crossnma to synthesize cross-design evidence and cross-format data using network meta-analysis (dichotomous outcome)

Tasnim Hamza, Guido Schwarzer and Georgia Salanti

2022-03-30

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 set up the package `crossnma` to synthesize all available evidence.

This document demonstrates how to use `crossnma` to synthesize cross-design evidence and cross-format data via Bayesian network meta-analysis and meta-regression (NMA and NMR). All models are implemented in JAGS (Plummer 2003).

We describe the workflow within the package using a worked example from a network meta-analysis of studies for treatments in relapsing remitting multiple sclerosis (RRMS). The primary outcome is the occurrence of relapses in two years (binary outcome). In the analysis, the relative effect will be the 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.

The synthesis models

We first introduce the model that synthesizes studies with individual-level (IPD) or/and aggregate data (AD) ignoring their design (unadjusted synthesis). Then, we present three possible models that account for the different study designs. In the table below we set the notation that will be used in the description of the four synthesis models.

Notation	Description	Argument in `crossnma.model()`
$i = 1, ..., np_j$	participant id	
$j = 1, ..., ns$	study id	`study`
$k = 1, ..., K$	treatment index	`trt`
$ns_{IPD}, ns_{AD}, ns_{RCT}, ns_{NRS}$	the number of studies. The index refers to the design or format of the study	
y_{ijk}	binary outcome (0/1)	`outcome`
p_{ijk}	probability of the event to occur	
r_{jk}	the number of events per arm	`outcome`
n_{jk}	the sample size per arm	`n`
b	the study-specific reference	*

Notation	Description	Argument in `crossnma.model()`
u_{jb}	The treatment effect of the study-specific reference b when $x_{ijk} = \bar{x}_j = 0$	
δ_{jbk}	log(OR) of treatment k relative to b	
x_{ijk}	the covariate	`cov1`, `cov2`, `cov3`
$\bar{x}_j$	the mean covariate for study j	
d_{Ak}	the basic parameters. Here, $d_{AA} = 0$ when A is set as the reference in the network	use `reference` to assign the reference treatment
z_j	study characteristics to estimate the bias probability π_j	`bias.covariate`
w	common inflation factor of variance for the NRS estimates	the element `var.infl` in `run.nrs`
ζ	common mean shift of the NRS estimates	the element `mean.shift` in `run.nrs`

*The study-specific reference b is assigned automatically to be the network reference for studies that have the network reference treatment. If not, it is assigned to the first alphabetically ordered treatment on the study.

Unadjusted network meta-regression (NMR)

We synthesize 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.

NMR model for IPD studies

$$y_{ijk} \sim Bernoulli(p_{ijk})$$

$$logit(p_{ijk}) = \begin{cases} u_{jb} + \beta_{0j}x_{ijk} & \text{if } k = b \\ u_{jb} + \delta_{jbk} + \beta_{0j}x_{ijk} + \beta^{w}_{1,jbk}x_{ijk} + (\beta^{B}_{1,jbk} - \beta^{w}_{1,jbk})\bar{x}_j & \text{if } k \neq b \end{cases} \tag{1}$$

NMR model for AD studies

$$r_{jk} \sim Binomial(p_{.jk}, n_{jk})$$

$$logit(p_{.jk}) = \begin{cases} u_{jb} & \text{if } k = b \\ u_{jb} + \delta_{jbk} + \beta^{B}_{1,jbk}\bar{x}_j & \text{if } k \neq b \end{cases} \tag{2}$$

Using non-randomized studies (NRS) to construct priors for the treatment effects

First, the (network) meta-regression with only NRS data estimates the relative treatment effects with posterior distribution of mean $\tilde{d}^{NRS}_{Ak}$ and variance V^{NRS}_{Ak} (use `run.nrs` in `crossnma.model()` to control this process). The posteriors of NRS results are then used as priors for the corresponding basic parameters in the RCT model, $d_{Ak} \sim N(\tilde{d}^{NRS}_{Ak}, V^{NRS}_{Ak})$. We can adjust for potential biases associated with NRS by either shifting the mean of the prior distribution with a bias term ζ or by dividing the prior variance with a common inflation factor $w, 0 < w < 1$ controls NRS contribution. The assigned priors become $d_{Ak} \sim N(\tilde{d}_{Ak}^{NRS} + \zeta, V_{Ak}^{NRS}/w)$.

Bias-adjusted model 1

We incorporate judgments about study risk of bias (RoB) in bias-adjusted model 1 and model 2. Each judgment about the risk of bias in a study is summarized by the index R_j which takes binary values 0 (no bias) or 1 (bias). In bias-adjusted model 1, we extend the method introduced by Dias et al. (2010) by adding a treatment-specific bias term $\gamma_{2,jbk}R_j$ to the relative treatment effect on both the AD and IPD parts of the model. A multiplicative model can also be employed, where treatment effects are multiplied by $\gamma_{1,jbk}^{R_j}$. We can add either multiplicative bias effects, additive bias effects, or both (in this case, δ_{jbk} should be dropped from the additive part). The models in previous section are extended to adjust for bias as follows.

NMR model for IPD studies

$$logit(p_{ijk}) = \begin{cases} u_{jb} + \beta_{0j}x_{ijk} & \text{if } k = b \\ u_{jb} + \overbrace{\delta_{jbk}\gamma_{1,jbk}^{R_j}}^{\text{multiplicative}} + \overbrace{\delta_{jbk} + \gamma_{2,jbk}R_j}^{\text{additive}} + \beta_{0j}x_{ijk} + \beta_{1,jbk}^{w}x_{ijk} + (\beta_{1,jbk}^{B} - \beta_{1,jbk}^{w})\bar{x}_j & \text{if } k \neq b \end{cases} \tag{3}$$

NMR model for AD studies

$$logit(p_{.jk}) = \begin{cases} u_{jb} & \text{if } k = b \\ u_{jb} + \overbrace{\delta_{jbk}\gamma_{1,jbk}^{R_j}}^{\text{multiplicative}} + \overbrace{\delta_{jbk} + \gamma_{2,jbk}R_j}^{\text{additive}} + \beta_{1,jbk}^{B}\bar{x}_j & \text{if } k \neq b \end{cases} \tag{4}$$

The bias indicator R_j follows the following distribution

$$R_j \sim Bernoulli(\pi_j)$$

The bias probabilities π_j are study-specific and can be estimated in two different ways. They are either given informative beta priors ($Beta(a_1, a_2)$) that are set according to the risk of bias for each study.

$$\pi_j \sim Beta(a_1, a_2)$$

The hyperparameters a_1 and a_2 should be chosen in a way that reflects the risk of bias for each study. The degree of skewness in beta distribution can be controlled by the ratio a_1/a_2 . When a_1/a_2 equals 1 (or $a_1 = a_2$), there is no skewness in the beta distribution (the distribution is reduced to a uniform distribution), which is appropriate for studies with unclear risk of bias. When the ratio a_1/a_2 is closer to 1, the more the mean of probability of bias (expected value of $\pi_j = a_1/(a_1 + a_2)$) gets closer to 1 and the study acquires 'major' bias adjustment. The default beta priors are as follows: high bias RCT `pi.high.rct='dbeta(10,1)'`, low bias RCT `pi.low.rct='dbeta(1,10)'`, high bias NRS `pi.high.nrs='dbeta(30,1)'` and low bias NRS `pi.low.nrs='dbeta(1,30)'`. Alternatively, we can use the study characteristics z_j to estimate π_j through a logistic transformation (internally coded).

We combine the multiplicative and the additive treatment-specific bias effects across studies by assuming they are exchangeable $\gamma_{1,jbk} \sim \mathrm{N}(g_{1,bk}, \tau_{1,\gamma}^2)$,$\gamma_{2,jbk} \sim \mathrm{N}(g_{2,bk}, \tau_{2,\gamma}^2)$) or common $\gamma_{1,jbk} = g_{1,bk}$ and $\gamma_{2,jbk} = g_{2,bk}$. Dias et al. (2010) proposed to model the mean bias effect $(g_{1,bk}, g_{2,bk})$ based on the treatments being compared.

$$g_{m,bk} = \begin{cases} g_m & \text{if } b \text{ is inactive treatment} \\ 0 \text{ or } (-1)^{dir_{bk}} g_m^{act} & \text{if } b \text{ and } k \text{ are active treatments} \end{cases} \tag{5}$$

where $m = 1, 2$. This approach assumes a common mean bias for studies that compare active treatments with an inactive treatment (placebo, standard or no treatment). For active vs active comparisons, we could assume either a zero mean bias effect or a common bias effect g_m^{act}. The direction of bias dir_{bk} in studies that compare active treatments with each other should be defined in the data. That is set to be either 0, meaning that bias favors b over k, or 1 , meaning that k is favored to b. In `crossnma.model()`, the bias direction is specified by providing the unfavoured treatment for each study, `unfav`. To select which mean bias effect should be applied, the user can provide the `bias.group` column as data. Its values can be 0 (no bias adjustment), 1 (to assign for the comparison mean bias effect g_m) or 2 (to set bias g_m^{act}).

Another parameterisation of the logistic model with additive bias effect is

NMR model for IPD studies

$$logit(p_{ijk}) = \begin{cases} u_{jb} + \beta_{0j} x_{ijk} & \text{if } k = b \\ u_{jb} + (1 - R_j)\delta_{jbk} + \delta_{jbk}^{bias} R_j + \beta_{0j} x_{ijk} + \beta_{1,jbk}^{w} x_{ijk} + (\beta_{1,jbk}^{B} - \beta_{1,jbk}^{w})\bar{x}_j & \text{if } k \neq b \end{cases} \tag{6}$$

NMR model for AD studies

$$logit(p_{.jk}) = \begin{cases} u_{jb} & \text{if } k = b \\ u_{jb} + (1 - R_j)\delta_{jbk} + \delta_{jbk}^{bias} R_j + \beta_{1,jbk}^{B} \bar{x}_j & \text{if } k \neq b \end{cases} \tag{7}$$

Then the bias-adjusted relative treatment effect $\delta_{jbk}^{bias} = \delta_{jbk} + \gamma_{jbk}$ can be assumed exchangeable across studies $ *{jbk}^{bias} N(g*{bk}+d_{Ak}-d_{Ab}, ^2/q_j)$ or fixed as $ *{jbk}^{bias}= g*{bk}+d_{Ak}-d_{Ab}$. In this parameterisation, instead of assigning prior to the between-study heterogeneity in bias effect τ_γ, we model the RoB weight $q_j = \tau^2/(\tau^2 + \tau_\gamma^2)$ for each study. This quantity $0 < q_j < 1$ quantifies the proportion of the between-study heterogeneity that is not explained by accounting for risk of bias. The values of v determine the extent studies at high risk of bias will be down-weighted on average. Setting $v = 1$ gives $E(q_j) = v/(v+1) = 0.5$, which means that high risk of bias studies will be penalized by 50% on average. In `crossnma.model()`, the user can assign the average down-weight $E(q_j)$ to the argument `down.wgt`.

Bias-adjusted model 2

Another way to incorporate the RoB of the study is by replacing δ_{jbk} by a "bias-adjusted" relative treatment effect θ_{jbk}. Then θ_{jbk} is modeled with a bimodal normal distribution as described in Section 2.5. For more details see Verde (2020).

NMR model for IPD studies

$$logit(p_{ijk}) = \begin{cases} u_{jb} + \beta_{0j} x_{ijk} & \text{if } k = b \\ u_{jb} + \theta_{jbk} + \beta_{0j} x_{ijk} + \beta_{1,jbk}^{w} x_{ijk} + (\beta_{1,jbk}^{B} - \beta_{1,jbk}^{w})\bar{x}_j & \text{if } k \neq b \end{cases} \tag{8}$$

NMR model for AD studies

$$logit(p_{jk}) = \begin{cases} u_{jb} & \text{if } k = b \\ u_{jb} + \theta_{jbk} + \beta_{1,jbk}^{B} \bar{x}_j & \text{if } k \neq b \end{cases} \tag{9}$$

where the bias-adjusted relative treatment effect (θ_{jk}) are modeled via random-effects model with a mixture of two normal distributions.

$$\theta_{jbk} \sim (1 - \pi_j) N(d_{Ak} - d_{Ab}, \tau^2) + \pi_j N(d_{Ak} - d_{Ab} + \gamma_{jbk}, \tau^2 + \tau_\gamma^2)$$

Alternatively, we can summarize these relative effects assuming a common-effect model

$$\theta_{jbk} = d_{Ak} - d_{Ab} + \pi_j \gamma_{jbk}$$

Assumptions about the model parameters

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

Parameter	Assumptions	Argument in `crossnma.model()`
Relative treatment effect (δ_{jbk})	Random-effects: $\delta_{jbk} \sim N(d_{Ak} - d_{Ab}, \tau^2)$	`trt.effect='random'`
	Common-effect: $\delta_{jbk} = d_{Ak} - d_{Ab}$	`trt.effect='common'`
Covariate effect (β_{0j})	Independent effects: $\beta_{0j} \sim N(0, 10^2)$	`reg0.effect='independent'`
	Random-effects: $\beta_{0j} \sim N(B_0, \tau_0^2)$	`reg0.effect='random'`
Within-study covariate-treatment interaction ($\beta^W_{1,jbk}$)	Independent effects: $\beta^W_{1,jbk} \sim N(0, 10^2)$	`regw.effect='independent'`
	Random-effects: $\beta^W_{1,jbk} \sim N(B^W_{1,Ak} - B^W_{1,Ab}, \tau^2_W)$	`regw.effect='random'`
	Common-effect: $\beta^W_{1,jbk} = B^W_{1,Ak} - B^W_{1,Ab}$	`regw.effect='common'`
Between-study covariate-treatment interaction ($\beta^B_{1,jbk}$)	Independent effects: $\beta^B_{1,jbk} \sim N(0, 10^2)$	`regb.effect='independent'`
	Random-effects: $\beta^B_{1,jbk} \sim N(B^B_{1,Ak} - B^B_{1,Ab}, \tau^2_B)$	`regb.effect='random'`
	Common-effect: $\beta^B_{1,jbk} = B^B_{1,Ak} - B^B_{1,Ab}$	`regb.effect='common'`
Bias effect ($\gamma_{m,jbk}$), $m = 1, 2$	Random-effects: $\gamma_{m,jbk} \sim N(g_{m,bk}, \tau^2_{m,\gamma})$	`bias.effect='random'`
	Common-effect: $\gamma_{m,jbk} = g_{m,bk}$	`bias.effect='common'`

Mean bias effect $g_{m,bk}$	The treatment k is active. $g_{m,bk} = g_m$ (b inactive), $g_{m,bk} = 0$ (b active & no bias) $g_{m,bk} = g^{act}_m$(b active & bias)	`unfav=0`, `bias.group=1`
		`unfav=1`, `bias.group=0`
		`unfav=1`, `bias.group=2`

Bias probability (π_j)	$\pi_j \sim Beta(a_1, a_2)$	`pi.high.nrs`, `pi.low.nrs`, `pi.high.rct`, `pi.low.rct`
	$\pi_j = e + fz_j$	`bias.covariate`

Synthesis of studies comparing drugs for relapsing-remitting multiple sclerosis

Description of the data

The data we use are fictitious but have been developed to resample to real RCTs with IPD and aggregate data included in Tramacere and Filippini (2015). The studies provide either aggregate data `stddata` (2 RCTs) or as individual participant data `ipddata` (3 RCTs and 1 cohort study). Both datasets compare in total four drugs which are anonymized.

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

```
head(ipddata)
#>    id relapse treat design age sex rob unfavored bias.group year
#> 1   1       0     D    rct  22   0 low         1          1 2002
#> 3   1       0     D    rct  37   0 low         1          1 2002
#> 6   1       0     D    rct  34   1 low         1          1 2002
#> 7   1       0     D    rct  38   1 low         1          1 2002
#> 9   1       0     D    rct  31   0 low         1          1 2002
#> 10  1       0     D    rct  38   0 low         1          1 2002
```

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

The aggregate data has the standard format for meta-analysis with the exact same variable names

```
head(stddata)
#>   id relapse   n treat design  age sex  rob unfavored bias.group year
#> 1  1      19  25     A    rct 34.3 0.2 high         0          1 2010
#> 2  1      11  25     C    rct 34.3 0.3 high         1          1 2010
#> 3  2      97 126     A    rct 30.0 0.4 high         0          1 2015
#> 4  2      89 125     C    rct 30.0 0.5 high         1          1 2015
```

Analysis

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

Unadjusted network meta-analysis

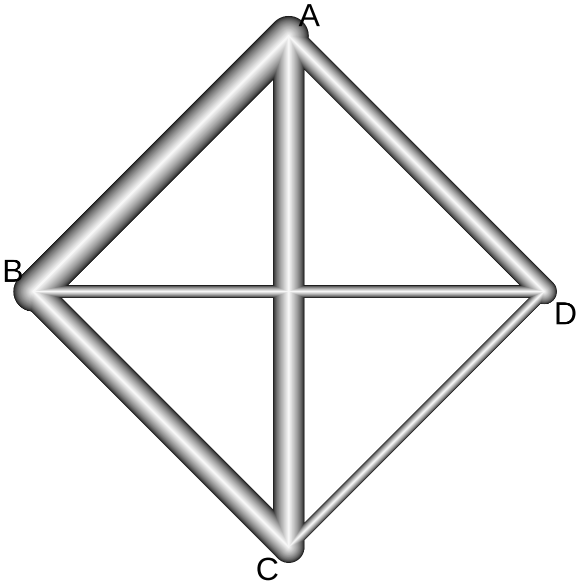
We start by providing the essential variables which - as stated earlier - must have equal names in both data sets. Next, we give the names of the datasets on participant-level (argument prt.data) and aggregate data (argument std.data). The reference treatment needs to be assigned (we set it to drug A). 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 as method.bias = 'naive'.

Optionally, we can specify a prior to the common heterogeneity of the treatment effect across studies. We indicate that distribution in the argument prior as tau.trt='dunif(0,3)', see below.

```
# jags model: code+data
mod1 <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = NULL, trt.effect = "random",
  #---------- bias adjustment ----------
  method.bias = "naive",
  #---------- assign a prior ----------
  prior = list(tau.trt='dunif(0,3)')
  )
```

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.

```
netgraph(mod1)
```



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

```
# run jags
jagsfit1 <- crossnma.run(mod1,
  n.adapt = 50, n.iter = 500, n.burnin = 200,
  thin = 1, n.chains = 2)
#> NOTE: Stopping adaptation
```

We summarize the estimated parameters in the following table.

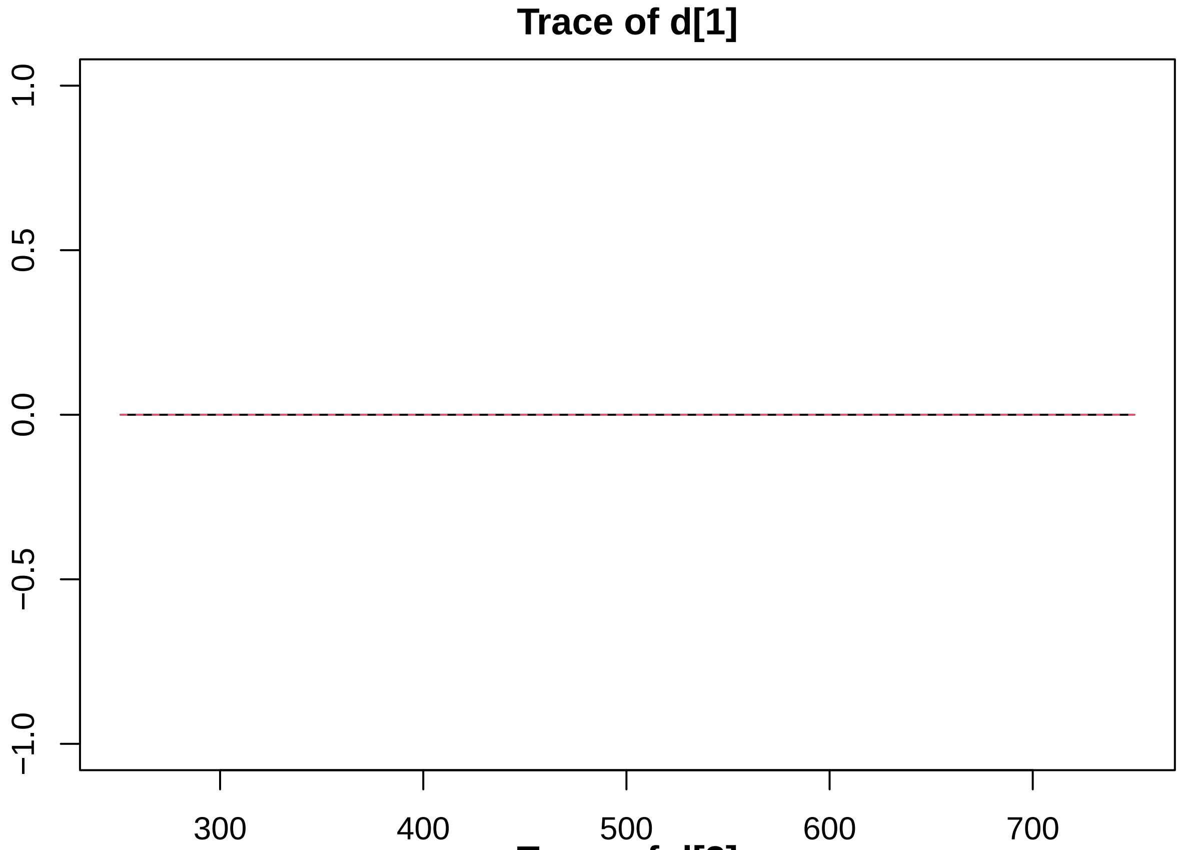
```
knitr::kable(summary(jagsfit1, exp = FALSE))
```

	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
d.A	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	NaN	0
d.B	-0.9201955	0.4366660	-1.9106193	-0.8866423	-0.1096985	1.005	329
d.C	-0.1941044	0.4484781	-1.1549939	-0.1599743	0.6581318	1.018	341
d.D	-1.0903841	0.5647798	-2.3171071	-1.0596529	-0.0611785	1.012	413
tau	0.6346353	0.3861362	0.1279521	0.5608700	1.7356238	1.089	74

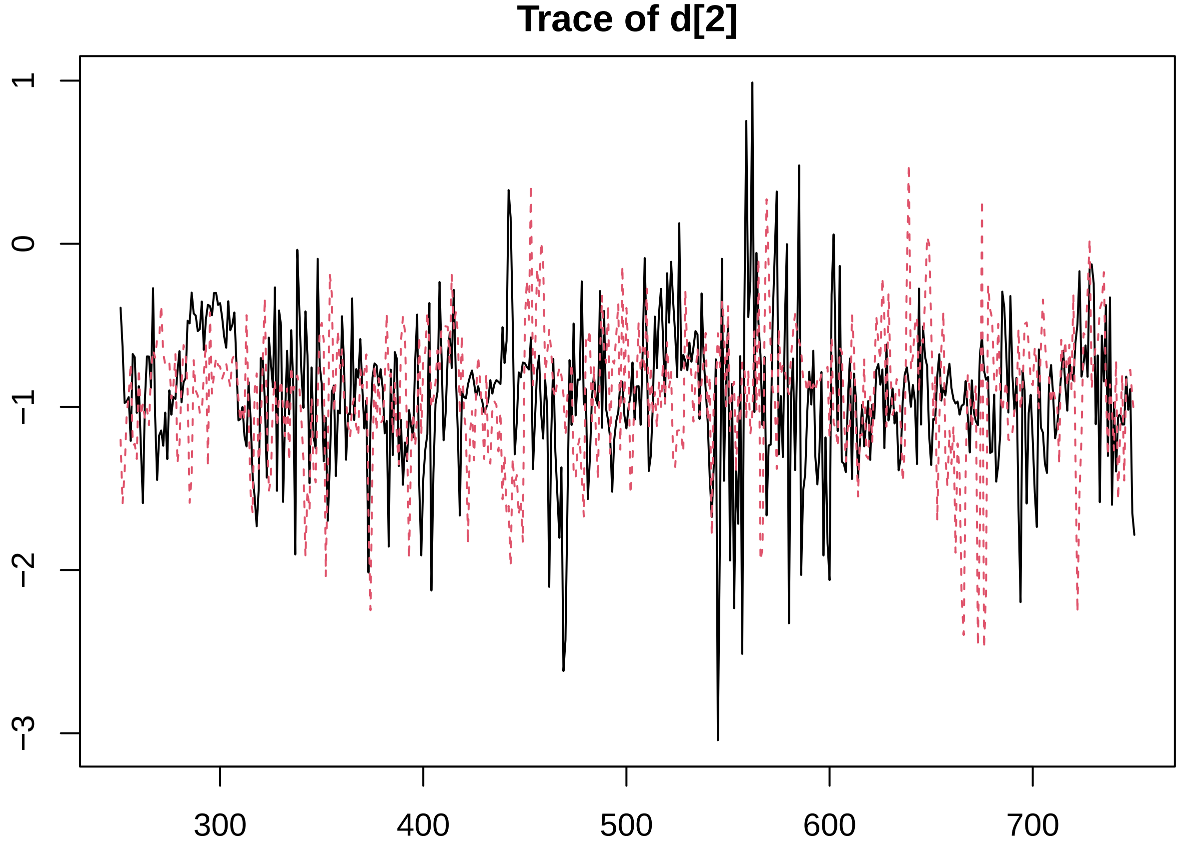
The estimated OR of B vs A can be obtained as exp(d.B) and similarly for exp(d.C) and exp(d.D) are the ORs of C and D relative to A, respectively. The value of tau refers to the estimates of the heterogeneity standard deviation in the relative treatment effects across studies.

We need also to assess the convergence of the MCMC chains either by checking the Gelman and Rubin statistic, Rhat (it should be approximately 1) in the table above or visually inspect the trace plot.

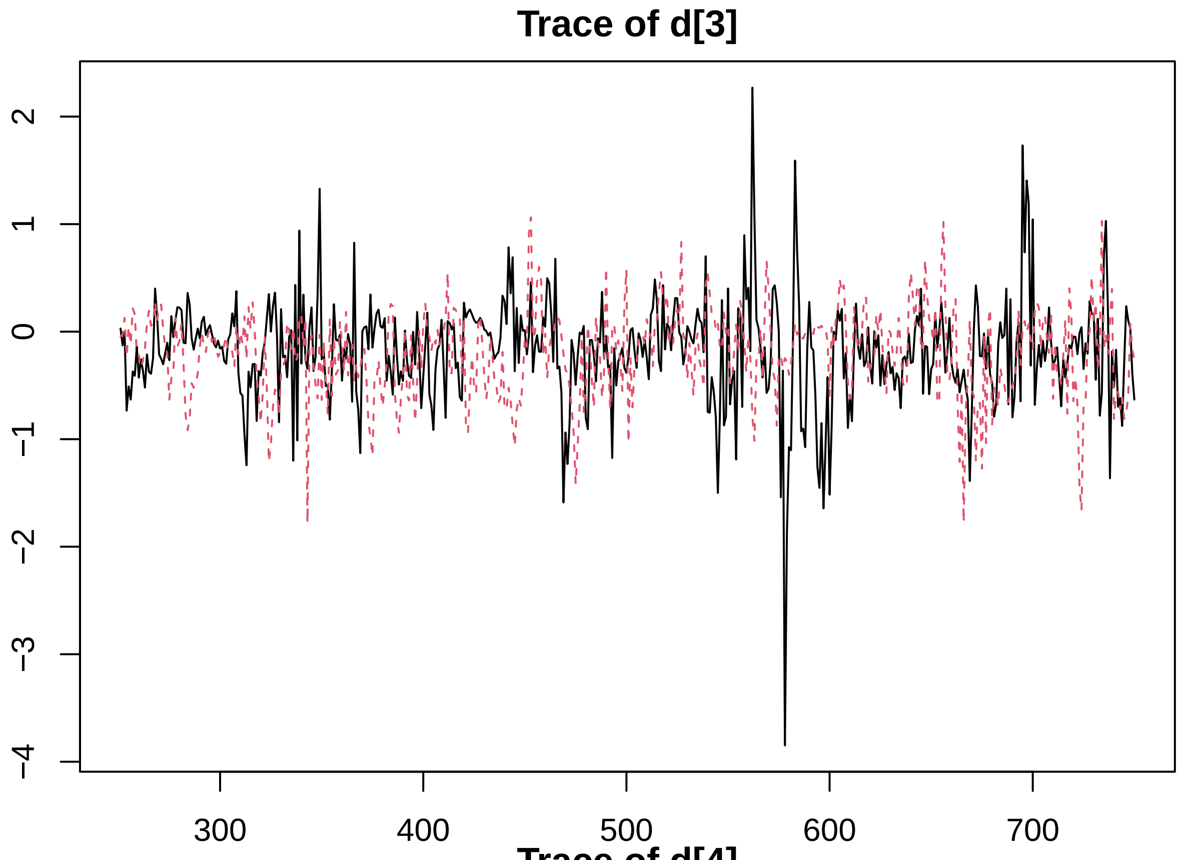
```
par(mar = rep(2, 4))
coda::traceplot(jagsfit1$samples)
```

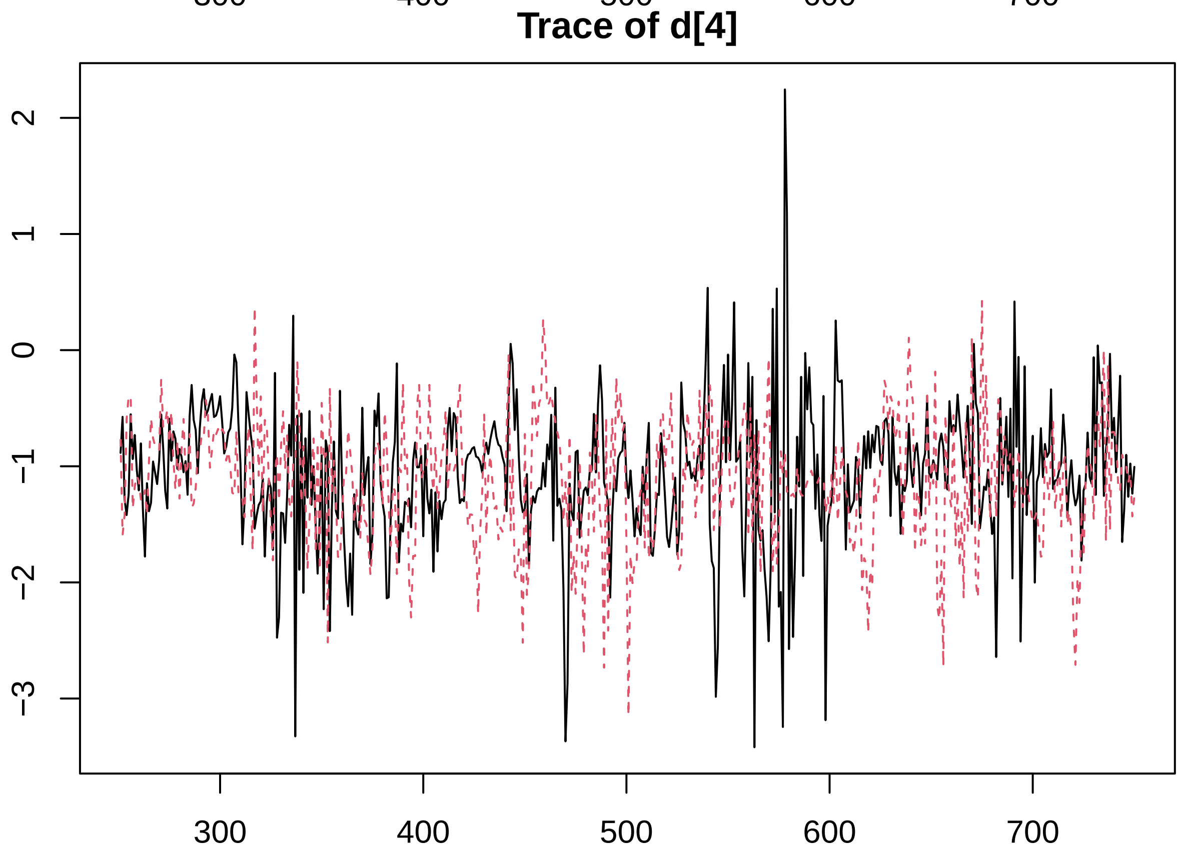


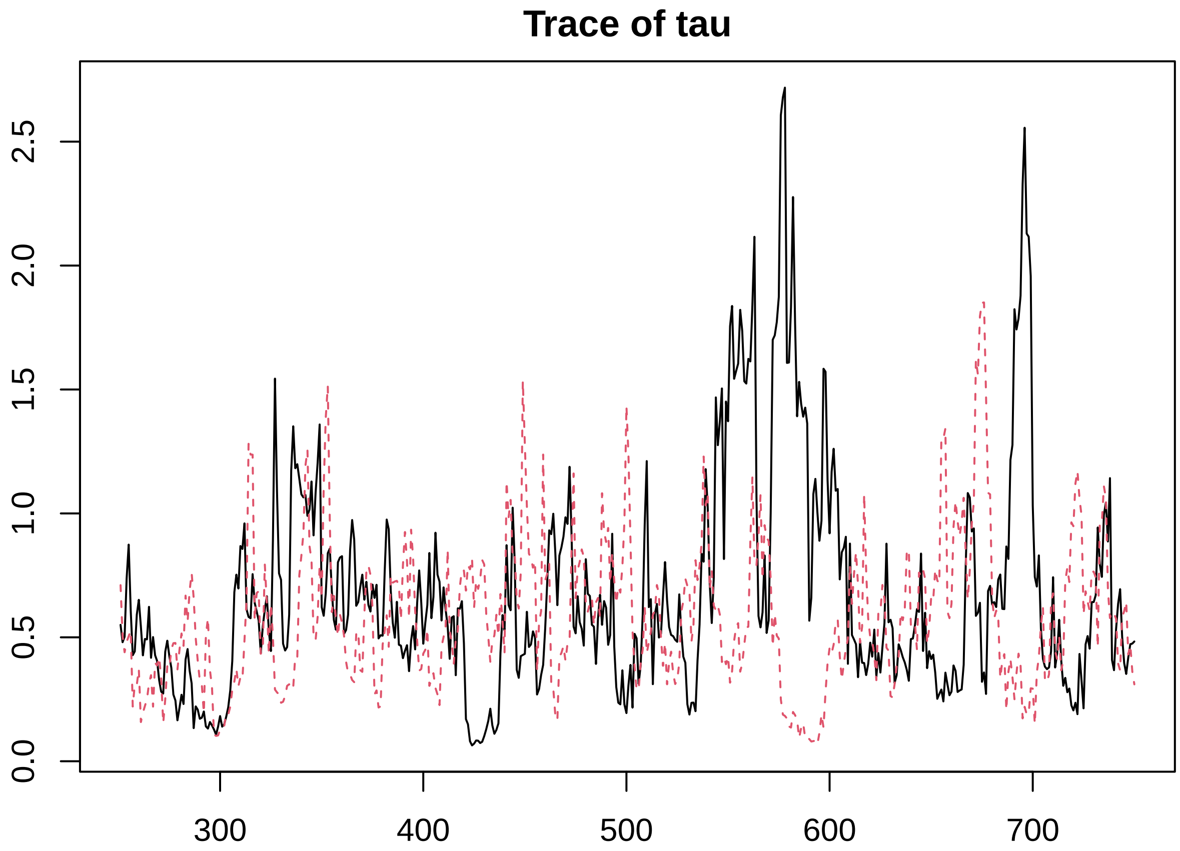
Trace of d[1]



Trace of d[2]


Trace of d[3]
2
1
0
−1
−2
−3
−4
300
400
500
600
700


Trace of d[4]
2
1
0
−1
−2
−3
300
400
500
600
700



Unadjusted network meta-regression

In this part, we set argument `cov1 = age` to run a NMR model with one covariate. Again, datasets `ipddata` and `stddata` must use the same variable name.

```
# jags model: code+data
mod2 <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random",
  #---------- bias adjustment ----------
  method.bias = "naive",
  #----------  meta-regression ----------
  cov1 = age,
  split.regcoef = FALSE
  )
```

We could add two more covariates to the NMR model using arguments `cov2` and `cov3`.

The MCMC is run under the same set up as in the network meta-analysis.

```
# run jags
jagsfit2 <- crossnma.run(mod2,
  n.adapt = 50, n.iter = 500, n.burnin = 200,
  thin = 1, n.chains = 2)
#> NOTE: Stopping adaptation
```

and the output table is presented below

```
knitr::kable(summary(jagsfit2, exp = FALSE))
```

	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
b_1	-0.0025060	0.1155742	-0.2905483	-0.0044398	0.2580818	1.005	846
d.A	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	NaN	0
d.B	-0.7924699	0.5307499	-1.9432979	-0.7997837	0.1730272	1.017	31
d.C	-0.2219132	0.5368606	-1.2958327	-0.2250837	0.7897003	1.077	44
d.D	-1.0874143	0.7013984	-2.5847284	-1.0542939	0.2405581	1.028	59
tau	0.5264074	0.3703657	0.0241210	0.4674878	1.4352358	1.048	56
tau.b_1	0.1212107	0.1581617	0.0011051	0.0399946	0.5331789	1.021	12

Now, we additionally estimate 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. Here, we obtain a single estimate because we choose to not split the within- and between-study age coefficients ($\beta^{w}_{1,jbk} = \beta^{B}_{1,jbk} = \beta_{1,jbk}$) to improve the convergence of MCMC.

The league table summarizes the relative effect with the 95% credible interval of each treatment on the top compared to the treatment on the left. All estimates are computed for participant age 38. We can display the table in wide format

```
league(jagsfit2, exp = TRUE, prt.cov1.value = 38)
#>   A                      B                     C                     
#> A "A"                    "0.45 (0.14 to 1.19)" "0.80 (0.27 to 2.20)" 
#> B "2.23 (0.84 to 6.98)"  "B"                   "1.73 (0.68 to 5.51)" 
#> C "1.25 (0.45 to 3.65)"  "0.58 (0.18 to 1.48)" "C"                   
#> D "2.87 (0.79 to 13.26)" "1.30 (0.31 to 6.15)" "2.17 (0.61 to 10.65)"
#>   D                    
#> A "0.35 (0.08 to 1.27)"
#> B "0.77 (0.16 to 3.22)"
#> C "0.46 (0.09 to 1.64)"
#> D "D"
```

or in long format

```
league(jagsfit2, exp = TRUE, prt.cov1.value = 38, direction = "long")
#>    Treatment Comparator    median        lci       uci
#> 1          A          A 1.0000000 1.00000000  1.000000
#> 2          B          A 0.4494262 0.14324198  1.188899
#> 3          C          A 0.7984495 0.27366987  2.202738
#> 4          D          A 0.3484384 0.07541666  1.271959
#> 5          A          B 2.2250597 0.84111527  6.982311
#> 6          B          B 1.0000000 1.00000000  1.000000
#> 7          C          B 1.7297067 0.67657651  5.511979
#> 8          D          B 0.7681926 0.16265490  3.221160
#> 9          A          C 1.2524278 0.45398118  3.654038
#> 10         B          C 0.5781327 0.18142323  1.478034
#> 11         C          C 1.0000000 1.00000000  1.000000
#> 12         D          C 0.4600641 0.09386825  1.636040
#> 13         A          D 2.8699481 0.78618899 13.259704
#> 14         B          D 1.3017574 0.31044721  6.147986
#> 15         C          D 2.1736104 0.61123233 10.653236
#> 16         D          D 1.0000000 1.00000000  1.000000
```

Using non-randomized studies (NRS) to construct priors for the treatment effects

To run NMA with a prior constructed from NRS, two additional arguments are needed: we indicate using NRS as a prior by setting `method.bias='prior'`. That means that the model runs internally NMA with only NRS data which are then used to construct informative priors. This requires defining MCMC settings (the number of adaptations, iterations, burn-ins, thinning and chains) in the argument `run.nrs`.

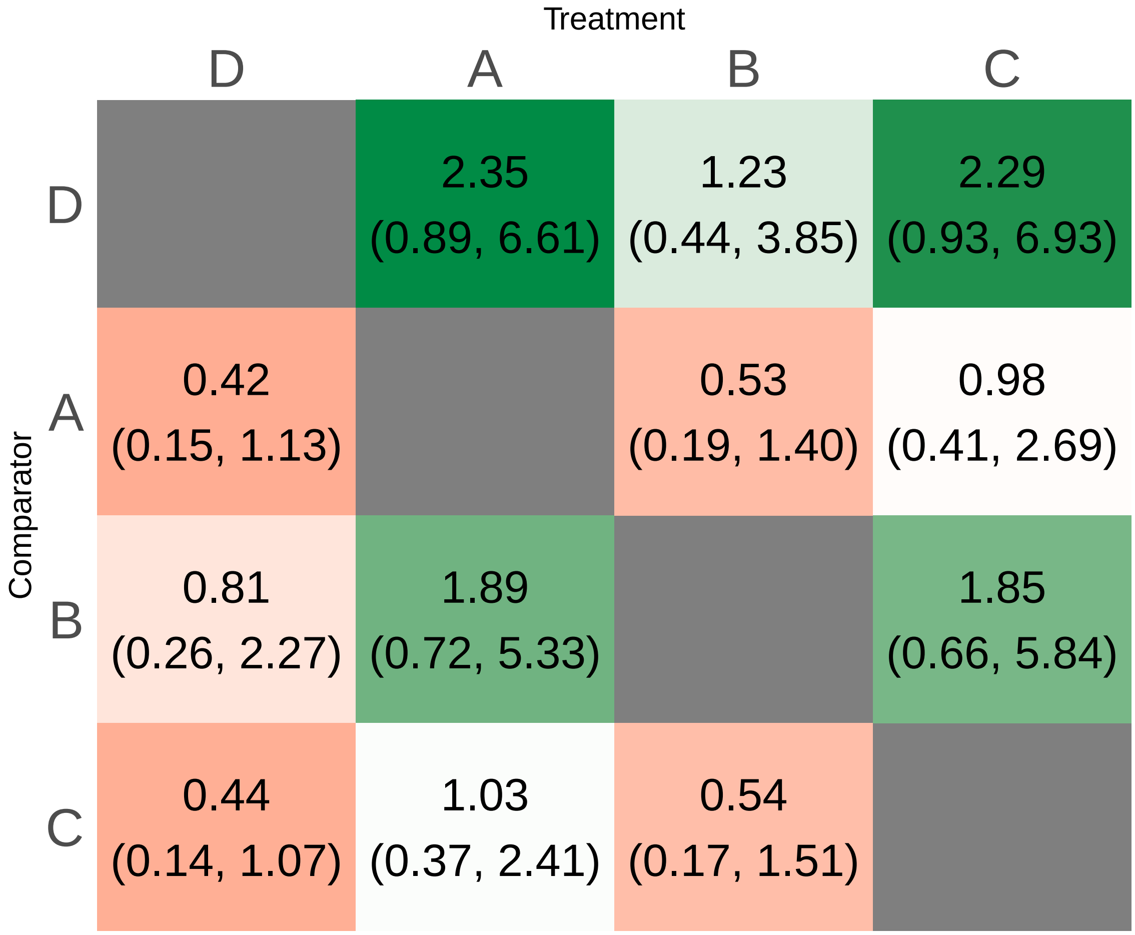
In this method, the prior for the basic parameters is set to a normal distribution. For basic parameters not examined in the NRS, the code sets a minimally informative prior `d~dnorm(0, 1e-2)`. To account for possible bias, the means of the distribution can be shifted by `mean.shift` and/or the variance 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 <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "D", trt.effect = "random",
  #----------  meta-regression ----------
  cov1 = age,
  split.regcoef = FALSE,
  #---------- bias adjustment ----------
  method.bias = "prior",
  run.nrs =
    list(trt.effect = "common",
         var.infl = 0.6, mean.shift = 0,
         n.adapt = 500, n.iter = 10000, n.burnin = 4000,
         thin = 1, n.chains = 2)
  )
#> Compiling model graph
#>    Resolving undeclared variables
#>    Allocating nodes
#> Graph information:
#>    Observed stochastic nodes: 150
#>    Unobserved stochastic nodes: 3
#>    Total graph size: 475
#>
#> Initializing model
```

```
# run jags
jagsfit3 <- crossnma.run(mod3,
  n.adapt = 50, n.iter = 500, n.burnin = 200,
  thin = 1, n.chains = 2)
#> NOTE: Stopping adaptation
```

The heat plot summarizes the relative effect with the 95% credible interval of each treatment on the top compared to the treatment on the left. All estimates are computed for participant age 38.

```
heatplot(jagsfit3, exp = TRUE, prt.cov1.value = 38,
  cell.text.size = 6, trt.name.size = 20, axis.title.size = 12)
```



Comparator \ Treatment	D	A	B	C
D		2.35 (0.89, 6.61)	1.23 (0.44, 3.85)	2.29 (0.93, 6.93)
A	0.42 (0.15, 1.13)		0.53 (0.19, 1.40)	0.98 (0.41, 2.69)
B	0.81 (0.26, 2.27)	1.89 (0.72, 5.33)		1.85 (0.66, 5.84)
C	0.44 (0.14, 1.07)	1.03 (0.37, 2.41)	0.54 (0.17, 1.51)	

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.

To fit the model, we set `method.bias='adjust1'` and we need to provide the bias variable `bias=rob` in the datasets. The direction of bias is determined by the column `unfav=unfavored` which indicates the unfavoured treatment. The mean bias effect can be indicated by `bias.group`, 0 (`bias.group=0`), g (`bias.group=1`) or g^{act} (`bias.group=2`). By default, the effect of bias is assumed to be additive `bias.type='add'` and equal across studies `bias.effect='common'`. We also use the `year` of study publication to estimate the study-probability of bias, `bias.covariate = year`.

```
# jags model: code+data
mod4 <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random",
  #---------- bias adjustment ----------
  method.bias = 'adjust1',
  bias.type = 'add',
  bias.effect = 'common',
  bias = rob,
  unfav = unfavored,
  bias.group = bias.group,
  bias.covariate = year
)
```

```
# run jags
jagsfit4 <- crossnma.run(mod4,
  n.adapt = 50, n.iter = 500, n.burnin = 200,
  thin = 1, n.chains = 2)
#> NOTE: Stopping adaptation
```

The results are presented below

```
knitr::kable(summary(jagsfit4, exp = FALSE))
```

	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
d.A	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	NaN	0
d.B	-0.7942201	0.4070236	-1.6813085	-0.7829139	0.0314850	1.061	394
d.C	0.0710663	0.4705409	-0.8131723	0.0040243	1.0824685	1.620	259
d.D	-1.0476234	0.4833161	-1.9966401	-1.0527117	-0.1459673	1.005	339
g	-0.5368519	6.9329043	-16.8908410	-0.8253826	15.8801668	1.292	195
tau	0.5424540	0.3463539	0.0715710	0.4657351	1.4878692	1.019	74

The parameter `g` refers to the mean bias effect, common for all studies.

Bias-adjusted model 2

The arguments for `method.bias='adjust2'` are similar to the ones used before in `method.bias='adjust1'`.

```
# jags model: code+data
mod5 <- crossnma.model(treat, id, relapse, n, design,
  prt.data = ipddata, std.data = stddata,
  reference = "A", trt.effect = "random",
  #---------- bias adjustment ----------
  method.bias = 'adjust2',
  bias.type = 'add',
  bias = rob,
  unfav = unfavored,
  bias.group = bias.group,
)
```

```
# run jags
jagsfit5 <- crossnma.run(mod5,
  n.adapt = 50, n.iter = 500, n.burnin = 200,
  thin = 1, n.chains = 2)
#> NOTE: Stopping adaptation
```

```
knitr::kable(summary(jagsfit5, exp = FALSE))
```

	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
d.A	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	NaN	0
d.B	-0.8769505	0.3628223	-1.7120684	-0.7976142	-0.3659198	1.103	102
d.C	-0.2251466	0.3702792	-0.9983325	-0.1601902	0.3278068	1.137	85
d.D	-1.2834063	0.5186745	-2.5620925	-1.1567583	-0.4674291	1.101	51
g	0.4357752	0.5306283	-0.4206426	0.3886777	1.5416454	1.104	32
tau	0.4416808	0.3822280	0.0109361	0.3499329	1.3827847	1.454	17

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.

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

Tramacere, Del Giovane, I, and G Filippini. 2015. "Immunomodulators and Immunosuppressants for Relapsing-remitting Multiple Sclerosis: A Network Meta-analysis." *Cochrane Database of Systematic Reviews*, no. 9. https://doi.org/10.1002/14651858.CD011381.pub2.

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.