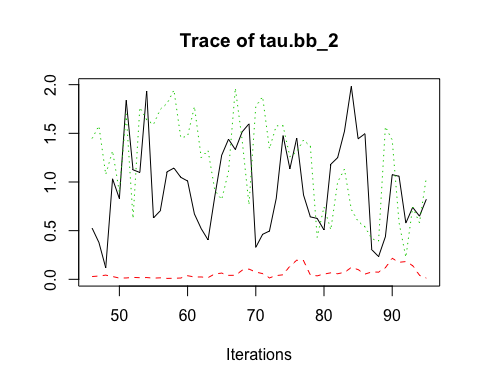
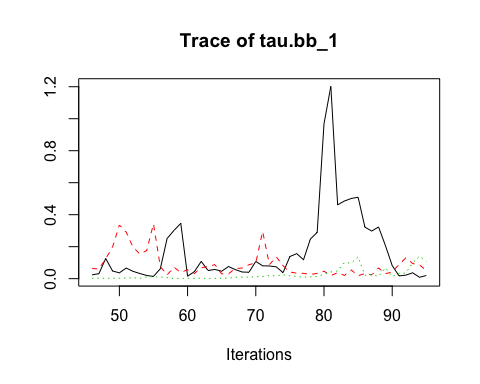
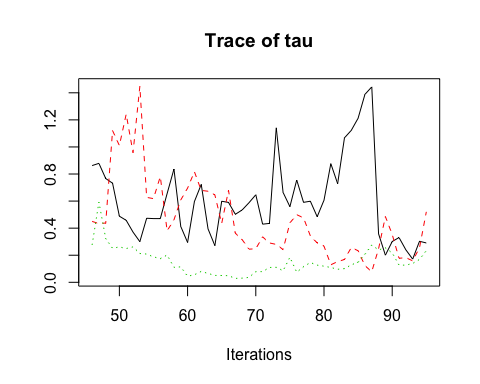
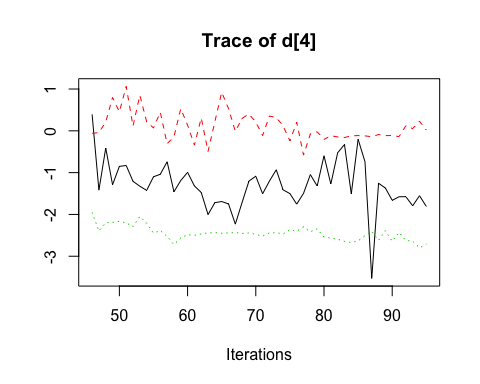
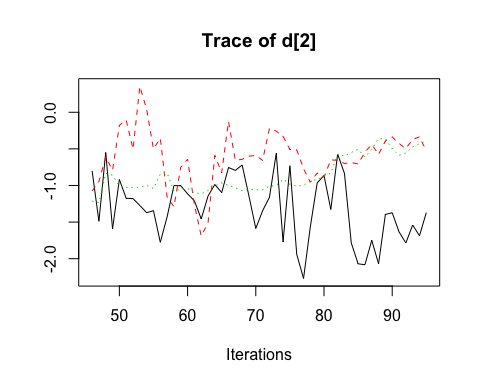
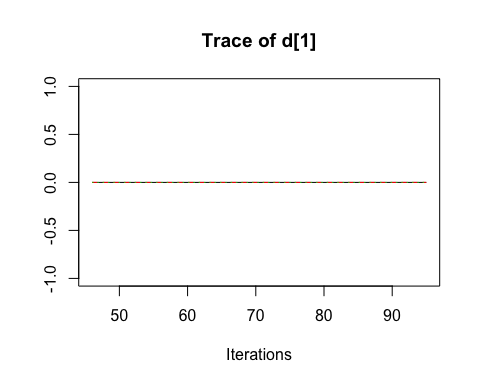
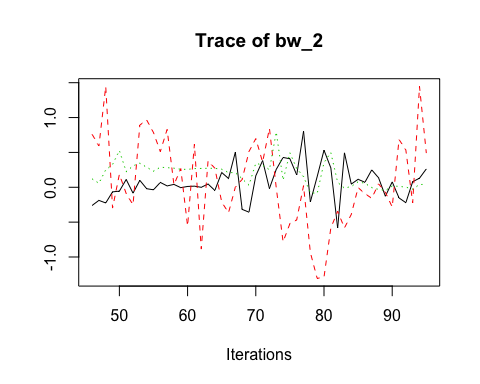
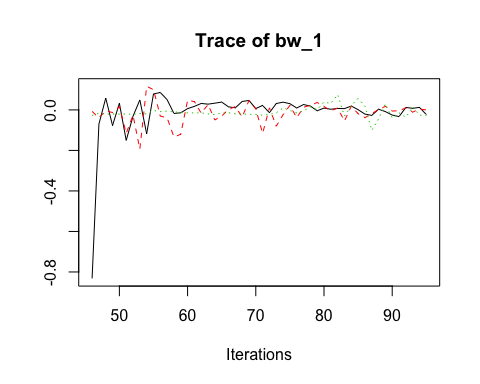
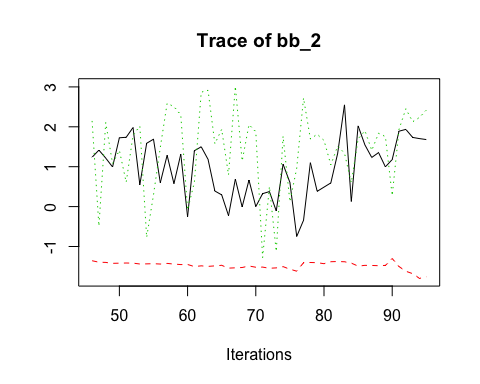
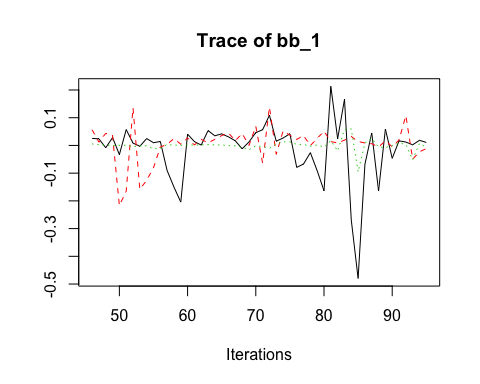
*Table of estimates*

knitr::kable(summary(jagsfit3))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Mean | SD | 2.5% | 50% | 97.5% | Rhat | n.eff |
| bb\_1 | 0.997 | 1.076 | 0.839 | 1.004 | 1.123 | 1.142 | 262 |
| bb\_2 | 1.357 | 4.348 | 0.198 | 1.794 | 13.805 | 3.296 | 68 |
| bw\_1 | 0.988 | 1.083 | 0.883 | 0.995 | 1.079 | 1.177 | 270 |
| bw\_2 | 1.118 | 1.499 | 0.444 | 1.094 | 2.360 | 1.184 | 82 |
| d.A | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | NaN | 0 |
| d.B | 0.395 | 1.579 | 0.139 | 0.394 | 0.849 | 1.727 | 42 |
| d.C | 0.269 | 2.591 | 0.057 | 0.236 | 1.076 | 3.472 | 23 |
| d.D | 0.298 | 3.068 | 0.068 | 0.266 | 1.831 | 4.777 | 68 |
| g | 1.257 | 2.132 | 0.508 | 0.886 | 4.349 | 6.022 | 13 |
| tau | 0.409 | 0.313 | 0.048 | 0.302 | 1.221 | 1.612 | 24 |
| tau.bb\_1 | 0.096 | 0.157 | 0.002 | 0.043 | 0.490 | 1.363 | 22 |
| tau.bb\_2 | 0.737 | 0.620 | 0.013 | 0.636 | 1.892 | 2.434 | 111 |
| tau.bw\_1 | 0.076 | 0.155 | 0.003 | 0.047 | 0.209 | 1.250 | 65 |
| tau.bw\_2 | 0.498 | 0.448 | 0.011 | 0.397 | 1.642 | 2.145 | 39 |

*Trace plot of estimates*

coda::traceplot(jagsfit3$samples)



## RoB-adjusted model 2

**Set up JAGS model**

The arguments for method.bias='adjust2' are similar to the ones that were used above for method.bias='adjust1'. However, there are two main differences between them. First, in 'adjust2', the relative treatment effects are modeled with a bi-modal normal distribution instead of a univariate normal distribution as in 'adjust1' method. Second, we can use a logistic model with a year as a covariate bias.covariate = c('year','year') to estimate the probability of bias in 'adjust2'. In 'adjust1', instead we assume a () to estimate the probability of bias. More details about this method can be found in [@verde\_2020] and in Section 2.4.

# jags model: code+data  
mod4 <- crosnma.model(prt.data=prt.data,  
 std.data=std.data,  
 trt=c('trt','trt'),  
 study=c('study','study'),  
 outcome=c('outcome','outcome'),  
 n='n',  
 design=c('design','design'),  
 reference='A',  
 trt.effect='random',  
 #---------- meta-regression ----------  
 covariate = list(c('age','sex'),c('age','sex')),  
 #---------- bias adjustment ----------  
 method.bias='adjust1',  
 bias=c('bias','bias'),  
 bias.type='add',  
 bias.effect='common',  
 bias.covariate = c('year','year'),#  
 #---------- assign a prior ----------  
 prior=list(tau.trt='dunif(0,3)',  
 pi.high.rct='dbeta(5,1)',  
 pi.low.rct='dbeta(1,20)',  
 pi.high.nrs='dbeta(30,1)',  
 pi.low.nrs='dbeta(1,2)'  
 )  
)

**Run JAGS**

# run jags  
jagsfit4 <- crosnma.run(model=mod4,  
 n.adapt = 20,  
 n.iter=50,  
 thin=1,  
 n.chains=3)  
## NOTE: Stopping adaptation

**Output**

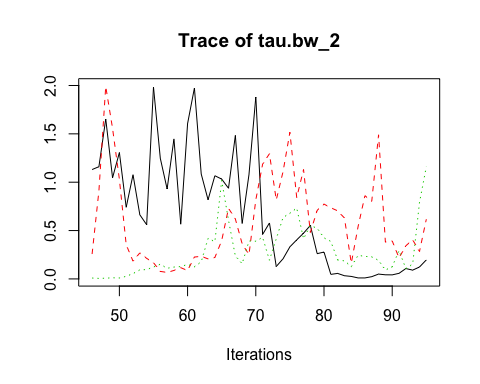
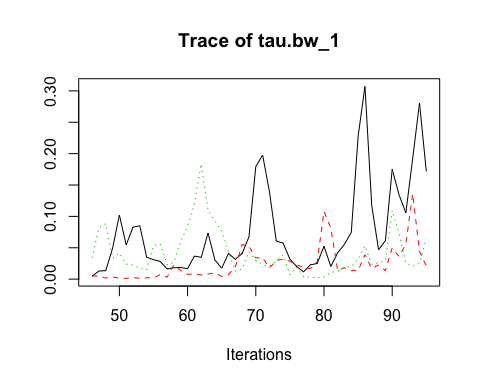
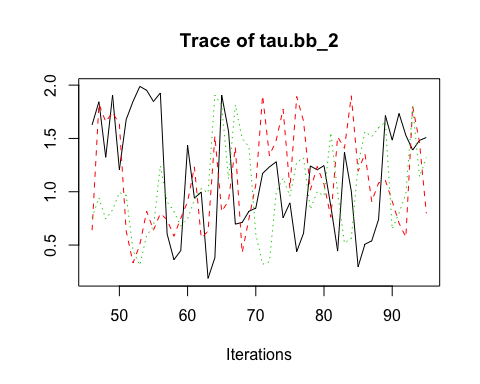
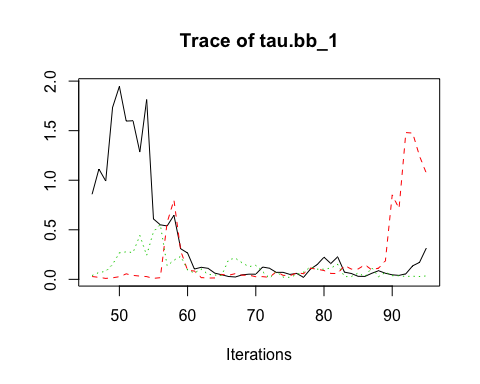
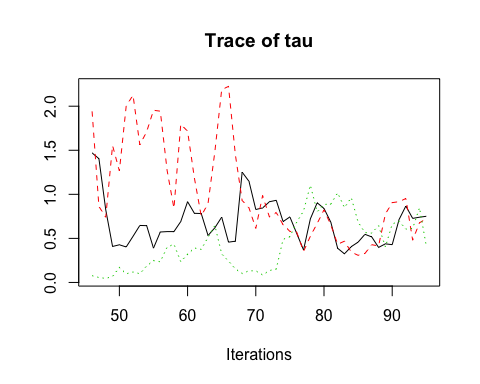
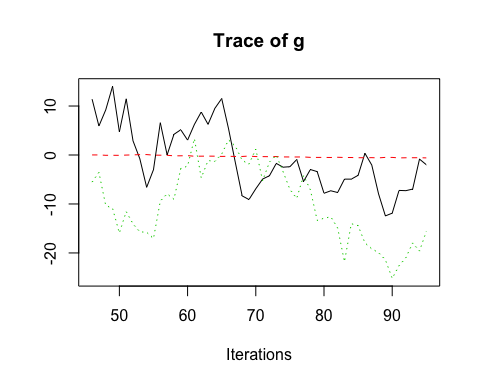
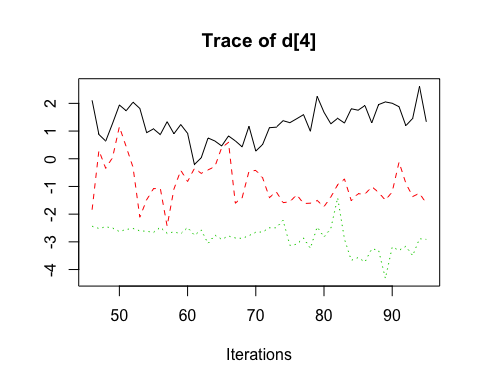
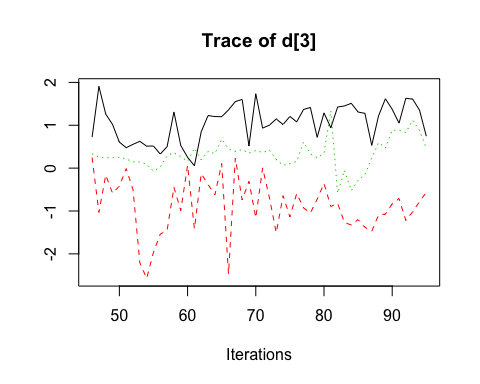
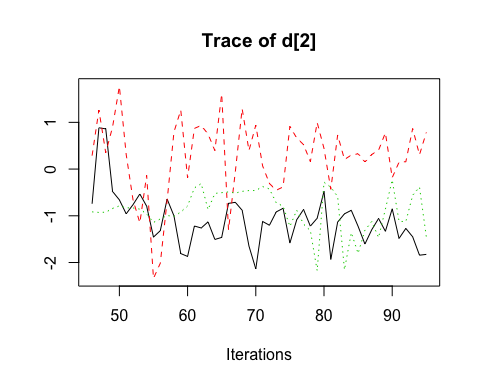
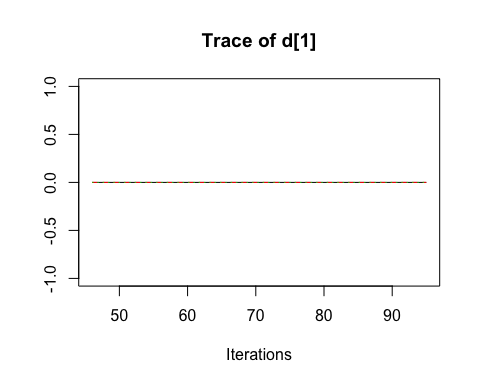
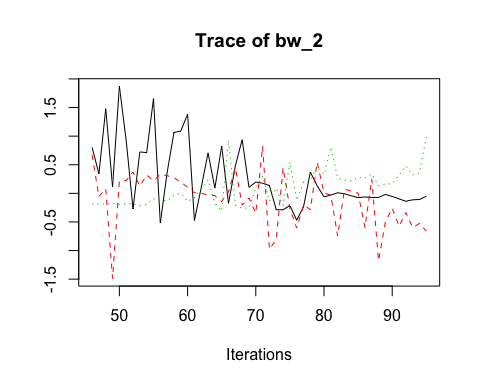
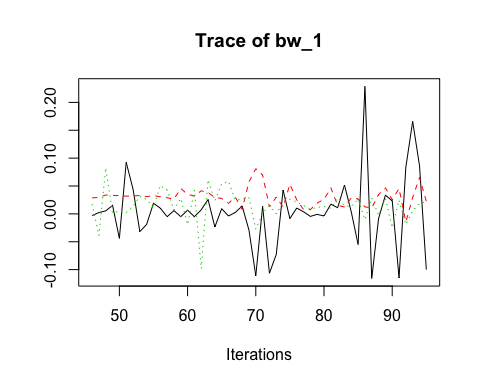
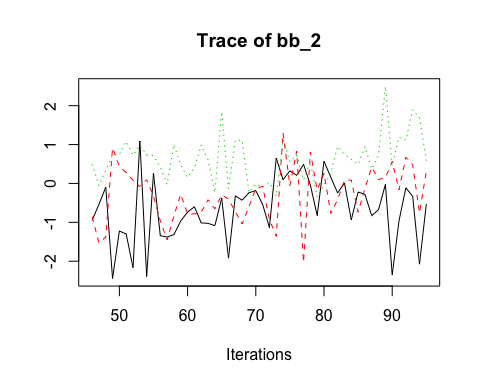
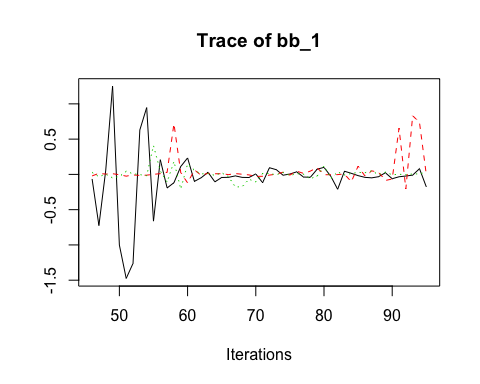
*Table of estimates*

knitr::kable(summary(jagsfit4))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Mean | SD | 2.5% | 50% | 97.5% | Rhat | n.eff |
| bb\_1 | 1.001 | 1.322 | 0.507 | 0.997 | 2.061 | 1.183 | 207 |
| bb\_2 | 0.902 | 2.418 | 0.122 | 0.933 | 4.094 | 1.615 | 117 |
| bw\_1 | 1.016 | 1.042 | 0.903 | 1.019 | 1.088 | 1.220 | 195 |
| bw\_2 | 1.085 | 1.620 | 0.466 | 1.005 | 3.222 | 1.151 | 62 |
| d.A | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | NaN | 0 |
| d.B | 0.565 | 2.369 | 0.129 | 0.461 | 3.535 | 1.965 | 64 |
| d.C | 1.181 | 2.548 | 0.191 | 1.296 | 5.009 | 3.083 | 54 |
| d.D | 0.434 | 5.969 | 0.030 | 0.319 | 7.732 | 5.090 | 39 |
| g | 0.027 | 1624.263 | 0.000 | 0.506 | 22870.622 | 1.667 | 11 |
| tau | 0.715 | 0.465 | 0.083 | 0.650 | 1.967 | 1.483 | 42 |
| tau.bb\_1 | 0.241 | 0.402 | 0.014 | 0.080 | 1.598 | 1.173 | 18 |
| tau.bb\_2 | 1.100 | 0.465 | 0.330 | 1.010 | 1.906 | 1.011 | 56 |
| tau.bw\_1 | 0.047 | 0.053 | 0.002 | 0.029 | 0.194 | 1.307 | 35 |
| tau.bw\_2 | 0.520 | 0.481 | 0.010 | 0.387 | 1.715 | 1.173 | 25 |

*Trace plot of estimates*

coda::traceplot(jagsfit4$samples)



# References