Automating network meta-analysis

Gert van Valkenhoef Guobing Lu Bert de Brock Hans Hillege
A. E. Ades Nicky J. Welton

June 25, 2012

Abstract

Mixed Treatment Comparison (MTC) (also called network meta-analysis) is an extension of traditional meta-analysis to allow the simultaneous pooling of data from clinical trials comparing more than two treatment options. Typically, MTCs are performed using general purpose Markov Chain Monte Carlo (MCMC) software such as WinBUGS, requiring a model and data to be specified using a specific syntax. It would be preferable if, for the most common cases, both could be derived from a well-structured data file that can be easily checked for errors. Automation is particularly valuable for simulation studies in which the large number of MTCs that have to be estimated may preclude manual model specification and analysis. Moreover, automated model generation raises issues that provide additional insight into the nature of MTC. We present a method for the automated generation of Bayesian homogeneous variance random effects consistency models, including the choice of basic parameters and trial baselines, priors, and starting values for the Markov chain(s). We validate our method against the results of five published MTCs. The method is implemented in freely available open source software. This means that performing an MTC no longer requires manually writing a statistical model. This reduces time and effort, and facilitates error checking of the data set.

1 Introduction

Meta-analysis refers to statistical methods that combine evidence from multiple clinical trials in order to derive a pooled estimate of the relative effect of treatments. Traditional meta-analysis (Hedges and Vevea, 1998; Normand, 1999) has focused on pair-wise comparisons of treatments based upon summary measures of relative effect as reported in the original trials. Mixed Treatment Comparison (MTC) is a recently developed method that allows the simultaneous comparison of more than two treatments (Lu and Ades, 2004; Salanti et al., 2008). MTCs allow the use of both direct and indirect evidence for comparisons. In this paper, we focus on the Bayesian approach to MTC, which also allows the straightforward calculation of the rank-probabilities of a set of alternative treatments.

Specifying a Bayesian MTC model involves writing a Directed Acyclic Graph (DAG) model for general purpose Markov Chain Monte Carlo (MCMC) software such as WinBUGS (Lunn et al., 2000) or JAGS (Plummer, 2009). In addition, prior distributions have to be specified for a number of the parameters and the data have to be supplied in a specific format. Together, the DAG, priors and data form a Bayesian Hierarchical Model (BHM). Moreover, due to the nature of MCMC estimation, over-dispersed starting values have to be chosen for a number of independent chains so that convergence can be assessed (Gelman and Rubin, 1992; Brooks and Gelman, 1998). Currently, there is no software that automatically generates MTC models, although there are some tools to aid in the process. For example, the UK National Institute for Health and Clinical Excellence (NICE) Decision Support Unit offers a technical support document (Dias et al., 2011) that includes example WinBUGS code that applies to a broad range of data sets. However, most of the decisions mentioned above still have to be made, and having a fixed default value in the example code may lead to misleading results if the user is not aware of the need to modify this value for the situation at hand. Moreover, the format in which data are presented to BUGS can be hard to read (especially for large data sets) and does not facilitate error checking. For example, treatments are referred to by number rather than a name and data can be presented either in

tabular format, in which it is difficult to keep track of which column corresponds to which variable, or in list format, in which one has to check that indices match between several lists.

The effort that is required to manually specify MTC models is for the most part unnecessary, and automated model generation would enable the analyst to focus on more interesting aspects of the problem. The input data can then be presented in a more structured format that facilitates error checking. Moreover, in some cases, the number of MTCs to be carried out may necessitate such an approach, for example in simulation based studies where each iteration requires estimating an MTC, or in decision support applications where many criteria need to be considered. To address this, we present how Bayesian MTC models can be generated automatically, an endeavor that also provides insight into the nature of MTC. Specifically, we show how homogeneous variance random effects consistency models for both dichotomous and continuous outcomes can be generated. To fully automate model generation, we show how to specify the DAG, priors that limit bias, and starting values that are unlikely to lead to misdiagnosing convergence. Generating the DAG for inconsistency models is discussed in van Valkenhoef et al. (2011).

Note that what we present in this paper is the automated generation of MTC models that can be run in WinBUGS or JAGS. The analyst is still expected to choose the run length of the MCMC simulation, check whether the posterior has converged (and increase the run length if needed), assess the model fit, and interpret the results. For application in simulation studies our methods can easily be combined with an automated convergence checking routine that extends the run length as needed. However, there is always the risk of erroneously concluding that convergence has been reached so, if feasible, convergence checking is best done by visual inspection of the relevant plots (Brooks and Gelman, 1998).

2 Background

In the Bayesian framework, an MTC is implemented as a BHM and estimated using MCMC simulation (Lu and Ades, 2006; Salanti et al., 2008). This section explains the structure of the BHM and the considerations that have to be made when estimating it through MCMC simulation. MTC models are an extension of a Bayesian formulation of pair-wise meta-analysis. For clarity, we initially limit the discussion to random-effects homogeneous variance consistency models for dichotomous variables (Lu and Ades, 2004, 2006; Salanti et al., 2008). The extension to continuous variables (Salanti et al., 2008) is discussed thereafter. We will assume that the outcome data are reported per arm, rather than as treatment contrasts against a common baseline.

2.1 Consistency models for dichotomous variables

For a dichotomous variable (the occurrence or non-occurrence of an event), for every clinical trial i, for each included treatment x, we have the sample size $n_{i,x}$, and the number of events $r_{i,x}$ that occurred. The events are assumed to arise from a binomial process with success probability $p_{i,x}$:

$$r_{i,x} \sim \text{Bin}(p_{i,x}, n_{i,x}) \ . \tag{1}$$

The success probability can be transformed to the log odds scale through the $\text{logit}(p) = \log(p/(1-p))$ function. On the log odds scale, relative effects are assumed additive and normally distributed, drastically simplifying the model. The inverse transformation, logit^{-1}, is used to define $p_{i,x}$ in terms of log odds scale random variables:

$$\text{logit}(p_{i,x}) = \mu_i + \delta_{i,b(i),x} \Leftrightarrow p_{i,x} = \text{logit}^{-1}(\mu_i + \delta_{i,b(i),x}) \ , \tag{2}$$

where $b(i)$ is the baseline arm chosen for i, μ_i is the effect of $b(i)$ in i and $\delta_{i,b(i),x}$ is the *random effect* of x relative to $b(i)$ in i. If $b(i) = x$, we set $\delta_{i,b(i),x} = 0$. Otherwise,

$$\delta_{i,b(i),x} \sim \mathcal{N}(d_{b(i),x}, \sigma^2) \ , \tag{3}$$

where $d_{b(i),x}$ is the *relative effect* of x compared to $b(i)$, the quantity of interest, and σ^2 is the *random effects variance*, a measure of the heterogeneity between trials. Because we assume σ to be the same for

all $d_{x,y}$, this is a *homogeneous variance* model. In such a model, the covariances between comparisons in multi-arm trials work out to $\sigma^2/2$ (Salanti et al., 2008):

$$\begin{pmatrix} \delta_{i,b(i),x} \\ \vdots \\ \delta_{i,b(i),z} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} d_{b(i),x} \\ \vdots \\ d_{b(i),z} \end{pmatrix}, \begin{pmatrix} \sigma^2 & \sigma^2/2 & \cdots \\ \sigma^2/2 & \ddots & \sigma^2/2 \\ \cdots & \sigma^2/2 & \sigma^2 \end{pmatrix} \right) \quad . \tag{4}$$

Finally, we assume that comparisons are *consistent* (or, more fundamentally, we assume that trials are exchangeable conditional on the true value of the hyper parameters $d_{\cdot,\cdot}$ and σ, and consistency is implied (Lu and Ades, 2009)). That is, if we compare x and y indirectly through z, the result will be consistent with the direct comparison:

$$d_{x,y} = d_{x,z} - d_{y,z} \quad . \tag{5}$$

The right hand side parameters are called the *basic parameters*, for which we estimate probability distributions. Any other relative effect can be calculated using the consistency assumption. Hence $d_{x,y}$, a *functional parameter*, is completely defined in terms of the basic parameters on the right hand side. There are ways to test whether consistency holds (Lumley, 2002; Lu and Ades, 2006; Dias et al., 2010; Lu et al., 2011), but these are beyond the scope of this paper.

[Figure 1 about here.]

The model as discussed above forms a DAG, as shown in Figure 1, with the $r_{i,x}$ nodes at the very bottom (these are *sinks*) and $n_{i,x}$, $d_{x,y}$, μ_i and σ at the top (*sources*). Now, because the model is Bayesian, we have to specify prior distributions for the source nodes (except $n_{i,x}$, which have a fixed value), that reflect our beliefs before seeing the data. In addition, to perform MCMC estimation of the BHM, we have to specify a starting point for each of the stochastic nodes. Stochastic nodes are nodes that have probability distributions such as (3) and unlike (2). The MCMC simulation then proceeds by making random jumps in the parameter space using a procedure that is *guaranteed* to converge (under certain regularity assumptions) on the posterior distribution in the limit of infinitely many iterations. Most MCMC software includes a *tuning* or *adaptive* phase (Spiegelhalter et al., 2003; Plummer, 2009; Graves, 2008), during which the sampling algorithms are optimized to ensure efficient exploration of the parameter space. While tuning, the MCMC simulation will not converge, so after this initial phase tuning is turned off to allow the model to converge. Then, we run the model only as long as is needed to accurately estimate the quantities of interest, discarding an initial sub-sequence of the samples called the *burn-in* period. In that case, we say approximate convergence has been reached.

[Figure 2 about here.]

To assess approximate convergence using the Brooks-Gelman-Rubin diagnostic (Brooks and Gelman, 1998), the model is run N_C times in parallel, with distinct starting points, each for $2N_I$ iterations. The starting points have to be over-dispersed relative to the target distribution for the assessment to be valid. The results of the last N_I iterations of all N_C runs (called *chains*), are then analyzed and the within-chain and between-chain variance are compared to estimate the Potential Scale Reduction Factor (PSRF). A PSRF close to 1 indicates approximate convergence has been reached. The first N_I iterations are called burn-in iterations, and are discarded. To conclude that approximate convergence has been reached in MTC, all of the parameters of interest (source nodes) should have a PSRF below a certain threshold α, and visual inspection of plots of the PSRF and time series should not contradict this conclusion. If this is not the case, the simulation phase should be extended. Although it has been suggested that values below 1.2 are acceptable (Brooks and Gelman, 1998), we suggest to set $1 < \alpha \leq 1.05$ to be conservative. For the type of model discussed here, that condition is usually achieved reasonably quickly (i.e. within 50,000 iterations), and so we can afford to be conservative. The full process of estimating an MTC is shown in Figure 2. Note that there are valid alternative work flows, that will not be described here.

2.2 Continuous variables

When considering continuous outcomes, the Equations 1 and 2 are replaced by a normal likelihood (Salanti et al., 2008). Now, for a trial i and treatment x we take the sample mean $m_{i,x}$, sample standard deviation

$s_{i,x}$, and sample size $n_{i,x}$:

$$m_{i,x} \sim \mathcal{N}(\mu_i + \delta_{i,b(i),x}, s^2_{i,x}/n_{i,x}) \ . \tag{6}$$

Where, again, we model the observed effect in each arm in terms of a baseline effect μ_i for the trial and a random effect $\delta_{i,b(i),x}$. This model requires that all studies have measured the outcome on comparable scales, or that they have been transformed onto a common scale, for example the standardized mean difference.

2.3 Maximum likelihood estimators in single trials

Now, we briefly review the maximum likelihood estimators for both types of data, as they will be needed later on. First, define estimators $\hat{\theta}$ for the log odds and $\hat{v}^2$ for its variance:

$$\hat{\theta}_{i,x} = \text{logit}(\frac{r'_{i,x}}{n'_{i,x}}) \ ; \ \ \hat{v}^2_{i,x} = \frac{1}{r'_{i,x}} + \frac{1}{n'_{i,x} - r'_{i,x}} \ ; \tag{7}$$

$$r'_{i,x} = r_{i,x} + \frac{1}{2} \ ; \ \ n'_{i,x} = n_{i,x} + 1 \ ,$$

where $r'_{i,x}$ and $n'_{i,x}$ are corrected to ensure ratios are always defined. Usually the correction is only applied when the uncorrected ratio is undefined but, since we don't use these estimators for inference, the difference is not relevant. For continuous outcomes the estimators are:

$$\hat{\theta}_{i,x} = m_{i,x} \ ; \ \ \hat{v}^2_{i,x} = \frac{s^2_{i,x}}{n_{i,x}} \ . \tag{8}$$

For the relative effect of y when compared to x the estimators (for dichotomous or continuous data) are (DerSimonian and Laird, 1986):

$$\hat{\delta}_{i,x,y} = \hat{\theta}_{i,y} - \hat{\theta}_{i,x} \ ; \ \ \hat{s}^2_{i,x,y} = \hat{v}^2_{i,y} + \hat{v}^2_{i,x} \ . \tag{9}$$

3 Methods

To automatically generate MTC models, the BHM has to be specified, consisting of the DAG, prior distributions and data. Moreover, over-dispersed starting values have to be chosen for N_C independent chains. In the following, we show how to specify the DAG for consistency models and how priors that limit bias, and starting values that are unlikely to lead to misdiagnosing convergence can be chosen to fully automate MTC model generation. Finally, we provide a worked example of the methods.

3.1 Generating the model structure

Generating the DAG for inconsistency models has previously been discussed (van Valkenhoef et al., 2011), and involves finding the formulation that maximizes the number of potential inconsistencies that can be estimated, the Inconsistency Degrees of Freedom (ICDF). However, the problem is considerably simpler for consistency models, and rather than being forced to search for a DAG that maximizes the ICDF, we can try to specify one that has an easily understandable structure. Arguably, the most easily understood structure is the one where we parameterize each treatment effect relative to a common baseline (e.g. placebo). In such a model, the basic parameters form a star shaped graph, and all functional parameters are a linear combination of just two basic ones. The algorithm proposed for inconsistency models (van Valkenhoef et al., 2011) does exactly the opposite: due to the specific search algorithm it uses to find a suitable parametrization, it is likely to end up with a line graph, or something close to it.

It has been shown that for inconsistency models, an incorrect choice of baseline treatments for the individual studies may result in some of the parameters being under-constrained (Lu and Ades, 2006; van Valkenhoef et al., 2011). In Appendix A, we show that this problem cannot occur for consistency models. Thus, regardless of how the basic parameters and study baselines are chosen, the model is always well defined. Therefore, we are completely free to choose a parametrization that is easily understandable. In fact, in a consistency model it is not even necessary for the basic parameters to have been measured directly, and thus we could choose a star shaped graph for the basic parameters. However, the method

we propose below to choose starting values depends on the basic parameters being measured directly as a simplifying assumption, so we restrict the basic parameters to the directly measured comparisons. Note that after the samples for the basic parameters have been obtained, it is straightforward to express the result relative to any of the included treatments by post-processing the samples, so this restriction does not hamper inference.

Given that the basic parameters are restricted to directly measured comparisons, and that they must form a spanning tree of the evidence graph, we would like for the linear equations that define the functional parameters to involve as few basic parameters as possible. The diameter of a spanning tree is the length of the longest of the paths between its vertices. Thus, the minimum diameter spanning tree provides the parametrization that minimizes the length of the longest equation. The minimum diameter spanning tree can be found efficiently (Hassin and Tamir, 1995). This provides an automated way of specifying the basic parameters even if the evidence structure does not have a common comparator. For the study baselines, we just choose (in each study) the treatment that has the most connections in the spanning tree of the basic parameters, again with the aim of minimizing the length of equations. If there is more than one such treatment we may choose an arbitrary one, e.g. alphabetical ordering of the treatments can be used to break ties.

3.2 Choosing priors

Prior distributions have to be specified for all the basic relative effect parameters $d_{x,y}$, the baseline effects μ_i and the random effects variance σ^2. For the variance parameters, either a uniform or an inverse-Gamma prior may be chosen (Lu and Ades, 2004). However, the uniform prior is more commonly used (e.g. Lu and Ades, 2006):

$$\sigma \sim \mathcal{U}(0, u) \ ,$$

where u has to be appropriately chosen so that it is larger than the standard deviation that may exist. But how to do this has not been discussed in the literature. Normally, the analyst should provide this value, but doing so is often difficult: what would constitute an unreasonably large deviation in the log odds ratio for treatment response? No single value of u will be appropriate for every situation, and thus a simple heuristic that over-estimates the measurement scales based on minimal information is a reasonable compromise. It should be noted that in most cases the posterior is not sensitive to the variance prior, as long as it is sufficiently wide (Lu and Ades, 2004). A simple technique is to calculate the maximum likelihood estimators $\hat{\delta}_{i,x,y}$ (Equation 9) for all possible combinations of i, x, and y and find the maximal one, δ^+. Setting $u = \delta^+$ guarantees $u > \sigma$ and this choice of u does not depend on the chosen DAG, so that alternative DAGs have the same prior.

For the relative effect parameters and the baseline effects, independent identical normal priors are usually specified (Lu and Ades, 2004, 2006; Salanti et al., 2008), i.e.:

$$d_{x,y}, \mu_i \sim \mathcal{N}(\nu, \eta^2) \ ,$$

where $\nu = 0$ and $\eta^2 = 1000$ is often used for dichotomous outcomes (Lu and Ades, 2004, 2006; Salanti et al., 2008). The large variance is chosen so that the data will dominate the prior information in the final result (i.e. we specify a *vague* prior). Again, what constitutes a large variance depends on the scale of the data, and therefore we set $\eta^2 = (ku)^2$. Choosing 0 as the prior mean for $d_{x,y}$ is done so any bias due to the prior is conservative: the bias is in the direction of no effect. For μ_i a prior mean of 0 is more arbitrary. For dichotomous data, it introduces a slight bias towards $p_{i,b(i)} = 0.5$, which means values near 0 and 1 will be (slightly) shrunk towards 0.5. For continuous data, the bias is towards 0. Thus, it is important that η^2 is sufficiently large to minimize this bias.

To choose a value for the scaling factor k, we looked at a number of studies in the literature (Lu and Ades, 2004; Welton et al., 2009; Dias et al., 2010). In those studies, η^2 was either 10^3 or 10^4, and the upper bound for the variance prior ranged from 2 to 50. If the η^2 were generated using a scaling factor k, the value of k would range from 3.16 to 15.8. Somewhat arbitrarily, we choose $k = 15$ as the default, which is in the upper range of values seen in published analyses, since the goal is to ensure the variance is large enough, regardless of the measurement scale. This choice is validated in Section 5 by comparing the posteriors we obtain against those reported in the original analyses.

3.3 Choosing starting values

Starting values have to be over-dispersed relative to the target density in order to ensure complete exploration of the parameter space and assess convergence. They have to be chosen for all stochastic nodes that are not bound to data (i.e., $\delta_{i,x,y}$, μ_i, $d_{x,y}$ and σ). Note that for the $d_{x,y}$ this includes only the basic parameters and not the functional ones. Starting values can be generated by creating an over-dispersed approximate distribution from which they are drawn (Gelman and Rubin, 1992). Our strategy is to use analytic techniques for pair-wise comparisons (DerSimonian and Laird, 1986) to get a maximum likelihood estimate ξ and standard error τ for each node, then sample that node's starting values from $\mathcal{N}(\xi, c\tau^2)$ with $c \gg 1$. To estimate μ_i, take $\xi = \hat{\theta}_{i,b(i)}$ and $\tau^2 = \hat{v}^2_{i,b(i)}$ (Equations 7, 8). For $\delta_{i,x,y}$, $\xi = \hat{\theta}_{i,y} - \hat{\theta}_{i,x}$ and $\tau^2 = \hat{v}^2_{i,y} + \hat{v}^2_{i,x}$ (Equation 9). For the $d_{x,y}$, we do a pair-wise random effects pooling (DerSimonian and Laird, 1986) of all trials that include x and y to find ξ and τ^2. Note that in our generated models, the basic parameters are always directly measured (Section 3.1), and that we use a correction to account for zero cells (Section 2.3). When only one trial measures the basic parameter in question, the pooling method will estimate the between-trials error to be equal to the standard error of the mean of that one trial. Starting values for σ can be obtained by sampling from its prior distribution.

[Figure 3 about here.]

It is important that the starting values are sufficiently over-dispersed so that the PSRF is not close to 1 before approximate convergence has been reached, in which case it will increase later on when an unexplored part of the parameter space is found (see Figure 3). We also don't want c to be too large, as this would slow down the exploration of the parameter space significantly. However, we need not be overly concerned with this issue, as the models considered here tend to converge quickly, even if the starting values are not ideal. The value c is configurable in our implementation, but we set $c = 2.5$ by default.

3.4 Worked example

[Table 1 about here.]

[Figure 4 about here.]

We now illustrate the methods with a worked example of 5 treatments for Parkinson's disease (Dias et al., 2011). There is data on mean off-time reduction from 7 trials (see Table 1). The evidence network is shown in Figure 4. First, to determine the basic parameters, we find the minimum diameter spanning tree (the solid lines in Figure 4). In this case, the basic parameters are $d_{D,A}$, $d_{D,B}$, $d_{D,C}$, and $d_{D,E}$. This is the only paremeterization in which all functional parameters can be expressed in terms of at most two basic ones (i.e. treatment D is a common comparator). Then, we choose the study baselines by identifying the treatment with the most connections in the spanning tree (basic parameters). In this case, treatment D is connected to all four other treatments, so any trial that includes D will have D as the baseline treatment. For example, study 3 would be parameterized as:

$$\begin{pmatrix} \delta_{3,D,A} \\ \delta_{3,D,B} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} d_{D,A} \\ d_{D,B} \end{pmatrix}, \begin{pmatrix} \sigma^2 & \sigma^2/2 \\ \sigma^2/2 & \sigma^2 \end{pmatrix} \right) \ .$$

For the remaining trials, we choose the alphabetically first treatment as the baseline. For example, trial 1 would be parameterized as:

$$\delta_{1,A,C} \sim \mathcal{N}(-d_{D,A} + d_{D,C}, \sigma^2) \ .$$

Then, to choose the prior distributions, we calculate a maximum likelihood estimate for *all* relative effects in all studies, including $\hat{\delta}_{1,A,C} = -0.31$, $\hat{\delta}_{1,C,A} = 0.31$, etc. For three-arm studies, there are six such relative effects. In this case, the maximum is 2.3, so we set $u = 2.3$ and $\eta^2 = (15 \cdot 2.3)^2 = 1.2 \cdot 10^3$, leading to the priors:

$$\sigma \sim U(0, 2.3) \ ; \ d_{x,y}, \mu_i \sim \mathcal{N}(0, 1.2 \cdot 10^3) \ .$$

Finally, we use maximum likelihood estimators to derive sampling distributions from which to draw starting values. For the baseline effect μ_1 in study 1, where A is the baseline, we draw starting values from

$$\mathcal{N}(\hat{\theta}_{1,A}, c \cdot \hat{v}^2_{1,A}) = \mathcal{N}(-1.22, 2.5 \cdot 3.7^2/54) \ .$$

For the relative effect $\delta_{1,A,C}$ we draw starting values from

$$\mathcal{N}(\hat{\theta}_{1,C} - \hat{\theta}_{1,A}, c \cdot (\hat{v}^2_{1,A} + \hat{v}^2_{1,C})) = \mathcal{N}(-0.31, 2.5 \cdot (0.25 + 0.19)) \ .$$

Finally, for the basic parameters we perform random effects pooling of the relevant studies. For example, to determine the distribution of starting values for $d_{D,C}$, we pool studies 4 and 5. Then, the distribution is $\mathcal{N}(-0.04, 0.45)$.

4 Implementation

The methods presented in this paper are implemented in GeMTC (`http://drugis.org/gemtc`), enabling generation of JAGS and BUGS models. The data are stored in an XML format, an example of which is shown in Figure 5.

[Figure 5 about here.]

A simple graphical user interface is provided to facilitate data entry and manipulation of data files, as well as model generation. The main screen (see Figure 6) enables the user to create, load, and save datasets in the GeMTC XML format. Each dataset is opened in a tab named after the dataset (‘`cipriani-efficacy.gemtc`’ in Figure 6). In the left panel there are two tabs where the user can add, edit, and remove treatments and studies. A treatment consists of a short identifier and an optional description. A study consists of an identifier and at least two included treatments. At the top of the right panel, the user chooses the type of data (dichotomous or continuous), and a description for the dataset. The rest of the right panel consists of the data table, where data for each arm can be input in the appropriate format. For dichotomous measurements, this is the number of events and the sample size; for continuous measurements, the mean, standard deviation, and sample size. In the current version of GeMTC (0.12.1), data have to be input manually, as copy-paste support and import from various formats are not yet available.

[Figure 6 about here.]

After the dataset is complete, the statistical model can be generated by clicking ‘Generate’ in the tool bar. If the studies do not form a connected evidence network, the user is advised of this and asked to correct the dataset. Otherwise, the user is presented with a settings dialog (Figure 7). Here, various parameters can be configured, such as the syntax type (BUGS or JAGS), the model type (only the consistency model is described in this paper), the scaling factor c for the initial values, and the number of chains, tuning iterations and simulation iterations for the MCMC simulation. Unfortunately, priors are currently not configurable from this screen, and the user will need to modify the generated BUGS or JAGS code to change the priors.

[Figure 7 about here.]

After configuring the parameters for model generation, the user can click ‘OK’ to generate the model. This opens a new window, which shows the various components of the model in separate tabs (see Figure 8). JAGS users can use the ‘save’ button to save these components to a series of files, after which the ‘`*.script`’ file can be used to run the model in JAGS (some BUGS versions also support this). BUGS users can copy-paste the contents of each tab to BUGS and run the model from the BUGS user interface.

[Figure 8 about here.]

Generating the BUGS or JAGS models entails taking the data, applying the methods discussed in the previous section to determine the basic parameters and study baselines, choose the priors and generate starting values, and then generating BUGS or JAGS syntax code from that abstract representation. A number of files have to be generated: the model code, a data file, an initial values file for each chain and a script file that specifies how the model should be run (this last file is not necessary for WinBUGS). To generate the model code, we use a template based on (Dias et al., 2011), which we generalized to

also apply to inconsistency models (though that is outside the scope of this paper). The template and an example of the resulting code are shown in Figure 9. The BUGS and JAGS model specification languages are not fully compatible, but we have been careful to ensure that the template is compatible with both. The data and initial values files are output in S-Plus/R format for both BUGS and JAGS, structured according to (Dias et al., 2011). For BUGS, care has to be taken with matrices, since it stores matrices in row-major order, whereas JAGS (like S-Plus and R) stores matrices in column-major order. Generating the script file is, again, a simple matter of variable substitution on a template. However, in this case, separate templates are required for JAGS and BUGS since their scripting languages are entirely different.

[Figure 9 about here.]

5 Results

In this section, we validate the approach presented above by reproducing a number of previously published analyses. The first MTC is a smoking cessation network comparing three forms of counseling with self-help consisting of 24 trials (Lu and Ades, 2006). The second MTC consists of 28 trials comparing 8 thrombolytic treatments after acute myocardial infarction (Lu and Ades, 2006). The third, a comparison of the efficacy of 12 second-generation anti-depressants, includes 111 clinical trials with 24,693 patients in total (Cipriani et al., 2009). These three MTCs synthesize dichotomous data on the log odds ratio scale. The remaining two MTCs were selected to evaluate our methods for continuous datasets. The fourth dataset concerns the effect of psychological interventions on coronary heart disease, and assessed eight outcomes, three of which were expressed as mean change from baseline: diastolic blood pressure, systolic blood pressure and total cholesterol (Welton et al., 2009). The fifth MTC compares 5 treatments for Parkinson's disease on mean off-time reduction using 7 trials (Dias et al., 2011), which was included because there is relatively little data, meaning that priors may have a greater influence on the result. The first two MTCs (smoking cessation and thrombolytic treatments) were also analyzed by Dias et al. (2010).

[Table 2 about here.]

In Table 2 we contrast the priors specified for the original analyses with those generated using the heuristics described in Section 3.2, using $k = 15$. As can be seen, the priors used in the literature vary widely, and the ones generated by our algorithm appear to be no less reasonable. In two cases (systolic blood pressure and total cholesterol) the generated priors are much narrower than those reported in the original paper, but the posterior is unaffected (see below).

[Figure 10 about here.]

Figure 10 shows the evidence graphs of the smoking cessation, thrombolytics, Parkinson and anti-depressants data sets. The basic parameters chosen by our algorithm are shown as solid lines, and the functional ones as dashed lines. The coronary heart disease data sets are not shown, as they were analyzed in a pair-wise fashion. In most cases, the algorithm finds a single comparator against which all other treatments are compared. In the thrombolytics data set, however, no single comparator exists and a different solution is identified: some treatments are parameterized relative to ASPAC and others relative to AtPA.

[Table 3 about here.]

[Table 4 about here.]

[Table 5 about here.]

[Table 6 about here.]

[Table 7 about here.]

Fixing $k = 15$ and $c = 2.5$, and using 4 chains with 20,000 tuning, 20,000 burn-in and 20,000 inference iterations, we reproduced the five published MTCs. The results are shown in Tables 3 through 7, where we show the posterior summaries reported in the original paper side by side with the results of our analysis. Convergence was assessed using the Brooks-Gelman-Rubin diagnostic and was adequate for all models. In all cases, the results of our generated models are very similar to those of the original analyses, and they are almost identical for the anti-depressants data (Table 5), as is to be expected since the priors should have very little influence on the posterior in such a dense dataset. There were no cases where our choice of prior led to truncation of the density for the variance parameter or to biased estimates for the relative effects, nor did we detect slow convergence or spuriously low PSRFs.

6 Discussion

In this paper, we described how to generate MTC consistency models fully automatically based solely on the data set. We showed that parametrization of consistency models is indeed easy, and thus that we can optimize for understandability of the model (this is a marked difference with inconsistency models). Moreover, we proposed heuristics to safely choose defaults for priors and starting values. The methods were validated against published MTCs and the generated models give nearly identical results, but at a significant reduction in effort for the analyst.

The methods have been implemented in GeMTC (`http://drugis.org/gemtc`), which provides a graphical interface to manipulate data sets and can generate MTC models for either JAGS or BUGS. The software can also generate inconsistency models (based on previous work), and is used in the ADDIS decision support system (`http://drugis.org/addis`). An R package is being worked on, and is currently available in experimental form (`http://drugis.org/gemtc`). All of this software has been released under an open source license, and the datasets referred to in this paper are distributed with the software.

The presented results, though encouraging, do not guarantee correct results for all problem instances. Specifically, priors are chosen heuristically, and can not in any sense be shown to be 'correct'. But if default values are to be given, we can at least try to ensure that they don't result in overly precise estimates. This is what the given heuristics do and arguably that is an improvement over template code that gives a fixed default value. The prior for the random effects variance σ may be problematic in sparse datasets, as it can lead to over-estimation of the variance. Thus, one should be careful when using the software with sparse data sets. Naturally, if real prior information is available, the defaults should be replaced. Moreover, human judgment is still needed to choose the appropriate run length, assess convergence, and interpret the results. For use in simulation studies where each iteration requires the estimation of one or more MTCs, our methods can be combined with automated convergence checking to run the MTCs fully unsupervised. Wrongly concluding that convergence is adequate is a real risk in that case, but running a larger number of chains with properly over-dispersed starting values can help to minimize that risk.

The software vastly simplifies the task of performing an MTC by automating the process of model specification. Compared to existing 'generic' code for MTCs, the software presented in this paper facilitates error-checking of the data set, specifies vague priors appropriate for the problem rather than giving a fixed default, and automatically specifies multiple chains with over-dispersed starting values. The `mvmeta` package for the Stata statistical software enables MTC in a frequentist framework (White, 2011). However, if there is no common comparator in the MTC this has to be handled by augmenting the data set with fictional arms with high variance, which is not very elegant and requires a decision as to what constitutes a sufficiently high variance. Moreover, one might prefer the Bayesian approach, and thus an automated solution for Bayesian MTC is still desirable.

The presented methods are limited to specific types of MTC model, and future work should address this limitation. First, we have assumed that arm-level outcome data is available, but this is often not the case. Continuous outcomes are often reported as relative effects compared to a control treatment, in which case the relative effects are correlated and these correlations have to be modelled in the likelihood function for multi-arm trials (Dias et al., 2011). There are various formats in which such data may be reported, and the correlations may or may not be reported. This gives rise to some challenges in the data modeling and user interface, and several new variants of the likelihood function. We showed that study baselines can be chosen arbitrarily, so whether data are arm-based or contrast-based should not affect the basic model structure for consistency models. However, contrast-based data will create additional

problems for inconsistency models.

Second, we did not discuss fixed effects models, but generating them entails a trivial modification of the template. A greater challenge lies in heterogeneous variance random effects models, as problems arise in sparse data sets, where some comparisons are informed by only one study. For those comparisons, the random effects variance can not be estimated, and the model generation code should detect these cases and potentially offer a solution. Moreover, for some datasets more specialized likelihoods are needed, such as for time-to-event data (Lu et al., 2007) and in some cases it may be necessary to adjust for covariates using a meta-regression model (Salanti et al., 2009). Future work should investigate how these types of model can be generated automatically.

Finally, it has not been discussed in the literature how node split models for the assessment of inconsistency (Dias et al., 2010) can be generated. Automatically generating these models would be useful, as for each MTC analysis there will be a relatively large number of node split models that have to be specified: potentially one model for each comparison present in the dataset.

Acknowledgements

This study was partly performed in the context of the Escher project (T6-202), a project of the Dutch Top Institute Pharma.

References

Brooks, S. P. and Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *J Comput Graph Stat*, 7(4):434–455. doi:10.1080/10618600.1998.10474787.

Cipriani, A., Furukawa, T. A., Salanti, G., Geddes, J. R., Higgins, J. P. T., Churchill, R., Watanabe, N., Nakagawa, A., Omori, I. M., McGuire, H., Tansella, M., and Barbui, C. (2009). Comparative efficacy and acceptability of 12 new-generation antidepressants: a multiple-treatments meta-analysis. *Lancet*, 373(9665):746 – 758. doi:10.1016/S0140-6736(09)60046-5.

DerSimonian, R. and Laird, N. (1986). Meta-analysis in clinical trials. *Controlled Clin Trials*, 7(3):177–188. doi:10.1016/0197-2456(86)90046-2.

Dias, S., Welton, N. J., Caldwell, D. M., and Ades, A. E. (2010). Checking consistency in mixed treatment comparison meta-analysis. *Stat Med*, 29(7-8, Sp. Iss. SI):932–944. doi:10.1002/sim.3767.

Dias, S., Welton, N. J., Sutton, A. J., and Ades, A. E. (2011). NICE DSU technical support document 2: A generalised linear modelling framework for pairwise and network meta-analysis of randomised controlled trials. Technical report. updated August 2011. Available from: http://www.nicedsu.org.uk/.

Fowler, M. (2003). *UML Distilled*. Addison-Wesley, 3rd edition.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Stat Sci*, 7(4):457–472. doi:10.1214/ss/1177011136.

Graves, T. L. (2008). *An Introduction to YADAS*. Available from: http://yadas.lanl.gov/.

Hassin, R. and Tamir, A. (1995). On the minimum diameter spanning tree problem. *Inform Process Let*, 53(2):109–111. doi:10.1016/0020-0190(94)00183-Y.

Hedges, L. V. and Vevea, J. L. (1998). Fixed- and random-effects models in meta-analysis. *Psychol Methods*, 3(4):486–504. doi:10.1037/1082-989X.3.4.486.

Lu, G. and Ades, A. (2009). Modeling between-trial variance structure in mixed treatment comparisons. *Biostatistics*, 10(4):792–805. doi:10.1093/biostatistics/kxp032.

Lu, G. and Ades, A. E. (2004). Combination of direct and indirect evidence in mixed treatment comparisons. *Stat Med*, 23(20):3105–3124. doi:10.1002/sim.1875.

Lu, G. and Ades, A. E. (2006). Assessing evidence inconsistency in mixed treatment comparisons. *J Am Stat Assoc*, 101(474):447–459. doi:10.1198/016214505000001302.

Lu, G., Ades, A. E., Sutton, A. J., Cooper, N. J., Briggs, A. H., and Caldwell, D. M. (2007). Meta-analysis of mixed treatment comparisons at multiple follow-up times. *Stat Med*, 26(20):3681–3699. doi:10.1002/sim.2831.

Lu, G., Welton, N. J., Higgins, J. P. T., White, I. R., and Ades, A. E. (2011). Linear inference for mixed treatment comparison meta-analysis: A two-stage approach. *Res Synth Methods*, 2(1):43–60. doi:10.1002/jrsm.34.

Lumley, T. (2002). Network meta-analysis for indirect treatment comparisons. *Stat Med*, 21(16):2313–2324. doi:10.1002/sim.1201.

Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000). WinBUGS – a Bayesian modelling framework: concepts, structure, and extensibility. *Stat Comput*, 10(4):325–337. doi:10.1023/A:1008929526011.

Normand, S.-L. T. (1999). Meta-analysis: formulating, evaluating, combining, and reporting. *Stat Med*, 18(3):321–359. doi:10.1002/(SICI)1097-0258(19990215)18:3<321::AID-SIM28>3.0.CO;2-P.

Plummer, M. (2009). *JAGS Version 1.0.3 manual*. Available from: http://www-fis.iarc.fr/~martyn/software/jags.

Salanti, G., Higgins, J. P. T., Ades, A. E., and Ioannidis, J. P. A. (2008). Evaluation of networks of randomized trials. *Stat Methods Med Res*, 17(3):279–301. doi:10.1177/0962280207080643.

Salanti, G., Marinho, V., and Higgins, J. P. T. (2009). A case study of multiple-treatments meta-analysis demonstrates that covariates should be considered. *J Clin Epidemiol*, 62(8):857–864. doi:10.1016/j.jclinepi.2008.10.001.

Spiegelhalter, D., Thomas, A., Best, N., and Lunn, D. (2003). *WinBUGS User Manual, version 1.4*. Available from: http://www.mrc-bsu.cam.ac.uk/bugs.

van Valkenhoef, G., Tervonen, T., de Brock, B., and Hillege, H. (2011). Algorithmic parametrization of mixed treatment comparisons. *Stat Comput*. (in press). doi:10.1007/s11222-011-9281-9.

Welton, N. J., Caldwell, D. M., Adamopoulos, E., and Vedhara, K. (2009). Mixed Treatment Comparison Meta-Analysis of Complex Interventions: Psychological Interventions in Coronary Heart Disease. *Am J Epidemiol*, 169(9):1158–1169. doi:10.1093/aje/kwp014.

White, I. R. (2011). Multivariate random-effects meta-regression: Updates to mvmeta. *Stata J*, 11(2):255–270.

A Proof: choice of basic parameters and study baselines is arbitrary in consistency models

A parameter is under-constrained if either (a) it is never used to define any random effect (Equation 3) or (b) it always co-occurs with another variable. We show here that this problem cannot occur for consistency models.

Let $G_i = (T_i, E_i)$ be the (fully connected) evidence graph for trial i, and $G = \bigcup_i G_i = (T, E)$ the (possibly sparse) evidence graph of the treatment network. Now, if R is any spanning tree of G (determining the $|T| - 1$ basic parameters), and whatever $b(i) \in T_i$ we choose as the baseline in the individual trials, the basic parameters are fully constrained. To see this, note that a trial i is expressed as a star-shaped graph H_i in which all included treatments are compared to $b(i)$. If we take the union $H = \bigcup_i H_i = (T, E_H)$, this represents all comparisons that are explicitly expressed in the model. Since H is the union of spanning trees of the graphs that G is the union of, H is connected. Each contrast $\{x, y\} \in E_H$ generates a (unique, undirected, simple) path between x and y in R, denoted as

$\text{path}_R(\{x, y\})$. Because H is connected, the set P of these paths visits all treatments T and thus uses all the basic parameters at least once. Now, if $\{x, y\}$ and $\{y, z\}$ are basic parameters, there is at least one path $p \in P$ that contains one, but not the other: for $\{y, w\} \in H$, $\text{path}_R(\{y, w\})$ contains $\{x, y\}$ or $\{y, z\}$, but not both. Finally, if any two treatment contrasts co-occur, the entire path in R between them must co-occur. However, for any adjacent contrasts we can find a $p \in P$ so that they don't co-occur, so for any two basic parameters we can find a $p \in P$ so that they don't co-occur. In summary: each basic parameter is used at least once and no two basic parameters always co-occur.

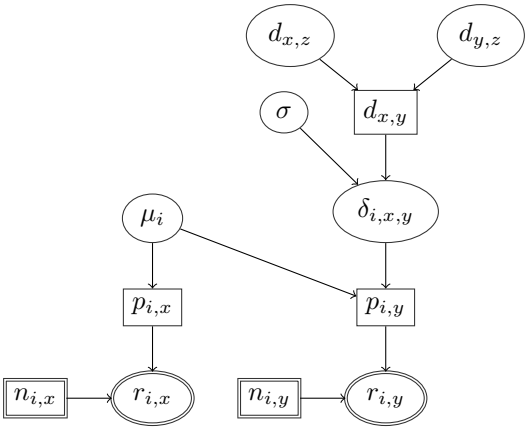


Figure 1: The basic structure of an MTC model is a DAG, here shown graphically for a subset of a full MTC model. Elliptical nodes represent density functions, whereas rectangular nodes represent deterministic functions. Nodes with a double border have data associated with them. Here, one study (study i) that includes two treatments (x and y) in a model with three treatments (x, y and z) is shown. The parameter $d_{x,y}$ is a functional parameter, defined as a deterministic function of the basic parameters $d_{x,z}$ and $d_{y,z}$.

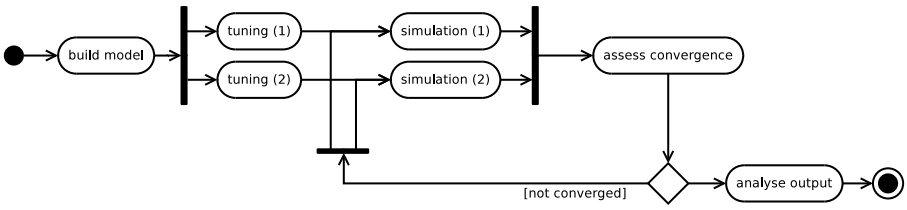
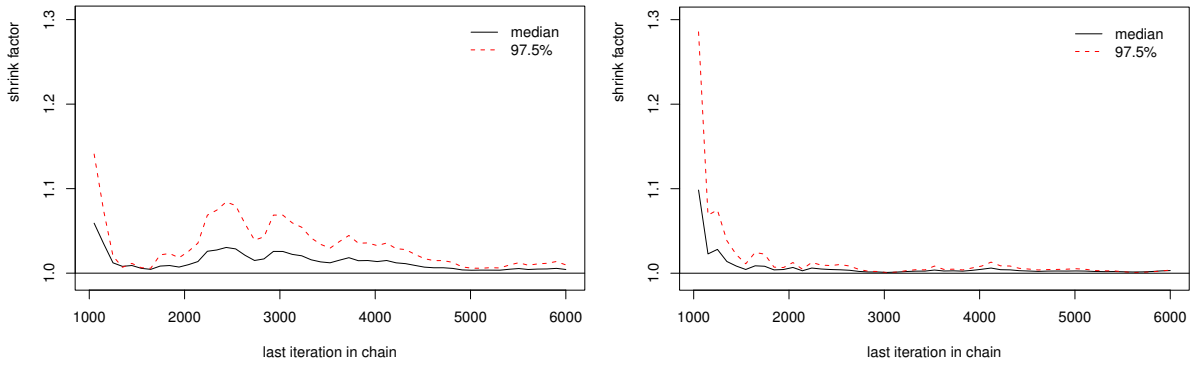


Figure 2: The process of running an MTC with two parallel chains. Notation: UML-2 activity diagram (Fowler, 2003).



(a) Undesirable

(b) Desirable

Figure 3: Undesirable and desirable convergence behaviour illustrated. In (a), after 1,500 iterations we might incorrectly conclude the simulation has sufficiently converged. The more gradual but consistent decline of (b) is preferable.

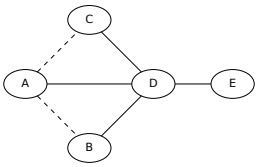


Figure 4: Evidence network for the Parkinson's disease trials, with basic parameters shown as solid lines and the functional parameters as dashed lines

```
<network description="Response to treatment">
   <treatments>
      <treatment id="Pla">Placebo</treatment>
      <treatment id="Flu">Fluoxetine</treatment>
   </treatments>
   <studies>
      <study id="Fictional et al, 2009">
         <measurement treatment="Pla" responders="38" sample="97" />
         <measurement treatment="Flu" responders="42" sample="95" />
      </study>
   </studies>
</network>
```

Figure 5: The MTC input data format (XML) illustrated with an artificial data set containing one (fictional) study "Fictional et al, 2009" comparing two treatments (Placebo and Fluoxetine) on a dichotomous outcome (response to treatment)

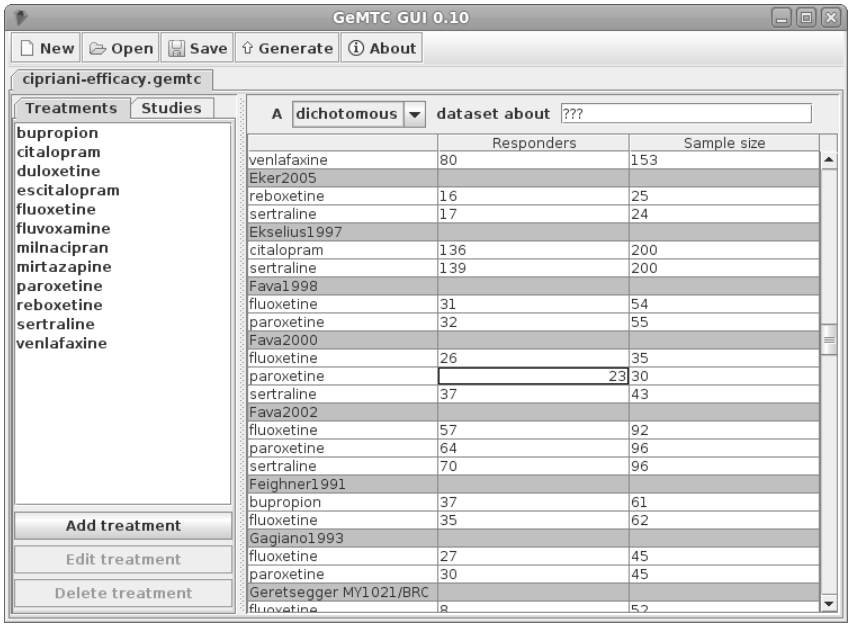


Figure 6: Creating and editing an MTC data file using the GeMTC GUI

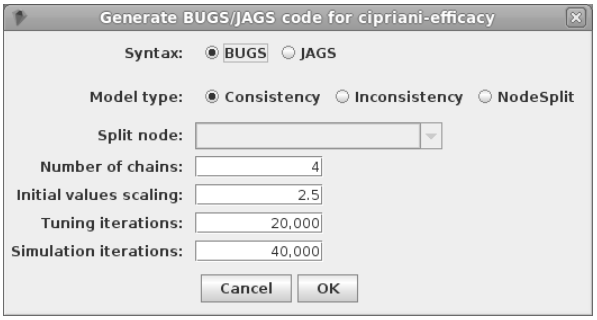


Figure 7: Configuring the model generation using the GeMTC GUI

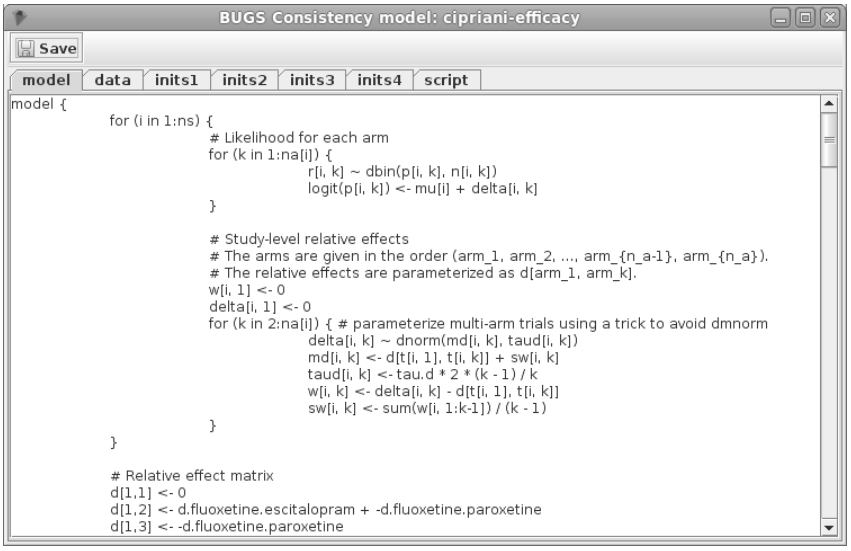


Figure 8: Generating BUGS model files using the GeMTC GUI

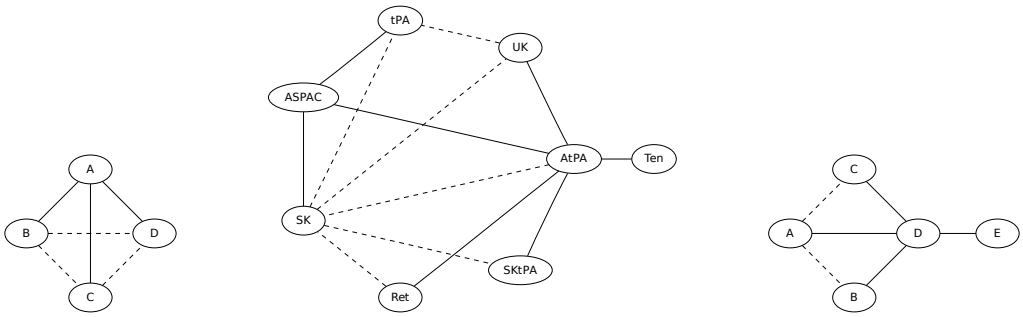
```
model {
  for (i in 1:ns) {
    # Likelihood for each arm
    for (k in 1:na[i]) {
      %armLikelihood%
    }

    # Study-level relative effects
    w[i, 1] <- 0
    delta[i, 1] <- 0
    # parameterize multi-arm trials using a trick
    # to avoid using the multi-variate normal
    for (k in 2:na[i]) {
      delta[i, k] ~ dnorm(md[i, k], taud[i, k])
      md[i, k] <- d[t[i, 1], t[i, k]] + sw[i, k]
      taud[i, k] <- tau.d * 2 * (k - 1) / k
      w[i, k] <- delta[i, k] - d[t[i, 1], t[i, k]]
      sw[i, k] <- sum(w[i, 1:k-1]) / (k - 1)
    }
  }

  # Relative effect matrix
  %relativeEffectMatrix%

  # Study baseline priors
  for (i in 1:ns) {
    mu[i] ~ dnorm(0, %priorPrecision%)
  }

  # Variance prior
  sd.d ~ dunif(0, %upper%)
  tau.d <- pow(sd.d, -2)

  # Effect parameter priors
  %parameters%
}
```

(a) Template

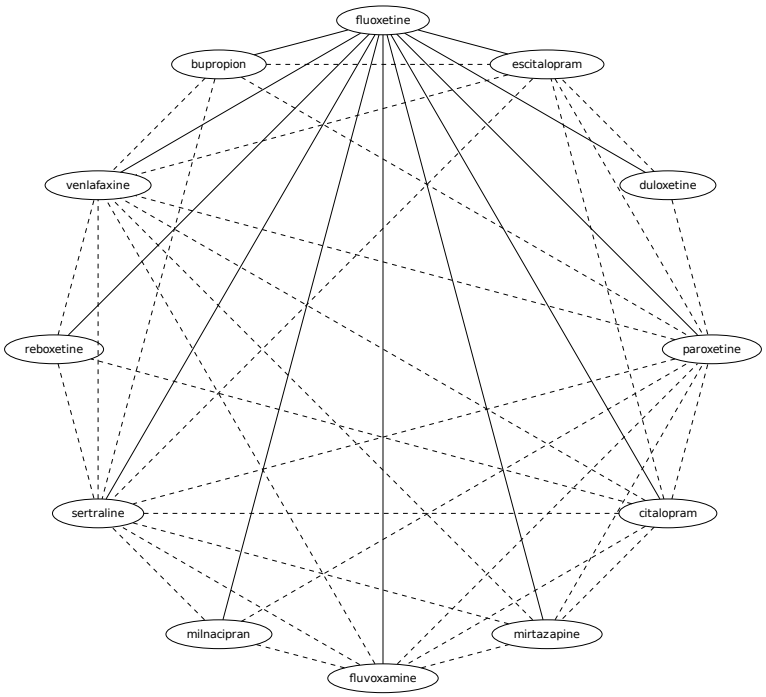
```
model {
  for (i in 1:ns) {
    # Likelihood for each arm
    for (k in 1:na[i]) {
      r[i, k] ~ dbin(p[i, k], n[i, k])
      logit(p[i, k]) <- mu[i] + delta[i, k]
    }

    # Study-level relative effects
    w[i, 1] <- 0
    delta[i, 1] <- 0
    # parameterize multi-arm trials using a trick
    # to avoid using the multi-variate normal
    for (k in 2:na[i]) {
      delta[i, k] ~ dnorm(md[i, k], taud[i, k])
      md[i, k] <- d[t[i, 1], t[i, k]] + sw[i, k]
      taud[i, k] <- tau.d * 2 * (k - 1) / k
      w[i, k] <- delta[i, k] - d[t[i, 1], t[i, k]]
      sw[i, k] <- sum(w[i, 1:k-1]) / (k - 1)
    }
  }

  # Relative effect matrix
  d[1, 1] <- 0
  d[1, 2] <- d.A.B
  d[2, 1] <- -d.A.B
  d[2, 2] <- 0

  # Study baseline priors
  for (i in 1:ns) {
    mu[i] ~ dnorm(0, 9.0*10^2)
  }

  # Variance prior
  sd.d ~ dunif(0, 2.0*10^0)
  tau.d <- pow(sd.d, -2)

  # Effect parameter priors
  d.A.B ~ dnorm(0, 9.0*10^2)
}
```

(b) Example

Figure 9: General BUGS/JAGS model template used to generate BUGS/JAGS code (a). Template variables (written as `%variableName%`) are replaced with the appropriate text by our code generation procedure. The resulting BUGS/JAGS code is illustrated for a pair-wise meta-analysis with dichotomous data (b). Note that the shown model (b) is specific to JAGS, as the priors are written in scientific notation as `9.0*10^2`, whereas for BUGS they would be written as `9.0E-2`.



(a) Smoking cessation (b) Thrombolytics (c) Parkinson



(d) Anti-depressants

Figure 10: Evidence graphs of the various data sets, with the basic parameters (spanning tree) shown as solid lines and the functional parameters as dashed lines

Table 1: Mean off-time reduction data from 7 trials studying 5 treatments form Parkinson's disease Dias et al. (2011).

Study	Treatment	Mean	Std. dev.	sample size
1	A	-1.22	3.70	54
	C	-1.53	4.28	95
2	A	-0.70	3.70	172
	B	-2.40	3.40	173
3	A	-0.30	4.40	76
	B	-2.60	4.30	71
	D	-1.20	4.30	81
4	C	-0.24	3.00	128
	D	-0.59	3.00	72
5	C	-0.73	3.00	80
	D	-0.18	3.00	46
6	D	-2.20	2.31	137
	E	-2.50	2.18	131
7	D	-1.80	2.48	154
	E	-2.10	2.99	143

Table 2: Prior values employed in the literature ('Theirs'), and those based on our heuristics ('Ours'). u is the upper limit of the variance prior $\sigma \sim U(0, u)$, and η^2 is the variance for the normal priors $\mu, d \sim \mathcal{N}(0, \eta^2)$. For their priors, $\hat{k}$ is the value for k such that $\eta^2 = (ku)^2$. For our priors, η^2 is calculated using $k = 15$. MD = Mean Difference, OR = Odds Ratio, NR = Not Reported.

			Theirs			Ours	
Dataset	Outcome	Scale	u	η^2	$\hat{k}$	u	η^2
Smoking cessation (Lu and Ades, 2004)	Cessation	OR	2	10^3	15.8	3.52	$2.8 \cdot 10^3$
Smoking cessation (Dias et al., 2010)	Cessation	OR	10	10^4	10.0	3.52	$2.8 \cdot 10^3$
Thrombolytic drugs (Lu and Ades, 2004)	Death (30/35 days)	OR	2	10^3	15.8	1.36	$4.1 \cdot 10^2$
Thrombolytic drugs (Dias et al., 2010)	Death (30/35 days)	OR	10	10^4	10.0	1.36	$4.1 \cdot 10^2$
Anti-depressants (Cipriani et al., 2009)	Response	OR	NR	NR		1.98	$8.8 \cdot 10^2$
Coronary heart disease (Welton et al., 2009)	Diastolic blood pressure	MD	10	10^3	3.16	8.20	$1.5 \cdot 10^4$
	Systolic blood pressure	MD	50	10^3	0.63	8.60	$1.7 \cdot 10^4$
	Total cholesterol	MD	10	10^3	3.16	0.82	$1.5 \cdot 10^2$
	Cardiac mortality	OR	2	10^3	15.8	2.67	$1.6 \cdot 10^3$
	Non-fatal MI	OR	2	10^3	15.8	2.39	$1.3 \cdot 10^3$
	Total mortality	OR	2	10^3	15.8	2.47	$1.4 \cdot 10^3$
Parkinson's (Dias et al., 2011)	Off-time reduction	MD	5	10^4	20	2.30	$1.2 \cdot 10^3$

Table 3: Results for the smoking cessation dataset (Lu and Ades, 2006), homogeneous variance consistency model. 'Theirs' are the posterior mean and standard deviation for the log odds ratio, as reported in the original paper. 'Ours' are the equivalent calculated based on our generated model for the same data. 'MC Error' is the Monte Carlo error of our estimates. The analysis was done using 4 independent chains, with 20,000 tuning (adaptive), 20,000 burn-in and 20,000 inference iterations.

Parameter	Theirs	Ours	MC Error
d_{AB}	0.494 (0.399)	0.494 (0.405)	0.002
d_{AC}	0.844 (0.236)	0.838 (0.240)	0.001
d_{AD}	1.101 (0.437)	1.100 (0.442)	0.002
σ^2	0.731	0.751	0.003

Table 4: Results for the thrombolytics dataset (Lu and Ades, 2006), homogeneous variance consistency model. 'Theirs' are the posterior mean and standard deviation for the log odds ratio, as reported in the original paper. 'Ours' are the equivalent calculated based on our generated model for the same data. 'MC Error' is the Monte Carlo error of our estimates. The analysis was done using 4 independent chains, with 20,000 tuning (adaptive), 20,000 burn-in and 20,000 inference iterations.

Parameter	Theirs	Ours	MC Error
SK–AtPA	-0.219 (0.126)	-0.224 (0.136)	0.003
SK–tPA	-0.010 (0.088)	-0.020 (0.093)	0.002
SK–SK+tPA	-0.056 (0.134)	-0.057 (0.141)	0.001
SK–Ten	-0.212 (0.199)	-0.218 (0.214)	0.003
SK–Ret	-0.167 (0.142)	-0.171 (0.154)	0.003
SK–UK	-0.236 (0.232)	-0.233 (0.240)	0.008
SK–ASPAC	0.040 (0.102)	0.037 (0.107)	0.002
σ^2	0.020	0.023	0.001

Table 5: Results for the efficacy of second-generation anti-depressants dataset (Cipriani et al., 2009). 'Theirs' are the posterior median and 95% credibility interval for the mean difference, as reported in the original paper. 'Ours' are the equivalent calculated based on our generated model for the same data. 'MC Error' is the Monte Carlo error of our estimates. The analysis was done using 4 independent chains, with 20,000 tuning (adaptive), 20,000 burn-in and 20,000 inference iterations.

Parameter	Theirs	Ours	MC Error
Fluoxetine–Bupropion	1.08 (0.90, 1.29)	1.08 (0.90, 1.30)	0.002
Fluoxetine–Citalopram	1.10 (0.93, 1.31)	1.10 (0.93, 1.31)	0.001
Fluoxetine–Duloxetine	0.99 (0.79, 1.24)	0.99 (0.78, 1.24)	0.002
Fluoxetine–Escitalopram	1.32 (1.12, 1.55)	1.32 (1.12, 1.55)	0.002
Fluvoxamine–Fluoxetine	1.02 (0.81, 1.30)	1.02 (0.80, 1.29)	0.003
Milnacipran–Fluoxetine	0.99 (0.74, 1.31)	0.99 (0.74, 1.32)	0.003
Mirtazapine–Fluoxetine	0.73 (0.60, 0.88)	0.73 (0.60, 0.88)	0.002
Paroxetine–Fluoxetine	0.98 (0.86, 1.12)	0.99 (0.86, 1.13)	0.002
Reboxetine–Fluoxetine	1.48 (1.16, 1.90)	1.48 (1.15, 1.91)	0.002
Sertraline–Fluoxetine	0.80 (0.69, 0.93)	0.80 (0.69, 0.94)	0.002
Venlafaxine–Fluoxetine	0.78 (0.68, 0.90)	0.78 (0.68, 0.90)	0.001

Table 6: Results for the diastolic blood pressure dataset (Welton et al., 2009), pair-wise random effects model. 'Theirs' are the posterior median and standard deviation for the mean difference, as reported in the original paper. 'Ours' are the equivalent calculated based on our generated model for the same data. 'MC Error' is the Monte Carlo error of our estimates. The analysis was done using 4 independent chains, with 20,000 tuning (adaptive), 20,000 burn-in and 20,000 inference iterations.

Outcome	Parameter	Theirs	Ours	MC Error
Diastolic bloodpressure	d	-1.377 (-3.312, 0.6232)	-1.382 (-3.340, 0.617)	0.004
	σ	1.966 (0.223, 5.085)	1.966 (0.215, 5.088)	0.014
Systolic bloodpressure	d	-1.316 (-4.24, 2.326)	-1.318 (-4.184, 2.176)	0.008
	σ	3.438 (1.042, 8.21)	3.407 (1.063, 7.288)	0.016
Cholesterol	d	-0.3197 (-0.4975, -0.1316)	-0.320 (-0.498, -0.131)	0.000
	σ	0.277 (0.154, 0.4982)	0.277 (0.154, 0.495)	0.001

Table 7: Results for the Parkinson dataset (Dias et al., 2011), homogeneous variance consistency random effects model. 'Theirs' are the posterior mean and standard deviation for the log odds ratio, as reported in the original paper. 'Ours' are the equivalent calculated based on our generated model for the same data. The analysis was done using 4 independent chains, with 20,000 tuning (adaptive), 20,000 burn-in and 20,000 inference iterations.

Parameter	Theirs	Ours	MC Error
d_{AB}	-1.84 (-2.91, -0.85)	-1.84 (-2.86, -0.88)	0.004
d_{AC}	-0.50 (-1.78, 0.75)	-0.50 (-1.74, 0.71)	0.007
d_{AD}	-0.53 (-1.77, 0.71)	-0.53 (-1.76, 0.67)	0.008
d_{AE}	-0.83 (-2.35, 0.69)	-0.84 (-2.31, 0.63)	0.009
σ	0.28 (0.01, 1.55)	0.28 (0.01, 1.39)	0.006