**crosnma for synthesize Cross-design evidence and Cross-format data using Network Meta-Analysis**

**Tasnim Hamza and Georgia Salanti**

**2021-06-21**

* [1 Introduction](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#introduction)
* [2 The synthesis model of individual participant data (IPD) and aggregate data (AD)](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#the-synthesis-model-of-individual-participant-data-ipd-and-aggregate-data-ad)
  + [2.1 Naive synthesis](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#naive-synthesis)
  + [2.2 Using non-randomized studies (NRS) as a prior](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#using-non-randomized-studies-nrs-as-a-prior)
  + [2.3 Bias-adjusted model 1](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#bias-adjusted-model-1)
  + [2.4 Bias-adjusted model 2](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#bias-adjusted-model-2)
  + [2.5 Assumptions about the model parameters](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#assumptions-about-the-model-parameters)
* [3 Application to the analysis of a network of studies comparing drugs for relapsing-remitting multiple sclerosis](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#application-to-the-analysis-of-a-network-of-studies-comparing-drugs-for-relapsing-remitting-multiple-sclerosis)
* [4 R implementation](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#r-implementation)
  + [4.1 Naïve synthesis](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#na%C3%AFve-synthesis)
    - [4.1.1 Naïve network meta-analysis](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#na%C3%AFve-network-meta-analysis)
    - [4.1.2 Naïve network meta-regression](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#na%C3%AFve-network-meta-regression)
  + [4.2 Using non-randomized studies (NRS) as a prior](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#using-non-randomized-studies-nrs-as-a-prior-1)
  + [4.3 Bias-adjusted model 1](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#bias-adjusted-model-1-1)
  + [4.4 Bias-adjusted model 2](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#bias-adjusted-model-2-1)
* [References](file:///Users/th19g983/Google%20Drive/GenericModelNMA/crosnma/doc/crosnma.html#references)

**library**(crosnma)

**library**(rjags)

*#> Loading required package: coda*

*#> Warning: package 'coda' was built under R version 3.6.2*

*#> Linked to JAGS 4.3.0*

*#> Loaded modules: basemod,bugs*

**load.module**('mix')

*#> module mix loaded*

**1 Introduction**

In network meta-analysis we synthesize all relevant available evidence about health outcomes from competing treatments. That evidence might come from different study designs and in different formats: from non-randomized studies (NRS) or randomized controlled trials (RCT) as individual participant data (IPD) or as aggregate data (AD). We establish the (crosnma) package to utilize all available evidence.

This document demonstrates how to use crosnma to perform a synthesize from cross-design evidence and cross-format data. The package has been developed for conducting Bayesian network meta-analysis and meta-regression. All models are implemented in JAGS.

We describe the workflow within the package using 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 using the data from 6 different studies in different formats and different designs.

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

We introduce the model to synthesis RCT and NRS where each one can be either available on individual-level (IPD) or/and on study-level (AD). We describe the four synthesis models for network meta-regression. However, we can run network meta-analysis by declining covariate terms from the models. Let’s begin by setting the notations that will be used in the description of the four models.

| **Notation** | **Description** | **Argument in crosnma.model()** |
| --- | --- | --- |
| i=1,...,npji=1,...,npj | participant id |  |
| j=1,...,nsj=1,...,ns | study id |  |
| k=1,...,Kk=1,...,K | treatment index |  |
| nsIPD,nsAD,nsRCT,nsNRSnsIPD,nsAD,nsRCT,nsNRS | the number of studies. The index refers to the design or format of the study |  |
| yijkyijk | binary outcome (0/1) | outcome |
| pijkpijk | probability of the event to occur |  |
| rjkrjk | the number of events per arm | outcome |
| njknjk | the sample size per arm | n |
| bb | the study-specific reference | \* |
| ujuj | The treatment effect of the study-specific reference |  |
| δjkδjk | log(OR) of treatment k relative to bb |  |
| xijkxijk | the covariate | covariate |
| x¯jx¯j | the mean covariate for study jj |  |
| dAkdAk | the basic parameters. Here, dAA=0dAA=0 when A set as the reference in the network | use reference to assign the reference treatment |
| zjzj | study characteristics to estimate the bias probability πjπj | bias.covariate |
| wjwj | inflation factor of variance for the NRS estimates | the element var.infl in run.nrs |
| ϑjϑj | mean shift of the NRS estimates | the element mean.shift in run.nrs |

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

**2.1 Naive synthesis**

We synthesis the evidence from RCT and NRS without acknowledging the differences between them. We combine the IPD data from RCT and NRS in one model and we do the same in another model with the AD information. Then, we combine the estimates from both parts as described in Section 2.5. See the full details in Section 2.2.1 in Hamza T (2021).

**model IPD only**

yijk∼Bernoulli(pijk)yijk∼Bernoulli(pijk)

logit(pijk)={uj+β0,jxijkuj+δjk+(β0,j+βw1,jk)xijk+(βB1,jk−βw1,jk)x¯.jif k=bif k≠blogit(pijk)={uj+β0,jxijkif k=buj+δjk+(β0,j+β1,jkw)xijk+(β1,jkB−β1,jkw)x¯.jif k≠b

**model AD only**

rjk∼Binomial(pjk,njk)rjk∼Binomial(pjk,njk)

logit(pjk)={ujuj+δjk+βB1,jkx¯jif k=bif k≠blogit(pjk)={ujif k=buj+δjk+β1,jkBx¯jif k≠b

**2.2 Using non-randomized studies (NRS) as a prior**

First we estimate the relative treatment effects using only the NRS (use run.nrs in crosnma.model() to control this process). Second, we use the NRS estimates (d̂ NRSAk,V̂ NRSAkd^AkNRS,V^AkNRS) as a prior information for the basic parameters of RCT data, dAk∼N(d̂ NRSAk,V̂ NRSAk)dAk∼N(d^AkNRS,V^AkNRS). To control the NRS influence in the RCT estimates, we can either inflate the NRS variance by dividing by wjwj (the inflated variance is V̂ NRSAk/wjV^AkNRS/wj) or shift the NRS means by ϑϑ.

**2.3 Bias-adjusted model 1**

We incorporate the risk of bias (RoB) into the IPD and the AD models by adding the bias term γjRjγjRj to both models as follows:

**model IPD only**

yijk∼Bernoulli(pijk)yijk∼Bernoulli(pijk)

logit(pijk)={uj+β0,jxijkuj+δjk+(β0,j+βw1,jk)xijk+(βB1,jk−βw1,jk)x¯.j+γjRjif k=bif k≠blogit(pijk)={uj+β0,jxijkif k=buj+δjk+(β0,j+β1,jkw)xijk+(β1,jkB−β1,jkw)x¯.j+γjRjif k≠b

**model AD only**

rjk∼Binomial(pjk,njk)rjk∼Binomial(pjk,njk)

logit(pjk)={ujuj+δjk+βB1,jkx¯j+γjRjif k=bif k≠blogit(pjk)={ujif k=buj+δjk+β1,jkBx¯j+γjRjif k≠b

The bias indicator

Rj∼Bernoulli(πj)Rj∼Bernoulli(πj)

**2.4 Bias-adjusted model 2**

Another way to incorporate the RoB of the study is by replacing \_{jk} by the bias-adjusted relative treatment effect θjkθjk. Then θjkθjk is modeled by a bimodal normal distribution as described in Table 2. We can estimate the bias probability πjπj by either assigning a beta distribution or by using the study characteristics zjzj through a logistic transformation, see Table 2.

**model IPD only**

yijk∼Bernoulli(pijk)yijk∼Bernoulli(pijk)

logit(pijk)={uj+β0,jxijkuj+θjk+(β0,j+βw1,jk)xijk+(βB1,jk−βw1,jk)x¯jif k=bif k≠blogit(pijk)={uj+β0,jxijkif k=buj+θjk+(β0,j+β1,jkw)xijk+(β1,jkB−β1,jkw)x¯jif k≠b

**model AD only**

rjk∼Binomial(pjk,njk)rjk∼Binomial(pjk,njk)

logit(pjk)={ujuj+θjk+βB1,jkx¯jif k=bif k≠blogit(pjk)={ujif k=buj+θjk+β1,jkBx¯jif k≠b

**2.5 Assumptions about the model parameters**

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

| **Parameter** | **Assumptions** | **Argument in crosnma.model()** |
| --- | --- | --- |
| relative treatment effect (δjkδjk) | Random-effects: δjk∼N(dAk−dAb,τ2δ)δjk∼N(dAk−dAb,τδ2) | trt.effect='random' |
|  | Common-effect: δjk=dAk−dAbδjk=dAk−dAb | trt.effect='common' |
| Covariate effect β0,jβ0,j | Independent effects: β0,j∼N(0,102)β0,j∼N(0,102) | reg0.effect='independent' |
|  | Random-effects: β0,j∼N(B0,τβ0)β0,j∼N(B0,τβ0) | reg0.effect='random' |
| Within-study covariate-treatment interaction (βW1,jkβ1,jkW) | Random-effects: βW1,jk∼N(BW1,Ak−BW1,Ab,τβW1)β1,jkW∼N(B1,AkW−B1,AbW,τβ1W) | regw.effect='random' |
|  | Common-effect: βW1,jk=BW1,Ak−BW1,Abβ1,jkW=B1,AkW−B1,AbW | regw.effect='common' |
| Between-study covariate-treatment interaction (βB1,jkβ1,jkB) | Random-effects: βB1,jk∼N(BB1,Ak−BB1,Ab,τβB1)β1,jkB∼N(B1,AkB−B1,AbB,τβ1B) | regb.effect='random' |
|  | Common-effect: βB1,jk=BB1,Ak−BB1,Abβ1,jkB=B1,AkB−B1,AbB | regb.effect='common' |
| bias-adjusted relative treatment effect (θjkθjk) | Random-effects: θjk∼(1−πj)N(dAk−dAb,τ2δ)+πjN(dAk−dAb+γj,τ2δ+τ2γ)θjk∼(1−πj)N(dAk−dAb,τδ2)+πjN(dAk−dAb+γj,τδ2+τγ2) | trt.effect='random' |
| Bias effect (γjγj) | Random-effects: γj∼N(Γ,τγ)γj∼N(Γ,τγ) | bias.effect='random' |
|  | Common-effect: γj=Γγj=Γ | bias.effect='common' |
| Bias probability (πjπj) | πj∼Beta(a,b)πj∼Beta(a,b) |  |
|  | πj=a+bzjπj=a+bzj |  |

**3 Application to the analysis of a network of studies comparing drugs for relapsing-remitting multiple sclerosis**

Both datasets have 6 studies. These studies are either available as study-level data std.data (2 studies) or as individual participant data prt.data (four studies). The prt.data have been manipulated by re-sampling from the original data set. Three of the prt.data are RCTs and one comes from an observational study (NRS). The two std.data are RCTs. Both datasets consist of four drugs which are anonymized.

The prt.data contains 2950 rows, each row 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 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 (can be set as low, high or unclear) bias and the year of publication year.

We display the full study-level data:

**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*

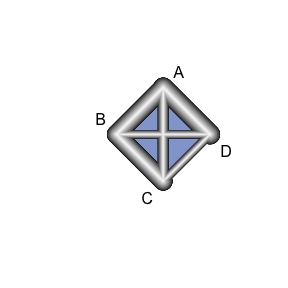
For all studies, we provide the following information on each arm level: the total number of participants n and the number of those who relapsed outcome, the treatment label trt. At study-level, we have the study id study, the design of the study design, the mean age age, the percentage of males sex,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

**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 |

**4 R implementation**

There are two steps to run the NMA/NMR model. The first step is to create JAGS model using crosnma.model() which creates the JAGS code and the data. In the second step, the output of that function will be used in crosnma.run() to run the analysis through JAGS (Plummer 2003).

**4.1 Naïve synthesis**

**4.1.1 Naïve network meta-analysis**

**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 needs to be given respectively in prt.data and std.data. Next, the reference treatment needs to be assigned (we set it to drug A). By choosing trt.effect=random, we are assigning a normal distribution to each relative treatment effect to allow the synthesis across studies, see the table in Section 2.1. Finally, the different designs; RCT and NRS are combined with the information taken at face-value; 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 us to set the number of adaptations, iterations, thinning and chains.

*# run jags*

jagsfit1 <- **crosnma.run**(model=mod1,

n.adapt = 20,

n.iter=100,

n.burnin = 40,

thin=1,

n.chains=2)

*#>* **NOTE***: Stopping adaptation*

**Output**

*Table of estimates*

We summarize the estimated parameters in the following table.

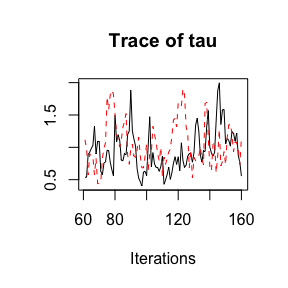
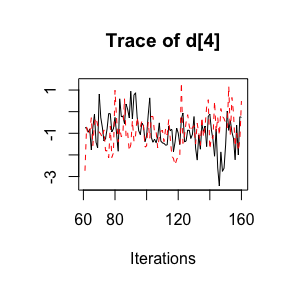
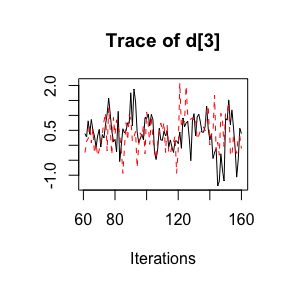
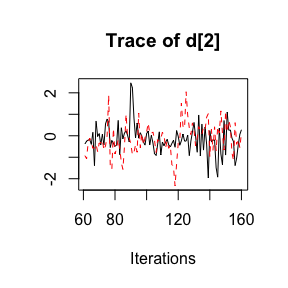
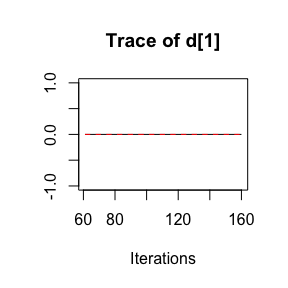
knitr::**kable**(**summary**(jagsfit1,expo=T))

|  | **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 | 0.858 | 2.026 | 0.225 | 0.814 | 3.217 | 0.999 | 108 |
| d.C | 1.472 | 1.803 | 0.393 | 1.480 | 4.816 | 0.996 | 100 |
| d.D | 0.404 | 2.198 | 0.085 | 0.399 | 2.239 | 0.995 | 124 |
| tau | 0.995 | 0.343 | 0.468 | 0.924 | 1.824 | 1.037 | 40 |

The estimated parameters are: exp(d.B) which is the estimated OR of B vs A, exp(d.C) and exp(d.D) are the OR of C and D relative to A, respectively. tau refers to the estimates of the heterogeneity standard deviation in the relative treatment effect across studies.

*Trace plot of estimates* We need also to check the convergence of the MCMC chains, for example, by visually inspecting the trace plot.

coda::**traceplot**(jagsfit1$samples)



**4.1.2 Naïve network meta-regression**

In this part, we run NMR model by adding age as a covariate from both datasets. We set a list of 2 covariates in prt.data and std.data, respectively, covariate=list(c('age'),c('age')). The first element in the list indicates the names of the covariate column in prt.data and the second element refers to the corresponding names in std.data.

**Set up JAGS model**

*# 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'),**c**('age')),

split.regcoef = F,

*#---------- bias adjustment ----------*

method.bias='naive'

)

**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=100,

n.burnin = 40,

thin=1,

n.chains=2)

*#>* **NOTE***: Stopping adaptation*

**Output**

*Table of estimates*

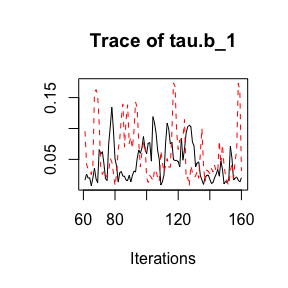
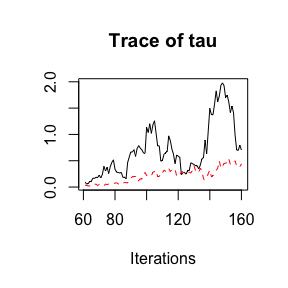
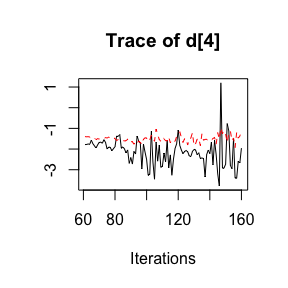
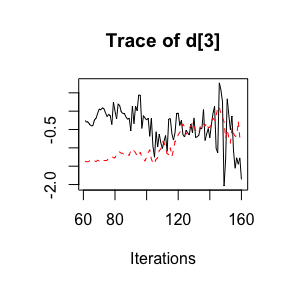
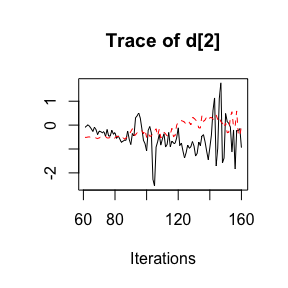
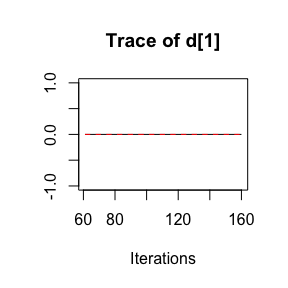
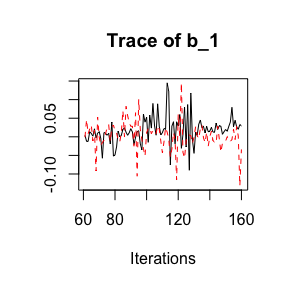
knitr::**kable**(**summary**(jagsfit2))

|  | **Mean** | **SD** | **2.5%** | **50%** | **97.5%** | **Rhat** | **n.eff** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| b\_1 | 1.010 | 1.039 | 0.927 | 1.011 | 1.094 | 1.066 | 200 |
| d.A | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | NaN | 0 |
| d.B | 0.700 | 1.692 | 0.235 | 0.675 | 1.751 | 1.283 | 41 |
| d.C | 0.524 | 1.691 | 0.241 | 0.551 | 1.301 | 1.572 | 17 |
| d.D | 0.162 | 1.810 | 0.038 | 0.194 | 0.336 | 1.891 | 135 |
| tau | 0.465 | 0.437 | 0.032 | 0.314 | 1.728 | 1.889 | 8 |
| tau.b\_1 | 0.052 | 0.038 | 0.010 | 0.039 | 0.158 | 1.089 | 52 |

Here, we additionally get the estimate of b\_1 which indicates the mean effect of age and tau.b\_1 which refers to the heterogeneity standard deviation in the effect of age across studies.

*Trace plot of estimates*

coda::**traceplot**(jagsfit2$samples)



**4.2 Using non-randomized studies (NRS) as a prior**

To combine both designs, we use NRS evidence to construct a prior for the relative treatment effects, and the RCT to form the likelihood, see Section 2.2.2 in (Hamza T 2021).

**Set up JAGS model**

In this case, two additional arguments are needed to control combining RCT and NRS.

1. We indicate using NRS as prior by setting method.bias='prior'.
2. That means we need to run NMA with only NRS data. This requires the following MCMC settings: the number of adaptations, iterations, burn-ins, thinning and chains, assigned to run.nrs.

In this method, the prior for the basic parameters is set to a normal distribution that is either a minimally informative prior d~dnorm(0, 1e-4) (for those comparisons for which we don’t have NRS ) or the NRS mean and variance of the effects estimated in the NRS. The mean can be shifted by mean.shift to reflect the potential bias in NRS and/or the variance can be inflated by var.infl to control the influence of NRS on the final estimation. Both should be provided in run.nrs.

*# 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',

*#---------- bias adjustment ----------*

method.bias='prior',

run.nrs=**list**(n.adapt = 10,

n.iter=5000,

n.burnin = 400,

thin=1,

n.chains=2))

*#> 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**

*# run jags*

jagsfit3 <- **crosnma.run**(model=mod3,

n.adapt = 20,

n.iter=100,

n.burnin = 40,

thin=1,

n.chains=2)

*#>* **NOTE***: Stopping adaptation*

**Output**

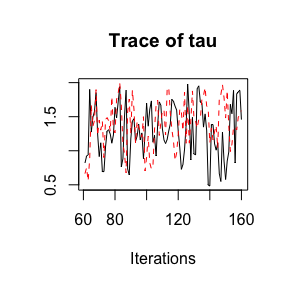
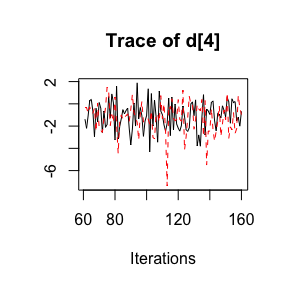
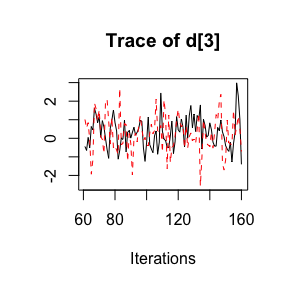
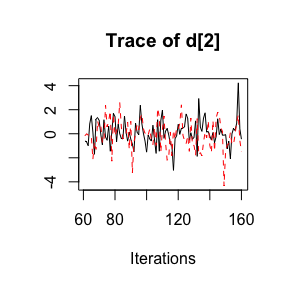
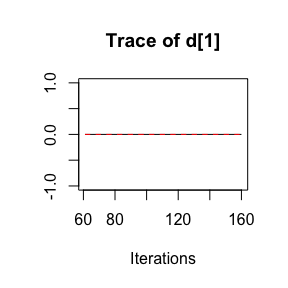
*Table of estimates*

knitr::**kable**(**summary**(jagsfit3))

|  | **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 | 1.001 | 3.218 | 0.120 | 0.977 | 10.573 | 1.017 | 200 |
| d.C | 1.308 | 2.519 | 0.244 | 1.345 | 8.240 | 1.016 | 152 |
| d.D | 0.305 | 3.929 | 0.022 | 0.316 | 3.135 | 1.005 | 357 |
| tau | 1.335 | 0.366 | 0.645 | 1.344 | 1.933 | 1.039 | 96 |

*Trace plot of estimates*

coda::**traceplot**(jagsfit3$samples)



**4.3 Bias-adjusted model 1**

In this part, the overall relative treatment effects are estimated from both NRS and RCT with adjustment to study-specific bias following the method introduced by (Dias et al. 2010) and extended here; 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 first element of the vector refers to the column name of bias in prt.data and the second in std.data. By default, the effect of bias is assumed to be additive bias.type='add' and equal across studies bias.effect='common'.

Optionally, we can give a list of the different priors to control the estimates of the various model 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 a beta distribution (Beta(a,b)Beta(a,b)) as priors for the probability of bias on each study. The default priors are as follows: 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 a/ba/b controls the skewness of the beta distribution. The closer to 1 the ratio a/b, the more the mean of probability of bias gets closer to 1 and the study acquires ‘major’ bias adjustment.

*# 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',

*#---------- 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*

jagsfit4 <- **crosnma.run**(model=mod4,

n.adapt = 20,

n.iter=100,

n.burnin = 40,

thin=1,

n.chains=2)

*#>* **NOTE***: Stopping adaptation*

**Output**

*Table of estimates*

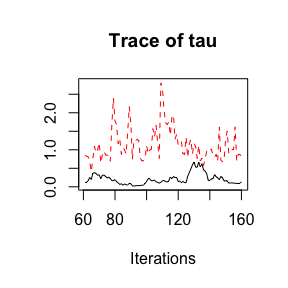
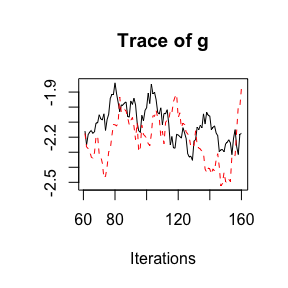
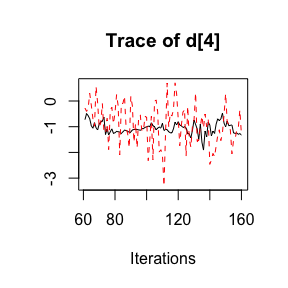
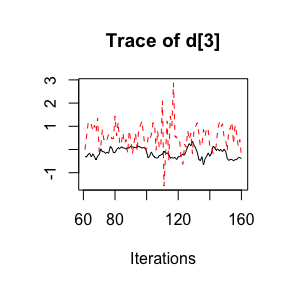
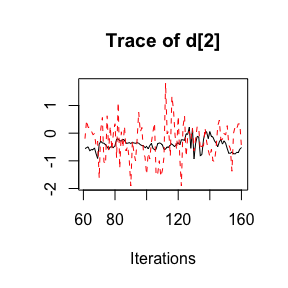
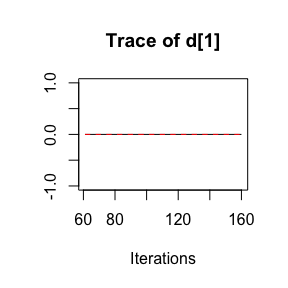
knitr::**kable**(**summary**(jagsfit4))

|  | **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 | 0.702 | 1.669 | 0.225 | 0.686 | 1.891 | 1.234 | 90 |
| d.C | 1.220 | 1.753 | 0.620 | 1.078 | 3.442 | 2.168 | 110 |
| d.D | 0.372 | 1.795 | 0.114 | 0.356 | 1.332 | 1.245 | 77 |
| g | 0.115 | 1.159 | 0.084 | 0.116 | 0.148 | 1.284 | 14 |
| tau | 0.671 | 0.572 | 0.036 | 0.635 | 1.944 | 3.531 | 33 |

The parameter g refers to the mean bias effect on each study.

*Trace plot of estimates*

coda::**traceplot**(jagsfit4$samples)



**4.4 Bias-adjusted model 2**

**Set up JAGS model**

The arguments for method.bias='adjust2' are similar to the ones used before in method.bias='adjust1'. The only additional argument is bias.covariate = c('year','year') which indicates using the year of study publication to estimate the study-probability of bias. The bias probabilities will be linked to year through logistic model ( internally coded). For more details see (Verde 2020) and our extension in Section 2.4.

*# jags model: code+data*

mod5 <- **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',

*#---------- bias adjustment ----------*

method.bias='adjust2',

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*

jagsfit5 <- **crosnma.run**(model=mod5,

n.adapt = 20,

n.iter=100,

n.burnin = 40,

thin=1,

n.chains=2)

*#>* **NOTE***: Stopping adaptation*

**Output**

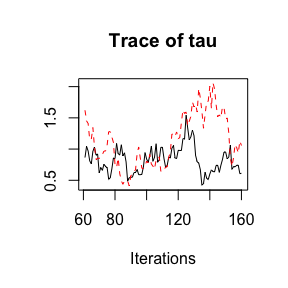
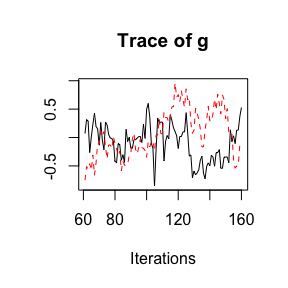
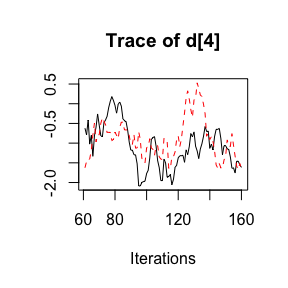
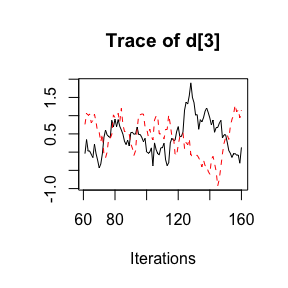
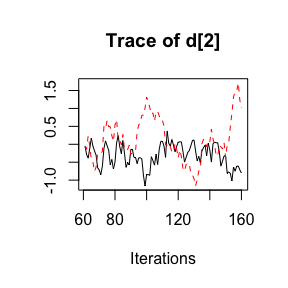
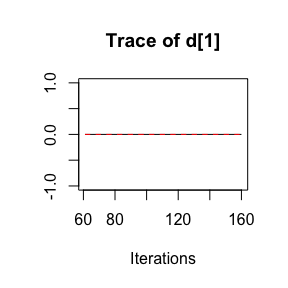
*Table of estimates*

knitr::**kable**(**summary**(jagsfit5))

|  | **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 | 0.912 | 1.703 | 0.418 | 0.867 | 3.432 | 1.470 | 27 |
| d.C | 1.525 | 1.633 | 0.648 | 1.538 | 3.739 | 0.998 | 13 |
| d.D | 0.374 | 1.733 | 0.141 | 0.360 | 1.195 | 1.043 | 11 |
| g | 0.986 | 1.456 | 0.518 | 0.971 | 2.118 | 1.096 | 26 |
| tau | 0.973 | 0.365 | 0.492 | 0.868 | 1.771 | 1.478 | 17 |

*Trace plot of estimates*

coda::**traceplot**(jagsfit5$samples)



**References**

Dias, Sofia, N. J. Welton, V. C. C. Marinho, G. Salanti, J.P.T Higgins, and A. E. Ades. 2010. “Estimation and Adjustment of Bias in Randomized Evidence by Using Mixed Treatment Comparison Meta-Analysis.” *Journal of the Royal Statistical Society* 173: 613–29.

Hamza T, Kuhle J, Pellegrini F. 2021. “Flexible Generic Framework for Evidence Synthesis in Health Technology Assessment.” *Unpublished*.

Plummer, Martyn. 2003. “JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling.”

Verde, Pablo Emilio. 2020. “A Bias-Corrected Meta-Analysis Model for Combining, Studies of Different Types and Quality.” *Biometrical Journal. Biometrische Zeitschrift*, September. <https://doi.org/10.1002/bimj.201900376>.