bmeta — a package for Bayesian meta-analysis and meta-regression in R

Ding Tao Gianluca Baio

Department of Statistical Science, University College London

1 The R package bmeta

The R package bmeta provides a collection of functions for analysing a relatively large class of models using Bayesian meta-analysis in R. The package includes functions for the calculation of various effect size or outcome measures (i.e. odds ratios, mean difference and incidence rate ratio) for different types of data, based on MCMC simulations. Users are allowed to fit fixed- and random-effects models with a pre-specified set of different priors to the data. When effects of additional covariates are observed, meta-regression can be performed to adjust for moderators. Table 1 presents all the models included in the package for selection with explanation.

Table 1: Models provided by bmeta

Outcome	Model name	Description	Type
Binary	std.norm	meta-analysis with normal prior	fixed/random
	std.dt	meta-analysis with t-distribution prior	fixed/random
	reg.norm	meta-regression with normal prior	fixed/random
	reg.dt	meta-regression with t-distribution prior	fixed/random
Continuous	std.ta	meta-analysis for studies reporting two arms separately	fixed/random
	std.mv	meta-analysis for studies reporting mean difference and variance only	fixed/random
	reg.ta	meta-regression for studies reporting two arms separately	fixed/random
	reg.mv	meta-regression for studies reporting mean difference and variance only	fixed/random
Count	std	meta-analysis	fixed
	std.unif	meta-analysis with uniform prior	random
	std.hc	meta-analysis with half-Cauchy prior	random
	reg	meta-regression	fixed
	reg.unif	meta-regression with uniform prior	random
	reg.hc	meta-regression with half-Cauchy prior	random

Furthermore, the package includes functions for plots which display modelling output in a standard manner (i.e. posterior or forest plots). Trace plots and some other diagnostic plots are also available for assessing model fit and performance.

bmeta works by allowing the user to specify the set of options in a standardised way in the R command terminal. Currently bmeta implements 23 models: those shown in Table 1 and a 'null' model based on assuming independence across the studies (no pooling); when the user has selected the required configuration (random vs fixed effects; choice of outcome and prior distributions; presence of covariates), bmeta writes a suitable JAGS file in the working directory. This is used to call the package R2Jags and perform the actual analysis, but can also be considered as a sort of "template" — the user can then modify to extend the modelling by changing priors in a way that is not automatically done by bmeta and use it for future references. For example, in the working directory, users can find a text file containing codes for a specific model:

```
## a. binary standard fixed-effects meta-analysis with normal prior
model {
for (s in 1:S){
y0[s]~dbin(pi0[s],n0[s])
y1[s]~dbin(pi1[s],n1[s])
logit(pi0[s])<-alpha[s]
logit(pi1[s])<-alpha[s]+delta
alpha[s]~dnorm(0,0.0001)
}
###prior###
delta~dnorm(0,0.0001)
rho<-exp(delta)
}
```

This basic template includes a vague prior for delta, which can be modified if there is some external evidence on this parameter, i.e. Normal(0,1). It is also possible to replace the prior for delta by Student t-distribution:

```
delta~dt(0,0.5,v)
v~dunif(0,8)
```

where v represents the degree of freedom of the distribution. Notice that as the prior distribution has been respecified, initial values (i.e. inits<-function(){...}) need to be modified accordingly. In the first case, inits is defined as:

```
inits<-function(){alpha=rnorm(S),delta=rnorm(1)}
```

Whereas in the second case, inits is replaced by:

```
inits<-function(){alpha=rnorm(S),v=runif(1)}
```

In order for bmeta to work properly, users need to ensure that the data are in a R 'list'. For example, the user may start with data on the set of studies to be analysed in csv format. These can be read in R

```
> data=read.csv("data.csv")
> data

     study year    y1   se1     y0    se0   X0
1    Hwang 2004 17.60  1.20  16.30   1.28    1
2  Ashrafi 2005  6.17  0.94  10.96   1.82    0
3   Ensieh 2010 10.88  1.07  13.37   1.37    0
4     Agha 2010  8.56  0.58  11.95   1.10    1
5      Kim 2012 14.20  0.57  14.70   0.57    1
6   Bulent 2012 17.11  0.75  18.11   1.04    1
```

and will then need to be pre-processed before they can be applied to bmeta. For example, the user could re-format the original values using the following code:

```
> data <- list(y0=data$y0,y1=data$y1,se0=data$se0,se1=data$se1)
> data
$y0
[1] 16.30 10.96 13.37 11.95 14.70 18.11
```

```
$y1
[1] 17.60  6.17 10.88  8.56 14.20 17.11

$se0
[1] 1.28 1.82 1.37 1.10 0.57 1.04

$se1
[1] 1.20 0.95 1.078 0.58 0.57 0.745
```

1.1 Models for binary data

Apart from a 'null' model, there are 8 models in bmeta for binary data, with 4 models for meta-analysis and another 4 for meta-regression. The bmeta package produces a pooled estimate (and 95% Credible Interval, CrI) in the form of odds ratio. The 'null' model is presented at the end of this section.

1.1.1 Modelling approach

Suppose for a set of $s = 1, 2, ..., S$ independent studies, we observe the total number of events out of the case arm and that for the control arm. Let π_{sk}, n_{sk} denote the probability of an event and the total number of individuals, respectively, in the kth arm and sth study and the number of events, y_{sk}, in study s arm k is assumed to follow a Binomial distribution:

$$y_{sk} \sim \text{Binomial}(\pi_{sk}, n_{sk}) \tag{1}$$

As the outcome is binary (individuals can either develop a disease/risk factor or not), a logit link function is then used to ensure that the probabilities can lie within a range of 0 and 1.

$$\text{logit}(\pi_{sk}) = \begin{cases} \alpha_s, & \text{if } k\text{=0 (control arm)} \\ \alpha_s + \delta_s, & \text{if } k\text{=1 (case arm)} \end{cases}$$

where α_s is the log-odds of the control group in sth study and δ_s is the study-specific intervention effects (on log scale).

While the logistic regression is a suitable model to be used here, other link functions (i.e. probit) can also be considered if appropriate. This can be easily done by modifying the model template produced by bmeta.

However, when effects of covariates are observed (cases where meta-regression needs to be applied):

$$\text{logit}(\pi_{sk}) = \begin{cases} \alpha_s + \boldsymbol{\beta X}, & \text{if } k\text{=0 (control arm)} \\ \alpha_s + \boldsymbol{\beta X} + \delta_s, & \text{if } k\text{=1 (case arm)} \end{cases}$$

where $\boldsymbol{X}$ (i.e. $X_0, X_1, X_2, \ldots, X_J$) is a matrix with covariates information (i.e. binary, continuous or categorical variables). For example, we may assume that certain studies use study population with a higher BMI, which may result in reduced treatment effect compared with studies that use individuals with a lower BMI. Heterogeneity due to differences in study-level BMI may be an issue if this factor is not considered for modelling. $\boldsymbol{\beta}$ (i.e. $\beta_0, \beta_1, \beta_2, \ldots, \beta_J$) is a vector of regression parameters of interest.

It should be noted that if a covariate is categorical, a 'baseline' category needs to be specified and then dummies for each of the rest categories should be created in order for the model to run properly.

For example, assume we suspect that there is impact of ethnicity on treatment effect and 4 different ethnic groups (i.e. 'White', 'Black', 'Asian' and 'Mixed') are reported by all the studies (i.e. one ethnic group for a single study). We then need to choose a 'baseline' category, i.e. 'Asian', and then create dummy (indicator) variables for the remaining three categories. Thus, in this meta-regression model, $J = 3$ covariates X_0,X_1,X_2 should be used, each representing the incremental effect of the ethnic group in comparison to the baseline.

Users need to pay special attention to categorical variables and format them into the right form before calling `bmeta` to perform analyses. This applies to all meta-regression models included in `bmeta`.

For fixed-effects models, it is assumed that all studies estimate the same intervention effect whereas for the random-effects models, we assume there is exchangeability between studies and therefore, the 'true' intervention effect is allowed to follow a distribution with some variability.

$$\begin{aligned} \delta_s &= \delta & \text{Fixed-effects model} \\ \delta_s &\sim \text{Normal}(\mu, \sigma^2) & \text{Random-effects models} \end{aligned}$$

where δ and μ are the pooled log-odds ratio for fixed- and random-effects models, respectively; σ is the standard deviation between studies when random-effects models are applied.

The pooled odds ratio for fixed-effects models is computed as $\rho = \exp(\delta)$ whereas that for random-effects models is $\rho = \exp(\mu)$. The study-specific estimates only exist when random-effects models are used and are calculated as $\gamma_s = \exp(\delta_s)$.

The specification of the model is completed by considering the baseline case — including minimally informative prior distributions for some model parameters and for binary data, `bmeta` provides two types of prior, namely, Normal and Student t-distribution prior. Notice that for all the meta-regression models, apart from the priors specified for individual cases, we consider by default independent minimally informative prior for the vector of regression parameters of interest:

$$\beta_0, \beta_1, \ldots, \beta_J \overset{iid}{\sim} \text{Normal}(0, 0.0001) \tag{2}$$

The underlying assumption here is that the covariates generate no effects on the main study estimate. We use 0.0001 here because models in `JAGS` require users to input the *precision* rather than *variance*. Thus, indexing a Normal distribution with a scale parameter 0.0001 implies a variance of 1/0.0001.

For ***fixed-effects meta-analysis with normal prior*** and ***fixed-effects meta-regression with normal prior***, the priors are specified as below:

$$\alpha_s \sim \text{Normal}(0, 0.0001) \tag{3}$$

$$\delta \sim \text{Normal}(0, 0.0001) \tag{4}$$

For ***fixed-effects meta-analysis with t-distribution prior*** and ***fixed-effects meta-regression with t-distribution prior***, while others remain the same, the prior for δ is replaced by:

$$\begin{aligned} \delta &\sim \text{t}(0, 0.5, v) \\ v &\sim \text{Uniform}(0, 8) \end{aligned}$$

The Student t-distribution has heavier tails and is therefore more robust to outliers.

For ***random-effects meta-analysis with normal prior*** and ***random-effects meta-regression with normal prior***, we include the same prior for α_s according to (3) but instead of assigning prior to δ, we give vague priors to μ and σ:

$$\begin{aligned} \mu &\sim \text{Normal}(0, 0.0001) \\ \sigma &\sim \text{Uniform}(0, 5) \end{aligned}$$

Similarly for ***random-effects meta-analysis with t-distribution prior*** and ***random-effects meta-regression with t-distribution prior***, the prior for σ still applies whereas that for μ is modified as:

$$\begin{aligned}
\mu &\sim \text{t}(0, 0.5, v) \\
v &\sim \text{Uniform}(0, 8)
\end{aligned}$$

1.1.2 Independence model

The independent model (without pooling effects) using Bayesian methods is developed to examine results for individual studies separately. While the Binomial distribution is still used to model events in the two arms according to (1), a new parameter α_{1s} is introduced to model the probability of contracting a disease for cases.

$$\begin{aligned}
\text{logit}(\pi_{0s}) &= \alpha_{0s} \\
\text{logit}(\pi_{1s}) &= \alpha_{1s} \\
\log(\text{OR}) &= \alpha_{1s} - \alpha_{0s} \\
\text{OR} &= \exp(\alpha_{1s} - \alpha_{0s}) \\
\alpha_{0s}, \alpha_{1s} &\overset{iid}{\sim} \text{Normal}(0, 1.45)
\end{aligned}$$

The prior for α_{0s}, α_{1s} is different from the assumptions made for the previous pooling models as for most observational studies or trials, the odds ratios are very unlikely to go beyond 10. For example, assume α_{1s} and α_{0s} follow the same distribution, Normal(0,σ^2), then log(OR)=$\alpha_{1s} - \alpha_{0s} \sim \text{Normal}(0, 2\sigma^2)$. Suppose the assumption that the odds ratio is unlikely to go beyond 10 holds, we compute σ^2 by

$$\begin{aligned}
0 + 1.96\sqrt{2\sigma^2} &= \log(10) \\
\sigma^2 &= 0.69
\end{aligned}$$

After conversion, the precision (inverse of 0.69) equals to 1.45 here. Notice that there is no link function between π_{0s} and π_{1s}, consequently, the pooling effect does not exist in this model.

1.1.3 Examples

In this section, we use a dataset ‘PE’ as an example to specify how our models for binary data work. We aim at investigating the relative risk of developing pre-eclampsia (PE) by comparing treatment and control group of 9 diuretics trials. Notice that the dataset contains an additional covariate ‘Region’ (dummy variable) to indicate whether the study was conducted in Europe (Region=1) or US (Region=0). Suppose ***random-effects meta-analysis with normal prior*** and ***random-effects meta-regression with normal prior*** are selected as our modelling approach for this dataset.

The data structure of PE in `csv` file is shown below and it contains the minimum information that users need to input:

```
> data = read.csv("PE.csv")
> data

  Study Year  y1   n1  y0   n0 Region
1     1 2003  14  131  14  136      1
2     2 2005  21  385  17  134      0
```

```
3      3 2006  14   57  24   48      0
4      4 2007   6   38  18   40      1
5      5 2010  12 1011  35  760      0
6      6 2011 138 1370 175 1336      1
7      7 2012  15  506  20  524      1
8      8 2013   6  108   2  103      0
9      9 2014  65  153  40  102      1
```

Notice that `y1`, `n1`, `y0` and `n0` correspond to events in case arm, total number of cases, events in control arms and total number of controls, respectively.

For now, we do not consider the covariate effect and therefore use ***random-effects meta-analysis with normal prior*** for modelling. Below is an illustration of coding in `R` to format the data input:

```
> data.list <- list(y0=data$y0,y1=data$y1,n0=data$n0,n1=data$n1)
```

We then call `bmeta` and specify the model to generate output:

```
> x <- bmeta(data=data.list,outcome="bin",model="std.norm",type="ran")
> x
```

Below is the summary printed by `bmeta`:

```
Inference for Bugs model at "model.txt", fit using jags,
 2 chains, each with 10000 iterations (first 5000 discarded),
 n.thin = 5 n.sims = 2000 iterations saved
          mu.vect sd.vect   2.5%   97.5%  Rhat n.eff
alpha[1]   -2.092   0.256 -2.622  -1.621 1.003   510
alpha[2]   -2.004   0.255 -2.504  -1.524 1.004   400
alpha[3]   -0.096   0.280 -0.646   0.448 1.001  2000
alpha[4]   -0.348   0.309 -0.956   0.277 1.001  2000
alpha[5]   -3.095   0.170 -3.441  -2.775 1.004   440
alpha[6]   -1.888   0.078 -2.041  -1.741 1.005   350
alpha[7]   -3.225   0.226 -3.706  -2.796 1.001  2000
alpha[8]   -3.456   0.507 -4.586  -2.594 1.014   110
alpha[9]   -0.382   0.203 -0.786  -0.016 1.001  2000
delta[1]   -0.122   0.349 -0.797   0.591 1.004   400
delta[2]   -0.820   0.324 -1.462  -0.202 1.008   200
delta[3]   -0.955   0.383 -1.724  -0.237 1.006   290
delta[4]   -1.129   0.465 -2.105  -0.279 1.001  2000
delta[5]   -1.227   0.319 -1.848  -0.610 1.004   750
delta[6]   -0.317   0.119 -0.555  -0.089 1.010   160
delta[7]   -0.341   0.326 -0.985   0.300 1.005   470
delta[8]    0.291   0.637 -0.801   1.736 1.001  2000
delta[9]    0.043   0.255 -0.425   0.563 1.001  2000
gamma[1]    0.941   0.346  0.451   1.806 1.004   400
gamma[2]    0.464   0.154  0.232   0.817 1.008   200
gamma[3]    0.414   0.162  0.178   0.789 1.006   290
gamma[4]    0.358   0.162  0.122   0.757 1.001  2000
gamma[5]    0.309   0.102  0.158   0.543 1.004   750
gamma[6]    0.733   0.088  0.574   0.915 1.010   160
gamma[7]    0.750   0.249  0.373   1.350 1.005   470
gamma[8]    1.679   1.410  0.449   5.676 1.001  2000
gamma[9]    1.079   0.285  0.654   1.756 1.001  2000
mu         -0.505   0.315 -1.133   0.128 1.004   490
rho         0.634   0.216  0.322   1.137 1.004   490
tau.delta   3.162   4.130  0.419  12.238 1.003   550
deviance  102.424   6.206 92.344 116.729 1.001  2000

For each parameter, n.eff is a crude measure of effective
sample size,
and Rhat is the potential scale reduction factor (at
convergence, Rhat=1).
```

```
DIC info (using the rule, pD = var(deviance)/2)
pD = 19.3 and DIC = 121.7
DIC is an estimate of expected predictive error (lower
deviance is better).
```

All parameters in this table exactly correspond to the model parameters introduced in the previous section. The parameter `gamma[]` represents the study-specific odds ratio and the pooled odds ratio, `rho`, is 0.634 with a 95% CrI (0.322, 1.137). The between study variance is the inverse of `tau.delta`, which equals to 0.316.

The object `x` automatically includes an element `mod0`, in which the results from the 'null' models are also stored. This can be accessed using the command

```
> x$mod0
```

Now we consider the effects of the covariate 'Region' and perform ***random-effects meta-regression with normal prior***. The covariate 'Region' is a dummy variable, and 'US' is considered to be the 'baseline' category (Region=0) whereas 'EU' is represented by 'Region=1'. It should be noted that for categorical variables that have more than 2 categories (i.e. n categories), a reference group needs to be selected and dummies for each of the **rest** $n-1$ categories should be created. The matrix 'X' that includes information on covariates can be specified using the `cbind` command in `R`, i.e. `X=cbind(data$cov1, data$cov2)`.

Similar as before, we format the data input and specify the model to be used:

```
> data.list <- list(y0=data$y0,y1=data$y1,n0=data$n0,n1=data$n1,X=cbind(
    data$Region))
> x <- bmeta(data=data.list,outcome="bin",model="reg.norm",type="ran")
> x
```

Below is the output produced by `bmeta`, which is largely the same as the previous table. We can see that `rho` slightly decreases with a narrower 95% CrI and the between study variance decreases to 0.248. The underlying assumption for the regression models is that there is no additional effects of exposure on the outcome of interest and the prior for coefficients of covariates in the regression is vague with large variability, which may have some influence on the results. However, if more information on effects of ethnicity is available, i.e. the probability of developing PE is 5% higher for women in the US, estimates with higher accuracy are expected by replacing more specific prior for regression coefficients, i.e. $\beta_0 \sim \text{Normal}(-2.94, 0.01)$. Notice that the effective sample size (`n.eff`) for certain model parameters are particularly low compared with the actual number of iterations stored for inference, which may indicate potential problem for model convergence. Therefore, we may want to check model convergence using diagnostic plots in `bmeta` and this is discussed in detail in Section 2.

```
Inference for Bugs model at "model.txt", fit using jags,
 2 chains, each with 10000 iterations (first 5000 discarded),
 n.thin = 5 n.sims = 2000 iterations saved
          mu.vect sd.vect    2.5%   97.5%  Rhat n.eff
alpha[1]   -1.992  24.462 -48.857  47.494 1.001  2000
alpha[2]   -2.161  30.575 -64.628  56.345 1.001  2000
alpha[3]   -0.244  30.579 -62.231  58.416 1.001  2000
alpha[4]   -0.244  24.476 -47.506  49.622 1.001  2000
alpha[5]   -3.259  30.584 -65.119  55.519 1.001  2000
alpha[6]   -1.773  24.465 -48.818  47.928 1.001  2000
alpha[7]   -3.095  24.462 -50.188  46.389 1.001  2000
alpha[8]   -3.509  30.575 -65.122  55.610 1.001  2000
alpha[9]   -0.244  24.472 -47.180  49.428 1.001  2000
delta[1]   -0.139   0.349  -0.796   0.578 1.011   150
delta[2]   -0.817   0.306  -1.415  -0.230 1.015   180
```

```
delta[3]    -0.938   0.361  -1.682  -0.278 1.004   510
delta[4]    -1.100   0.449  -2.055  -0.321 1.018    97
delta[5]    -1.161   0.305  -1.770  -0.593 1.111    19
delta[6]    -0.312   0.123  -0.550  -0.069 1.001  2000
delta[7]    -0.368   0.317  -0.977   0.239 1.001  2000
delta[8]     0.097   0.587  -0.868   1.444 1.059    34
delta[9]     0.012   0.253  -0.472   0.517 1.009   170
gamma[1]     0.926   0.345   0.451   1.782 1.011   150
gamma[2]     0.463   0.144   0.243   0.794 1.015   180
gamma[3]     0.417   0.149   0.186   0.757 1.004   510
gamma[4]     0.366   0.158   0.128   0.726 1.018    97
gamma[5]     0.328   0.099   0.170   0.553 1.111    19
gamma[6]     0.737   0.090   0.577   0.933 1.001  2000
gamma[7]     0.727   0.232   0.376   1.270 1.001  2000
gamma[8]     1.341   1.084   0.420   4.236 1.059    34
gamma[9]     1.045   0.271   0.624   1.677 1.009   170
mu          -0.528   0.286  -1.098   0.034 1.007  1700
rho          0.615   0.189   0.334   1.034 1.007  1700
tau.delta    4.030   8.892   0.500  15.752 1.074    30
deviance   103.264   6.452  92.761 117.361 1.018    92

For each parameter, n.eff is a crude measure of effective
sample size,and Rhat is the potential scale reduction
factor (at convergence, Rhat=1).

DIC info (using the rule, pD = var(deviance)/2)
pD = 20.6 and DIC = 123.9
DIC is an estimate of expected predictive error (lower
deviance is better).
```

1.2 Models for continuous data

Apart from 2 'null' models, there are 8 models in bmeta for continuous data, with 4 models for meta-analysis and another 4 for meta-regression. bmeta produces a pooled estimate (and 95% Credible Interval, CrI) in the form of mean difference. We assume that for continuous data, there are two types of studies: the first type of studies report detailed information (mean and standard deviation/error) of both cases and controls while for the second type of studies, only limited amount of information (mean difference and variance) can be retrieved. Different models are designed for these two types of studies.

1.2.1 Modelling approach for studies reporting two arms

Suppose we include a set of $s = 1, 2, ..., S$ independent studies and let y_{sk}, se_{sk} denote the observed mean difference before and after experiment and standard error in the kth arm in study s and we model this mean difference by a Normal distribution:

$$y_{sk} \sim \text{Normal}(\alpha_{sk}, prec_{sk}) \tag{5}$$

The precision required by JAGS is computed as follows:

$$prec_{sk} = 1/(se_{sk})^2$$

where the standard error of a sample (size n) is the sample standard deviation divided by $\sqrt{n}$.

The link function between case and control arm is presented below:

$$\alpha_{s1} = \alpha_{s0} + \delta_s \tag{6}$$

where α_{s1}, α_{s0} represent mean difference before and after experiment for case and control arm, respectively.

However, when covariates are observed (cases where meta-regression needs to be used), the parameter α_{s0} is modelled by including covariate information and the link function (equation 6) still holds:

$$\alpha_{s0} = \gamma_{s0} + \boldsymbol{\beta X} + \delta_s$$

where $\boldsymbol{X}$ (i.e. $X_0, X_1, X_2, \ldots, X_J$) and $\boldsymbol{\beta}$ (i.e. $\beta_0, \beta_1, \beta_2, \ldots, \beta_J$) are defined exactly the same as introduced before in the binary data section.

Similarly, different assumptions for fixed- and random-effects models are applied here:

$$\begin{aligned} \delta_s &= \delta && \text{Fixed-effects model} \\ \delta_s &\sim \text{Normal}(\mu, \sigma^2) && \text{Random-effects models} \end{aligned}$$

where δ and μ are the pooled mean difference between case and control arm for fixed- and random-effects models, respectively; σ is the standard deviation between studies when random-effects models are applied. The study-specific mean difference for random-effects models is represented by δ_s.

The model is completed by giving priors to some model parameters and again, for all the meta-regression models, vague priors for the vector of regression parameters of interest, $\beta_0, \beta_1, \beta_2, \ldots, \beta_J$, are assigned as (2).

For ***fixed-effects meta-analysis***, the priors are specified as below:

$$\begin{aligned} \alpha_{s0} &\sim \text{Normal}(0, 0.0001) && (7) \\ \delta &\sim \text{Normal}(0, 0.0001) && (8) \end{aligned}$$

While the prior for δ is the same, we now need to include prior for γ_{s0} instead for ***fixed-effects meta-regression***:

$$\gamma_{s0} \sim \text{Normal}(0, 0.0001) \qquad (9)$$

For ***random-effects meta-analysis***, α_{s0} is still modelled according to (7), however, as we allow a distribution of true effect, vague priors for μ and σ need to be specified:

$$\begin{aligned} \mu &\sim \text{Normal}(0, 0.0001) \\ \sigma &\sim \text{Uniform}(0, 10) \end{aligned}$$

Users would expect that the only difference between ***random-effects meta-regression*** and the above regression model is that a vague prior for γ_{s0} is assigned rather than for α_{s0}, which can be referred to (9). The assumptions for the above models are that there is no exposure/intervention effect for the case arm and we do not observe any change in mean for the control group compared with the baseline.

1.2.2 Modelling approach for studies reporting mean difference and variance

In this section, we discuss models for studies that only report mean difference and variance. Suppose there are still $s = 1, 2, ..S$ studies included. Let y_s, $prec_s$ denote the observed mean difference between case and control arm and precision (inverse of variance) for study s.

Users need to ensure that the variance used here is the sum of variances for case and control group and do not confuse with the concept of the pooled sample variance S^2_{pooled} because only with the assumption of homogeneity of variance (i.e. σ_0=σ_1=σ) in the two arms, the variance σ_s^2 for study s can be computed as:

$$\sigma_s^2 \;=\; \frac{n_0+n_1}{n_0 n_1} S_{pooled}^2$$

where n_0 and n_1 are number of individuals in control and case group.

A Normal distribution is still used to model the observed mean difference:

$$y_s \;\sim\; \text{Normal}(\delta_s, prec_s)$$

where δ_s represents the 'true' mean difference between case and control group. Thus,

$$\begin{aligned} \delta_s &= \mu && \text{Fixed-effects model} \\ \delta_s &\sim \text{Normal}(\mu, \sigma^2) && \text{Random-effects models} \end{aligned}$$

where μ is the pooled mean difference for both types of models, σ is the between-study standard deviation and only exists when a random-effects model is used.

However, in cases where covariates are observed:

$$\begin{aligned} \delta_s &= \alpha_s + \boldsymbol{\beta X} && \\ \alpha_s &= \alpha && \text{Fixed-effects model} \\ \alpha_s &\sim \text{Normal}(\mu, \sigma^2) && \text{Random-effects models} \end{aligned}$$

where α_s is the study-specific 'true' mean difference and is equal to α for fixed-effects models as all the studies are assumed to measure the same effect; μ and σ are defined exactly the same as above in the meta-analysis; $\boldsymbol{X}$ is the covariate matrix and $\boldsymbol{\beta}$ is the vector of regression parameters of interest. We include vague priors for the vector of regression parameters of interest, $\beta_0, \beta_1, \beta_2, \ldots, \beta_J$, according to (2) for all the meta-regression models.

For ***fixed-effects meta-analysis***, the prior is specified as below:

$$\mu \;\sim\; \text{Normal}(0, 0.0001)$$

Whereas for ***fixed-effects meta-regression***, the prior is replaced by:

$$\alpha \;\sim\; \text{Normal}(0, 0.0001)$$

The two random-effects models require the same priors:

$$\begin{aligned} \mu &\sim \text{Normal}(0, 0.0001) \\ \sigma &\sim \text{Uniform}(0, 10) \end{aligned}$$

The assumption for the models introduced here is that there is no intervention effect on the case arm comparing with the control arm.

1.2.3 Independence model

The independent model (without pooling effects) using Bayesian methods is developed to examine results for individual studies separately. For continuous data, there are two independent models due to the differences in the reporting format of various studies. For studies reporting information on both arms separately, the observed mean and standard errors in both arms are still modelled by

(5). However, a new parameter $diff_s$ is introduced to model the mean difference between case and control group.

$$\begin{aligned}\mu_{0s} &\sim \text{Normal}(0, 0.0001)\\ \mu_{1s} &\sim \text{Normal}(0, 0.0001)\\ diff_s &= \mu_{1s} - \mu_{0s}\end{aligned}$$

The prior for μ_{0s}, μ_{1s} is fairly vague with zero mean and large variance. As no link function exists between μ_{0s} and μ_{1s}, there is no pooling effect in this model.

For studies that report mean difference and variance only, the 'null' model is slightly different:

$$\begin{aligned}y_s &\sim \text{Normal}(\mu_s, 1/prec_s)\\ \mu_s &\sim \text{Normal}(0, 0.0001)\end{aligned}$$

where y_s represents the observed mean difference and μ_s is introduced to model the 'true' mean difference between case and control group for each study. Again, we use vague prior for this model parameter.

1.2.4 Examples

In this section, we use a dataset 'IVF' as an example to specify how our models for continuous data work. We aim at comparing the effects of two different treatments (luteal long agonist and GnRH antagonists) on ovarian stimulation in women with polycystic ovary syndrome (PCOS). Suppose that ***random-effects meta-analysis for studies reporting information of two arms*** and ***random-effects meta-analysis for studies reporting mean difference and variance only*** are selected as our modelling framework for this dataset. The data structure of IVF in `csv` file is shown below:

```
    study year    y1       se1    y0      se0
1   Hwang 2004 17.60 1.2043325 16.30 1.280000
2 Ashrafi 2005  6.17 0.9466112 10.96 1.820734
3  Ensieh 2010 10.88 1.0673498 13.37 1.371455
4    Agha 2010  8.56 0.5784635 11.95 1.103335
5     Kim 2012 14.20 0.5714910 14.70 0.566022
6  Bulent 2012 17.11 0.7488067 18.11 1.037960
```

Columns 4-7 are the minimum information users need to provide for ***random-effects meta-analysis for studies reporting information of two arms*** to work properly, including mean and standard errors of both arms. In cases where standard deviations rather than standard errors are observed, data transformation is necessary and standard error is calculated as standard deviation divided by the square root of the number of individuals in that arm. This computation procedure can be implemented by either `Excel` or `R`.

We then format the data and specify the model for `bmeta` to generate output:

```
> data = read.csv("Data-ctns.csv")
> data.list <- list(y0=data$y0,y1=data$y1,se0=data$se0,se1=data$se1)
> x <- bmeta(data=data.list,outcome="ctns",model="std.ta",type="ran")
> x
```

The output printed by `bmeta` is shown below:

```
Inference for Bugs model at "model.txt", fit using jags,
 2 chains, each with 10000 iterations (first 5000 discarded),
 n.thin = 5 n.sims = 2000 iterations saved
          mu.vect  sd.vect   2.5%  97.5%  Rhat n.eff
```

```
alpha0[1]  17.122     1.179 14.690 19.329 1.001  2000
alpha0[2]   9.476     1.527  6.853 12.790 1.002   910
alpha0[3]  13.028     1.140 10.688 15.244 1.004   490
alpha0[4]  11.331     1.048  9.450 13.519 1.001  2000
alpha0[5]  14.858     0.544 13.792 15.909 1.001  2000
alpha0[6]  18.297     0.891 16.576 20.041 1.002   880
alpha1[1]  16.828     1.113 14.644 19.021 1.001  1600
alpha1[2]   6.598     0.909  4.779  8.309 1.001  2000
alpha1[3]  11.048     0.965  9.123 12.832 1.001  2000
alpha1[4]   8.728     0.567  7.633  9.854 1.005   350
alpha1[5]  14.027     0.547 12.981 15.140 1.001  2000
alpha1[6]  17.032     0.706 15.688 18.364 1.001  2000
delta[1]   -0.295     1.453 -2.673  2.882 1.001  1800
delta[2]   -2.878     1.657 -6.644 -0.220 1.002   910
delta[3]   -1.981     1.259 -4.643  0.366 1.005   450
delta[4]   -2.604     1.161 -5.050 -0.549 1.003   730
delta[5]   -0.831     0.761 -2.280  0.721 1.001  2000
delta[6]   -1.265     0.994 -3.180  0.748 1.002  1700
mu         -1.615     1.125 -4.053  0.723 1.001  2000
tau.delta 320.324 5367.733  0.028 62.911 1.031   110
deviance   33.663     4.625 25.921 43.572 1.004   410

For each parameter, n.eff is a crude measure of effective
sample size, and Rhat is the potential scale reduction
factor (at convergence, Rhat=1).
DIC info (using the rule, pD = var(deviance)/2)
pD = 10.7 and DIC = 44.3
DIC is an estimate of expected predictive error (lower
deviance is better).
```

All parameters in this table exactly correspond to the model parameters introduced in the previous section. The parameters `alpha0[]` and `alpha1[]` represent the study-specific mean for control and case arm, respectively and the pooled mean difference, `mu`, is -1.615 with a 95% CrI (-4.053, 0.723). The between study variance is the inverse of `tau.delta`, which equals to 0.003 and is very small.

Using the previous example again but this time, we implement the other type of model—***random-effects meta-analysis for studies reporting mean difference and variance only***. Suppose now only mean difference between two arms and variance are retrieved (we need to convert variance into precision first, which is computed as the inverse of variance):

```
     study year     y       var      prec
1    Hwang 2004  1.30 3.0888167 0.3237486
2  Ashrafi 2005 -4.79 4.2111455 0.2374651
3   Ensieh 2010 -2.49 3.0201244 0.3311122
4     Agha 2010 -3.39 1.5519691 0.6443427
5      Kim 2012 -0.50 0.6469829 1.5456359
6   Bulent 2012 -1.00 1.6380733 0.6104733
```

In the next step, we format the data and call `bmeta` to perform analysis:

```
> data = read.csv("Data-ctns.csv")
> data.list <- list(y=data$y,prec=data$prec)
> x <- bmeta(data=data.list,outcome="ctns",model="std.mv",type="ran")
> x
```

Below is the summary table produced by `bmeta`:

```
Inference for Bugs model at "model.txt", fit using jags,
 2 chains, each with 10000 iterations (first 5000 discarded),
 n.thin = 5 n.sims = 2000 iterations saved
          mu.vect sd.vect   2.5%   97.5%  Rhat n.eff
```

```
delta[1]  -0.335   1.435 -2.728   2.889 1.002  2000
delta[2]  -2.795   1.654 -6.648  -0.307 1.004   450
delta[3]  -1.964   1.279 -4.834   0.424 1.001  2000
delta[4]  -2.520   1.120 -4.733  -0.568 1.007   240
delta[5]  -0.846   0.766 -2.321   0.708 1.003   570
delta[6]  -1.248   1.012 -3.186   0.943 1.001  2000
mu        -1.608   1.095 -3.881   0.489 1.001  2000
tau.delta 50.187 852.105  0.034 165.473 1.003   740
deviance  22.003   3.268 16.603  29.019 1.008   200

For each parameter, n.eff is a crude measure of effective
sample size, and Rhat is the potential scale reduction
factor (at convergence, Rhat=1).
DIC info (using the rule, pD = var(deviance)/2)
pD = 5.3 and DIC = 27.3
DIC is an estimate of expected predictive error (lower
deviance is better).
```

Apparently, the two random-effects model generate similar results (-1.608 in the current table and -1.615 in the previous one), which is expected as the same dataset is analysed. Compared with the previous model, the current one require fewer model parameters, therefore, the Deviance Information Criteria (DIC) of the current model is much smaller.

1.3 Models for count data

Apart from a 'null' model, there are 6 models in `bmeta` for count data, with 3 models for meta-analysis and another 3 for meta-regression. It should be noted that `bmeta` produces a pooled estimate (and 95% Credible Interval, CrI) in the form of incidence rate ratio (IRR).

1.3.1 Modelling approach for meta-analysis

The model structure for count data is different from that for binary and continuous data as count data are generally fitted by a Poisson distribution. Suppose there are $s = 1, 2, ..., S$ independent studies included and the total number of events in the follow-up period (measured by person-time) is observed for both the case and control arm. Let y_{sk}, p_{sk} denote the number of events and total follow-up person time, respectively, for kth arm in sth study. A Poisson distribution is used to model the event rate:

$$y_{sk} \sim \text{Poisson}(\lambda_{sk}) \tag{10}$$

where λ_{sk} is the expectation of y_{sk}.

A log transformation for λ_{sk} is applied here, which not only reflects the multiplicative effects of event rate and follow-up person-time, but also make the transformed variables more closely approximate to the assumption for linear regression model (i.e. normally distributed residuals):

$$\log(\lambda_{sk}) = \begin{cases} \xi_s + \log(p_s), & \text{if } k\text{=0 (control arm)} \\ \xi_s + \log(p_s) + \delta_s, & \text{if } k\text{=1 (case arm)} \end{cases}$$

where ξ_s is the event rate for the control arm in sth study and δ_s is the study-specific intervention effect. Notice that these two parameters are all on a log scale.

If covariate effects exist:

$$\xi_{s0} = \theta_s + \boldsymbol{\beta X}$$

where ξ_{s0} is the event rate for controls including covariate effects whereas θ_s is considered to be the 'true' baseline event rate for control arm (both on log scale); $\boldsymbol{\beta}$ (vector of regression parameters of interest) and $\boldsymbol{X}$ (covariate matrix) are defined exactly the same as in other meta-regression models.

As model assumptions for fixed- and random-effects models differ, the following equations hold:

$$\begin{array}{rcll} \delta_s & = & \delta & \text{Fixed-effects model} \\ \delta_s & \sim & \text{Normal}(\mu, \sigma^2) & \text{Random-effects models} \end{array}$$

where δ and μ are the pooled log incidence rate ratio (IRR) for fixed- and random-effects models, respectively. The pooled IRR on natural scale is then equal to $\exp(\delta)$ for fixed-effects models and $\exp(\mu)$ for random-effects models. The study-specific IRRs only exist when random-effects models are applied and is computed as $\exp(\delta_s)$ (δ_s on log scale).

We include minimally informative priors for regression parameters of interest ($\beta_1, \beta_2, ...\beta_J$) applied in all other meta-regression models, which can be referred to (2). It should be noted that `bmeta` provides two types of prior for modelling count data when random-effects models are selected, namely, Uniform and half-Cauchy prior.

All the fixed-effects models (i.e. ***fixed-effects meta-analysis***, ***fixed-effects meta-regression***) designed for count data include the same prior for the pooled estimate δ:

$$\delta \sim \text{Normal}(0, 0.0001)$$

However, for ***fixed-effects meta-analysis***, we include vague prior for ξ_{s0}:

$$\xi_{s0} \sim \text{Uniform}(-5, 5) \tag{11}$$

Whereas for ***fixed-effects meta-regression***, the same prior needs to be specified for θ_s instead:

$$\theta_s \sim \text{Uniform}(-5, 5) \tag{12}$$

Notice that as the log scale is used, this Uniform prior actually indicates that the event rate for the control group ranges from e^{-5} to e^5 after transforming back to natural scale, which is very vague. The assumptions for the priors introduced above are that we do not expect any effect of exposure on the case arm and for control arm, the event rate can range from close to zero to infinity.

It is also worth commenting that for all the random-effects models, a common vague prior for μ is given:

$$\mu \sim \text{Normal}(0, 0.0001)$$

Moreover, for ***random-effects meta-analysis with uniform prior*** and ***random-effects meta-analysis with half-Cauchy prior***, the prior for ξ_{s0} is still based on (11). However, for the first model, we include a Uniform prior for the between-study standard deviation σ:

$$\sigma \sim \text{Uniform}(0, 10) \tag{13}$$

whereas for the latter model, a half-Cauchy prior is given for the same parameter:

$$\sigma = \frac{|Z_\sigma|}{\sqrt{\varepsilon_\sigma}} \tag{14}$$

$$Z_\sigma \sim \text{Normal}(0, \sigma^2_{Z_\sigma}) \tag{15}$$

$$\varepsilon_\sigma \sim \text{Gamma}(0.5, 0.5) \tag{16}$$

$$\sigma^2_{Z_\sigma} = \frac{1}{B^2_\sigma} \tag{17}$$

$$B_\sigma \sim \text{Uniform}(0, 0.5) \tag{18}$$

We introduce the half-Cauchy prior because in the previous model, σ is assumed to follow a Uniform distribution and this potentially indicates a problem: the simulation process is likely to be largely influenced by k where k is the upper bound of a Uniform distribution, i.e. Uniform(0,k). With its heavier tail, the half-Cauchy distribution allows for outliers and accommodates small variances close to zero.

Finally, for ***random-effects meta-regression with uniform prior*** and ***random-effects meta-regression with half-Cauchy prior***, in addition to prior for the pooled estimate μ, we still include prior for θ_s according to (12). Users would expect that the only difference in priors assigned for these two models is in terms of σ, which can be referred to (13) and (14-18), respectively.

1.3.2 Independence model

The independent model (without pooling effects) using Bayesian methods is developed to examine results for individual studies separately. While the events are still modelled using a Poisson distribution based on (10), the link function between the two arms does not exist.

$$\begin{aligned}
\xi_{0s}, \xi_{1s} &\overset{iid}{\sim} \text{Uniform}(-5, 5) \\
\text{IRR}_s &= \exp(\log(\xi_{1s}) - \log(\xi_{0s})) \\
\log(\text{IRR}_s) &= \log(\xi_{1s}) - \log(\xi_{0s})
\end{aligned}$$

The incidence rate ratio IRR_s (or on log scale) is compared between the two arms.

1.3.3 Examples

In this section, we use a dataset 'COPD' as an example to specify how our models for count data work. We aim at investigating the effects of LAMS treatment (versus placebo) on the exacerbation counts of chronic obstructive pulmonary disease (COPD). Suppose that ***random-effects meta-regression with half-Cauchy prior*** is selected as our modelling framework for this dataset.

The data structure of COPD in `csv` file is shown below.

```
> data
        Study  year    y0       p0   y1       p1
1   Ambrosino  2008    26    49.52   19    49.52
2     Bateman 2010a  1247   653.00  623   670.00
3     Bateman 2010b  1568  1802.81 1235  1789.89
4    Brusasco  2003   298   200.00  215   201.00
5    Casaburi  2002   352   371.00  418   550.00
6        Chan  2007   259   281.52  494   561.18
7      Cooper  2010   194   415.35  208   441.19
8   Donaldson  2012   111    52.00   47    48.00
9     Donohue  2010   150   209.00  110   207.50
10     Dusser  2006   862   510.00  550   500.00
11    Freeman  2007    44    37.39   23    42.00
12 Niewoehner  2005   480   457.50  388   457.00
13     Powrie  2007   134    73.00   60    69.00
14    Tashkin  2008 10220 12024.00 8719 11944.00
15     Tonnel  2008   395   216.00  209   199.50
```

Columns 2-3 specify the study characteristics and Columns 4-7 present the observations from each study, which need to be used to populate the model. Notice that the total follow-up person-time for an arm is simply calculated as the total number of person times the total follow-up period in that arm.

We then format the data and call `bmeta` to perform analysis:

```
> data= read.csv(”Data-count.csv”)
> data.list <- list(y0=data$y0,y1=data$y1,p0=data$p0,p1=data$p1)
> x <- bmeta(data=data.list,outcome=”count”,model=”std.hc”,type=”ran”)
> x
```

Below is the output table obtained:

```
Inference for Bugs model at ”model.txt”, fit using jags,
 2 chains, each with 10000 iterations (first 5000 discarded),
 n.thin = 5 n.sims = 2000 iterations saved
              mu.vect sd.vect      2.5%     97.5%  Rhat n.eff
IRR             0.695   0.049     0.601     0.795 1.002  2000
delta[1]       -0.348   0.192    -0.743     0.033 1.001  2000
delta[2]       -0.706   0.048    -0.804    -0.611 1.001  2000
delta[3]       -0.237   0.039    -0.312    -0.163 1.001  2000
delta[4]       -0.335   0.087    -0.511    -0.165 1.001  2000
delta[5]       -0.233   0.072    -0.374    -0.092 1.002  1200
delta[6]       -0.073   0.074    -0.218     0.074 1.002  1100
delta[7]       -0.049   0.095    -0.244     0.135 1.001  2000
delta[8]       -0.640   0.148    -0.943    -0.361 1.003   500
delta[9]       -0.318   0.113    -0.545    -0.097 1.001  2000
delta[10]      -0.425   0.053    -0.529    -0.323 1.002  2000
delta[11]      -0.556   0.183    -0.927    -0.218 1.001  2000
delta[12]      -0.222   0.066    -0.350    -0.093 1.002  1200
delta[13]      -0.634   0.132    -0.908    -0.382 1.002  1400
delta[14]      -0.153   0.014    -0.181    -0.125 1.001  2000
delta[15]      -0.536   0.081    -0.698    -0.381 1.001  2000
gamma[1]        0.719   0.140     0.476     1.034 1.001  2000
gamma[2]        0.494   0.024     0.448     0.543 1.001  2000
gamma[3]        0.789   0.030     0.732     0.850 1.001  2000
gamma[4]        0.718   0.062     0.600     0.848 1.001  2000
gamma[5]        0.794   0.057     0.688     0.912 1.002  1200
gamma[6]        0.933   0.069     0.804     1.077 1.002  1100
gamma[7]        0.957   0.091     0.784     1.145 1.001  2000
gamma[8]        0.533   0.079     0.389     0.697 1.003   500
gamma[9]        0.732   0.083     0.580     0.907 1.001  2000
gamma[10]       0.655   0.035     0.589     0.724 1.002  2000
gamma[11]       0.583   0.106     0.396     0.804 1.001  2000
gamma[12]       0.803   0.053     0.705     0.911 1.002  1200
gamma[13]       0.535   0.070     0.403     0.683 1.002  1400
gamma[14]       0.859   0.012     0.834     0.883 1.001  2000
gamma[15]       0.587   0.047     0.497     0.684 1.001  2000
lambda0[1]     26.365   4.414    18.590    35.747 1.001  2000
lambda0[2]   1240.876  36.175  1173.622  1311.805 1.001  2000
lambda0[3]   1571.858  39.859  1494.476  1650.800 1.001  2000
lambda0[4]    298.655  16.713   266.903   332.350 1.001  2000
lambda0[5]    354.024  18.667   319.584   393.143 1.003   700
lambda0[6]    263.992  16.012   232.795   296.331 1.002   960
lambda0[7]    199.781  13.716   173.828   227.643 1.001  2000
lambda0[8]    106.013  10.064    87.474   126.101 1.000  2000
lambda0[9]    150.702  11.896   128.050   175.792 1.002  1700
lambda0[10]   859.586  28.700   804.886   915.314 1.002  1200
lambda0[11]    40.750   5.858    30.746    53.164 1.001  2000
lambda0[12]   482.546  21.329   440.681   524.530 1.001  1500
lambda0[13]   128.853  10.992   108.531   151.444 1.001  2000
lambda0[14] 10220.067 102.120 10029.084 10425.303 1.003   670
lambda0[15]   391.724  19.431   355.894   432.160 1.001  2000
lambda1[1]     18.679   3.535    12.488    26.171 1.001  2000
lambda1[2]    628.538  24.216   580.875   676.311 1.001  2000
lambda1[3]   1231.074  35.841  1161.709  1302.374 1.001  2000
lambda1[4]    214.929  14.778   186.336   245.057 1.001  2000
lambda1[5]    415.615  20.161   377.441   455.778 1.001  2000
lambda1[6]    489.024  21.481   447.026   532.795 1.001  2000
```

```
lambda1[7]     202.135  13.958   175.735   229.942 1.001  2000
lambda1[8]      51.797   6.555    39.631    65.391 1.005   390
lambda1[9]     108.954   9.759    90.609   127.546 1.001  2000
lambda1[10]    551.270  22.724   508.664   598.171 1.001  2000
lambda1[11]     26.338   4.302    18.636    35.471 1.001  2000
lambda1[12]    386.211  19.268   350.014   425.179 1.001  2000
lambda1[13]     64.756   7.207    51.748    79.736 1.003   640
lambda1[14]   8715.651  93.364  8531.931  8899.159 1.006   330
lambda1[15]    211.949  14.103   185.401   240.948 1.001  2000
mu              -0.366   0.070    -0.508    -0.230 1.002  2000
tau.delta       19.308   8.948     6.963    41.924 1.001  2000
deviance       254.157   7.823   240.854   271.611 1.001  2000

For each parameter, n.eff is a crude measure of effective sample
size, and Rhat is the potential scale reduction factor (at
convergence, Rhat=1).
DIC info (using the rule, pD = var(deviance)/2)
pD = 30.6 and DIC = 284.8
DIC is an estimate of expected predictive error (lower deviance
is better).
```

All parameters in this table exactly correspond to the model parameters introduced in the previous section. The study-specific incidence rate ratios are represented by `gamma[]` while the pooled incidence rate ratio (`IRR`) is 0.695 with a 95% CrI (0.601,0.795).

2 Graphical summaries

In this section, we discuss the graphing functions of bmeta and show some sample coding and plots. Notice that bmeta not only produces plots that display summary from modelling in a standard format (i.e. posterior distribution plot and forest plot) but also some other diagnostic plots which examine the model fit and performance.

Table 2: Plots produced by bmeta

Plots	Function
posterior.plot	Posterior distribution plot of the summary estimate and between-study standard deviation
forest.plot	Graphical display of study-specific estimates and the pooled estimate. Need to install package 'forestplot' from R library to implement command
funnel.plot	Scatter plot to present publication bias
diag.plot	Diagnostic plot to present Gelman-Rubin statistic (Rhat) or effective sample size from MCMC simulations
traceplot.bmeta	Trace plot to examine the model convergence for each node
acf.plot	Autocorrelation plot to examine model convergence for each node

2.1 Posterior distribution plots

As all models included in bmeta employ entirely Bayesian methods, it is natural to examine the pooled estimates by looking at their posterior distributions, which is basically a plot of pooled estimate obtained from posterior distribution against density (the area represented by all the bars adds up to 1). For binary and count data, users have the option to select whether to display the summary estimate on log (scale="log") or natural scale (scale="exp"). When log scale (for binary and count data only) is used, a vertical reference line $x = 0$ is added whereas when natural scale is selected, the reference line is $x = 1$. Therefore, whether the experiment favours intervention or control is clearly exhibited in both ways. Moreover, there is a horizontal line in bold to show the 95% credible interval of the pooled estimate. Users may need to adjust the scale of $x-$axis for the plot to look more decent and the argument 'xlim' is designed with this function. If it is not specified, bmeta automatically uses a scale of $(-3, 3)$ for binary and count data and a scale of $(-5, 5)$ for continuous outcome. Additional arguments that can be used include 'xlab' and 'main', which enable users to change the label of the $x-$axis and the title of the plot, respectively.

Moreover, the posterior plot for between-study standard deviation (SD) is also provided by implementing the argument heterogeneity=T and this argument can be used only for random-effects models. Notice that for binary data, the between-study SD is on a logit scale whereas for count data, it is on a log scale. However, no matter what scale is used, it is expected that there is low heterogeneity when between-study SD is close to zero whereas the heterogeneity is high when between-study SD is close to infinity.

Suppose we use the sample dataset for binary outcome introduced previously as an example and by typing the following command after executing bmeta(. . .) (suppose random-effects meta-analysis with normal prior is used here), we get Figure 1.

```
> ### Posterior plot on log scale with a limit for x-axis set at (-2,2)
> posterior.plot(x,xlim=c(-2,2))
```

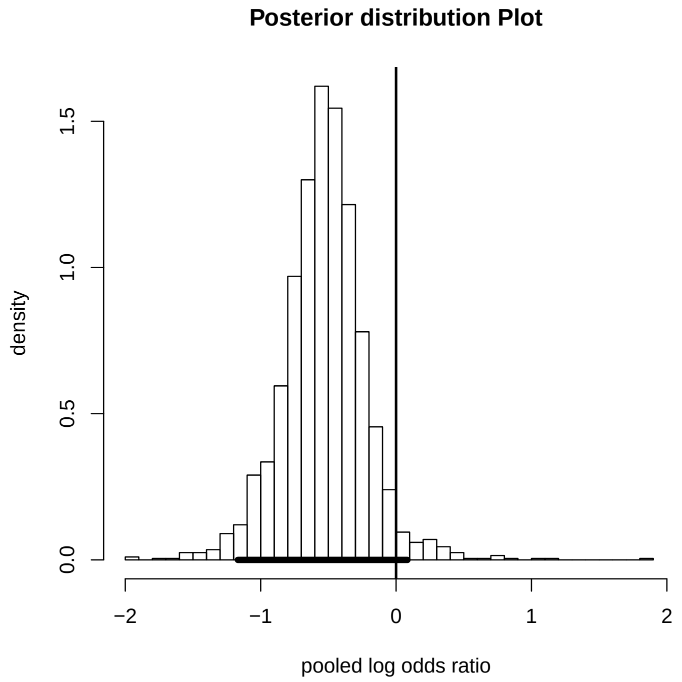

Figure 1: Posterior distribution for pooled log odds ratio

In the above plot, as the scale is not specified, the log scale (for binary and count data only) is automatically used by `bmeta`, with a vertical reference line $x = 0$ added. However, if the natural scale is preferred, then the following command needs to be executed and the correct label for the x-axis is 'pooled odds ratio' (see Figure 2).

```
> ### Posterior plot on natural scale with label and adjusted scale for x-axis
> posterior.plot(x,scale="exp",xlab="pooled odds ratio",xlim=c(0,2.5))
```

In the above two graphs, the relative likelihood for summary estimate to take on a given value sampled from the posterior distribution is exhibited, with the maximum likelihood of a certain value indicated by the highest 'bar'. It can be concluded that for this dataset, the experiment tends to favour intervention as the maximum likelihood value of the summary estimate as well as its 95% CrI (represented by the bold line) locates almost entirely to the left of the vertical reference line regardless of what scale is used.

Suppose we further want to examine the heterogeneity between studies and by programming the following code, Figure 3 is created. This plot indicates that the 95% CrI of the between-study SD is (0.3, 1.5), which is reasonably small. If the maximum likelihood of the between-study SD is very close to zero with a narrow 95% CrI, this may suggest that the heterogeneity between studies are small enough to allow us to switch to fixed-effects models instead.

```
> ### Posterior plot to examine heterogeneity between studies
> posterior.plot(x,heterogeneity=T,xlim=c(0,3),xlab="between-study standard
   deviation")
```

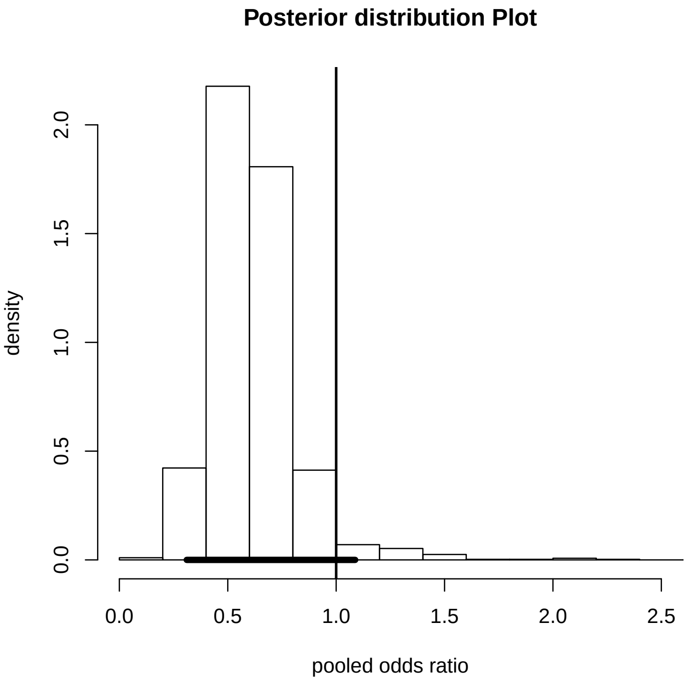

Figure 2: Posterior distribution for pooled odds ratio

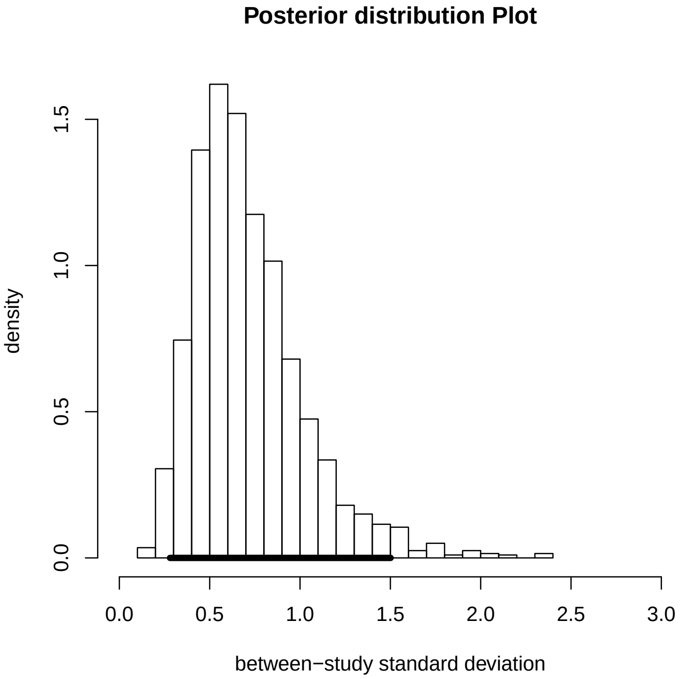

Figure 3: Posterior plot to examine heterogeneity between studies

2.2 Forest plot

Forest plot has been considered to be the standard plot for graphical illustration of output generated from meta-analysis. In a forest plot, a study-specific estimate is normally represented by either a box or a circle, which is displayed somewhere on a line that represents the 95% Credible Interval (CrI) of the estimate. The summary estimate is drawn like a diamond and a vertical reference line is added to show whether the outcome favours intervention or control. As the Bayesian fixed-effects models do not provide study-specific estimates, estimates from independence models are used and displayed on the forest plot for all fixed-effects models. It should be noted that bmeta calls the R package 'forestplot' to produce graphs, therefore, users need to ensure that the package 'forestplot' is installed (this is automatically done when installing bmeta from CRAN). The argument options for forest plot are listed in Table 3.

Table 3: Argument options for forest plot

Argument	Function
x	A bmeta object with results
title	Title of the plot
xlab	Label of the x-axis
log	Results displayed on log scale (TRUE) or natural scale (FALSE), FALSE by default
add.null	Whether to display results from 'null' model or not, only available for random-effects models, FALSE by default
study.label	Labels for each individual studies and summary estimate
clip	Lower and upper limits for clipping credible intervals to arrows
lines	Select the colour for the lines of the intervals
box	Select the colour for the mean estimates
summary	Select the colour for the summary estimate
box.symb	Select the symbol used to plot the mean. Options are "box" (default) or "circle"
lab.cex	Define the size of the text for the label. Defaults at 0.8 of normal size
xlab.cex	Define the size of the text for x-label. Defaults at 1 of the normal size
ticks.cex	Define the size of the text for the x-axis ticks. Defaults at 0.8 of the normal size
. . .	Other arguments

We use the sample dataset for count outcome introduced previously as an example and by typing the following command after executing bmeta(. . .), a basic forest plot is created (Figure 4).

```
> forest.plot(x)
```

The boxes representing study-specific estimates may be a bit too large and we can make them smaller by using the argument 'boxsize' (see Figure 5).

```
> ### Forest plot with a smaller boxsize
> forest.plot(x,boxsize=0.2)
```

However, we may further want to add study characteristics (i.e. author, year) and label the graph using a proper title. These can be achieved by executing the following codes, which produces Figure 6.

```
> ### Forest plot with a smaller boxsize, study label and title of the plot
> forest.plot(x,boxsize=0.2,study.label=c(paste0(data$study,",",data$year),"
   Summary estimate"),title="Forest plot for count data")
```

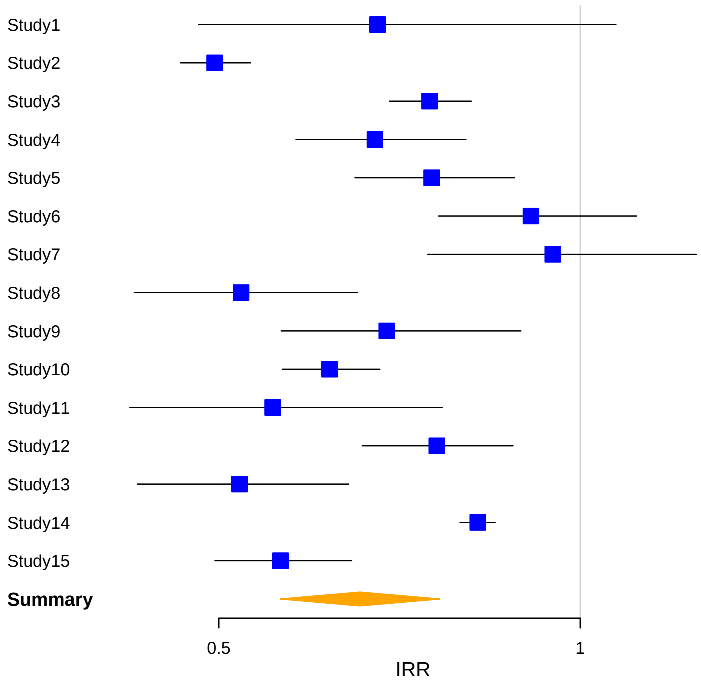

Figure 4: Forest plot for count data (pooled incidence rate ratio)

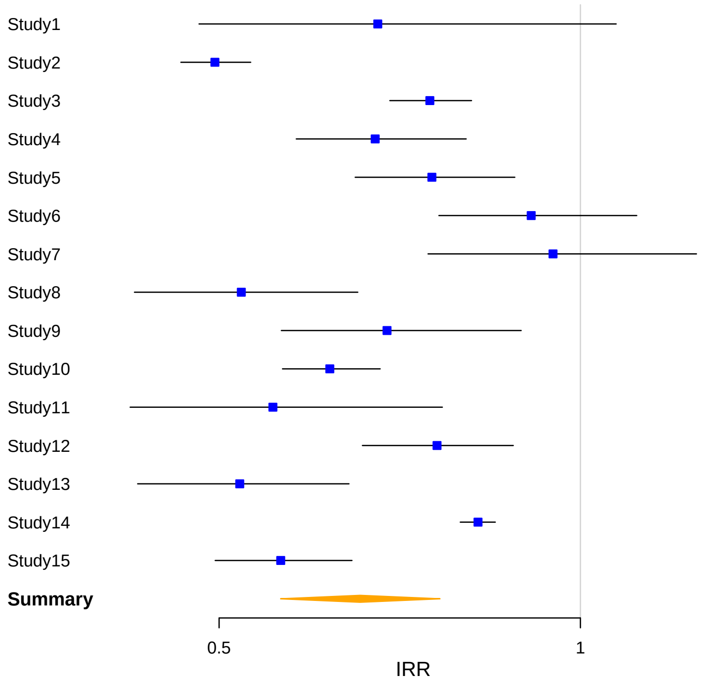

Figure 5: Forest plot for count data (pooled incidence rate ratio)

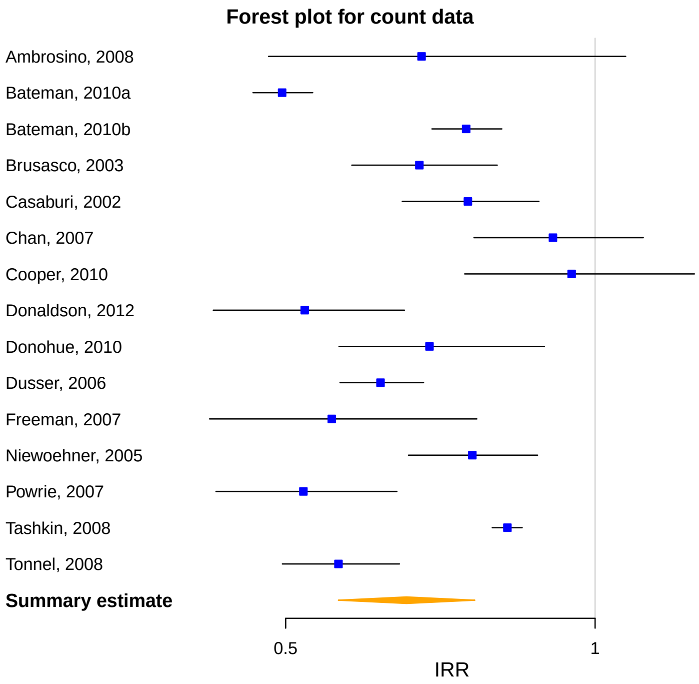

Figure 6: Forest plot for count data (pooled incidence rate ratio)

If users want results to be presented on a log scale rather than a natural scale, then use the argument ‘log=T’ and Figure 7 is what users can expect.

```
> ### Forest plot presenting results on log scale
> forest.plot(x,boxsize=0.2,study.label=c(paste0(data$study,”,”,data$year),”
   Summary estimate”),title=”Forest plot for count data”,log=T)
```

For random-effects models, bmeta also produces a two-line forest plot so that results from modelling and no-pooling effects model are displayed on the same graph. This plot can be used to examine whether the observed study-specific estimates before pooling are reasonably close to that after modelling. If there are huge differences between study-specific estimates before and after modelling, we may suspect that substantial amount of heterogeneity exits among distinct studies and thus consider if it makes sense to pool them together. By adding the argument ‘add.null=T’ to the previous command, Figure 8 is created.

```
> ### Two line forest plot presenting results from modelling and no-pooling
   effects model
> forest.plot(x,boxsize=0.2,study.label=c(paste0(data$study,”,”,data$year),”
   Summary estimate”),title=”Two line forest plot for count data”,add.null=T)
```

The distances between the two lines can be adjusted to make the graph look more decent by using the argument line.margin. We may also want the ‘lines’ and ‘boxes’ to be shifted rightwards so the main information is more centred. This can be achieved by resetting the lower and upper limits for clipping credible intervals to arrows (i.e. use the argument clip=c(. . .)). The following command produces Figure 9. This plot indicates that estimates from the no-pooling effects model and random-effects model are reasonably close for the majority of the studies included. Therefore, it seems sensible to obtain a pooled estimate from these studies.

```
> ### Two line forest plot with adjusted line.margin
```

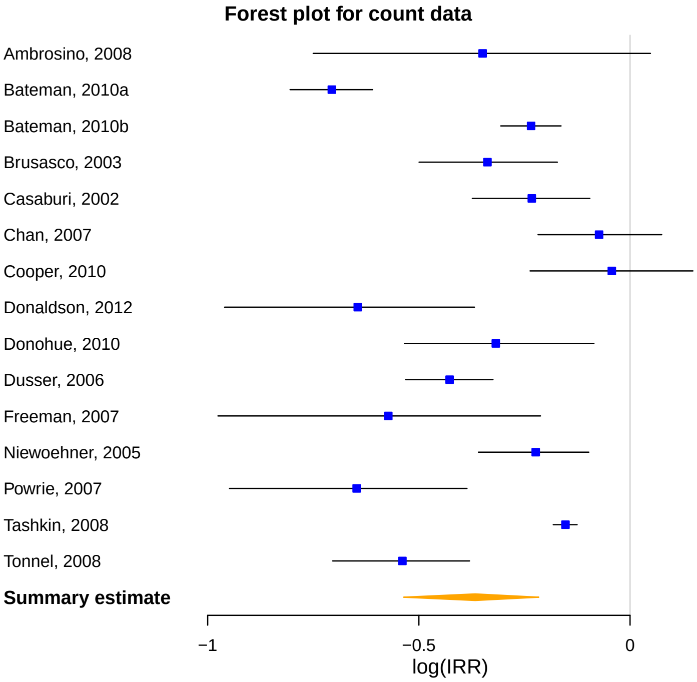

Figure 7: Forest plot for binary data (pooled log incidence rate ratio)

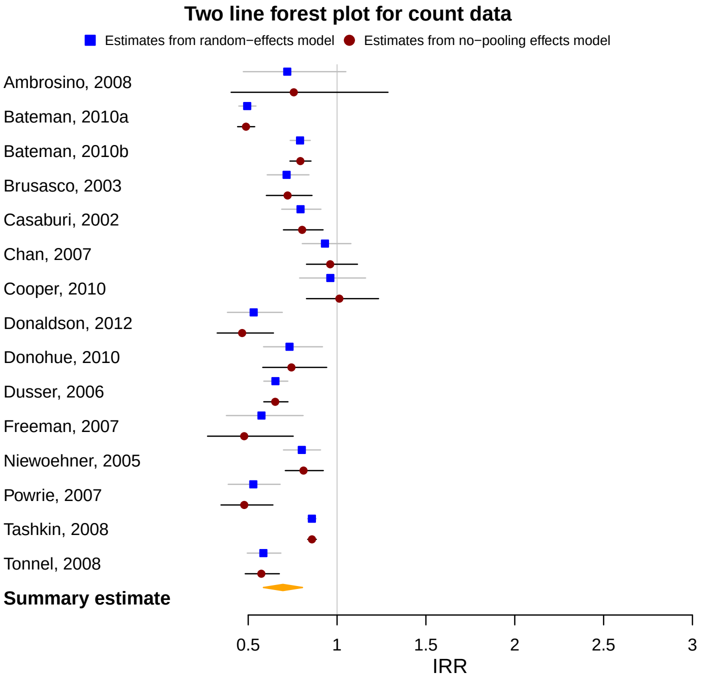

Figure 8: Two line forest plot for count data (pooled incidence rate ratio)

```
> forest.plot(x,boxsize=0.2,study.label=c(paste0(data$study,",",data$year),"
    Summary estimate"),title="Two line forest plot for count data",add.null=T,
    line.margin=0.25,clip=c(0,1.5))
```

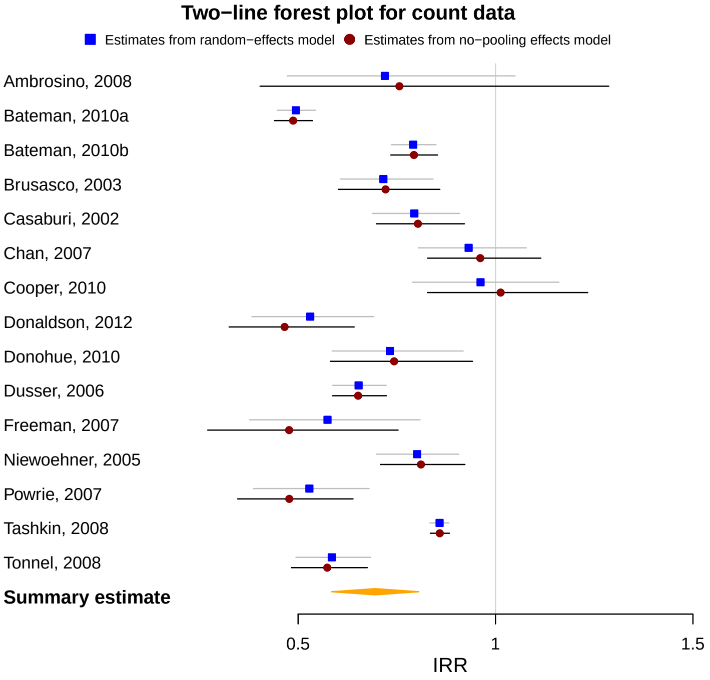

Figure 9: Two line forest plot for count data (pooled incidence rate ratio)

2.3 Funnel plot

Funnel plot is commonly used to examine publication bias and it is a scatter plot of study effect against some measure of study size. The funnel plot function in `bmeta` displays study effect on log scale against the standard error of the study effect. The effect estimates from smaller studies are often observed to scatter widely at the bottom because spontaneous estimates and larger standard errors are more likely to be expected. While the spread of effect estimates from larger studies should be narrower as results are less likely to be obtained by chance. It is suggested that when bias and between-study heterogeneity are completely absent, the scatter resembles a symmetrical funnel and the triangle area formed by connecting the centred summary estimate with its 2.5% and 97.5% quantiles on either side includes about 95% of the studies if the fixed-effects model assumption holds (i.e. all the studies estimate the same effect).

The study-specific estimates for all types of models to be displayed in a funnel plot are obtained from the independence models. As the publication bias needs to be examined based on fixed-effects model assumption, it is aware that the summary estimate is readily usable only when users select fixed-effects models. However, `bmeta` overcomes this issue by implementing the corresponding fixed-effects models at background to obtain the pooled estimate for all the random-effects models. For example, if users select random-effects meta-analysis with normal prior, then fixed-effects meta-analysis with normal prior is implemented at background. The four arguments to control the presentation of the plot are '`xlim`','`xlab`','`ylab`' and '`title`', which is to large extent self-explanatory.

By programming the following codes, a basic funnel plot is obtained (see Figure 10) based on the results from dataset for binary outcome. Notice that users can easily change the title and label for $x-$ and $y-$axis (i.e. xlab="study effect").

```
> ### Funnel plot
> funnel.plot(x)
```

We can further adjust the x-axis by using the argument xlim to make the plot look more decent (see Figure 11).

```
> ### Funnel plot
> funnel.plot(x,xlim=c(-2,1))
```

It is indicated that the scatter is approximately symmetrical with two dots lying outside the triangle, suggesting some extent of violation of the fixed-effects model assumption. However, if severe asymmetry of the plot is observed (i.e. arise from heterogeneity, reporting bias or chance), further investigation of possible causes is strongly recommended as the appropriateness of a simple meta-analysis is doubted.

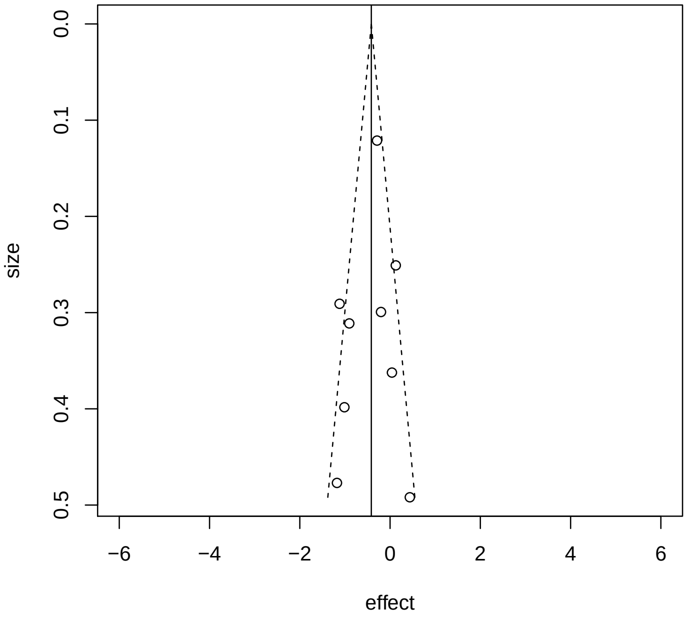

Figure 10: Funnel plot

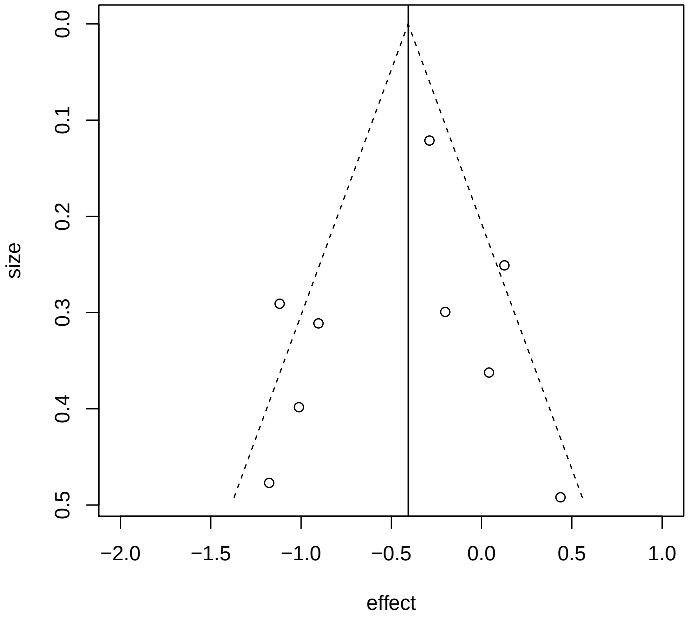

Figure 11: Funnel plot with adjusted x-axis

2.4 Diagnostic plot

Apart from the above 3 plots which provide graphical summary of modelling output, bmeta also produces some other plots for assessing model fit and performance. The diagnostic plot is designed to examine the convergence for model parameters. Both the Gelman-Rubin statistic and the effective sample size can be selected as the diagnostic criteria. It is suggested that a large Gelman-Rubin statistic and a small effective sample size (i.e. much smaller than the actual number of simulated iterations) may indicate problems of convergence for parameters. The diagnostic plot is simply implemented by the following codes:

```
> ### Diagnostic plot displaying Gelman-Rubin statistics (by default)
> diag.plot(x)

> ### Diagnostic plot displaying effective sample size
> funnel.plot(x,”n.eff”)
```

The first command generates Figure 12 where the Gelman-Rubin statistic (Rhat) is used as diagnostic statistics whereas the second plot (Figure 13) shows the number of effective simulated iterations against the total number of simulations (represented by dash line at the top) for each parameter. These plots display diagnostic statistics for all the model parameters. Notice that if not specified, the Gelman-Rubin statistic is set as the default option for this plot.

The interpretation of the plots is as follows: for the Gelman-Rubin statistic, there is a cut-off point of 1.1, values below which indicate convergence of the MCMC procedure to the target posterior distributions. Similarly, the effective sample size is computed as the equivalent number of independent samples that the MCMC runs produce (this is because the MCMC samples tend to be correlated and so to observe 1000 points may not produce the same information as 1000 independent values from a distribution). Thus, the lower the value (in comparison to the actual number of iterations saved to produce the summary statistics), the more relevant the presence of autocorrelation in the procedure.

Notice that, in general, some of the monitored nodes may be deterministic and so may be associated with an effective sample size of exactly 1. In other cases, while the Gelman-Rubin statistic suggests convergence for a given node, the effective sample size may not be very large. This would perhaps indicate the need of running the simulation procedure for a larger number of iterations on which the summary statistics can be based. In addition, these diagnostic plots can be used in conjunction with the trace plots, which we describe next.

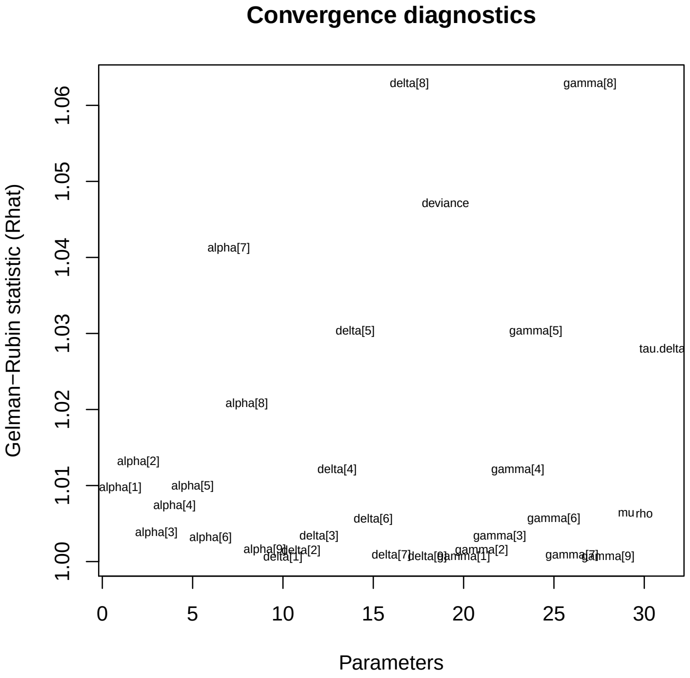

Figure 12: Diagnostic plot based on Gelman-Rubin statistics

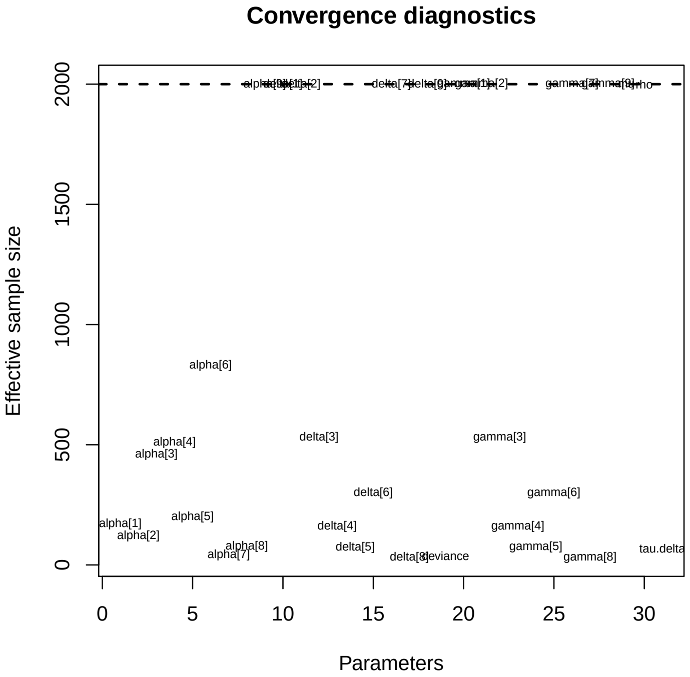

Figure 13: Diagnostic plot based on effective sample size

2.5 Trace plot

A plot of the sampled values versus the simulation index (termed a "trace plot") is considered to be very effective for assessing Markov Chain convergence for model parameters. It allows users to see if the chain has converged to a stationary distribution, in other words, if the chain still needs a longer burn-in period to achieve convergence. Moreover, one can tell whether a chain is mixing well, that is, the chain can traverse the posterior space rapidly by jumping from one to another remote region in the posterior in few steps. The 'perfect' trace plot can be described as that the chain appears to fluctuate around a certain value with small deviations, indicating a right distribution of the chain may have been obtained. Trace plot in bmeta can be produced easily by the following codes (the second argument specifies the model parameter for graphical display):

```
> ### Trace plot of model parameter alpha[1]
> traceplot.bmeta(x,"alpha[1]")
```

A typical trace plot of a certain node (here is alpha[1]) is displayed below. Note that the center of the chains appear to be around a value of 0.9 for this node, with some fluctuations. This suggests that the chains starting from different initial values are mixing well and moreover, these chains are not stuck in certain areas of the parameter space and therefore may have already reached convergence. If bad mixing of the chains is observed, users may need to run the simulation for a larger number of iterations until convergence is achieved, otherwise, we may doubt over the accuracy of the posterior estimates.

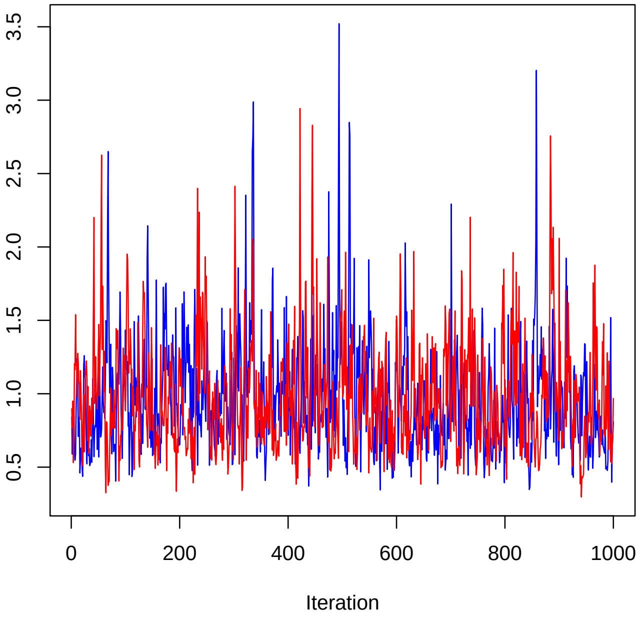

Figure 14: Trace plot

2.6 Autocorrelation plot

Autocorrelation plot is often used to check dependency among Markov chain samples. As the distribution of the current observation always depends on that of the previous one (a critical property of MCMC simulation), the iterations of a Markov chain are observed to be correlated. However, it is expected that the kth lag autocorrelation (defined as the correlation between every draw and its kth lag) gets smaller as k increases (i.e. the 5th and 100th draws should be less correlated than 5th and 10th draws). Therefore, if autocorrelation continues to be high for larger values of k, we would suspect that there is a slow mixing of chains and high degree of correlation between draws. The following codes program the required plot (the second argument specifies the model parameter for graphical display):

```
> ### Autocorrelation plot of model parameter alpha[1]
> acf.plot(x,"alpha[1]")
```

A sample autocorrelation plot of a certain node (here is `alpha[1]`) is presented below. This plot shows a relatively satisfactory autocorrelation as the correlation (represented by each vertical line) between every draw of this node and its kth lag decreases when k increases and even for a small value of k (i.e. k=5), the correlation is low (apart from the first one, all the rest vertical lines lie within the border formed by the two dash lines).

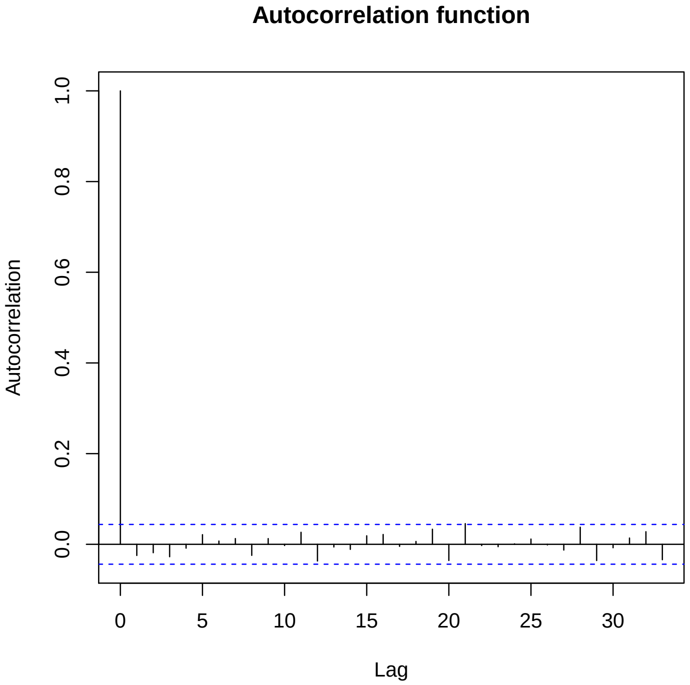

Figure 15: Autocorrelation function plot