Journal of Statistical Software

August 2017, Volume 80, Issue 5. doi: 10.18637/jss.v080.i05

Performing Arm-Based Network Meta-Analysis in R with the pcnetmeta Package

Lifeng Lin
University of Minnesota

Jing Zhang
University of Maryland

James S. Hodges
University of Minnesota

Haitao Chu
University of Minnesota

Abstract

Network meta-analysis is a powerful approach for synthesizing direct and indirect evidence about multiple treatment comparisons from a collection of independent studies. At present, the most widely used method in network meta-analysis is contrast-based, in which a baseline treatment needs to be specified in each study, and the analysis focuses on modeling relative treatment effects (typically log odds ratios). However, population-averaged treatment-specific parameters, such as absolute risks, cannot be estimated by this method without an external data source or a separate model for a reference treatment. Recently, an arm-based network meta-analysis method has been proposed, and the R package **pcnetmeta** provides user-friendly functions for its implementation. This package estimates both absolute and relative effects, and can handle binary, continuous, and count outcomes.

Keywords: absolute effect, arm-based method, Bayesian inference, network meta-analysis.

1. Introduction

In diverse scientific fields, such as social and medical research, summaries of cumulative knowledge are increasingly based on the results of meta-analyses (Hunter and Schmidt 1996; Lindholm, Carlberg, and Samuelsson 2005; Cooper, Hedges, and Valentine 2009). Meta-analysis is a statistical method for combining and contrasting a collection of estimated effect sizes, such as odds ratios, from multiple independent studies (DerSimonian and Laird 1986). Various approaches for treatment comparisons in meta-analysis have been introduced (Hedges and Olkin 1985; Sutton, Abrams, Jones, Jones, Sheldon, and Song 2000; Higgins and Green 2008).

Traditional meta-analysis focuses on *direct* pairwise comparisons between two treatments

in the collected studies. In some cases, however, not enough studies directly compare two treatments of interest. For example, suppose A, B, and C are three treatments for a disease, and researchers aim to compare A vs. B, while published studies only compare A vs. C or B vs. C. Although direct evidence is not available, the comparisons A vs. C and B vs. C provide *indirect* evidence. Based on this idea, network meta-analysis, also known as mixed treatment comparisons, was developed to *simultaneously* compare multiple treatments by synthesizing direct and indirect evidence (Lumley 2002; Lu and Ades 2004, 2006; Salanti, Higgins, Ades, and Ioannidis 2008; Lu and Ades 2009; Zhang *et al.* 2014). This technique has been widely applied in medical research (e.g., Psaty *et al.* 2003; Elliott and Meyer 2007; Cipriani *et al.* 2009). Currently, two types of approaches are used for network meta-analysis. The first approach is contrast-based. It focuses on modeling treatment contrasts (relative effect sizes, typically log odds ratios) within each study (Lu and Ades 2004). Another approach is arm-based (e.g., Zhang *et al.* 2014; Zhang, Chu, Hong, Neaton, Virnig, and Carlin 2017; Hong, Chu, Zhang, and Carlin 2016a,b), which focuses on describing population-averaged absolute effect sizes for each treatment arm.

We use an illustrative example to show the difference between the contrast- and arm-based approaches. Suppose that the outcome in a network meta-analysis is binary, and y_{ik} and n_{ik} are the numbers of events and participants, respectively, in treatment group k in the ith study. For such data, both the contrast- and arm-based models use the binomial likelihood $y_{ik} \sim \text{Binomial}(n_{ik}, p_{ik})$; they differ in the way they model the underlying absolute risks p_{ik} in each study's treatment group. Specifically, the contrast-based method needs to specify a baseline treatment $b(i)$ in the ith study. For convenience, we simply denote $b(i)$ as b. Then, the Bayesian hierarchical model for this approach is (Lu and Ades 2004, 2009):

$$\begin{aligned} g(p_{ik}) &= \mu_i + X_{ik}\delta_{ibk}; \\ \delta_{ibk} &\sim N(d_{bk}, \sigma^2_{bk}), \end{aligned}$$

where $g(\cdot)$ is a link function and X_{ik} is a dummy variable taking the value 0 if $k = b$ or 1 if $k \neq b$. Also, μ_i is the baseline effect for treatment b in the ith study, and δ_{ibk} is the relative effect of treatment k compared with the baseline b on the g-transformed scale. Note that this model treats the μ_i's as nuisances and uses non-informative priors for them. This model is described as contrast-based because it focuses on the overall relative effects d_{hk} between treatment pairs (h, k), which are estimated using the evidence consistency equation $d_{hk} = d_{bk} - d_{bh}$. This model does not permit a back-transformation from the relative effects to absolute effects, unless the absolute effect of a given "reference" treatment group can be accurately estimated from external data, or can be estimated using a separate model to analyze the existing data in the "reference" treatment group (Welton, Sutton, Cooper, Abrams, and Ades 2012; Dias, Welton, Sutton, and Ades 2013b).

Population-averaged absolute effects are preferred in some situations such as cost-effectiveness analysis and patient decisions (Dias *et al.* 2013b). For example, consider two scenarios comparing treatments A and B according to one-year survival rates: (i) $p_{\text{A}} = 0.8$ vs. $p_{\text{B}} = 0.5$; (ii) $p_{\text{A}} = 0.004$ vs. $p_{\text{B}} = 0.001$. Both scenarios yield an odds ratio of 4.0, but patients would prefer treatment A in scenario (i) more strongly than in scenario (ii). Therefore, an absolute effect or absolute difference is preferred in this case. Compared with the contrast-based method, the arm-based model provides a straightforward way to estimate absolute effects and various

types of relative effects. The model is specified as (Zhang *et al.* 2014, 2017):

$$g(p_{ik}) = \mu_k + \nu_{ik};$$
$$(\nu_{i1}, \nu_{i2}, \ldots, \nu_{iK})^\top \sim MVN(\mathbf{0}, \mathbf{\Sigma}_K),$$

where $\mathbf{\Sigma}_K$ is the variance-covariance matrix of the vector of random effects specific to study i. The μ_k's are treatment-specific parameters reflecting absolute effects. Based on the absolute effects, various types of relative effects can be obtained. In some cases (e.g., the missingness of treatment arms is not at random), the effect size estimates produced by the arm-based method may be less biased than those given by the contrast-based method (Zhang *et al.* 2017). Moreover, the arm-based method can use information contained in single-arm studies, in which only one treatment group is available or of interest. Single-arm studies cannot be included in the contrast-based model but they may provide valuable information for treatment comparisons and enhance the robustness of a network meta-analysis (Lin, Chu, and Hodges 2016). Although the arm-based approach has many advantages, the convergence of Markov chain Monte Carlo (MCMC) algorithms for parameter estimation may be slower compared with the contrast-based approach. The estimates using the arm-based models may not converge well if some treatments are only included in a few (say, less than three) studies. Also, some researchers have concerns about the arm-based model, for example, that absolute effects tend to be highly variable compared to relative effects, and that pooling arm-level data may not fully respect the randomization process in randomized controlled trials (Dias and Ades 2016). However, these concerns are mainly raised because the arm- and contrast-based models use different assumptions about treatment effects. Specifically, the contrast-based model assumes that relative effects are exchangeable across studies, while the arm-based model assumes absolute effects are exchangeable (Hong *et al.* 2016b). Although the assumption of exchangeable relative effects is popular in meta-analysis, the assumption of exchangeable absolute effects is also accepted in the literature (see, e.g., Van Houwelingen, Zwinderman, and Stijnen 1993; Shuster, Jones, and Salmon 2007; Senn 2010; Chu, Nie, Chen, Huang, and Sun 2012). More details of the arm-based model are discussed in Dias and Ades (2016) and Hong *et al.* (2016b).

Plenty of software packages are dedicated to conducting traditional meta-analysis (e.g., Rosenberg, Adams, and Gurevitch 2000; Borenstein, Hedges, Higgins, and Rothstein 2005; Viechtbauer 2014; Schwarzer 2015), but very limited software is available specifically for network meta-analysis. The R (R Core Team 2017) package **netmeta** (Rücker, Schwarzer, and Krahn 2015) provides models in a frequentist framework described in Rücker (2012). The R package **gemtc** (van Valkenhoef and Kuiper 2015) and the Stata (StataCorp 2015) module **network** (White 2015, 2017) perform contrast-based analyses. Neither package provides estimates for population-averaged treatment-specific parameters. This article introduces the R package **pcnetmeta** (Lin, Zhang, and Chu 2017), which performs network meta-analysis using the arm-based model and provides estimates for various effect sizes. This package is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=pcnetmeta`. It uses MCMC techniques on the R platform through **JAGS** (Plummer 2003, 2016). **JAGS** is a program for analyzing Bayesian hierarchical models using MCMC simulation, which is available for diverse computer platforms including Windows and Mac OS X.

The package **pcnetmeta** provides user-friendly functions to perform network meta-analysis for various types of data. Convergence of the MCMC routine can be assessed by the function outputs. The package also provides functions to draw network plots which illustrate the

comparisons between multiple treatments. In addition, plots for 95% credible intervals of treatment-specific and relative effect sizes are provided to visually display treatment effects and their comparisons.

This article is organized as follows. Section 2 presents an overview of arm-based Bayesian hierarchical models as implemented in the **pcnetmeta** package. Section 3 illustrates the use of the package with several examples, and discusses the output structures. Finally, Section 4 closes with suggested future improvements.

2. Arm-based models for network meta-analysis

2.1. Arm-based model for binary outcomes

Suppose a network meta-analysis reviews I studies on K treatments, where each study only investigated a subset of the K treatments. Let T_i $(1 \leq i \leq I)$ be the set of treatments compared in the ith study. Also, in the ith study, the total number of events/participants in treatment group k $(k \in T_i)$ is denoted by y_{ik}/n_{ik}. Zhang *et al.* (2014) specified the following arm-based model using the probit link function:

$$\begin{aligned} y_{ik} &\sim \text{Binomial}(n_{ik}, p_{ik}), \qquad k \in T_i, \\ \Phi^{-1}(p_{ik}) &= \mu_k + \nu_{ik}, \\ (\nu_{i1}, \nu_{i2}, \cdots, \nu_{iK})^\top &\sim MVN(\mathbf{0}, \mathbf{\Sigma}_K), \end{aligned} \tag{1}$$

where $\Phi(\cdot)$ denotes the standard normal cumulative distribution function. In this model, μ_k is a fixed effect for treatment k, and the random effects $(\nu_{i1}, \nu_{i2}, \ldots, \nu_{iK})^\top$ are correlated within each study with variance-covariance matrix $\mathbf{\Sigma}_K$. Based on this model, the absolute risk of treatment k can be estimated as $p_k = \mathsf{E}[p_{ik}|\mu_k, \sigma_k] = \Phi\left(\mu_k/\sqrt{1+\sigma_k^2}\right)$ (Zeger, Liang, and Albert 1988), where σ_k^2 is the kth diagonal element in $\mathbf{\Sigma}_K$. Since this estimate is a marginal expectation of p_{ik} given μ_k and σ_k, we can interpret p_k as the population-averaged absolute risk of treatment k. With the estimates p_k for absolute risk, we can further estimate the risk difference (RD), the odds ratio (OR), and the risk ratio (RR), which are defined as $\text{RD}_{kl} = p_k - p_l$, $\text{OR}_{kl} = \frac{p_k/(1-p_k)}{p_l/(1-p_l)}$, and $\text{RR}_{kl} = p_k/p_l$, respectively. Other link functions may be also considered in the arm-based model, but they do not yield simple expressions for population-averaged absolute risks, though some approximations exist. For example, using the logit link we can approximate $p_k = \mathsf{E}[p_{ik}|\mu_k, \sigma_k] \approx \left[1 + \exp\left(-\mu_k/\sqrt{1+C^2\sigma_k^2}\right)\right]^{-1}$, where $C = \frac{16\sqrt{3}}{15\pi}$ (Zeger *et al.* 1988). This article and the package **pcnetmeta** use the probit link function for simplicity.

2.2. Arm-based model for continuous outcomes

Researchers can also perform network meta-analysis on studies with continuous outcomes (e.g., Kasiske, Lakatua, Ma, and Louis 1998; Philbrook, Barrowman, and Garg 2007; Zhang, Fu, and Carlin 2015). We continue to use i and k to index the studies and treatments and n_{ik} as the total number of participants receiving treatment k in the ith study. The summary data include sample mean $\bar{y}_{ik}$ and within-study sample standard deviation s_{ik}. We can specify the

arm-based model for continuous outcomes as

$$\begin{aligned} \bar{y}_{ik} &\sim N(\theta_{ik}, s_{ik}^2/n_{ik}), \qquad k \in T_i, \\ \theta_{ik} &= \mu_k + \nu_{ik}, \\ (\nu_{i1}, \nu_{i2}, \cdots, \nu_{iK})^\top &\sim MVN(\mathbf{0}, \mathbf{\Sigma}_K). \end{aligned} \tag{2}$$

In this model, μ_k is of interest because it can be interpreted as the overall effect of treatment k by noticing that $\mathsf{E}[\theta_{ik}|\mu_k, \sigma_k] = \mu_k$, and we may further estimate the effect difference between treatments k and l, which is defined as $d_{kl} = \mu_k - \mu_l$.

2.3. Arm-based model for count datasets

Dias, Sutton, Ades, and Welton (2013a) discussed methods for count datasets in network meta-analysis. Their models are contrast-based, but we can consider the corresponding arm-based versions.

In some network meta-analyses, the available data are in the form of counts over a certain time period. The total number of person-years at risk is supplied rather than the total number of participants. Let y_{ik} be the number of events in treatment group k in the ith study, and E_{ik} is the corresponding exposure time in person-years. Suppose λ_{ik} is the treatment-specific rate for treatment group k in the ith study, and we are interested in the population-averaged treatment-specific rate. We consider the following arm-based model with a Poisson likelihood and the log link function:

$$\begin{aligned} y_{ik} &\sim \text{Poisson}(E_{ik}\lambda_{ik}), \qquad k \in T_i, \\ \log(\lambda_{ik}) &= \mu_k + \nu_{ik}, \\ (\nu_{i1}, \nu_{i2}, \cdots, \nu_{iK})^\top &\sim MVN(\mathbf{0}, \mathbf{\Sigma}_K). \end{aligned} \tag{3}$$

A key assumption for this model is that the rates λ_{ik} are constant over the follow-up period. Based on this model, the population-averaged treatment specific rate can be estimated as $\lambda_k = \mathsf{E}[\lambda_{ik}|\mu_k, \sigma_k] = \exp(\mu_k + \sigma_k^2/2)$.

A similar model is available for studies that report the proportion of patients developing an event within a given follow-up period, where the follow-up time may differ for each study (e.g., Psaty *et al.* 2003; Elliott and Meyer 2007). In this case, the event probability depends on the length of follow-up. Specifically, we have y_{ik} and n_{ik} as the number of events and participants, respectively, in treatment group k in the ith study, and the study-specific follow-up times are denoted f_i. Following Dias *et al.* (2013a), we assume a latent Poisson process with rate λ_{ik} for each treatment group in each study; therefore the time T_{ik} until an event occurs is distributed as exponential with rate λ_{ik} and survivor function $\mathsf{P}(T_{ik} > f_i) = \exp(-\lambda_{ik} f_i)$. Thus, the event probability is $p_{ik} = 1 - \exp(-\lambda_{ik} f_i)$. Using the complementary log-log link $\text{cloglog}(t) = \log(-\log(1-t))$ for p_{ik}, we have $\text{cloglog}(p_{ik}) = \log(f_i) + \log(\lambda_{ik})$. Again, we can model $\log(\lambda_{ik})$ using model (3) above, and an arm-based model is constructed as follows:

$$\begin{aligned} y_{ik} &\sim \text{Binomial}(n_{ik}, p_{ik}), \qquad k \in T_i, \\ \text{cloglog}(p_{ik}) &= \log(f_i) + \log(\lambda_{ik}), \\ \log(\lambda_{ik}) &= \mu_k + \nu_{ik}, \\ (\nu_{i1}, \nu_{i2}, \cdots, \nu_{iK})^\top &\sim MVN(\mathbf{0}, \mathbf{\Sigma}_K). \end{aligned} \tag{4}$$

The parameter of interest is still the population-averaged treatment-specific rate $\lambda_k = \exp(\mu_k + \sigma_k^2/2)$.

2.4. Parameter estimation for arm-based models

Let $\boldsymbol{\nu}_i = (\nu_{i1}, \nu_{i2}, \ldots, \nu_{iK})^\top$. The full likelihood function for the arm-based model is

$$L \propto \prod_{i=1}^{I} \int_{\mathbb{R}^K} \left(\prod_{k \in T_i} L_c(y_{ik}; \mu_k, \nu_{ik}) \right) \times \frac{\exp\left(-\frac{1}{2}\boldsymbol{\nu}_i^\top \boldsymbol{\Sigma}_K^{-1} \boldsymbol{\nu}_i\right)}{(2\pi)^{K/2} |\boldsymbol{\Sigma}_K|^{1/2}} d\boldsymbol{\nu}_i,$$

where $L_c(y_{ik}; \mu_k, \nu_{ik})$ is the conditional likelihood given the random effects ν_{ik}. For example, for binary outcomes, the conditional likelihood is

$$L_c(y_{ik}; \mu_k, \nu_{ik}) = \left[g^{-1}(\mu_k + \nu_{ik})\right]^{y_{ik}} \left[1 - g^{-1}(\mu_k + \nu_{ik})\right]^{n_{ik} - y_{ik}},$$

where $g^{-1}(\cdot)$ is the inverse of the link function. To obtain the maximum likelihood estimates, the maximization problem is subject to the condition that $\boldsymbol{\Sigma}_K$ is positive definite. When the outcome is binary or count, the full likelihood function cannot be expressed in a closed form; for continuous outcomes, it may have a closed form. However, maximizing the likelihood may be unstable and converge slowly if the number of treatments K is large and the number of collected studies I is small.

Alternatively, we can, and **pcnetmeta** does, apply MCMC to obtain Bayesian estimates for parameters of interest. In arm-based models, vague $N(0, 1000)$ priors are used for the treatment-specific fixed effects μ_k. As suggested in Gelman and Hill (2007), we may assign an inverse-Wishart prior to the unstructured variance-covariance matrix $\boldsymbol{\Sigma}_K$ with the scale matrix being the $K \times K$ identity matrix $\mathbf{I}_K$ and degrees of freedom $K+1$, i.e., $\boldsymbol{\Sigma}_K^{-1} \sim \text{Wishart}(\mathbf{I}_K, K+1)$. This has the effect of setting a uniform prior on the individual correlation parameters. Alternatively, the separation strategy by Cholesky decomposition can be used to specify a vague prior to $\boldsymbol{\Sigma}_K$ (Barnard, McCulloch, and Meng 2000; Lu and Ades 2009; Wei and Higgins 2013). We denote this model as HET-COR because the variances of the random effects are heterogeneous.

To reduce model complexity, we may assume an exchangeable correlation structure for $\boldsymbol{\Sigma}_K$ (Zhang *et al.* 2017), that is, $\boldsymbol{\Sigma}_K = \mathbf{D}\mathbf{R}_{\text{ex}}\mathbf{D}$, where $\mathbf{D} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_K)$ and

$$\mathbf{R}_{\text{ex}} = \begin{bmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \cdots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \cdots & 1 \end{bmatrix}.$$

To guarantee that $\mathbf{R}_{\text{ex}}$ is positive definite, ρ must be greater than $-\frac{1}{K-1}$. A vague uniform prior on $\left(-\frac{1}{K-1}, 1\right)$ can be used for ρ. We denote models with this exchangeable correlation matrix as HET-EQCOR for heterogeneous variances σ_k^2. If we further assume homogeneity of variances, that is, $\sigma_k^2 = \sigma^2$ for all $k = 1, 2, \ldots, K$, the model is denoted as HOM-EQCOR.

To avoid overfitting, we can use the deviance information criterion (DIC) proposed by Spiegelhalter, Best, Carlin, and Van der Linde (2002) for model selection. A smaller penalized deviance implies a better model.

Finally, to implement the hierarchical models in **JAGS**, the package **pcnetmeta** automatically generates initial values for the parameters and specifies different random number generators (RNGs) for different chains. Zero is used to initialize the random effects ν_{ik}; the initial values for the variance parameters $\mathbf{\Sigma}_K$ are the means of their prior distributions. For example, since $\mathbf{\Sigma}_K^{-1} \sim \text{Wishart}(\mathbf{I}_K, K+1)$ in the HET-COR model, the initial value for $\mathbf{\Sigma}_K^{-1}$ is the mean of this Wishart distribution, $(K+1)\mathbf{I}_K$. For the fixed effects μ_k, the initial values are the naïve estimates of the corresponding absolute effects computed by simply pooling the data in each treatment group. For example, consider the continuous outcome in model (2), the initial values for μ_k's are $\sum_{\{i:k\in T_i\}} \bar{y}_{ik} n_{ik} / \sum_{\{i:k\in T_i\}} n_{ik}$.

3. Using the R package pcnetmeta

The R package **pcnetmeta** provides user-friendly functions to perform arm-based network meta-analysis using the models described above. Users can download its source file at https://CRAN.R-project.org/package=pcnetmeta, or directly install it within R by typing `install.packages("pcnetmeta")`. Note that the **pcnetmeta** package depends on the R packages **rjags** (Plummer 2016) and **coda** (Plummer, Best, Cowles, and Vines 2006). The **pcnetmeta** package does not include a copy of the **JAGS** library, so users must install **JAGS** separately. **JAGS** is freely available at its homepage http://mcmc-jags.sourceforge.net/. Also, the package **pcnetmeta** requires **JAGS** version $\geq$ 4.0.0; the earlier versions of **JAGS** may not guarantee exact reproducibility of the results. In this section, we introduce the basic usage of this package.

3.1. Data structure for network meta-analysis

To begin, we briefly introduce the necessary dataset structures. In the package **pcnetmeta**, four datasets, `smoke`, `parkinson`, `dietaryfat`, and `diabetes`, are provided as illustrative examples.

The dataset `smoke` contains 24 studies on smoking cessation with binary outcomes, reported in Hasselblad (1998) and Lu and Ades (2006). This network meta-analysis compares four treatments, labeled as: 1) no contact (NC); 2) self-help (SH); 3) individual counseling (IC); and 4) group counseling (GC). We display the dataset's first few rows below. The column `s.id` contains IDs for the 24 studies, and `t.id` labels the treatments included in each study. For example, Study 1 compares treatments 1, 3, and 4. The columns `r` and `n` are the number of events (successful cessation) and participants, respectively.

```
R> library("pcnetmeta")
R> data("smoke", package = "pcnetmeta")
R> head(smoke)
```

```
  s.id t.id  r   n
1    1    1  9 140
2    1    3 23 140
3    1    4 10 138
4    2    2 11  78
5    2    3 12  85
6    2    4 29 170
```

The dataset `parkinson` is a collection of studies with continuous outcomes, reported in Dias *et al.* (2013a). It contains 7 studies on 5 treatments. The outcome is the mean off-time reduction in patients given dopamine agonists as adjunct therapy in Parkinson's disease. One treatment is placebo, coded by 1, and the other four treatments are active drugs, coded 2 to 5. The dataset is displayed below. The columns `s.id`, `t.id` and `n` have the same meanings as in the dataset `smoke`, while `mean` and `sd` report the sample means and standard deviations of the continuous outcome.

```
R> data("parkinson", package = "pcnetmeta")
R> parkinson
```

```
   s.id t.id  mean   sd   n
1     1    1 -1.22 3.70  54
2     1    3 -1.53 4.28  95
3     2    1 -0.70 3.70 172
4     2    2 -2.40 3.40 173
5     3    1 -0.30 4.40  76
6     3    2 -2.60 4.30  71
7     3    4 -1.20 4.30  81
8     4    3 -0.24 3.00 128
9     4    4 -0.59 3.00  72
10    5    3 -0.73 3.00  80
11    5    4 -0.18 3.00  46
12    6    4 -2.20 2.31 137
13    6    5 -2.50 2.18 131
14    7    4 -1.80 2.48 154
15    7    5 -2.10 2.99 143
```

The datasets `dietaryfat` and `diabetes` serve as examples for models (3) and (4). This article uses `diabetes` to illustrate estimation of treatment-specific rates and rate ratios. This dataset was analyzed by Elliott and Meyer (2007) to assess the effects of antihypertensive agents on incident diabetes, and includes the follow-up times (in years) for each study. Twenty-two clinical studies are included, covering six different treatments: 1) diuretic; 2) placebo; 3) beta blocker (BB); 4) calcium-channel blocker (CCB); 5) angiotensin-converting-enzyme inhibitor (ACEI); and 6) angiotensin-receptor blocker (ARB). Users can apply `data()` in R to load the datasets `dietaryfat` and `diabetes`; we do not display the detailed datasets.

Note that `NA` is not allowed in a dataset for the package **pcnetmeta**, because the published articles collected in a network meta-analysis typically report all summary results (such as *both* mean and variance for continuous outcomes). Also, each row in the dataset represents one treatment group in a study, so a single-arm study is straightforwardly input as a single row in the dataset for analysis using the arm-based models in the **pcnetmeta** package.

3.2. Plotting the network

The function `nma.networkplot()` in package **pcnetmeta** provides a visual overview of treatment comparisons in network datasets. Calling this function produces a network graph in an R plot window. Each vertex in the network plot represents a treatment and each edge

between two nodes stands for a direct comparison between the corresponding two treatments. The usage of the function is as follows:

```
nma.networkplot(s.id, t.id, data, title = "", trtname, alphabetic = TRUE,
  weight.edge = TRUE, adjust.thick = 5, weight.node = TRUE,
  adjust.node.size = 10, node.col = "orange", edge.col = "black",
  text.cex = 1, adjust.figsizex = 1.1, adjust.figsizey = 1.1)
```

Users need to input `s.id` and `t.id` for study and treatment IDs respectively. The argument `title` gives the graph title, and `trtname` specifies the treatment names. If `trtname` is not specified, the treatment IDs given in `t.id` are used. The argument `alphabetic` is a logical value indicating whether to sort the treatment nodes alphabetically in the clockwise direction according to the treatment names; if `alphabetic = FALSE`, the nodes are sorted according to treatment IDs (`t.id`). The logical argument `weight.edge = TRUE` causes the edge thickness to be drawn proportional to the number of direct treatment comparisons; `weight.node = TRUE` causes the node size to be proportional to the number of direct comparisons which contain that treatment node.

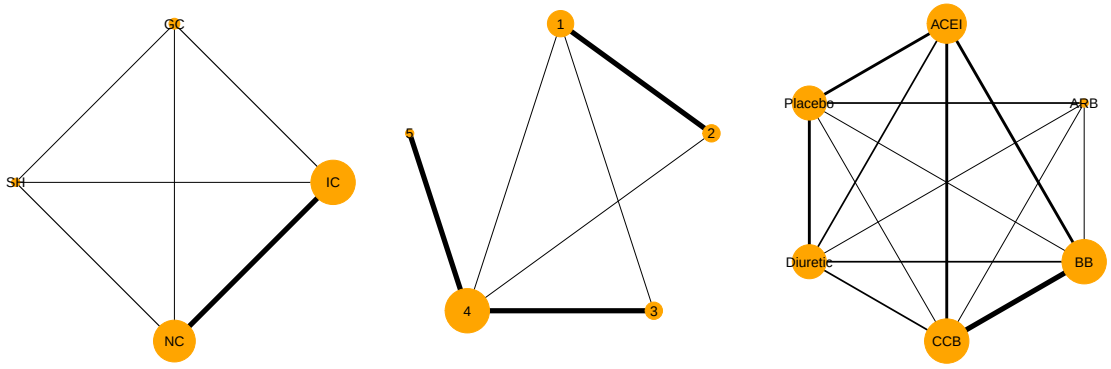
The following code produces network plots for the datasets `smoke`, `parkinson`, and `diabetes` respectively.

```
R> data("smoke", package = "pcnetmeta")
R> nma.networkplot(s.id, t.id, data = smoke,
+    trtname = c("NC", "SH", "IC", "GC"))
R> data("parkinson", package = "pcnetmeta")
R> nma.networkplot(s.id, t.id, data = parkinson)
R> data("diabetes", package = "pcnetmeta")
R> nma.networkplot(s.id, t.id, data = diabetes,
+    trtname = c("Diuretic", "Placebo", "BB", "CCB", "ACEI", "ARB"))
```

Figure 1 shows the resulting graphs. In the left panel for the smoking cessation data, every pair of treatment nodes is connected by an edge, so all pairs of treatments are directly compared. For the dataset `parkinson` in Figure 1b, we did not specify treatment names, so the function used the treatment IDs from 1 to 5 as the names. Its network plot shows no edge between some pairs of treatments, e.g., treatments 2 and 3. This means that no study directly compares treatments 2 and 3. As a result, if the aim is to compare the effects of treatments 2 and 3, only indirect evidence is available, e.g., from the comparisons of treatments 2 vs. 1 and 3 vs. 1. Figure 1c is the network plot for the dataset `diabetes`; all pairs of treatments are directly compared except ACEI and ARB.

3.3. Performing arm-based network meta-analysis

The major functions in package **pcnetmeta** are `nma.ab.bin()`, `nma.ab.cont()`, `nma.ab.py()`, and `nma.ab.followup()`, which perform arm-based network meta-analysis for different types of data using the models introduced in Section 2. In particular, `nma.ab.bin()` analyzes binary outcomes, while `nma.ab.cont()` is used for continuous outcomes. These two functions are based on models (1) and (2), respectively. Functions `nma.ab.py()` and `nma.ab.followup()` can be used when exposure times or follow-up times are available, and are based on the



(a) Smoking cessation (b) Parkinson's disease (c) Diabetes

Figure 1: Network plots generated by the function `nma.networkplot()`.

models (3) and (4), respectively. The commands in each of the following subsections may take around 5–20 minutes on an Intel 2.60 GHz processor. The actual runtime depends on the complexity of the treatment network and the user's processor.

Function `nma.ab.bin()` *for binary outcomes*

The arguments of the function `nma.ab.bin()` are as follows

```
nma.ab.bin(s.id, t.id, event.n, total.n, data, trtname,
  param = c("AR", "LOR", "LRR", "RD", "rank.prob"), model = "het_cor",
  prior.type, a = 0.001, b = 0.001, c = 10, higher.better = FALSE,
  digits = 4, n.adapt = 5000, n.iter = 1e+05, n.burnin = floor(n.iter/2),
  n.chains = 3, n.thin = max(1, floor((n.iter - n.burnin)/1e+05)),
  conv.diag = FALSE, trace = NULL, dic = FALSE, postdens = FALSE,
  mcmc.samples = FALSE)
```

As in `nma.networkplot()`, the arguments `s.id` and `t.id` are numeric or character vectors indicating study and treatment IDs. Users also specify each study's number of events and participants using `event.n` and `total.n` respectively. The argument `model` can be specified as `"het_cor"`, `"het_eqcor"`, or `"hom_eqcor"`, which corresponds to the models described in Section 2.4. When `model = "het_cor"` (the default), users can specify `prior.type = "invwishart"` (the default) to assign an inverse-Wishart prior to the variance-covariance matrix of random effects. Alternatively, by assigning `prior.type = "chol"`, the separation strategy by Cholesky decomposition is used for the variance-covariance matrix, and uniform priors $U(0, c)$ are assigned to the standard deviations and vague priors are assigned to the correlation components (Barnard *et al.* 2000; Lu and Ades 2009; Wei and Higgins 2013). When `"het_eqcor"` and `"hom_eqcor"` are used, the correlation matrix of the random effects has an exchangeable correlation structure. For the `"hom_eqcor"` and `"het_eqcor"` models, two types of priors for the random-effect variances can be used. A popular prior for variances is inverse-Gamma(ϵ, ϵ) where ϵ can be set to a low value (e.g., 0.001) (Spiegelhalter, Thomas, Best, and Lunn 2003). However, the posterior may be sensitive to the choice of ϵ, and a uniform prior for standard deviations is preferred in some cases (Gelman 2006). Users choose the prior by specifying `prior.type = "unif"` (the default) or `"invgamma"`, and the prior

parameters `a`, `b`, and `c`. If inverse-Gamma priors are used, inverse-Gamma(a, b) with density $\frac{b^a}{\Gamma(a)} x^{-a-1} e^{-b/x}$ is assigned to the variances of the random effects; if uniform priors are used, $U(0, c)$ is assigned to the standard deviations. The default prior parameters are `a = b = 0.001` and `c = 10`.

The argument `param` is a character string vector which indicates the effect sizes to be estimated. As in Section 2.1, `param` can include absolute risk (`"AR"`), odds ratio (`"OR"`), log odds ratio (`"LOR"`), risk ratio (`"RR"`), log risk ratio (`"LRR"`), risk difference (`"RD"`). In addition, researchers may be interested in treatment ranks (Salanti, Ades, and Ioannidis 2011). Users can estimate the rank probabilities of different treatments (i.e., probabilities of the treatment having ranks 1, 2, ..., K) by adding `"rank.prob"` to the argument `param`. When `"rank.prob"` is added, users need to specify the logical argument `higher.better`; `TRUE` indicates that a higher event probability implies a better treatment. For example, the event in the dataset `smoke` is smoking cessation, and a higher smoking cessation probability implies a better treatment. Many outcomes in medical studies are the events of developing a disease (e.g., Thijs, Lemmens, and Fieuws 2008), in which case a better treatment should lead to a lower event probability.

The arguments `n.adapt`, `n.iter`, `n.burnin`, `n.chains`, and `n.thin` control the MCMC algorithm run by **rjags** (Plummer 2016). The argument `n.adapt` is the number of iterations for adaptation (the default is 5,000); this is used to maximize the sampling efficiency. The argument `n.iter` determines the number of iterations in each MCMC chain, and `n.burnin` is the number of burn-in iterations at the beginning of each chain which are discarded. The argument `n.chains` is the number of MCMC chains; the default is 3. Additionally, `n.thin` is the thinning rate for MCMC chains, which is used to save memory and computation time if `n.iter` is large. For example, if `n.iter` is 10^6 and `n.thin` is 10, then only one sample would be kept in every 10 samples in each chains, and the remaining number of iterations is 10^5.

The argument `conv.diag` specifies whether to compute potential scale reduction factors (PSRFs) proposed by Gelman and Rubin (1992) for convergence diagnostics. The argument `trace` is a character string vector which can be chosen from the elements specified in `param` except for `"rank.prob"`. Trace plots of the specified effect sizes are saved in users' current working directory as .png files. A trace plot is a plot of the sampled parameter estimates at each iteration against iteration number. Both PSRFs and trace plots can be used to examine whether the MCMC chains are drawn from stationary distributions. Finally, if `dic = TRUE`, the function will provide the deviance information criterion (DIC) statistic proposed by Spiegelhalter *et al.* (2002); conventionally the model with smallest DIC is considered the best among the candidate models. The posterior density plot of treatment-specific effect sizes can be obtained as a .pdf file by setting `postdens = TRUE`.

The function `nma.ab.bin()` returns a list with effect size estimates, which lists the posterior mean, standard deviation, median, and a 95% credible interval (CI) with 2.5% and 97.5% quantiles as the lower and upper bounds.

Here is an example to demonstrate the function's usage. We call the function `nma.ab.bin()` on the dataset `smoke` as follows:

```
R> data("smoke", package = "pcnetmeta")
R> set.seed(12345)
R> smoke.out <- nma.ab.bin(s.id, t.id, r, n, data = smoke, trtname = c("NC",
+    "SH", "IC", "GC"), param = c("AR", "OR", "RR", "LOR", "LRR", "RD",
```

```
+    "rank.prob"), model = "het_cor", higher.better = TRUE, digits = 3,
+    n.adapt = 10000, n.iter = 200000, n.thin = 1, conv.diag = TRUE,
+    dic = TRUE, trace = "LOR", postdens = TRUE)
```

The following messages were outputted:

```
Start running MCMC...
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
   Observed stochastic nodes: 50
   Unobserved stochastic nodes: 29
   Total graph size: 646

Initializing model

  |++++++++++++++++++++++++++++++++++++++++++++++++++| 100%
  |**************************************************| 100%
  |**************************************************| 100%
Start calculating MCMC convergence diagnostic statistics...
Start calculating deviance information criterion statistics...
  |**************************************************| 100%
Start saving trace plots...
Start saving posterior density plot for absolute risk...
```

When a **JAGS** model is compiled, it may require an initial sampling phase during which the samplers adapt their behavior to maximize their efficiency (e.g., a Metropolis-Hastings random walk algorithm may change its step size) (Plummer 2016). The warning "adaptation incomplete" may occasionally occur if the number of iterations for the adaptation process (i.e., the argument `n.adapt`) is not sufficient, so the MCMC algorithm may not achieve the maximum efficiency. This warning generally has little impact on the posterior estimates of the treatment effects. To avoid this warning, users may increase `n.adapt`.

The results are saved in the object `smoke.out`, a list containing `AbsoluteRisk`, `OddsRatio`, `LogOddsRatio`, `RelativeRisk`, `LogRelativeRisk`, `RiskDifference`, `TrtRankProb`, and `DIC`. We can use these effect size names to display the corresponding estimates. For example, the estimates of absolute risks (posterior mean and standard deviation, and posterior median and 95% credible interval) can be displayed as

```
R> smoke.out$AbsoluteRisk
```

```
$Mean_SD
   Mean (SD)
NC 0.082 (0.014)
SH 0.167 (0.052)
IC 0.185 (0.028)
GC 0.229 (0.064)
```

```
$Median_CI
   Median (95% CI)
NC 0.081 (0.058, 0.114)
SH 0.159 (0.086, 0.291)
IC 0.183 (0.136, 0.245)
GC 0.221 (0.128, 0.373)
```

The argument `digits` in the function `nma.ab.bin()` can be used to change the number of digits to the right of the decimal point. Here, we used `digits = 3`. Each list element in the object `smoke.out` consists of two sublists: `Mean_SD` contains posterior sample means with sample standard deviations; `Median_CI` contains posterior medians with 95% CIs. For example, users can output the medians and 95% CIs for the log odds ratio as follows:

```
R> smoke.out$LogOddsRatio$Median_CI
```

```
   NC                     SH                        IC
NC --                     -0.761 (-1.610, 0.005) -0.924 (-1.430, -0.445)
SH 0.761 (-0.005, 1.610) --                       -0.165 (-0.929, 0.657)
IC 0.924 (0.445, 1.430)  0.165 (-0.657, 0.929)  --
GC 1.170 (0.402, 1.980)  0.404 (-0.614, 1.420)  0.237 (-0.516, 1.050)
   GC
NC -1.170 (-1.980, -0.402)
SH -0.404 (-1.420, 0.614)
IC -0.237 (-1.050, 0.516)
GC --
```

Since the log odds ratio is a relative effect size comparing a pair of treatments, the output estimates are displayed in a $K \times K$ matrix, where K is the number of treatments. In this example, $K = 4$. The element in the ith row and jth column is the estimated log odds ratio of treatment i compared to treatment j. To statistically test the difference between treatments, one can examine whether the effect size under the null hypothesis (i.e., the two treatments do not differ) is within the corresponding 95% CI; under the null hypothesis, ORs and RRs are 1, while LORs, LRRs, and RDs are 0. From the output, the 95% CI of the log odds ratio for SH vs. NC is $(-0.032, 1.610)$ which contains 0; therefore, there is not sufficient evidence to reject the null hypothesis. However, the 95% CI of the log odds ratio comparing GC to NC is $(0.399, 1.970)$, which does not contain 0; therefore, these two treatments differ significantly.

Also, in this example, we included `"rank.prob"` in the argument `param` to estimate treatment rank probabilities, and `dic` was specified as `TRUE` to calculate the DIC statistic. Recall that for the smoking cessation dataset, a higher event probability implies a better treatment, so we specified the argument `higher.better` as `TRUE`. Users can access the treatment rank probabilities and DIC statistics using

```
R> smoke.out$TrtRankProb
```

```
   rank1  rank2  rank3  rank4
NC 0.0000 0.0003 0.0272 0.9730
```

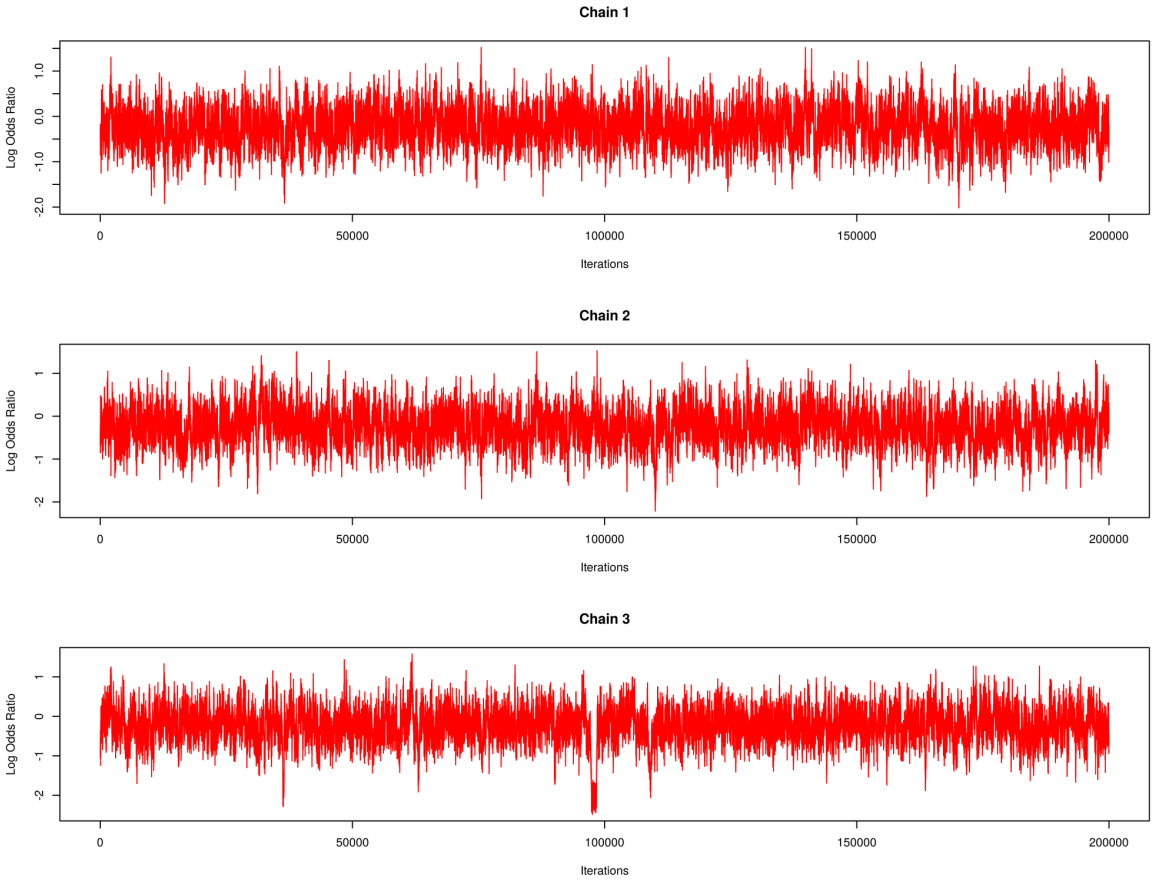


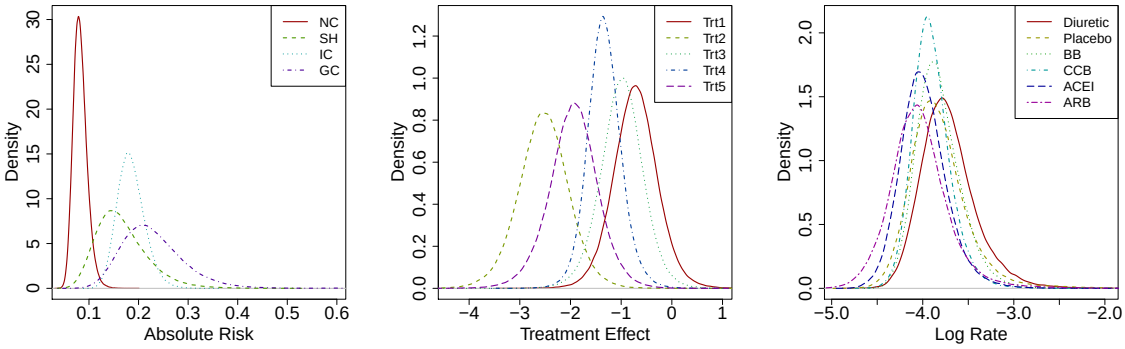
Figure 2: Trace plots generated by R function `nma.ab.bin()` for the log odds ratio comparing IC and GC in the smoking cessation data.

```
SH 0.1530 0.2410 0.5800 0.0257
IC 0.1980 0.5350 0.2670 0.0001
GC 0.6480 0.2240 0.1260 0.0017
```

```
R> smoke.out$DIC
```

```
D.bar 278.70025
pD     44.22552
DIC   322.92577
```

From the output, treatment GC has the highest probability of being the best treatment (66.2%). As for the DIC statistic, `D.bar` is the posterior expectation of the deviance, which reflects the model fit; it is usually lower when more parameters are used in the model. However, complex models may lead to overfitting. To balance between the number of parameters and fitting effects, `pD` is used to penalize `D.bar`; it reflects the number of effective parameters used in the model. DIC is the penalized deviance, calculated as the sum of `D.bar` and `pD`; a model with smaller DIC is preferred.



(a) Smoking cessation (b) Parkinson's disease (c) Diabetes

Figure 3: Posterior density plots generated by the functions in package **pcnetmeta**.

Trace plots of LOR are generated because `trace = "LOR"` was specified. Figure 2 shows the trace plots of the LOR comparing IC and GC. Since we used the default `n.chains = 3`, three trace plots are drawn. Each trace plot shows evidence that the posterior samples of LOR are drawn from the stationary distribution.

A posterior density plot (Figure 3a) for treatment-specific absolute risks is also generated by specifying `postdens = TRUE`. This density plot is smoothed by the R function `density()`. This plot shows visualized treatment effects, and we may also evaluate treatment differences. For example, NC clearly has lower event probability than IC and GC, and its posterior density only overlaps with the densities of IC and GC in tiny regions.

Function `nma.ab.cont()` *for continuous outcomes*

For continuous outcomes, the arguments of `nma.ab.cont()` are mostly similar to those of `nma.ab.bin()`. The major difference is that users need to specify the summaries of continuous outcomes (sample means and standard deviations) for the arguments `mean` and `sd` in `nma.ab.cont()`. Also, the effect sizes to be estimated include continuous treatment-specific effects and their differences. Users can specify the argument `param` as `"mu"` to estimate treatment-specific effects and `"diff"` to estimate effect differences. Also, `"rank.prob"` can be included in `param` to estimate treatment rank probabilities. The network dataset `parkinson` is used as an example:

```
R> data("parkinson", package = "pcnetmeta")
R> set.seed(12345)
R> parkinson.out <- nma.ab.cont(s.id, t.id, mean, sd, n, data = parkinson,
+    model = "hom_eqcor", prior.type = "unif", digits = 3, n.adapt = 10000,
+    n.iter = 100000, n.thin = 1, conv.diag = TRUE, trace = "mu",
+    postdens = TRUE)
```

In this example, we used the model `"hom_eqcor"`, which assumes an exchangeable correlation structure for the correlations between treatment effects. We display the medians and the corresponding 95% CIs for treatment-specific effects and effect differences as follows.

```
R> parkinson.out$TrtEffect$Median_CI
```

```
     Median (95% CI)
Trt1 -0.719 (-1.600,  0.174)
Trt2 -2.530 (-3.550, -1.520)
Trt3 -0.985 (-1.860, -0.144)
Trt4 -1.340 (-1.990, -0.617)
Trt5 -1.940 (-2.960, -0.969)
```

```
R> parkinson.out$EffectDiff$Median_CI
```

```
     Trt1                     Trt2                     Trt3
Trt1 --                       1.810 (0.744, 2.880)  0.274 (-0.839, 1.370)
Trt2 -1.810 (-2.880, -0.744) --                    -1.540 (-2.770, -0.324)
Trt3 -0.274 (-1.370, 0.839)  1.540 (0.324, 2.770)  --
Trt4 -0.618 (-1.590, 0.403)  1.190 (0.108, 2.330)  -0.338 (-1.260, 0.557)
Trt5 -1.210 (-2.520, -0.018) 0.600 (-0.833, 1.910) -0.916 (-2.330, 0.234)
     Trt4                     Trt5
Trt1 0.618 (-0.403, 1.590)   1.210 (0.018, 2.520)
Trt2 -1.190 (-2.330, -0.108) -0.600 (-1.910, 0.833)
Trt3 0.338 (-0.557, 1.260)   0.916 (-0.234, 2.330)
Trt4 --                      0.558 (-0.218, 1.750)
Trt5 -0.558 (-1.750, 0.218)  --
```

The effect difference in the ith row and jth column is calculated as the effect of treatment i minus that of treatment j. To statistically test the difference between two treatments, users may check whether 0 is within the corresponding 95% CI for the effect difference. For instance, from the output, the 95% CI for the effect difference between treatments 1 and 2 is $(-2.860, -0.762)$, which does not contain 0. Therefore, treatments 1 and 2 differ significantly. The posterior density plot (Figure 3b) for treatment-specific effects is obtained by specifying `postdens = TRUE`. From the density plot, the overlap region of densities for treatments 1 and 2 is fairly small, and this supports the above conclusion.

Functions `nma.ab.py()` *and* `nma.ab.followup()` *for count datasets*

For models (3) and (4), treatment effects can be related to the follow-up times of participants. Some studies report the total exposure times in person-years (e.g., Hooper *et al.* 2000) for each treatment group, and some report the mean follow-up time for each study (e.g., Psaty *et al.* 2003; Elliott and Meyer 2007). The functions `nma.ab.py()` and `nma.ab.followup()` can be used for these two types of datasets, corresponding to models (3) and (4), respectively. In these two functions, the argument `param` can include `"rate"` (treatment-specific rate), `"lograte"` (log rate), `"ratio"` (rate ratio), `"logratio"` (log rate ratio), and `"rank.prob"` (treatment rank probabilities). Since the two functions are similar, this article focuses on using the dataset `diabetes` to illustrate the usage and output of `nma.ab.followup()`. The function is called as follows:

```
R> data("diabetes", package = "pcnetmeta")
R> set.seed(12345)
R> diabetes.out <- nma.ab.followup(s.id, t.id, r, n, folup, data = diabetes,
```

```
+    trtname = c("Diuretic", "Placebo", "BB", "CCB", "ACEI", "ARB"),
+    model = "het_cor", digits = 3, n.adapt = 10000, n.iter = 200000,
+    n.thin = 2, conv.diag = TRUE, trace = "lograte", postdens = TRUE)
```

Log rate ratio (the treatment in a row compared to that in a column) for each pair of treatments can be displayed as follows:

```
R> diabetes.out$LogRateRatio$Median_CI
```

```
         Diuretic                Placebo                 BB
Diuretic --                      0.109 (-0.654, 0.819)   0.096 (-0.559, 0.835)
Placebo  -0.109 (-0.819, 0.654) --                      -0.016 (-0.625, 0.754)
BB       -0.096 (-0.835, 0.559) 0.016 (-0.754, 0.625)   --
CCB      -0.169 (-0.896, 0.434) -0.055 (-0.822, 0.530) -0.075 (-0.600, 0.442)
ACEI     -0.263 (-0.958, 0.392) -0.157 (-0.781, 0.405) -0.167 (-0.716, 0.429)
ARB      -0.324 (-1.130, 0.560) -0.212 (-1.020, 0.603) -0.228 (-0.917, 0.637)
         CCB                    ACEI                    ARB
Diuretic 0.169 (-0.434, 0.896)  0.263 (-0.392, 0.958)  0.324 (-0.560, 1.130)
Placebo  0.055 (-0.530, 0.822)  0.157 (-0.405, 0.781)  0.212 (-0.603, 1.020)
BB       0.075 (-0.442, 0.600)  0.167 (-0.429, 0.716)  0.228 (-0.637, 0.917)
CCB      --                     0.093 (-0.510, 0.641)  0.156 (-0.637, 0.760)
ACEI     -0.093 (-0.641, 0.510) --                     0.062 (-0.793, 0.774)
ARB      -0.156 (-0.760, 0.637) -0.062 (-0.774, 0.793) --
```

From the output, all of the 95% CIs contain 0, so the difference between any pair of treatments is not significant. The posterior density plot of the log rates is shown in Figure 3c, which also indicates that treatment-specific density curves do not differ much.

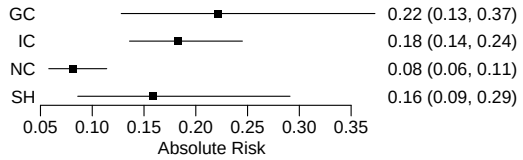
3.4. Plotting 95% credible intervals

When presenting network meta-analysis results, it is helpful to report the 95% CIs for effect sizes of interest. The package **pcnetmeta** provides functions `absolute.plot()` and `contrast.plot()` to draw the 95% CIs for treatment-specific and relative effect sizes, respectively. Users can simply call these functions on objects obtained from `nma.ab.bin()`, `nma.ab.cont()`, `nma.ab.py()`, and `nma.ab.followup()`. Here, we use the three objects obtained is the previous sections as examples to generate treatment-specific 95% CI plots:

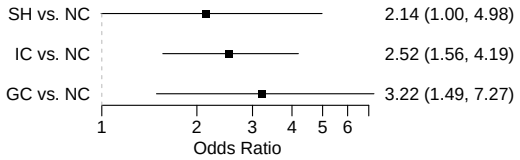
```
R> absolute.plot(smoke.out, width = 5, height = 1.5)
R> absolute.plot(parkinson.out, width = 5, height = 1.5)
R> absolute.plot(diabetes.out, width = 8, height = 2.5)
```

The generated plots are shown in the left panels of Figure 4. Contrast plots showing comparisons to a reference treatment can be generated by the following code:

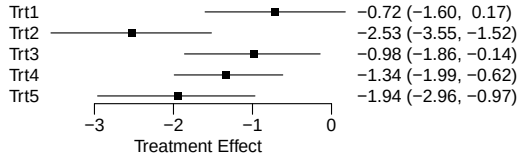
```
R> contrast.plot(smoke.out, reference = "NC", width = 5, height = 1.5)
R> contrast.plot(parkinson.out, reference = "Trt1", width = 5, height = 1.5)
R> contrast.plot(diabetes.out, reference = "Placebo", width = 8,
+    height = 2.5)
```



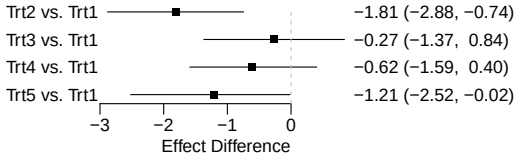
(a) Absolute risk plot for the smoking cessation dataset



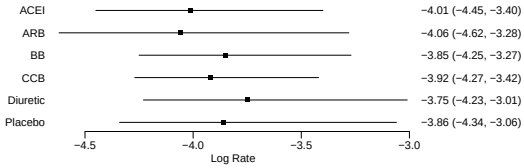
(b) Contrast plot for the smoking cessation dataset



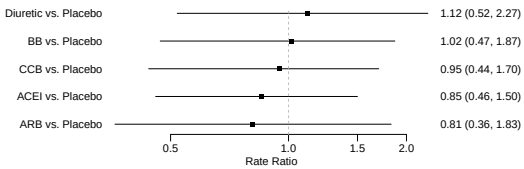
(c) Absolute treatment effect plot for the Parkinson's disease dataset



(d) Contrast plot for the Parkinson's disease dataset



(e) Absolute log rate plot for the diabetes dataset



(f) Contrast plot for the diabetes dataset

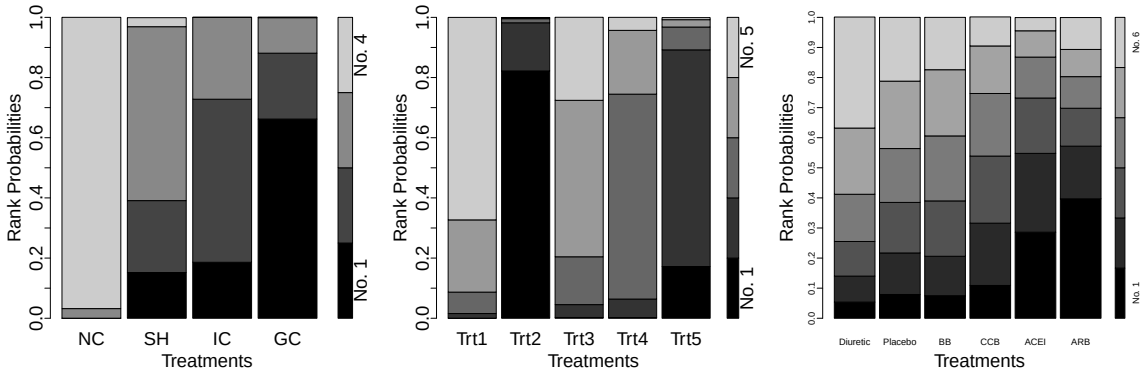
Figure 4: The left three panels show the plots for treatment-specific (absolute) effects generated by the function `absolute.plot()`; the right three panels show the plots for relative effects generated by the function `contrast.plot()`.

The argument `reference` specifies the reference treatment to be compared against. The right panels of Figure 4 display the contrast plots.

Figure 4a shows the treatment-specific 95% CI plot for the smoking cessation data. Clearly, the 95% CIs of IC and GC do not overlap with the 95% CI of NC. This indicates significant differences between IC vs. NC and GC vs. NC. This can be confirmed by Figure 4b: The 95% CIs of ORs comparing IC vs. NC and GC vs. NC do not intersect with the vertical line at OR = 1. Figure 4c is the treatment-specific 95% CI plot for the Parkinson's disease data. Note that the 95% CIs for treatments 1 and 2 overlap only in a tiny region. Correspondingly, the 95% CI for the effect difference between the two treatments in Figure 4d does not intersect with the vertical line at 0. Finally, Figures 4e and 4f show the 95% CI plots for treatment-specific and relative effects for the diabetes dataset. All of the treatment-specific 95% CIs have a large overlap with other CIs. The 95% CIs of the rate ratios compared with placebo intersect with the vertical line at 1; therefore, the active treatments do not differ significantly from placebo.

3.5. Plotting treatment rank probabilities

Function `rank.prob()` is used to graph the probabilities of each treatment having each of the different possible ranks among the treatments. Users can call this function for the objects



(a) Smoking cessation (b) Parkinson's disease (c) Diabetes

Figure 5: Plots of treatment rank probabilities generated by the function `rank.prob()`.

`smoke.out`, `parkinson.out`, and `diabetes.out` as follows:

```
R> rank.prob(smoke.out, cex.axis = 2, cex.lab = 2)
R> rank.prob(parkinson.out, cex.axis = 2, cex.lab = 2)
R> rank.prob(diabetes.out, cex.axis = 1, cex.lab = 2)
```

Figure 5 shows the plots of treatment rank probabilities. In the plots, each vertical bar represents probabilities that a specific treatment has different possible ranks. A darker area indicates the probability of having a higher rank, thus the black areas show the probabilities of having the best treatment. Therefore, from Figures 5a and 5b, treatments GC and Trt2 have much higher probabilities of being the best treatment, compared with other treatments in their respective studies. Figure 5c shows the rank probabilities plot for the dataset `diabetes`. Treatment ARB has the highest probability of being the best treatment, although the probability for treatment ACEI is close to the highest probability, so ARB and ACEI do not differ much.

4. Discussion

This article presents an overview of the R package **pcnetmeta**. Arm-based models are introduced to demonstrate the underlying methods of the functions. Practical usage of various functions is illustrated with examples of real network meta-analyses. Also, the package provides several plots for interpretation of network meta-analysis outputs.

MCMC convergence diagnostics have been extensively discussed in the literature (Cowles and Carlin 1996; Kass, Carlin, Gelman, and Neal 1998). The PSRFs and trace plots provided by the package **pcnetmeta** are used to examine whether the MCMC chains are drawn from stationary distributions; however, additional techniques are required to determine the effective sample size for adequate convergence (Robert and Casella 2004, p. 500). By specifying the argument `mcmc.samples = TRUE` in `nma.ab.bin()`, `nma.ab.cont()`, `nma.ab.followup()`, and `nma.ab.py()`, the MCMC posterior samples are saved in the output objects. Functions in other packages developed for MCMC convergence and sample-size adequacy, such as the R package **mcmcse** (Flegal and Hughes 2012), can be called for these posterior samples.

The current version of **pcnetmeta** does not detect inconsistency in arm-based network meta-analysis. Future work would add functions for network consistency assessment (Zhao, Hodges, Ma, Jiang, and Carlin 2016). Moreover, both the contrast-based and arm-based methods can be extended to handle individual patient data (IPD; Jansen 2012; Saramago, Chuang, and Soares 2014; Hong, Fu, Price, and Carlin 2015; Veroniki, Soobiah, Tricco, Elliott, and Straus 2015); a future update of the package may include functions for IPD.

Acknowledgments

LL and JZ were supported in part by the NIAID AI103012. HC was supported in part by the NIAID AI103012, NIDCR R03DE024750, NLM R21LM012197, NCI P30CA077598, NIMHD U54MD008620, and NIDDK U01DK106786.

References

Barnard J, McCulloch R, Meng XL (2000). "Modeling Covariance Matrices in Terms of Standard Deviations and Correlations, with Application to Shrinkage." *Statistica Sinica*, **10**(4), 1281–1312.

Borenstein M, Hedges LV, Higgins JPT, Rothstein H (2005). *Comprehensive Meta-Analysis, Version 2*. Biostat, Englewood. URL http://www.meta-analysis.com/.

Chu H, Nie L, Chen Y, Huang Y, Sun W (2012). "Bivariate Random Effects Models for Meta-Analysis of Comparative Studies With Binary Outcomes: Methods for the Absolute Risk Difference and Relative Risk." *Statistical Methods in Medical Research*, **21**(6), 621–633. doi:10.1177/0962280210393712.

Cipriani A, Furukawa TA, Salanti G, Geddes JR, Higgins JPT, Churchill R, Watanabe N, Nakagawa A, Omori IM, McGuire H, Tansella M, Barbui C (2009). "Comparative Efficacy and Acceptability of 12 New-Generation Antidepressants: A Multiple-Treatments Meta-Analysis." *The Lancet*, **373**(9665), 746–758. doi:10.1016/s0140-6736(09)60046-5.

Cooper H, Hedges LV, Valentine JC (2009). *The Handbook of Research Synthesis and Meta-Analysis*. 2nd edition. Russell Sage Foundation, New York.

Cowles MK, Carlin BP (1996). "Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review." *Journal of the American Statistical Association*, **91**(434), 883–904. doi:10.1080/01621459.1996.10476956.

DerSimonian R, Laird N (1986). "Meta-Analysis in Clinical Trials." *Controlled Clinical Trials*, **7**(3), 177–188. doi:10.1016/0197-2456(86)90046-2.

Dias S, Ades AE (2016). "Absolute or Relative Effects? Arm-Based Synthesis of Trial Data." *Research Synthesis Methods*, **7**(1), 23–28. doi:10.1002/jrsm.1184.

Dias S, Sutton AJ, Ades AE, Welton NJ (2013a). "Evidence Synthesis for Decision Making 2: A Generalized Linear Modeling Framework for Pairwise and Network Meta-Analysis of Randomized Controlled Trials." *Medical Decision Making*, **33**(5), 607–617. doi:10.1177/0272989x12458724.

Dias S, Welton NJ, Sutton AJ, Ades AE (2013b). "Evidence Synthesis for Decision Making 5: The Baseline Natural History Model." *Medical Decision Making*, **33**(5), 657–670. doi:10.1177/0272989x13485155.

Elliott WJ, Meyer PM (2007). "Incident Diabetes in Clinical Trials of Antihypertensive Drugs: A Network Meta-Analysis." *The Lancet*, **369**(9557), 201–207. doi:10.1016/s0140-6736(07)60108-1.

Flegal JM, Hughes J (2012). ***mcmcse**: Monte Carlo Standard Errors for MCMC*. R package version 1.0-1, URL https://CRAN.R-project.org/package=mcmcse.

Gelman A (2006). "Prior Distributions for Variance Parameters in Hierarchical Models (Comment on Article by Browne and Draper)." *Bayesian Analysis*, **1**(3), 515–534. doi:10.1214/06-ba117a.

Gelman A, Hill J (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, New York.

Gelman A, Rubin DB (1992). "Inference from Iterative Simulation Using Multiple Sequences." *Statistical Science*, **7**(4), 457–472.

Hasselblad V (1998). "Meta-Analysis of Multitreatment Studies." *Medical Decision Making*, **18**(1), 37–43. doi:10.1177/0272989x9801800110.

Hedges LV, Olkin I (1985). *Statistical Method for Meta-Analysis*. Academic Press, Orlando.

Higgins JPT, Green S (2008). *Cochrane Handbook for Systematic Reviews of Interventions*. John Wiley & Sons. doi:10.1002/9780470712184.fmatter.

Hong H, Chu H, Zhang J, Carlin BP (2016a). "A Bayesian Missing Data Framework for Generalized Multiple Outcome Mixed Treatment Comparisons." *Research Synthesis Methods*, **7**(1), 6–22. doi:10.1002/jrsm.1153.

Hong H, Chu H, Zhang J, Carlin BP (2016b). "Rejoinder to the Discussion of 'A Bayesian Missing Data Framework for Generalized Multiple Outcome Mixed Treatment Comparisons,' by S. Dias and A.E. Ades." *Research Synthesis Methods*, **7**(1), 29–33. doi:10.1002/jrsm.1186.

Hong H, Fu H, Price KL, Carlin BP (2015). "Incorporation of Individual-Patient Data in Network Meta-Analysis for Multiple Continuous Endpoints, with Application to Diabetes Treatment." *Statistics in Medicine*, **34**(20), 2794–2819. doi:10.1002/sim.6519.

Hooper L, Summerbell CD, Higgins JPT, Thompson RL, Clements G, Capps N, Davey Smith G, Riemersma R, Ebrahim S (2000). "Reduced or Modified Dietary Fat for Preventing Cardiovascular Disease." *Cochrane Database of Systematic Reviews*. doi:10.1002/14651858.cd002137.

Hunter JE, Schmidt FL (1996). "Cumulative Research Knowledge and Social Policy Formulation: The Critical Role of Meta-Analysis." *Psychology, Public Policy, and Law*, **2**(2), 324–347. doi:10.1037/1076-8971.2.2.324.

Jansen JP (2012). "Network Meta-Analysis of Individual and Aggregate Level Data." *Research Synthesis Methods*, **3**(2), 177–190. doi:10.1002/jrsm.1048.

Kasiske BL, Lakatua JDA, Ma JZ, Louis TA (1998). "A Meta-Analysis of the Effects of Dietary Protein Restriction on the Rate of Decline in Renal Function." *American Journal of Kidney Diseases*, **31**(6), 954–961. doi:10.1053/ajkd.1998.v31.pm9631839.

Kass RE, Carlin BP, Gelman A, Neal RM (1998). "Markov Chain Monte Carlo in Practice: A Roundtable Discussion." *The American Statistician*, **52**(2), 93–100. doi:10.1080/00031305.1998.10480547.

Lin L, Chu H, Hodges JS (2016). "Sensitivity to Excluding Treatments in Network Meta-Analysis." *Epidemiology*, **27**(4), 562–569. doi:10.1097/ede.0000000000000482.

Lin L, Zhang J, Chu H (2017). ***pcnetmeta**: Methods for Patient-Centered Network Meta-Analysis*. R package version 2.5, URL https://CRAN.R-project.org/package=pcnetmeta.

Lindholm LH, Carlberg B, Samuelsson O (2005). "Should β Blockers Remain First Choice in the Treatment of Primary Hypertension? A Meta-Analysis." *The Lancet*, **366**(9496), 1545–1553. doi:10.1016/s0140-6736(05)67573-3.

Lu G, Ades AE (2004). "Combination of Direct and Indirect Evidence in Mixed Treatment Comparisons." *Statistics in Medicine*, **23**(20), 3105–3124. doi:10.1002/sim.1875.

Lu G, Ades AE (2006). "Assessing Evidence Inconsistency in Mixed Treatment Comparisons." *Journal of the American Statistical Association*, **101**(474), 447–459. doi:10.1198/016214505000001302.

Lu G, Ades AE (2009). "Modeling Between-Trial Variance Structure in Mixed Treatment Comparisons." *Biostatistics*, **10**(4), 792–805. doi:10.1093/biostatistics/kxp032.

Lumley T (2002). "Network Meta-Analysis for Indirect Treatment Comparisons." *Statistics in Medicine*, **21**(16), 2313–2324. doi:10.1002/sim.1201.

Philbrook HT, Barrowman N, Garg AX (2007). "Imputing Variance Estimates Do Not Alter the Conclusions of a Meta-Analysis with Continuous Outcomes: A Case Study of Changes in Renal Function After Living Kidney Donation." *Journal of Clinical Epidemiology*, **60**(3), 228–240. doi:10.1016/j.jclinepi.2006.06.018.

Plummer M (2003). "**JAGS**: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling." In K Hornik, F Leisch, A Zeileis (eds.), *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*. Technische Universität Wien, Vienna, Austria. URL https://www.R-project.org/conferences/DSC-2003/Proceedings/Plummer.pdf.

Plummer M (2016). ***rjags**: Bayesian Graphical Models Using MCMC*. R package version 4-6, URL https://CRAN.R-project.org/package=rjags.

Plummer M, Best N, Cowles K, Vines K (2006). "**coda**: Convergence Diagnosis and Output Analysis for MCMC." *R News*, **6**(1), 7–11.

Psaty BM, Lumley T, Furberg CD, Schellenbaum G, Pahor M, Alderman MH, Weiss NS (2003). "Health Outcomes Associated with Various Antihypertensive Therapies Used as First-Line Agents: A Network Meta-Analysis." *JAMA*, **289**(19), 2534–2544. doi:10.1001/jama.289.19.2534.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Robert CP, Casella G (2004). *Monte Carlo Statistical Methods*. 2nd edition. Springer-Verlag, New York.

Rosenberg MS, Adams DC, Gurevitch J (2000). ***MetaWin**: Statistical Software for Meta-Analysis*. Sinauer Associates, Sunderland. URL http://www.metawinsoft.com/.

Rücker G (2012). "Network Meta-Analysis, Electrical Networks and Graph Theory." *Research Synthesis Methods*, **3**(4), 312–324. doi:10.1002/jrsm.1058.

Rücker G, Schwarzer G, Krahn U (2015). ***netmeta**: Network Meta-Analysis with R*. R package version 0.7-0, URL https://CRAN.R-project.org/package=netmeta.

Salanti G, Ades AE, Ioannidis JPA (2011). "Graphical Methods and Numerical Summaries for Presenting Results from Multiple-Treatment Meta-Analysis: An Overview and Tutorial." *Journal of Clinical Epidemiology*, **64**(2), 163–171. doi:10.1016/j.jclinepi.2010.03.016.

Salanti G, Higgins JPT, Ades AE, Ioannidis JPA (2008). "Evaluation of Networks of Randomized Trials." *Statistical Methods in Medical Research*, **17**(3), 279–301. doi:10.1177/0962280207080643.

Saramago P, Chuang LH, Soares MO (2014). "Network Meta-Analysis of (Individual Patient) Time to Event Data Alongside (Aggregate) Count Data." *BMC Medical Research Methodology*, **14**(1), 105. doi:10.1186/1471-2288-14-105.

Schwarzer G (2015). ***meta**: Meta-Analysis with R*. R package version 4.1-0, URL https://CRAN.R-project.org/package=meta.

Senn S (2010). "Hans van Houwelingen and the Art of Summing Up." *Biometrical Journal*, **52**(1), 85–94. doi:10.1002/bimj.200900074.

Shuster JJ, Jones LS, Salmon DA (2007). "Fixed vs Random Effects Meta-Analysis in Rare Event Studies: The Rosiglitazone Link With Myocardial Infarction and Cardiac Death." *Statistics in Medicine*, **26**(24), 4375–4385. doi:10.1002/sim.3060.

Spiegelhalter D, Thomas A, Best N, Lunn D (2003). *WinBUGS User Manual, Version 1.4*. MRC Biostatistics Unit, Cambridge. URL http://www.mrc-bsu.cam.ac.uk/bugs/.

Spiegelhalter DJ, Best NG, Carlin BP, Van der Linde A (2002). "Bayesian Measures of Model Complexity and Fit." *Journal of the Royal Statistical Society B*, **64**(4), 583–639. doi:10.1111/1467-9868.00353.

StataCorp (2015). *Stata Statistical Software: Release 14*. StataCorp LP, College Station, TX. URL http://www.stata.com/.

Sutton AJ, Abrams KR, Jones DR, Jones DR, Sheldon TA, Song F (2000). *Methods for Meta-Analysis in Medical Research*. John Wiley & Sons, Chichester.

Thijs V, Lemmens R, Fieuws S (2008). "Network Meta-Analysis: Simultaneous Meta-Analysis of Common Antiplatelet Regimens After Transient Ischaemic Attack or Stroke." *European Heart Journal*, **29**(9), 1086–1092. doi:10.1093/eurheartj/ehn106.

Van Houwelingen HC, Zwinderman KH, Stijnen T (1993). "A Bivariate Approach to Meta-Analysis." *Statistics in Medicine*, **12**(24), 2273–2284. doi:10.1002/sim.4780122405.

van Valkenhoef G, Kuiper J (2015). ***gemtc**: Network Meta-Analysis Using Bayesian Methods*. R package version 0.7-1, URL https://CRAN.R-project.org/package=gemtc.

Veroniki AA, Soobiah C, Tricco AC, Elliott MJ, Straus SE (2015). "Methods and Characteristics of Published Network Meta-Analyses Using Individual Patient Data: Protocol for a Scoping Review." *BMJ Open*, **5**(4), e007103. doi:10.1136/bmjopen-2014-007103.

Viechtbauer W (2014). ***metafor**: Meta-Analysis Package for R*. R package version 1.9-5, URL https://CRAN.R-project.org/package=metafor.

Wei Y, Higgins JPT (2013). "Bayesian Multivariate Meta-Analysis with Multiple Outcomes." *Statistics in Medicine*, **32**(17), 2911–2934. doi:10.1002/sim.5745.

Welton NJ, Sutton AJ, Cooper NJ, Abrams KR, Ades AE (2012). *Evidence Synthesis for Decision Making in Healthcare*. John Wiley & Sons, Chichester. doi:10.1002/9781119942986.

White I (2017). "**network**: Stata Module to Perform Network Meta-Analysis." Statistical Software Components, Boston College Department of Economics. URL https://ideas.repec.org/c/boc/bocode/s458319.html.

White IR (2015). "Network Meta-Analysis." *The Stata Journal*, **15**(4), 951–985. URL http://www.stata-journal.com/article.html?article=st0410.

Zeger SL, Liang KY, Albert PS (1988). "Models for Longitudinal Data: A Generalized Estimating Equation Approach." *Biometrics*, **44**(4), 1049–1060. doi:10.2307/2531734.

Zhang J, Carlin BP, Neaton JD, Soon GG, Nie L, Kane R, Virnig BA, Chu H (2014). "Network Meta-Analysis of Randomized Clinical Trials: Reporting the Proper Summaries." *Clinical Trials*, **11**(2), 246–262. doi:10.1177/1740774513498322.

Zhang J, Chu H, Hong H, Neaton JD, Virnig BA, Carlin BP (2017). "Bayesian Hierarchical Models for Network Meta-Analysis Incorporating Nonignorable Missingness." *Statistical Methods in Medical Research*. doi:10.1177/0962280215596185. Forthcoming.

Zhang J, Fu H, Carlin BP (2015). "Detecting Outlying Trials in Network Meta-Analysis." *Statistics in Medicine*, **34**(19), 2695–2707. doi:10.1002/sim.6509.

Zhao H, Hodges JS, Ma H, Jiang Q, Carlin BP (2016). "Hierarchical Bayesian Approaches for Detecting Inconsistency in Network Meta-Analysis." *Statistics in Medicine*, **35**(20), 3524–3536. doi:10.1002/sim.6938.

Affiliation:

Lifeng Lin, James S. Hodges, Haitao Chu
Division of Biostatistics
School of Public Health
University of Minnesota
Minneapolis, MN 55455, United States of America
E-mail: linl@umn.edu, hodge003@umn.edu, chux0051@umn.edu

Jing Zhang
Department of Epidemiology and Biostatistics
School of Public Health
University of Maryland
College Park, MD 20740, United States of America
E-mail: jzhang86@umd.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
August 2017, Volume 80, Issue 5 *Submitted:* 2015-05-20
doi:10.18637/jss.v080.i05 *Accepted:* 2016-08-19