

# crossnma: Cross-Design & Cross-Format Network Meta-Analysis and Regression

Tasnim Hamza, Guido Schwarzer and Georgia Salanti

## 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 set up the package **crossnma** to synthesize all available evidence for a binary outcome with the odds ratio as effect measure.

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.

## 2 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 <code>crossnma.model()</code>
$i = 1, \dots, np_j$	participant id	
$j = 1, \dots, ns$	study id	<b>study</b>
$k = 1, \dots, K$	treatment index	<b>trt</b>
$ns_{IPD}, ns_{AD}, ns_{RCT}, ns_{NRS}$	the number of studies. The index refers to the design or format of the study	
$y_{ijk}$	binary outcome (0/1)	<b>outcome</b>
$p_{ijk}$	probability of the event to occur	
$r_{jk}$	the number of events per arm	<b>outcome</b>
$n_{jk}$	the sample size per arm	<b>n</b>
$b$	the study-specific reference	*
$u_{jb}$	The treatment effect of the study-specific reference $b$ when $x_{ijk} = \bar{x}_j = 0$	
$\delta_{jbk}$	log(OR) of treatment $k$ relative to $b$	

Notation	Description	Argument in <code>crossnma.model()</code>
$x_{ijk}$	the covariate	<code>cov1, cov2, cov3</code>
$\bar{x}_j$	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 <code>reference</code> to assign the reference treatment
$z_j$	study characteristics to estimate the bias probability $\pi_j$	<code>bias.covariate</code>
$w$	common inflation factor of variance for the NRS estimates	the element <code>var.infl</code> in <code>run.nrs</code>
$\zeta$	common mean shift of the NRS estimates	the element <code>mean.shift</code> in <code>run.nrs</code>

\*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.

## 2.1 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

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

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

### NMR model for AD studies

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

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

## 2.2 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  $\tilde{d}_{Ak}^{NRS}$  and variance  $V_{Ak}^{NRS}$  (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,  $d_{Ak} \sim \mathcal{N}(\tilde{d}_{Ak}^{NRS}, V_{Ak}^{NRS})$ . We can adjust for potential biases associated with NRS by either shifting the mean of the prior distribution with a bias term  $\zeta$  or by dividing the prior variance with a common inflation factor  $w, 0 < w < 1$  controls NRS contribution. The assigned priors become  $d_{Ak} \sim \mathcal{N}(\tilde{d}_{Ak}^{NRS} + \zeta, V_{Ak}^{NRS}/w)$ .

## 2.3 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,  $\delta_{jbk}$  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

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

### NMR model for AD studies

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

The bias indicator  $R_j$  follows the following distribution

$$R_j \sim \text{Bernoulli}(\pi_j)$$

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

$$\pi_j \sim \text{Beta}(a_1, a_2)$$

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 `prior.pi.high.rct = 'dbeta(10, 1)'`, low bias RCT `prior.pi.low.rct = 'dbeta(1, 10)'`, high bias NRS `prior.pi.high.nrs = 'dbeta(30, 1)'` and low bias NRS `prior.pi.low.nrs = 'dbeta(1, 30)'`. Alternatively, we can use the study characteristics  $z_j$  to estimate  $\pi_j$  through a logistic transformation (internally coded).

We combine the multiplicative and the additive treatment-specific bias effects across studies by assuming they are exchangeable  $\gamma_{1,jbk} \sim \mathcal{N}(g_{1,bk}, \tau_{1,\gamma}^2)$ ,  $\gamma_{2,jbk} \sim \mathcal{N}(g_{2,bk}, \tau_{2,\gamma}^2)$  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.

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

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  $g_m^{\text{act}}$ . The direction of bias  $\text{dir}_{bk}$  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  $g_m^{\text{act}}$ ).

Another parameterisation of the logistic model with additive bias effect is

#### NMR model for IPD studies

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

#### NMR model for AD studies

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

Then the bias-adjusted relative treatment effect ( $\delta_{jbbk}^{bias} = \delta_{jbbk} + \gamma_{jbbk}$ ) can be assumed exchangeable across studies  $\delta_{jbbk}^{bias} \sim \mathcal{N}(g_{bk} + d_{Ak} - d_{Ab}, \tau^2/q_j)$  or fixed as  $\delta_{jbbk}^{bias} = g_{bk} + d_{Ak} - d_{Ab}$ . In this parameterisation, instead of assigning prior to the between-study heterogeneity in bias effect  $\tau_\gamma$ , 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 `crossma.model()`, the user can assign the average down-weight  $E(q_j)$  to the argument `down.wgt`.

## 2.4 Bias-adjusted model 2

Another way to incorporate the RoB of the study is by replacing  $\delta_{jbbk}$  by a ‘‘bias-adjusted’’ relative treatment effect  $\theta_{jbbk}$ . Then  $\theta_{jbbk}$  is modeled with a bimodal normal distribution as described in Section 2.5. For more details see Verde (2020).

#### NMR model for IPD studies

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

#### NMR model for AD studies

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

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

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

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

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

## 2.5 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 <code>crossma.model()</code>
Relative treatment effect ( $\delta_{j bk}$ )	Random-effects: $\delta_{j bk} \sim \mathcal{N}(d_{Ak} - d_{Ab}, \tau^2)$	<code>trt.effect='random'</code>
	Common-effect: $\delta_{j bk} = d_{Ak} - d_{Ab}$	<code>trt.effect='common'</code>
Covariate effect ( $\beta_{0j}$ )	Independent effects: $\beta_{0j} \sim \mathcal{N}(0, 10^2)$	<code>reg0.effect='independent'</code>
	Random-effects: $\beta_{0j} \sim \mathcal{N}(B_0, \tau_0^2)$	<code>reg0.effect='random'</code>
Within-study covariate-treatment interaction ( $\beta_{1, j bk}^W$ )	Independent effects: $\beta_{1, j bk}^W \sim \mathcal{N}(0, 10^2)$	<code>regw.effect='independent'</code>
	Random-effects: $\beta_{1, j bk}^W \sim \mathcal{N}(B_{1, Ak}^W - B_{1, Ab}^W, \tau_W^2)$	<code>regw.effect='random'</code>
	Common-effect: $\beta_{1, j bk}^W = B_{1, Ak}^W - B_{1, Ab}^W$	<code>regw.effect='common'</code>
Between-study covariate-treatment interaction ( $\beta_{1, j bk}^B$ )	Independent effects: $\beta_{1, j bk}^B \sim \mathcal{N}(0, 10^2)$	<code>regb.effect='independent'</code>
	Random-effects: $\beta_{1, j bk}^B \sim \mathcal{N}(B_{1, Ak}^B - B_{1, Ab}^B, \tau_B^2)$	<code>regb.effect='random'</code>
	Common-effect: $\beta_{1, j bk}^B = B_{1, Ak}^B - B_{1, Ab}^B$	<code>regb.effect='common'</code>
Bias effect ( $\gamma_{m, j bk}$ ), $m = 1, 2$	Random-effects: $\gamma_{m, j bk} \sim \mathcal{N}(g_{m, bk}, \tau_{m, \gamma}^2)$	<code>bias.effect='random'</code>
	Common-effect: $\gamma_{m, j bk} = g_{m, bk}$	<code>bias.effect='common'</code>
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) $g_{m, bk} = g_m^{act}$ ( $b$ active & bias)	<code>unfav=0, bias.group=1 unfav=1,</code> <code>bias.group=0 unfav=1,</code> <code>bias.group=2</code>
Bias probability ( $\pi_j$ )	$\pi_j \sim \text{Beta}(a_1, a_2)$  $\pi_j = e + fz_j$	<code>pi.high.nrs, pi.low.nrs,</code> <code>pi.high.rct, pi.low.rct</code> <code>bias.covariate</code>

### 3 Synthesis of studies comparing drugs for relapsing-remitting multiple sclerosis

#### 3.1 Description of the data

The data we use are fictitious but resemble real RCTs with IPD and aggregate data included in Tramacere and Filippini (2015). The studies provide either individual participant data `ipddata` (3 RCTs and 1 cohort study) or aggregate data `stddata` (2 RCTs). In total, four drugs are compared which are anonymized.

The `ipddata` contains 1994 participants / rows. We display the first few rows of the data set:

```
dim(ipddata)
#> [1] 1944 10
head(ipddata)
#>   id relapse treat design age sex rob unfavored bias.group year
#> 1  1      0     D    rct  22  1 low          1          1 2002
#> 2  1      0     D    rct  31  1 low          1          1 2002
#> 3  1      0     D    rct  34  1 low          1          1 2002
#> 4  1      0     D    rct  38  0 low          1          1 2002
#> 5  1      0     D    rct  46  0 low          1          1 2002
#> 6  1      0     D    rct  45  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 `unfavoured`.

The aggregate meta-analysis data must be in long arm-based format with the exact same variable names and an additional variable with the sample sizes:

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

## 3.2 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()` to run the analysis through JAGS.

### 3.2.1 Unadjusted network meta-analysis

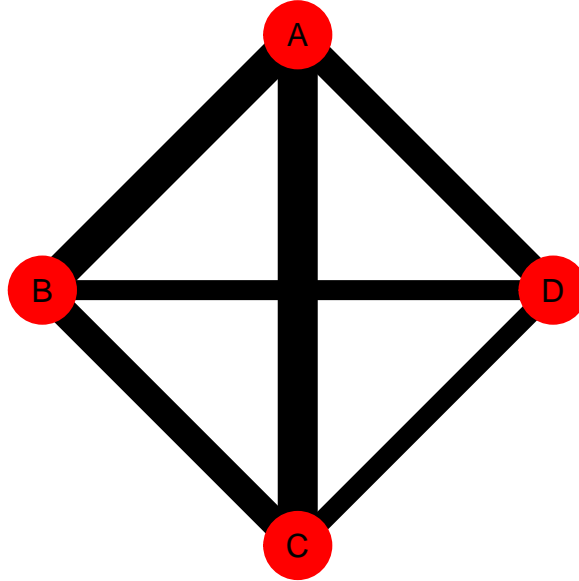
We start by providing the essential variables which - as stated earlier - must have equal names in both data sets. Next, we give the names of the datasets on participant-level (argument `prt.data`) and aggregate data (argument `std.data`). By default, binary data is analyzed using the odds ratio as a summary measure (`sm = "OR"`). The `reference` treatment can be assigned which by default is the first treatment (here: 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.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,
  trt.effect = "random",
  #----- bias adjustment -----
  method.bias = "naive",
  #----- assign a prior -----
  prior.tau.trt = "dunif(0, 3)"
)
#> Both designs are combined naively without acknowledging design differences.
```

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, cex.points = 7, adj = 0.5, plastic = FALSE)
```



Next, we fit the NMA model using `crossnma()`. We change the default settings for the number of iterations, burn-in and thinning.

```
# Run JAGS
jagsfit1 <- crossnma(mod1, n.iter = 5000, n.burnin = 2000, n.thin = 1)
jagsfit1
#>      Mean    SD  2.5%  50% 97.5% Rhat n.eff
#> exp(d.A)      .      .      .      .      .      .      .
#> exp(d.B) 0.470 0.194 0.322 0.468 0.692 1.009 865
#> exp(d.C) 0.629 0.211 0.416 0.629 0.944 1.015 582
#> exp(d.D) 0.341 0.265 0.202 0.341 0.571 1.004 671
#> tau      0.188 0.182 0.002 0.140 0.673 1.044 241
#> Mean and quantiles are exponentiated
```

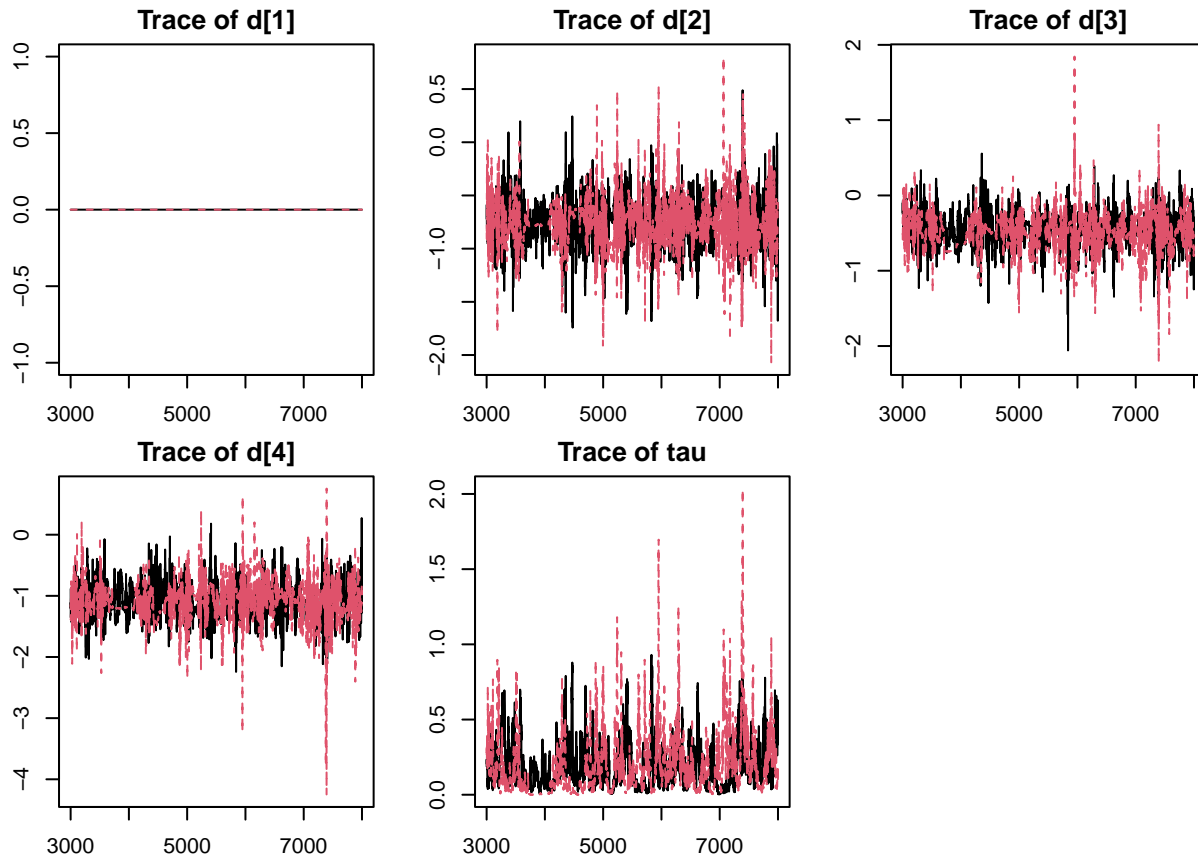
By default (argument `backtransf = TRUE`), estimated odds ratios, i.e.,  $\exp(d.B)$ ,  $\exp(d.C)$  and  $\exp(d.D)$ , are printed. The value of `tau` refers to the estimates of the heterogeneity standard deviation in the relative treatment effects `d.B`, `d.C` and `d.D` across studies.

We summarize the estimated parameters (argument `backtransf = FALSE`) in the following table.

	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
d.A	.	.	.	.	.	.	.
d.B	-0.755	0.194	-1.133	-0.759	-0.368	1.009	865
d.C	-0.463	0.211	-0.878	-0.463	-0.058	1.015	582
d.D	-1.076	0.265	-1.602	-1.077	-0.560	1.004	671
tau	0.188	0.182	0.002	0.140	0.673	1.044	241

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), mfrow = c(2, 3))
plot(jagsfit1)
```



### 3.2.2 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,
  trt.effect = "random",
  #----- bias adjustment -----
  method.bias = "naive",
  #----- meta-regression -----
  cov1 = age,
  split.regcoef = FALSE
)
```

*#> Both designs are combined naively without acknowledging design differences.*

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(mod2, n.iter = 5000, n.burnin = 2000, n.thin = 1)
```

and the output table is presented below

	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
d.A	.	.	.	.	.	.	.



	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
d.B	-0.899	0.347	-1.583	-0.894	-0.239	1.001	97
d.C	-0.468	0.377	-1.199	-0.464	0.283	1.006	127
d.D	-0.975	0.493	-1.896	-0.987	0.135	1.018	103
tau	0.220	0.193	0.008	0.170	0.726	1.024	219
b_1	0.000	0.067	-0.116	0.000	0.106	1.074	3300
tau.b_1	0.056	0.101	0.001	0.023	0.412	1.098	112

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 ( $\beta_{1,jbk}^w = \beta_{1,jbk}^B = \beta_{1,jbk}$ ) 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, cov1.value = 38, digits = 2)
#>
#>           A 0.41 (0.21 to 0.79) 0.63 (0.30 to 1.33)
#> 2.44 (1.27 to 4.87)           B 1.51 (0.75 to 3.38)
#> 1.59 (0.75 to 3.32) 0.66 (0.30 to 1.33)           C
#> 2.68 (0.87 to 6.66) 1.11 (0.34 to 2.79) 1.67 (0.56 to 4.57)
#>
#> 0.37 (0.15 to 1.14)
#> 0.90 (0.36 to 2.95)
#> 0.60 (0.22 to 1.79)
#>           D
```

or in long format

```
league(jagsfit2, cov1.value = 38, digits = 2, direction = "long")
#> Treatment Comparator median 2.5% 97.5%
#>      B      A 0.41 0.21 0.79
#>      C      A 0.63 0.30 1.33
#>      D      A 0.37 0.15 1.14
#>      A      B 2.44 1.27 4.87
#>      C      B 1.51 0.75 3.38
#>      D      B 0.90 0.36 2.95
#>      A      C 1.59 0.75 3.32
#>      B      C 0.66 0.30 1.33
#>      D      C 0.60 0.22 1.79
#>      A      D 2.68 0.87 6.66
#>      B      D 1.11 0.34 2.79
#>      C      D 1.67 0.56 4.57
```

### 3.2.3 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 arguments starts with `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  $d \sim \text{dnorm}(0, \{15 * \text{ML}\}^{-2})$ , where ML

is the largest maximum likelihood estimates of all relative treatment effects in all studies. To account for possible bias, the means of the distribution can be shifted by `run.nrs.mean.shift` and/or the variance can be inflated by `run.nrs.var.infl` to control the influence of NRS on the final estimation.

```
# 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.trt.effect= "common",
  run.nrs.var.infl = 0.6, run.nrs.mean.shift = 0,
  run.nrs.n.iter = 10000, run.nrs.n.burnin = 4000,
  run.nrs.n.thin = 1, run.nrs.n.chains = 2
)
```

```
# Run JAGS
jagsfit3 <- crossnma(mod3, n.iter = 5000, n.burnin = 2000, n.thin = 1)
```

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, cov1.value = 38,
  size = 6, size.trt = 20, size.axis = 12)
```

		Treatment			
		D	A	B	C
Comparator	D		2.62 (1.27 to 5.47)	1.05 (0.46 to 2.50)	1.66 (0.70 to 4.31)
	A	0.38 (0.18 to 0.79)		0.41 (0.21 to 0.83)	0.65 (0.30 to 1.40)
	B	0.95 (0.40 to 2.16)	2.42 (1.21 to 4.77)		1.54 (0.62 to 3.76)
	C	0.60 (0.23 to 1.43)	1.55 (0.71 to 3.39)	0.65 (0.27 to 1.61)	

### 3.2.4 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,
  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
)
```

```
#> Bias effect is assumed common across studies
```

```
# Run JAGS
```

```
jagsfit4 <- crossnma(mod4, n.iter = 5000, n.burnin = 2000, n.thin = 1)
```

The results are presented below

	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
d.A	.	.	.	.	.	.	.
d.B	-0.767	0.213	-1.189	-0.774	-0.310	1.012	1343
d.C	-0.474	0.224	-0.933	-0.473	-0.035	1.010	596
d.D	-1.073	0.280	-1.641	-1.070	-0.543	1.016	1028
tau	0.225	0.191	0.012	0.175	0.717	1.051	196
g	-0.076	19.786	-38.139	-0.318	39.382	1.000	6673

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

### 3.2.5 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,  
  trt.effect = "random",  
  #----- bias adjustment -----  
  method.bias = "adjust2",  
  bias.type = "add",  
  bias = rob,  
  unfav = unfavored,  
  bias.group = bias.group  
)
```

```
# Run JAGS
```

```
jagsfit5 <- crossnma(mod5, n.iter = 5000, n.burnin = 2000, n.thin = 1)
```

	Mean	SD	2.5%	50%	97.5%	Rhat	n.eff
d.A	.	.	.	.	.	.	.
d.B	-0.755	0.285	-1.297	-0.776	-0.163	1.009	387
d.C	-0.482	0.262	-1.074	-0.461	0.045	1.010	420
d.D	-1.094	0.359	-1.913	-1.070	-0.417	1.025	585
tau	0.291	0.247	0.001	0.225	0.941	1.018	100
g	0.009	0.366	-0.747	0.029	0.721	1.049	238

## 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*.