Chapter 8
Network Meta-Analysis

Network meta-analysis (also known as multiple treatment comparison or mixed treatment comparison) seeks to combine information from all randomised comparisons among a set of treatments for a given medical condition. It is therefore a key tool for evidence-based medicine [24] and is currently a very active research topic [1, 21, 31, 32].

Network meta-analysis is a generalisation of pairwise meta-analysis. It aims to answer, in a statistically principled way, the natural clinical question of how a number of existing treatments for a patient with a given diagnosis compare to each other. To do this, network meta-analysis combines direct and indirect evidence on treatment effects, as we explain in Sect. 8.1. However, problems of heterogeneity and potential inconsistency are ever present and potentially even more challenging than in pairwise meta-analysis. In Sect. 8.5 we therefore discuss graphical tools for presenting and understanding heterogeneity.

There are a variety of different methods and software for network meta-analysis [3]. In this chapter, we describe and illustrate a frequentist weighted least squares approach, described by Rücker [27, 28] and implemented in the R package **netmeta** [30]. This stable, computationally fast, and widely applicable approach is essentially equivalent to maximum likelihood estimation. Other approaches are described briefly in Appendix A.3.

8.1 Concepts and Challenges of Network Meta-Analysis

To introduce the ideas, suppose we wish to compare three treatments, say two active treatments A and B and a control C. Then each randomised comparison in a study having arms A and C provides a *direct* estimate of the difference of the treatment effects of A and C, measured on some scale, e.g., as a log odds ratio. Suppose we denote this $\hat{\theta}_{AC}^{\text{direct}}$. Other studies may provide information on the direct comparison

G. Schwarzer et al., *Meta-Analysis with R*, Use R!,
DOI 10.1007/978-3-319-21416-0_8

between treatment B and the same control C, denoted $\hat{\theta}_{BC}^{\text{direct}}$. Such studies provide *indirect* evidence for the comparison of A and B from the treatment difference $A-C$ and $B-C$ as follows:

$$\hat{\theta}_{AB}^{\text{indirect}} = \hat{\theta}_{AC}^{\text{direct}} - \hat{\theta}_{BC}^{\text{direct}} \tag{8.1}$$

with variance

$$\text{Var}\,(\hat{\theta}_{AB}^{\text{indirect}}) = \text{Var}\,(\hat{\theta}_{AC}^{\text{direct}}) + \text{Var}\,(\hat{\theta}_{BC}^{\text{direct}}). \tag{8.2}$$

Of course, in addition to this indirect evidence, we may have direct evidence from studies comparing A and B, denoted by $\hat{\theta}_{AB}^{\text{direct}}$. We wish to combine the direct and indirect evidence to get the most precise estimates of the treatment differences and associated standard errors. In order to do this we need to make some assumptions, principally

1. that the studies are independent, and
2. that the underlying effects are—in some sense—*consistent*. This is also known as transitivity assumption [31, 35].

We will see in Sect. 8.3.1 that the data available can be summarised in a *network graph*. Formally, a network is said to be consistent if the sum of direct treatment effects over all closed circuits in the graph is zero. In practice, consistency means that indirect evidence for the difference between any two treatments does not differ from the direct evidence. Consistency in our network of treatments A, B, and control C means that $\theta_{AB}^{\text{direct}} = \theta_{AB}^{\text{indirect}}$ for the comparison of A and B, $\theta_{AC}^{\text{direct}} = \theta_{AC}^{\text{indirect}}$ for the comparison of A and C, and $\theta_{BC}^{\text{direct}} = \theta_{BC}^{\text{indirect}}$ for the comparison of B and C. A consequence of this is that if data giving direct comparisons between A and B are available alongside data giving indirect comparisons, the extent of inconsistency, $\hat{\theta}_{AB}^{\text{direct}} - \hat{\theta}_{AB}^{\text{indirect}}$, can be assessed. Further, the proportion of the evidence coming from direct and indirect comparisons can be calculated and used to help interpret the results.

Example 8.1 Figure 8.1 shows a slightly more complex network with four treatments, named A, B, C and D. The treatments, called *nodes* in graph theory, which are joined with a line, called *edge* in graph theory, correspond to those for which direct evidence is available. Thus, we see that direct evidence is available for all comparisons except $A - C$ which must be estimated indirectly. Furthermore, for example, the comparison of A and D can draw on both direct information, and also indirect information following the paths $A \to B \to D$ and $A \to B \to C \to D$. Returning to the $A - C$ comparison without direct evidence, this is built up by combining information of both paths $A \to B \to C$ and $A \to D \to C$. □

The statistical problem is to estimate the treatment differences, in our example $\theta_{AB}, \theta_{AC}, \theta_{AD}, \theta_{BC}, \theta_{BD}$ and θ_{CD}, and their standard errors from the available studies. To do this we shall assume in the first instance that all the direct and indirect effects

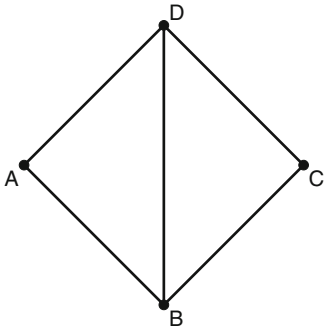

Fig. 8.1 A network with four treatments, A, B, C and D. *Lines* indicate we have data from one or more studies comparing the two treatments

are consistent with each other and that differences we see in the data are due to random error. We will explore how to assess this assumption and modify the analysis when it may not be realistic.

8.2 Model and Estimation in Network Meta-Analysis

There has been a considerable literature on frequentist estimation of treatment effects in network meta-analysis. The approach we now present draws on the following references: [19, 22, 27, 33]. We begin by considering data consisting of only two-arm trials. The extension to multi-arm studies is detailed on page 192.

Suppose there are n treatments, corresponding to the nodes in a network graph. Suppose further we wish to estimate the n treatment effects, which we denote by the $n \times 1$ vector $\boldsymbol{\theta}^{\text{treat}}$. To do this we have data from $K \geq (n-1)$ two-arm studies,[1] which we denote by $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_K)$ with associated standard errors $\mathbf{s} = (s_1, s_2, \ldots, s_K)$. Then our model is

$$\hat{\boldsymbol{\theta}} = \mathbf{X}\boldsymbol{\theta}^{\text{treat}} + \epsilon, \quad \epsilon \sim N(\mathbf{0}, \boldsymbol{\Sigma}), \tag{8.3}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix whose ith entry is s_i^2.

Note, $\hat{\boldsymbol{\theta}}$ is a $K \times 1$ vector in a network meta-analysis consisting only of two-arm trials. In the situation of multi-arm studies, this vector is of dimension $m \times 1$ with m denoting the total number of pairwise comparisons.

Example 8.2 Consider again the network shown in Fig. 8.1, and suppose we have $K = 5$ studies each providing a single pairwise treatment comparison: $A-B$, $B-C$, $C-D$, $A-D$ and $B-D$, i.e. the data consist of $m = 5$ pairwise treatment comparisons. Then $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1^{AB}, \hat{\theta}_2^{BC}, \hat{\theta}_3^{CD}, \hat{\theta}_4^{AD}, \hat{\theta}_5^{BD})^T$, and $\boldsymbol{\theta}^{\text{treat}} = (\theta_A, \theta_B, \theta_C, \theta_D)^T$ is the vector of

[1] A minimum of $n-1$ two-arm studies is necessary to create a connected network graph with n treatments (nodes).

treatment effects ($n = 4$). Based on this information, we can easily build the design matrix $\mathbf{X}$ and write down (8.3) for this special case:

$$\begin{pmatrix} \hat{\theta}_1^{AB} \\ \hat{\theta}_2^{BC} \\ \hat{\theta}_3^{CD} \\ \hat{\theta}_4^{AD} \\ \hat{\theta}_5^{BD} \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} \theta_A \\ \theta_B \\ \theta_C \\ \theta_D \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{pmatrix}$$
$$= \mathbf{X}\boldsymbol{\theta}^{\text{treat}} + \epsilon.$$

□

We note that, since—as usual in meta-analysis of summary data—we only have data on treatment differences, we cannot estimate each of the elements in $\boldsymbol{\theta}^{\text{treat}}$. However, as we describe in the next subsection, we can nevertheless estimate the fitted values from (8.3), and then use these to estimate the treatment comparisons and assess the extent of heterogeneity.

8.2.1 Fixed Effect Model

Here we describe details of the frequentist weighted least squares approach by Rücker [27]. Less technically minded readers may wish to skip to the worked example in Sect. 8.3.

We have already noted that (8.3) is over-parameterised, as we only have at most $(n - 1)$ independent treatment comparisons, but the model has a parameter for each of the n treatments, $\boldsymbol{\theta}^{\text{treat}}$. Thus the matrix $\mathbf{X}$ is not of full rank, so its inverse does not exist, and we cannot obtain the weighted least squares estimates of $\boldsymbol{\theta}^{\text{treat}}$ directly.

We could re-parameterise the model so that $\mathbf{X}$ is of full rank. However, as Rücker [27] shows we can avoid this by taking the approach outlined below. This approach was originally developed using graph-theoretical methods that have a long history in electrical network theory [7, 9, 26, 27, 34]. In fact, it is also equivalent to weighted least squares methodology developed for experimental design as far back as the 1930s [4, 5, 25, 27, 33, 36]. Indeed, network meta-analyses are examples of incomplete block designs [33]. The difference from agricultural designs is that the decisions about which comparisons to estimate, i.e. trials to perform, are not made in advance with the network meta-analysis in mind!

As before, we denote by n the number of different treatments (nodes) in the network and let m be the number of pairwise treatment comparisons (and hence the

number of studies, since thus far we are only considering two-arm studies).[2] We number the treatments $1, \ldots, n$ and pairwise comparisons $1, \ldots, m$ in an arbitrary order which remains unchanged throughout the calculation.

Then, let $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_m)^T$ and $\mathbf{s} = (s_1, \ldots, s_m)^T$ be the vectors of observed treatment differences and their standard errors, respectively. As is common in meta-analysis, we treat the standard errors as fixed.

As in (8.3), the network structure is defined by the design matrix $\mathbf{X}$ which is an $m \times n$ matrix. Here, if we only include two-arm studies, each row corresponds to a study and m is the number of studies. Then, as illustrated in Example 8.2, we have a "1" in the column that corresponds to the first treatment and a "-1" in the column that belongs to the second treatment. Each row of $\mathbf{X}$ should then sum to zero.

As previously mentioned, $\mathbf{X}^T\mathbf{X}$ is not of full rank, so to estimate the treatment effects we need to construct the Moore–Penrose pseudoinverse matrix [2, 28]. To do this, we define the $n \times n$ Laplacian matrix $\mathbf{L}$ (which plays a central role in graph theory) as

$$\mathbf{L} = \mathbf{X}^T\mathbf{W}\mathbf{X}, \tag{8.4}$$

where $\mathbf{W}$ is a diagonal matrix of dimension $m \times m$ whose diagonal elements are the inverse variance study weights $(1/s_1^2, \ldots, 1/s_m^2)$.

$\mathbf{L}$, the Laplacian matrix, has rank $n-1$ and is not invertible. However, its Moore–Penrose pseudoinverse $\mathbf{L}^+$ [10, 26] is defined and can be calculated by

$$\mathbf{L}^+ = (\mathbf{L} - \mathbf{J}/n)^{-1} + \mathbf{J}/n, \tag{8.5}$$

where $\mathbf{J}$ is the $n \times n$ matrix whose elements are all 1.

Estimation of Treatment Effects

Once we have $\mathbf{L}^+$, we can calculate our estimates of the fitted values $\hat{\boldsymbol{\theta}}^{nma}$ as

$$\begin{aligned} \hat{\boldsymbol{\theta}}^{nma} &= \mathbf{X}\mathbf{L}^+\mathbf{X}^T\mathbf{W}\hat{\boldsymbol{\theta}}, \\ &= \mathbf{H}\hat{\boldsymbol{\theta}}, \end{aligned} \tag{8.6}$$

where $\mathbf{H}$ is known as the *hat matrix* in regression. Equation (8.6) means that the elements of $\hat{\boldsymbol{\theta}}^{nma}$ (the network estimates) are linear combinations of the elements of $\hat{\boldsymbol{\theta}}$ (the observed estimates) with coefficients coming from the rows of $\mathbf{H}$. In other words, each network estimate is constituted by the observed estimates, weighted

[2] Note that if we allow multi-edge graphs, so, for example, if there are two studies comparing A and B we have two edges connecting nodes A and B, we may call m the number of *edges*.

with the elements of the corresponding row of $\mathbf{H}$. Accordingly, the elements of this row of $\mathbf{H}$ are interpreted as generalised weights.

We can also calculate the variance–covariance matrix of $\hat{\boldsymbol{\theta}}^{nma}$, $\mathbf{X}\mathbf{L}^{+}\mathbf{X}^{T}$. From this we can estimate all treatment contrasts and associated standard errors of interest. These estimates and standard errors are the same as those obtained by (weighted) maximum likelihood [25, 33, 36].

The hat matrix, $\mathbf{H}$, is a projection matrix which maps $\hat{\boldsymbol{\theta}}$ onto the consistent $(n-1)$-dimensional subspace. This gives the fitted values $\hat{\boldsymbol{\theta}}^{nma}$, which can be interpreted as the values that minimize the quadratic form

$$Q_{\text{total}} = (\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}^{nma})^{\top}\mathbf{W}(\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}^{nma}). \tag{8.7}$$

The Q_{total} statistic (8.7) therefore measures the extent of heterogeneity within the network. When all studies are two-arm studies, under the null hypothesis of no heterogeneity it is approximately χ^2-distributed with $m-(n-1)$ degrees of freedom. This therefore provides an approximate test of consistency. For a standard pairwise meta-analysis, Q_{total} corresponds to Cochran's Q statistic [12].

Variance Estimation

After fitting the network meta-analysis model, for any treatment comparison, we can calculate an estimate of the treatment effect using the direct evidence, and each piece of indirect evidence. The variance of the resulting treatment estimate is estimated by

$$V_{ij} = L^{+}_{ii} + L^{+}_{jj} - 2L^{+}_{ij}, \tag{8.8}$$

where V_{ij} denotes the variance of the resulting (potentially indirect) comparison of treatments i and j [7]. We note explicitly that by Eq. (8.8) variances are also estimated for pairs of treatments for which no direct comparison exists. Assuming that treatment estimates are consistent across the network, by using all the information in the data through the network meta-analysis, we obtain the most precise estimates of all comparisons.

Multi-Arm Studies

Usually, we have a number of multi-arm studies (i.e. studies with more than two treatment groups) to include in our network meta-analysis. We can do this most easily by including each multi-arm study in the dataset as a series of two-arm comparisons. However, the standard error of each two-arm comparison from a multi-arm study needs to be adjusted to reflect the fact that comparisons within multi-arm studies are correlated.

Consider a multi-arm study of p treatments with known variances. For this study, in the `netmeta` function, the user needs to supply treatment effects and standard errors for each of $p(p-1)/2$ possible comparisons. For instance, a three-arm study contributes three pairwise comparisons, $3(3-1)/2 = 3$, and a four-arm study six possible pairwise comparisons, $4(4-1)/2 = 6$.

We have to take care to account for within-study correlation before computing the weighted least squares estimator (8.6). To do this we inflate the standard errors for comparisons within each multi-arm study by back-calculation. Using these back-calculated standard errors in the weighted least squares estimator then gives results that correctly reflect the within-study correlation. To achieve this, we use a theorem by Gutman and Xiao [10, Theorem 7] to determine $\mathbf{L}_s^+$ and $\mathbf{L}_s = (\mathbf{L}_s^+)^+$ for each multi-arm study, also called sub-network s, with $s = 1, \ldots, S$, and S denoting the number of multi-arm studies, as described by [27]. The calculation proceeds as follows.

For multi-arm study s with $p_s > 2$ arms, for each of the $p_s(p_s-1)/2$ comparisons we have a variance for the comparison of each treatment contrast i and j ($i \neq j$), say v_{sij}^2. Let $\mathbf{V}_s$ be a $p_s \times p_s$ symmetric matrix with zeros on the diagonal and with (i,j) entry v_{sij}^2. Let $\mathbf{X}_s$ be the design matrix for the $p_s(p_s-1)/2$ comparisons in multi-arm study s, formed as in (8.3) above.

Then calculate

$$\mathbf{L}_s^+ = -\frac{1}{2p_s^2}\,\mathbf{X}_s^\top \mathbf{X}_s \mathbf{V}_s \mathbf{X}_s^\top \mathbf{X}_s, \tag{8.9}$$

and from this, calculate $\mathbf{L}_s = (\mathbf{L}_s^+)^+$ using (8.5). Denote the elements of $\mathbf{L}_s$ by l_{sij}. The adjusted variances for the comparison of treatment i and j are $-1/l_{sij}^{-1}$. Rücker et al. [28] show this method leads to results that are identical to those of the standard approach [19, 33].

Example 8.3 We illustrate the calculation for a network with a single four-arm study ($p_1 = 4$), so that $\mathbf{V} = \mathbf{V}_1$ has dimension 4×4. If v_{ij} is the variance of the comparison of treatment i and j, then we set

$$\mathbf{V} = \begin{pmatrix} 0 & v_{12} & v_{13} & v_{14} \\ v_{12} & 0 & v_{23} & v_{24} \\ v_{13} & v_{23} & 0 & v_{34} \\ v_{14} & v_{24} & v_{34} & 0 \end{pmatrix}.$$

Then,

$$\mathbf{X}_1 = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{pmatrix},$$

and we calculate $\mathbf{L}_1^+$ as given by (8.9). □

I-Squared for Network Meta-Analysis

The generalised heterogeneity statistic Q_{total} given in Eq. (8.7) is used to measure heterogeneity/inconsistency across the whole network. It can be partitioned in various ways, to help identify and understand the sources of heterogeneity. We return to this in Sect. 8.4.

We now consider the degrees of freedom of Q_{total}. Each p-arm study contributes $p-1$ degrees of freedom to the total Q_{total} statistic. The total degrees of freedom are given by the sum of the degrees of freedom contributed by each study minus $n-1$ (the number of treatments minus 1, which is the dimension of the consistent subspace). Denoting this by df, we can define a generalised I^2 statistic [12] as

$$I^2 = \max\left(\frac{Q_{\text{total}} - df}{Q_{\text{total}}}, 0\right). \tag{8.10}$$

8.2.2 Random Effects Model

A simple random effects model can be defined using the estimate of a common heterogeneity variance τ^2 for each pairwise treatment comparison. This is then added to each of the comparison variances, $s_i^2 + \hat{\tau}^2$, $i = 1, \ldots m$, before calculating $\mathbf{L}^+$ (8.5) which is used in (8.6). For multi-arm studies, the estimate $\hat{\tau}^2$ is added to the observed variance of each comparison before reducing the weights as described on page 192. Network meta-analysis is applied to the same observed treatment differences, now using the enlarged standard errors, as in standard pairwise meta-analysis.

In order to do this, we need an estimate of τ^2. For this we use a special case of the generalised DerSimonian–Laird estimate given in Jackson et al. [16] referred to in Sect. 7.3.1. It is estimated by

$$\hat{\tau}^2 = \max\left(\frac{Q_{\text{total}} - df}{tr\left((\mathbf{I} - \mathbf{H})\mathbf{U}\mathbf{W}\right)}, 0\right), \tag{8.11}$$

where $\mathbf{I}$ is the $m \times m$ identity matrix and *tr* denotes the trace of a matrix, which is the sum of its diagonal elements. The matrix $\mathbf{H}$ is defined in (8.6) and $\mathbf{W}$ is given in Sect. 8.2.1. $\mathbf{U}$ is obtained as a block diagonal matrix derived from the $m \times m$ matrix $\mathbf{X}\mathbf{X}^{\top}/2$ by selecting for each p-arm study a $p \times p$ block while setting the rest of the matrix elements to zero.

8.3 Using the R Package netmeta for Network Meta-Analysis

In this section we present an extended example using data from a network meta-analysis by Senn et al. [33] comparing different treatments for controlling blood glucose levels in patients with diabetes. This dataset comes with the R package **netmeta**.[3]

Example 8.4 To load the data, use the R code shown in Fig. 8.2. Patients enrolled in studies included in this dataset were treated with one of ten diabetes treatments, designed to reduce blood glucose levels. The effect measure was the mean difference of average plasma glucose concentration, referred to as HbA_{1c} and measured in mmol/mol. The full names of the various treatments can be obtained by typing

```
> help("Senn2013")
```

at the command line.

Variable `TE` contains the pairwise treatment effect comparing treatments `treat1` and `treat2`; variable `seTE` is the corresponding standard error. For example, the `DeFrozo1995` study is comparing metformin (`metf`) and placebo (`plac`). The average plasma glucose concentration is larger in the placebo group accordingly the mean difference is negative, −1.90.

The dataset contains one three-arm study (`Willms1999` in rows 3, 27, 28). The `netmeta` function which is described in the next subsection requires as input all pairwise comparisons for multi-arm studies. Therefore, we now show how the necessary information can be extracted from a publication using the `Willms1999` study as an example. This study reports sample sizes, group means, and corresponding standard deviations.

```
> willms <- data.frame(treatment=c("metf", "acar", "plac"),
+                      n=c(29, 31, 29),
+                      mean=c(-2.5, -2.3, -1.3),
+                      sd=c(0.862, 1.782, 1.831),
+                      stringsAsFactors=FALSE)
> willms
  treatment  n mean    sd
1      metf 29 -2.5 0.862
2      acar 31 -2.3 1.782
3      plac 29 -1.3 1.831
```

[3] To install R package **netmeta** use R command `install.packages("netmeta")`.

Using the `metacont` function we can calculate all three pairwise treatment comparisons:

```
> comp12 <- metacont(n[1], mean[1], sd[1], n[2], mean[2], sd[2],
+                    data=willms, sm="MD")
> comp13 <- metacont(n[1], mean[1], sd[1], n[3], mean[3], sd[3],
+                    data=willms, sm="MD")
> comp23 <- metacont(n[2], mean[2], sd[2], n[3], mean[3], sd[3],
+                    data=willms, sm="MD")
```

Next, we extract mean differences and corresponding standard errors from R objects `comp12`, `comp13` and `comp23`

```
> TE <- c(comp12$TE, comp13$TE, comp23$TE)
> seTE <- c(comp12$seTE, comp13$seTE, comp23$seTE)
```

and define R objects

```
> treat1 <- c(willms$treatment[1], willms$treatment[1],
+             willms$treatment[2])
> treat2 <- c(willms$treatment[2], willms$treatment[3],
+             willms$treatment[3])
```

with information on the two treatments.

These R objects can be combined in an R data set

```
> data.frame(TE, seTE=round(seTE, 4), treat1, treat2,
+            studlab="Willms1999")
    TE   seTE treat1 treat2    studlab
1 -0.2 0.3579   metf   acar Willms1999
2 -1.2 0.3758   metf   plac Willms1999
3 -1.0 0.4669   acar   plac Willms1999
```

which contains exactly the same information as given in rows 3, 27 and 28 of Fig. 8.2.

Note, a shortened version of this R code can be used to calculate the treatment comparison for a two-arm study reporting sample sizes, group means and standard deviations. Likewise, the code can be altered in order to calculate treatment comparisons for binary data using the `metabin` function instead of the `metacont` function.[4]

8.3.1 Basic Analysis and Network Plots

The `netmeta` function has the following arguments:

```
> args(netmeta)
function (TE, seTE, treat1, treat2, studlab, data = NULL,
```

[4] After finishing the book manuscript, a new R function `pairwise` to conduct these calculations automatically has been added to R package `meta`.

```
> # 1. Make R package netmeta available
> library(netmeta)
Loading required package: meta
Loading 'meta' package (version 4.0-2).
Loading 'netmeta' package (version 0.6-0).
> # 2. Load dataset
> data(Senn2013)
> data15 <- Senn2013
> # 3. Print dataset
> data15
      TE   seTE treat1 treat2            studlab
1  -1.90 0.1414   metf   plac       DeFronzo1995
2  -0.82 0.0992   metf   plac          Lewin2007
3  -0.20 0.3579   metf   acar         Willms1999
4  -1.34 0.1435   rosi   plac       Davidson2007
5  -1.10 0.1141   rosi   plac  Wolffenbuttel1999
6  -1.30 0.1268   piog   plac         Kipnes2001
7  -0.77 0.1078   rosi   plac        Kerenyi2004
8   0.16 0.0849   piog   metf       Hanefeld2004
9   0.10 0.1831   piog   rosi         Derosa2004
10 -1.30 0.1014   rosi   plac          Baksi2004
11 -1.09 0.2263   rosi   plac     Rosenstock2008
12 -1.50 0.1624   rosi   plac           Zhu2003
13 -0.14 0.2239   rosi   metf          Yang2003
14 -1.20 0.1436   rosi   sulf Vongthavaravat2002
15 -0.40 0.1549   acar   sulf          Oyama2008
16 -0.80 0.1432   acar   plac          Costa1997
17 -0.57 0.1291   sita   plac      Hermansen2007
18 -0.70 0.1273   vild   plac         Garber2008
19 -0.37 0.1184   metf   sulf           Alex1998
20 -0.74 0.1839   migl   plac       Johnston1994
21 -1.41 0.2235   migl   plac      Johnston1998a
22  0.00 0.2339   rosi   metf           Kim2007
23 -0.68 0.2828   migl   plac      Johnston1998b
24 -0.40 0.4356   metf   plac Gonzalez-Ortiz2004
25 -0.23 0.3467   benf   plac         Stucci1996
26 -1.01 0.1366   benf   plac         Moulin2006
27 -1.20 0.3758   metf   plac         Willms1999
28 -1.00 0.4669   acar   plac         Willms1999
```

Fig. 8.2 R code to load the diabetes example and view the data. For each treatment comparison (columns `treat1` and `treat2`) in the network, the estimated treatment effect on HbA_{1c}, `TE`, and its standard error, `seTE`, are shown. The right-hand column shows the study labels. Note `Willms1999` (rows 3, 27, 28) is a three-arm study

```
subset = NULL, sm = "", level = 0.95, level.comb = 0.95,
comb.fixed = TRUE, comb.random = FALSE, reference.group = "",
all.treatments = NULL, seq = NULL, tau.preset = NULL,
title = "", warn = TRUE)
```

Of these, the first four arguments: TE, seTE, treat1, and treat2 are mandatory. However, the study label argument studlab will be used in almost all analyses, and is essential for telling the software if there are multi-arm studies in the network. As with the Willms1999 study in Fig. 8.2, the treatment comparisons within a multi-arm study must have exactly the same study label, or the software will not be able to link them. A further important point is that it does not matter what order we put each study's treatment comparisons in; the software will re-order them (and change the sign of the treatment effect) as appropriate.

We can illustrate these points as follows.

```
> mn0 <- netmeta(TE, seTE, treat1, treat2, data=data15)
Warning messages:
1: In netmeta(TE, seTE, treat1, treat2, data = data15) :
  No information given for argument 'studlab'. Assuming that
  comparisons are from independent studies.
2: In netmeta(TE, seTE, treat1, treat2, data = data15) :
  Treatments within a comparison have been re-sorted in
  increasing order.
```

The first warning tells us that we did not give the study labels, and that the software is assuming that each row of the data matrix data15 comes from a different two-arm study. This is incorrect for this example, as we have one three-arm study, Willms1999. The second warning simply says that the software has re-ordered the treatment comparisons appropriately before performing any analyses.

As usual, we can print the object nm0 by entering it at the command line:

```
> mn0
Original data:

   treat1 treat2    TE   seTE
1    metf   plac -1.90 0.1414
2    metf   plac -0.82 0.0992
3    acar   metf  0.20 0.3579
4    plac   rosi  1.34 0.1435
*** Output truncated ***
Number of studies: k=28
Number of treatments: n=10
Number of pairwise comparisons: m=28
*** Output truncated ***
```

Comparing with Fig. 8.2, we see the number of studies is wrongly reported as 28, instead of 26. Furthermore, again comparing with Fig. 8.2, we see the fourth comparison (study Davidson2007) has been reversed, and the treatment effect correctly changed to 1.34.

The argument sm tells the software which summary measure to use for displaying the results of the network meta-analysis. Note that the treatment effects supplied to the software must be on the scale on which the analysis should be performed (e.g. log odds ratios, not odds ratios). For example, if log odds ratios are used as input to the netmeta function, odds ratios will be shown in printouts as well as forest plots if argument sm="OR" is used; otherwise, log odds ratios will be shown. The default

is `sm=""`, i.e. no information on the underlying summary measure is provided and thus no back-transformation is used in printouts and forest plots.

In many cases, the user will want to specify the reference group for making treatment comparisons. To do this, the argument `reference.group` (abbreviated `ref` or `r`) allows the user to specify a reference group. For example with the diabetes data we may specify `r="plac"`. To see all possible contrasts in the form of an effect matrix, the argument `all.treatments` (abbreviated `all` or `a`) is set to `TRUE`. This is the default.

The remaining arguments are the same as those we have met earlier in standard pairwise meta-analysis, and are discussed in detail in Part II. By default, only the results of a fixed effect network meta-analysis are printed. In order to also show results of a random effects analysis the argument `comb.random=TRUE` needs to be specified.

Before proceeding, we need to repeat the analysis, this time including the study labels. We also take the opportunity to specify the summary measure as mean difference (`MD`):

```
> mn1 <- netmeta(TE, seTE, treat1, treat2, studlab,
+                data=data15, sm="MD")
*** Warning message on study reordering omitted ***
```

8.3.2 A First Network Plot

We now show how to create a graphical representation of the network using the `netgraph` function.

```
> netgraph(mn1, seq=c("plac", "benf", "migl", "acar", "sulf",
+                     "metf", "rosi", "piog", "sita", "vild"))
```

The results are shown in Fig. 8.3. Note the use of the argument `seq` to specify the order of the sequence in which the treatments are shown anti-clockwise around the perimeter of the circle.[5] Treatments that are directly compared in at least one study are connected by a line. The thickness of this line is proportional to the inverse standard error of the direct treatment effect obtained using data from all studies which compared the two treatments. Thus, the thicker the line, the smaller the standard error, and the greater the evidence for that comparison. Of course, as this does not show the estimated treatment effect, this does not represent the statistical significance of the comparison.

From time to time we may wish to highlight a particular comparison, and this can be done with the argument `highlight`. We explore additional capabilities of the `netgraph` function in Sect. 8.3.4.

[5] We can also use argument `seq` in the `netmeta` function which would be considered in the `netgraph` and other functions, e.g. `print.netmeta`.

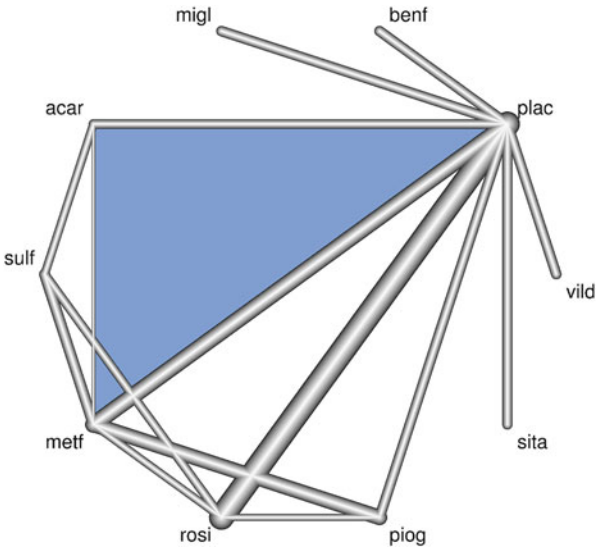

Fig. 8.3 A graph of the network for the diabetes data, generated using the `netgraph` function. The treatments are equally spaced on the perimeter of the circle. Any two treatments are connected by a line when there is at least one study comparing the two treatments. The thickness of the line is proportional to the inverse standard error of the direct treatment comparison. The *shading* indicates the three-arm study

8.3.3 A More Detailed Look at the Output

Next, we view the output stored in the R object `nm1` using the `print` function. We split this into several chunks which we discuss in turn.

```
> print(mn1, digits=2)
Original data (with adjusted standard errors for multi-arm studies):

              treat1 treat2    TE seTE seTE.adj narms multiarm
DeFronzo1995    metf   plac -1.90 0.14     0.14     2
Lewin2007       metf   plac -0.82 0.10     0.10     2
Willms1999      acar   metf  0.20 0.36     0.39     3        *
Davidson2007    plac   rosi  1.34 0.14     0.14     2
*** Output truncated ***
Moulin2006      benf   plac -1.01 0.14     0.14     2
Willms1999      metf   plac -1.20 0.38     0.41     3        *
Willms1999      acar   plac -1.00 0.47     0.82     3        *
...
```

First comes the data (treatments within a study have been re-ordered alphabetically by the program). Columns 1–5 correspond to the study labels, treatment groups (`treat1`, `treat2`), treatment effects (`TE`) and standard errors (`seTE`). The remaining three columns are only printed if at least one multi-arm study is

included in the meta-analysis. As discussed in Sect. 8.2.1, for multi-arm studies—`Willms1999` in the example—standard errors have to be adjusted accordingly. This is given in the column headed `seTE.adj`. Note that the standard error and adjusted standard error are the same for two-arm studies. The other two columns give the number of treatment arms per study (`narms`) and highlight multi-arm studies using a star (`multiarm`).

As we have already noted, it is important to use exactly identical study labels for all treatment comparisons belonging to the same multi-arm study (here `Willms1999`), otherwise the program will treat them as separate two-arm studies. The reason for this special caution with multi-arm studies is, as mentioned before, that the `netmeta` function automatically accounts for within-study correlation by reweighing all pairwise comparisons of each multi-arm study.

The next chunk of output gives a list of all the studies in alphabetical order with information on the numbers of treatment arms per study (column `narms`). Once again we see that `Willms1999` is the only study with three treatment arms.

```
...
Number of treatment arms (by study):
                    narms
Alex1998                2
*** Output truncated ***
Vongthavaravat2002      2
Willms1999              3
Wolffenbuttel1999       2
Yang2003                2
Zhu2003                 2
...
```

The next chunk of output gives results from the fixed effect network meta-analysis. For each study, and for each treatment comparison within that study, we have the treatment effect (here the mean difference) fitted by the network meta-analysis model, $\hat{\boldsymbol{\theta}}^{nma}$, see (8.6), and its 95 % confidence interval. Results are given to two decimal places; this is controlled by the `digits` argument in the `print` function. The corresponding contributions to the overall heterogeneity statistic, Q_{total}, see (8.7), are then shown, followed by the leverage (which is given by the corresponding diagonal element of the $m \times m$ hat matrix, $\mathbf{H}$).

```
...
Data utilised in network meta-analysis (fixed effect model):

                treat1 treat2    MD          95%-CI     Q leverage
DeFronzo1995      metf   plac -1.11 [-1.23; -1.00] 30.89     0.18
Lewin2007         metf   plac -1.11 [-1.23; -1.00]  8.79     0.36
Willms1999        acar   metf  0.29 [ 0.06;  0.51]  0.05     0.09
Davidson2007      plac   rosi  1.20 [ 1.11;  1.30]  0.93     0.11
*** Output truncated ***
Hermansen2007     plac   sita  0.57 [ 0.32;  0.82]  0.00     1.00
Garber2008        plac   vild  0.70 [ 0.45;  0.95]  0.00     1.00
*** Output truncated ***
Gonzalez...2004   metf   plac -1.11 [-1.23; -1.00]  2.69     0.02
```

```
Stucci1996       benf   plac -0.91 [-1.15; -0.66]  3.79     0.13
Moulin2006       benf   plac -0.91 [-1.15; -0.66]  0.59     0.87
Willms1999       metf   plac -1.11 [-1.23; -1.00]  0.04     0.02
Willms1999       acar   plac -0.83 [-1.04; -0.61]  0.04     0.02

Number of studies: k=26
Number of treatments: n=10
Number of pairwise comparisons: m=28
...
```

Each of the heterogeneity statistics approximately follows a χ^2_1-distribution. They are useful to identify studies whose data differ markedly from what the model predicts. When there is only a single study evaluating a treatment comparison—e.g. `Hermansen2007` and `Garber2008` are the only studies evaluating sitagliptin and vildagliptin, respectively—the model fits perfectly and the corresponding Q statistic is zero. Conversely, we see that the results of `DeFronzo1995` differ markedly from what the rest of the network predicts. We can confirm this by comparing the estimated metformin vs placebo comparison from the network in the output immediately above (-1.11) with the original data from this study in the printout on page 200 (-1.90). The leverage of comparison i is the ith diagonal element of the hat matrix $\mathbf{H}$. The model shows that this is the factor by which the variance of the estimate of a treatment comparison from a study is reduced by the information of the whole network. A small value of the leverage, close to 0, means a large variance reduction and thus a large gain in precision from the network. Conversely, a large value (close to 1) means almost no variance reduction and no gain in precision.

The mean leverage [27] depends only on the number of treatments n and the number of comparisons m: $(n-1)/m$. This can be checked by the following R commands

```
> mn1$n
[1] 10
> mn1$m
[1] 28
> mean(mn1$leverage.fixed)
[1] 0.3214286
> (mn1$n-1)/mn1$m
[1] 0.3214286
```

which are identical, and equal to $9/28 = 0.3214286$.

In the diabetes example the strongest gain from the network is seen in the `Gonzales-Ortiz2004` and two of the `Willms1999` comparisons with a leverage of 0.02. These comparisons have rather large standard errors and gain a lot from information through the network, particularly from other studies that have evaluated the same comparison, metformin or acarbose vs placebo. The largest possible leverage is 1.00, and two studies have this value: `Hermansen2007` and `Garber2008`. This occurs because they are the only studies evaluating sitagliptin and vildagliptin, respectively. Therefore these two drugs are not part of any loop

(circuit) in the network, see Fig. 8.3, and estimation cannot benefit from the additional data available within the network.

The next chunk of information gives the estimates for all the treatment contrasts in the network meta-analysis.

```
...
Fixed effect model

Treatment estimate (sm='MD'):
      acar  benf  metf  migl  piog  plac rosi  sita  sulf  vild
acar  0.00  0.08  0.29  0.12  0.24 -0.83 0.37 -0.26 -0.39 -0.13
*** Output truncated ***
vild  0.13  0.21  0.41  0.24  0.37 -0.70 0.50 -0.13 -0.26  0.00

Lower 95%-confidence limit:
      acar  benf  metf  migl  piog  plac  rosi  sita  sulf  vild
acar  0.00 -0.25  0.06 -0.21 -0.01 -1.04  0.15 -0.59 -0.61 -0.46
*** Output truncated ***
vild -0.20 -0.15  0.14 -0.11  0.08 -0.95  0.24 -0.49 -0.57  0.00

Upper 95%-confidence limit:
      acar  benf metf migl piog  plac rosi  sita  sulf  vild
acar  0.00  0.41 0.51 0.44 0.49 -0.61 0.60  0.07 -0.17  0.20
*** Output truncated ***
vild  0.46  0.56 0.69 0.60 0.66 -0.45 0.77  0.23  0.05  0.00
...
```

Above we have three $n \times n$ matrices (recall n is the number of treatments). These show the estimated treatment comparisons as well as lower and upper 95 % confidence limits. Results for each possible treatment comparisons are given.

Estimated treatment effects and confidence limits use information both from direct and indirect treatment comparisons. As the network is connected, all comparisons can be estimated.

The last chunk of output gives measures of heterogeneity/network inconsistency.

```
...
Quantifying heterogeneity/inconsistency:
tau^2 = 0.1087; I^2 = 81.4%

Test of heterogeneity/inconsistency:
     Q d.f.  p.value
 96.99   18 < 0.0001
```

Specifically we have the generalised DerSimonian–Laird estimator τ^2 [8, 15], Higgins' I^2, Cochran's Q_{total} (`Q`) with its degrees of freedom (`d.f.`), and a p-value for Q_{total} [12]. The results show that there is considerable heterogeneity in the network, and this needs to be explored further.

Finally, as an alternative to `print.netmeta` function, we can obtain a shorter summary of the analysis using the `summary.netmeta` function:

```
> print(summary(mn1))
Number of studies: k=26
```

```
Number of treatments: n=10
Number of pairwise comparisons: m=28
*** Output truncated ***
```

The resulting output includes matrices for treatment effects and confidence limits, as presented above.

8.3.4 Additional Network Plots

In Fig. 8.3 we have already seen a network plot produced by function `netgraph`. Here, we explore ways how to create alternative network plots using some of the additional functionality of `netgraph` [29].

We used the default settings to obtain Fig. 8.3. This placed the nodes (treatments) on a circle as, by default, argument `start.layout` is equal to `"circle"`.

We can use argument `iterate=TRUE` to further optimise the layout. First, "ideal" distances between each pair of nodes in the plane are specified. We followed a proposal in the literature to take the graph distance of nodes i and j, defined as the length, i.e. the number of edges, of the shortest path connecting i and j [14]. However, for most graphs this cannot be perfectly realised. Therefore, starting from an initial layout, the optimum is approximated in an iterative process called stress majorisation [14, 17, 23, 29], which is essentially a form of least squares optimisation. Users can choose whether network graphs generated for each iteration step are shown using argument `allfigures=TRUE`. A different starting layout than a circle can be chosen by argument `start.layout` (or abbreviated `start`): a starting layout obtained via eigenvectors of the Laplacian matrix (following Hall's algorithm [11]) or a random starting layout (`start="random"`). For Hall's algorithm, there are two possible options for setting the argument `start.layout`, `"eigen"`, or `"prcomp"`. These correspond to different procedures for computing eigenvectors by R function `eigen` or `prcomp` (via principal component analysis).

To see a series of random layouts, repeatedly execute the command (graph not shown):

```
> netgraph(mn1, start="random", iterate=TRUE,
+          col="darkgray", cex=1.5, multiarm=FALSE,
+          points=TRUE, col.points="green", cex.points=3)
```

If argument `points` is set to `TRUE`, nodes are marked. The marker type, size, and colour of the points can be specified using the arguments `pch.points`, `cex.points`, `col.points` which may be vectors. As usual, further generic plotting arguments are available, such as `cex` (determining the size of the treatment labels), `lwd` (determining the thickness of lines), `col` (determining the colour of lines), and others; see also help page of `netgraph` function.

In addition, the `netgraph` function allows us to identify multi-arm studies by coloured polygons. If argument `multiarm` is set to `FALSE` as in the command we have used, multi-arm studies are not identified in the figure.

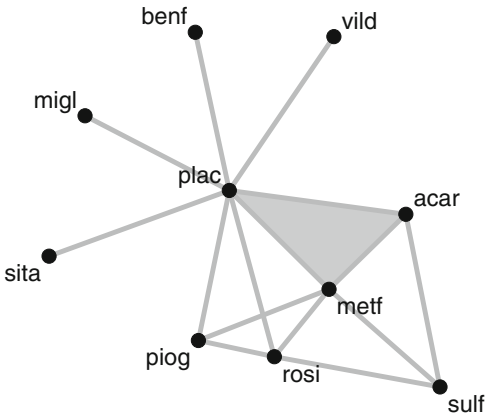

Fig. 8.4 Network graph for the diabetes data, drawn by the stress majorisation algorithm. The *shading* indicates the three-arm study

Using `multiarm=TRUE` (default) without explicitly specifying argument `col.multiarm` means that multi-arm studies are coloured using transparent colours from the R library **colorspace** if this is available, or colours from the rainbow scale otherwise. Unfortunately, transparent colours are not possible to display with all graphics devices. However, they can usually be incorporated into PDF-files. The function issues a warning if colours are not appropriately displayed.

As we have only one multi-arm study in our dataset, transparent colouring is not needed. Figure 8.4 is generated with the command

```
> netgraph(mn1, start="circle", iterate=TRUE,
+          col="darkgray", cex=1.5,
+          points=TRUE, col.points="black", cex.points=3,
+          col.multiarm="gray")
```

If you wish to view all the details of the iterations, then include the argument `allfigures=TRUE` (graphs not shown):

```
> netgraph(mn1, start="circle", iterate=TRUE,
+          col="darkgray", cex=1.5,
+          points=TRUE, col.points="black", cex.points=3,
+          col.multiarm="gray", allfigures=TRUE)
```

8.3.5 Forest Plots

Sometimes in a network meta-analysis the primary interest is to compare a number of treatments to a common treatment (also called reference or baseline treatment). This is usually placebo, usual care, no treatment, or a well-established standard treatment. When summarising the output, a reference treatment can be specified using the argument `reference.group` (abbreviated `ref`) in the `netmeta`, `summary.netmeta`, and the corresponding `print` functions.

```
> summary(mn1, ref="plac")
Number of studies: k=26
Number of treatments: n=10
Number of pairwise comparisons: m=28

Fixed effect model

Treatment estimate (sm="MD", reference.group="plac"):
          MD              95%-CI
acar -0.8274  [-1.0401; -0.6147]
benf -0.9052  [-1.1543; -0.6561]
metf -1.1141  [-1.2309; -0.9973]
migl -0.9439  [-1.1927; -0.6952]
piog -1.0664  [-1.2151; -0.9178]
plac  0.0000  [ 0.0000;  0.0000]
rosi -1.2018  [-1.2953; -1.1084]
sita -0.5700  [-0.8230; -0.3170]
sulf -0.4395  [-0.6188; -0.2602]
vild -0.7000  [-0.9495; -0.4505]
*** Output truncated ***
```

We see that in this printout all treatments are compared to the reference treatment, which is placebo.

A corresponding forest plot using reference group `"plac"` can be generated using the following R command:

```
> forest(mn1, ref="plac")
```

This produces a forest plot (graph not shown) for the fixed effect model because the analysis which created the R object `mn1` used the fixed effect model.

For the `forest.netmeta` function the argument `reference.group` is mandatory if this argument has not been used in the generation of the `netmeta` object. For example, the following R command will result in an error message.

```
> forest(mn1)
Error in forest.netmeta(mn1) :
  Argument 'reference.group' must match any of the following
  values: 'acar' - 'benf' - 'metf' - 'migl' - 'piog' - 'plac' -
          'rosi' - 'sita' - 'sulf' - 'vild'
```

Additional arguments are available in the `forest.netmeta` function to modify the figure. As always, the arguments of a function can be displayed using the `args` function.

```
> args(forest.netmeta)
function (x, pooled = ifelse(x$comb.random, "random", "fixed"),
    reference.group = x$reference.group, leftcols = "studlab",
    leftlabs = "Treatment", smlab = NULL, sortvar = x$seq, ...)
```

The argument `pooled` is used to explicitly specify whether the forest plot should be based on a fixed effect or random effects model. If `comb.random=FALSE` in the generation of the `netmeta` object, the fixed effect model is used. Otherwise, if `comb.random=TRUE`, the random effects model is used.

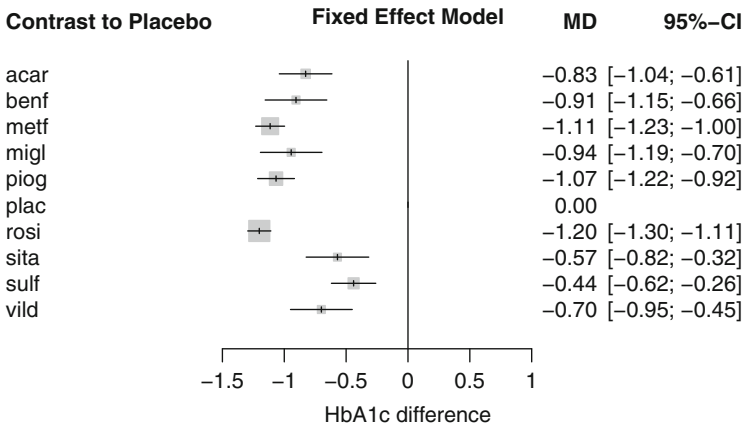

Fig. 8.5 Forest plot for the Senn data example, fixed effect model, with placebo as reference

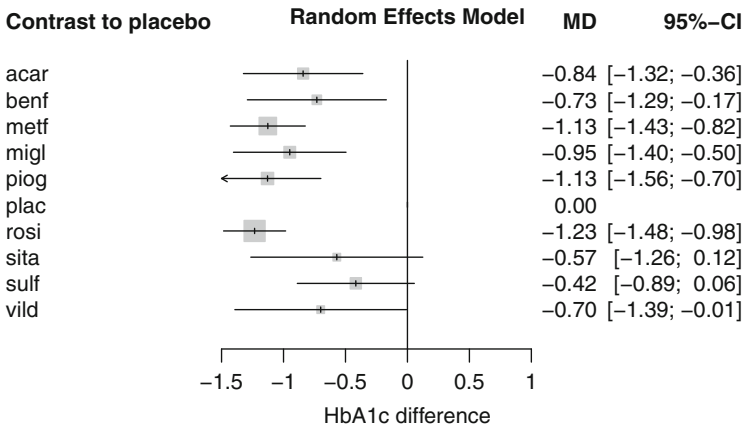

Fig. 8.6 Forest plot for the Senn data example, random effects model, with placebo as reference

Forest plots for fixed effect and random effects model are shown in Figs. 8.5 and 8.6, respectively. The following commands were used to produce these plots.

```
> forest(mn1, xlim=c(-1.5, 1), ref="plac",
+         leftlabs="Contrast to Placebo",
+         xlab="HbA1c difference")
> forest(mn1, xlim=c(-1.5, 1), ref="plac",
+         leftlabs="Contrast to placebo",
+         xlab="HbA1c difference",
+         pooled="random")
```

□

8.4 Decomposition of the Heterogeneity Statistic

The Q_{total} statistic (of the "whole network") can be decomposed into a Q statistic for assessing the heterogeneity between studies with the same design ("within designs") and a Q statistic for assessing the design inconsistency ("between designs"). Designs are defined by the subset of treatments compared in a study. For example, the comparison acarbose vs placebo in `Costa1997` and the comparison acarbose vs metformin vs placebo in `Willms1999` are two different designs in the Senn data example (see Fig. 8.2) even though the pairwise treatment comparison acarbose vs placebo is included in both designs.

Example 8.5 We can use the `decomp.design` function to calculate the Q statistics.

```
> round(decomp.design(mn1)$Q.decomp, 3)
                     Q df  pval
Whole network   96.986 18 0.000
Within designs  74.455 11 0.000
Between designs 22.530  7 0.002
```

We have used the fixed effect model for this analysis and we see there exists considerable heterogeneity/inconsistency within as well as between designs. We can further decompose the total within-design heterogeneity into the contribution from each design:

```
> print(decomp.design(mn1)$Q.het.design, digits=2)
           design     Q df    pval
1       acar:plac  0.00  0      NA
2       acar:sulf  0.00  0      NA
3       benf:plac  4.38  1 3.6e-02
4       metf:piog  0.00  0      NA
5       metf:plac 42.16  2 7.0e-10
6       metf:rosi  0.19  1 6.7e-01
7       metf:sulf  0.00  0      NA
8       migl:plac  6.45  2 4.0e-02
9       piog:plac  0.00  0      NA
10      piog:rosi  0.00  0      NA
11      plac:rosi 21.27  5 7.2e-04
12      plac:sita  0.00  0      NA
13      plac:vild  0.00  0      NA
14      rosi:sulf  0.00  0      NA
15 acar:metf:plac  0.00  0      NA
```

As we can see, 15 different designs are used in the 26 studies included in the network meta-analysis. Since there are only five designs for which we have more than one study, the remaining design specific Q statistics are equal to zero and have no degrees of freedom. Except for design `metf:rosi` ($p = 0.67$), for all the other four designs there is more heterogeneity between the contributing studies than we would expect by chance; in the case of `metf:plac` a substantial amount more ($p < 0.0001$). In a substantive application we would try and identify the sources of this and update the analysis appropriately.

The `decomp.design` function also gives the between-designs Q statistic based on a random effects model. This can be calculated based on a full design-by-treatment interaction random effects model [13]. Here, τ^2 is estimated by the method of moments [15]. Alternatively, the square-root of the between-study variance can be prespecified with `tau.preset`. This Q statistic can be displayed by entering:

```
> round(decomp.design(mn1)$Q.inc.random, 3)
                     Q df  pval tau.within
Between designs 2.194  7 0.948       0.38
```

□

8.5 The Net Heat Plot

We now introduce the net heat plot, proposed by Krahn, König, and Binder [18–20]. This is a graphical presentation which displays in a single plot two types of information. These are:

(1) for each network estimate, the contribution of each design to this estimate, and
(2) for each network estimate, the extent of inconsistency due to each design.

Taking (1) first, we have already seen in (8.6) that the elements in a row of the matrix **H** describe the contribution of the treatment comparison in each column to the network estimate in the row. The hat matrix **H** of dimension $m \times m$ was defined based on all individual pairwise treatment comparisons in the network, i.e. $m = 28$ treatment comparisons in the Senn data example. By contrast, the net heat plot is based on a condensed hat matrix with rows and columns corresponding to treatment comparisons within designs instead of single comparisons. This hat matrix therefore has lower dimension.

Example 8.6 Figure 8.7 shows a net heat plot for the fixed effect analysis of the Senn data example which was created using the following `netheat` command.

```
> netheat(mn1)
```

The rows and columns correspond to treatment comparisons within designs; treatment comparisons for which there is only one source of evidence are omitted. Pairwise comparisons corresponding to the three-arm design are designated by "_" following the treatment comparison label. The grey squares have area proportional to the contribution from the treatment comparison in the column to the treatment comparison in the row. For example, for the `acar:plac` treatment comparison, the largest grey square is on the diagonal indicating that the direct comparison is the greatest source of information. However, there are also moderate size squares in the same row for the `metf:sulf`, `rosi:sulf`, `plac:rosi`, `metf:plac`, and `acar:sulf` comparisons, indicating these are important sources of additional indirect evidence.

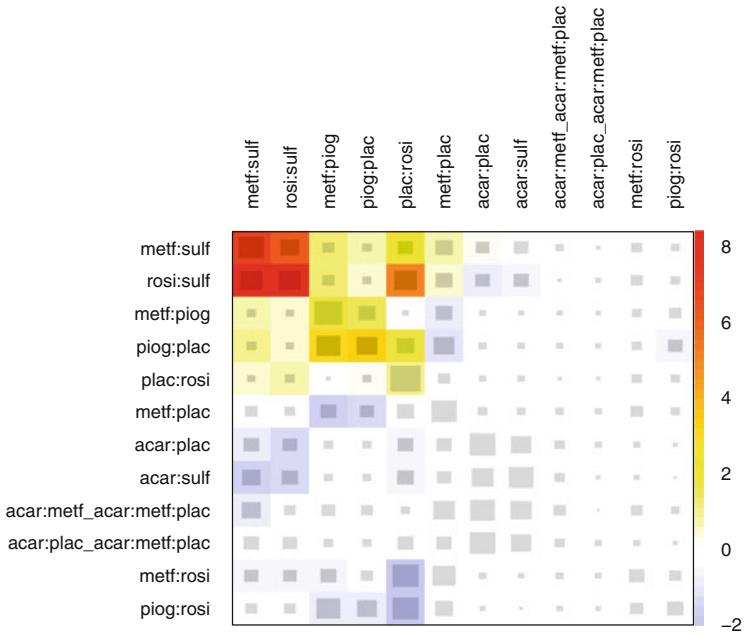

Fig. 8.7 Net heat plot of the Senn data example based on a fixed effect model

Now we consider (2). Recall that above we have decomposed the heterogeneity statistic Q_{total} into a within and between design component. We can further decompose the between design component into the contribution for each design. These are displayed in colour on the top-left to bottom-right diagonal of Fig. 8.7, with the largest heterogeneity shown in red in the top left corner. These correspond to the `metf:sulf` and `rosi:sulf` designs, which are together responsible for the majority of between-design heterogeneity.

We now turn to the off-diagonal colours. For each treatment comparison row, these are determined by the change in design inconsistency when a particular design is detached, i.e. after removing the consistency assumption for that specific design. For example, consider the top row of Fig. 8.7, corresponding to the `metf:sulf` treatment comparison. The (1,2) position is coloured red, which corresponds to a score of 6–8 in the colour scale on the right-hand side of the plot. This means that if we "remove" the assumption of consistency for the design in column 2 (`rosi:sulf`) and re-estimate the between-design inconsistency contribution to the design `metf:sulf`, it decreases. This means that the evidence for the treatment comparison `metf:sulf` from the design `rosi:sulf` is inconsistent with the other evidence.

Conversely, blue indicates that the evidence for the treatment comparison in the row from the design in the column is consistent. For example, for the

acar:sulf treatment comparison, the indirect evidence from the metf:sulf and rosi:sulf designs supports the direct evidence (coloured blue in the plot).

If the colours of a column corresponding to a design are identical to the colours on the diagonal, the detaching of the effect of this design removes the total inconsistency in the network.

As we have already noted, the diagonal colours show the designs metf:sulf, rosi:sulf, metf:piog, piog:plac, and plac:rosi contribute the most to the between-design inconsistency. The contributions of each design can also be printed as follows:

```
> round(decomp.design(mn1)$Q.inc.design, 2)
acar:plac      acar:sulf      benf:plac      metf:piog
     0.04           0.01           0.00           1.75
metf:plac      metf:rosi      metf:sulf      migl:plac
     0.20           0.01           6.62           0.00
piog:plac      piog:rosi      plac:rosi      plac:sita
     3.39           0.04           1.05           0.00
plac:vild      rosi:sulf acar:metf:plac acar:metf:plac
     0.00           9.29           0.01           0.13
```

We reiterate the point made above, that designs where only one treatment is involved in other designs of the network or where the removal of corresponding studies would lead to a splitting of the network do not contribute to the inconsistency assessment and are not incorporated into the net heat plot in Fig. 8.7. These are the four designs benf:plac, migl:plac, plac:sita, and plac:vild.

We see that the metf:sulf design contributes 6.62, and the rosi:sulf design contributes 9.29, consistent with the red colouring of the first and second diagonals of Fig. 8.7.

We have already commented on the red entries in positions (1,1), (1,2), (2,1) and (2,2) of Fig. 8.7. However, entries (4,3) and (4,4) are coloured orange, indicating inconsistency of evidence from the designs metf:piog and piog:plac.

Now consider the design plac:rosi (fifth row). There are six studies of this design in the network, and the within-design heterogeneity for this statistic, shown in Sect. 8.4 is large (21.27 on 5 degrees of freedom, $p < 0.001$). However, its large evidence base means it provides a lot of direct information (a large grey square in position (5,5)) as well as providing a lot of indirect information (as shown by the number of large grey squares in the plac:rosi column).

The colours in the metf:plac column are very light yellow in rows 1 and 2, then blue in rows 3 and 4, and then white. This means that relaxing the consistency assumption for this design slightly decreases the inconsistency contribution to the metf:sulf and rosi:sulf comparisons. However, the evidence from the metf:plac design is consistent with that from the metf:piog and piog:plac designs.

Overall, the heterogeneity in the network cannot be traced to one design. However, the single largest reduction in the whole network inconsistency is achieved by removing the rosi:sulf design (see the design inconsistency Q statistics above).

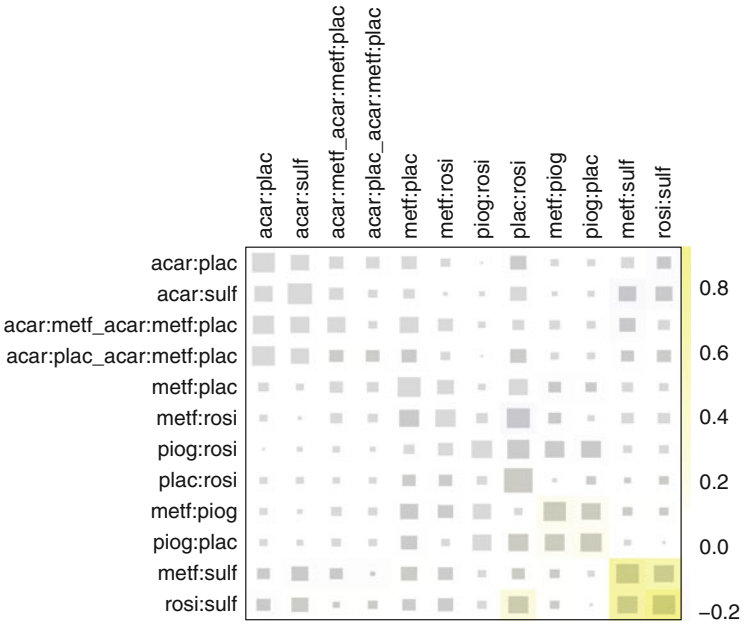

Fig. 8.8 Net heat plot of the Senn data example from a random effects model

Given the extent of heterogeneity, we may conclude a random effects analysis is more appropriate. The corresponding net heat plot is shown in Fig. 8.8, and shows a marked reduction in inconsistency. It is obtained by the following command:

```
> netheat(mn1, random=TRUE)
```

□

8.5.1 *Bland–Altman Plot to Assess the Effect of Heterogeneity on Estimated Treatment Comparisons*

In the Senn data example there is a substantial amount of heterogeneity. This is reflected in the fact that the estimated common heterogeneity variance $\hat{\tau}^2$ is larger than most of the study-specific sampling variances. Accordingly, alongside the fixed effect model, we might consider a random effects model.

Example 8.7 The results for the fixed effect and random effects model can be compared in a Bland–Altman plot [6] as follows:

```
> # Set seed so results are reproducible
> set.seed(125)
> fe <- mn1$TE.nma.fixed
> re <- mn1$TE.nma.random
```

```
> plot(jitter((fe+re)/2, 5), jitter(fe-re, 5),
+      xlim=c(-1.2, 1.2),
+      ylim=c(-0.25, 0.25),
+      xlab="Mean treatment effect (in fixed effect and random
            effects model)",
+      ylab="Difference of treatment effect (fixed effect minus
            random effects model)")
> abline(h=0)
```

The `jitter` function is used to separate overlapping points by adding a small random error to each x- and y-value. As the plot contains a random element, we use the `set.seed` command to generate a random, but reproducible, graph.

Figure 8.9 shows that estimated treatment effects of fixed effect and random effects model are similar with two exceptions. Estimated treatment effects in the random effects model are somewhat larger in the studies by `Stucci1996` and `Moulin2006` (bottom left of the plot).

A comparison of standard errors for fixed effect and random effects model shows that standard errors in the random effects model are much larger.

```
> summary(mn1$seTE.nma.random / mn1$seTE.nma.fixed)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.826   2.265   2.588   2.502   2.681   3.231
```

On average, standard errors in the random effects model (and accordingly confidence intervals) are about 2.5 times as large as those in the fixed effect model. However, treatment effect estimates are broadly similar. The results can be further illustrated using forest plots if desired (not shown). □

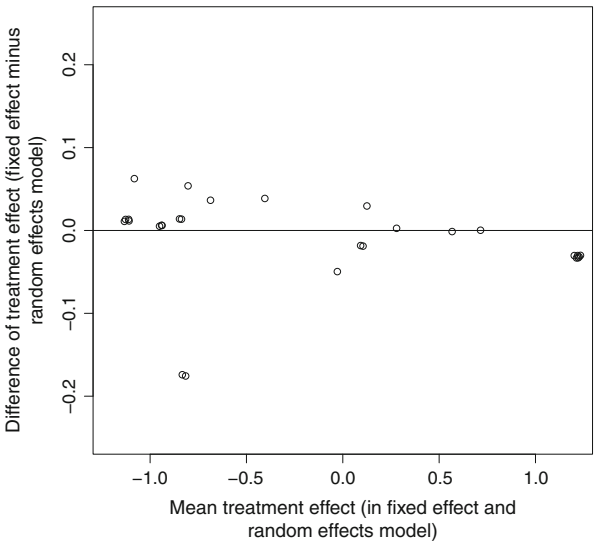

Fig. 8.9 Bland–Altman plot comparing individual treatment effects for fixed effect and random effects model

8.6 Summary

Network meta-analysis is a potentially powerful tool for using all the evidence in a particular area to estimate and compare treatment effects.

In this chapter, we have described a weighted least squares estimation approach which is implemented in the R package **netmeta**. The software can handle both single-arm and multi-arm studies; for the latter it accounts for the correlation appropriately. We have illustrated this approach using an example from diabetes [33], showing how the network can be graphed and a range of analyses explored.

Two important aspects of network meta-analysis are the extent of information gained on a particular treatment comparison through the indirect evidence and the extent of heterogeneity. Information about both of these is conveyed in the *net heat* plot; in addition the software provides a decomposition of heterogeneity within designs and between designs.

If there is clinically relevant heterogeneity, it should be explored further. As covariate adjustment is not currently possible with the software, one approach is to perform study specific (ideally individual participant data) analyses with appropriate covariate adjustment before using the software presented here to perform the network meta-analysis.

References

1. Achana, F., Hubbard, S., Sutton, A., Kendrick, D., Cooper, N.: An exploration of synthesis methods in public health evaluations of interventions concludes that the use of modern statistical methods would be beneficial. J. Clin. Epidemiol. (2013). doi:10.1016/j.jclinepi.2013.09.018
2. Albert, A.E.: Regression and the Moore-Penrose Pseudoinverse. Mathematics in Science and Engineering. Academic, New York (1972). ISBN:0-12-048450-1
3. Bafeta, A., Trinquart, L., Seror, R., Ravaud, P.: Analysis of the systematic reviews process in reports of network meta-analysis: methodological systematic review. Br. Med. J. **347**, f3675 (2013)
4. Bailey, R.A.: Designs for two-colour microarray experiments. J. R. Stat. Soc. Ser. C Appl. Stat. **56**(4), 365–394 (2007)
5. Bailey, R.A., Cameron, P.J.: Combinatorics of optimal designs. In: Huczynska, S., Mitchell, J.D., Roney-Dougal, C.M. (eds.) Surveys in Combinatorics. Mathematical Society Lecture Notes, vol. 365, pp. 19–73. Cambridge University Press, London (2009)
6. Bland, J.M., Altman, D.G.: Statistical methods for assessing agreement between two methods of clinical measurement. Lancet **327**(8476), 307–310 (1986). doi:10.1016/S0140-6736(86)90837-8
7. Bollobás, B.: Modern Graph Theory. Springer, Heidelberg/New York (2002)
8. DerSimonian, R., Laird, N.: Meta-analysis in clinical trials. Control. Clin. Trials **7**, 177–188 (1986)
9. Doyle, P.G., Snell, J.L.: Random Walks and Electric Networks. The Carus Mathematical Monographs. Mathematical Association of America, Washington, DC (1999)
10. Gutman, I., Xiao, W.: Generalized inverse of the Laplacian matrix and some applications. Bulletin T.CXXIX de l'Académie Serbe des Sciences et des Arts **29**, 15–23 (2004)

11. Hall, K.M.: An r-dimensional quadratic placement algorithm. Manage. Sci. **17**, 219–229 (1970)
12. Higgins, J.P.T., Thompson, S.G.: Quantifying heterogeneity in a meta-analysis. Stat. Med. **21**, 1539–1558 (2002)
13. Higgins, J.P.T., Jackson, D., Barrett, J.K., Lu, G., Ades, A.E., White, I.R.: Consistency and inconsistency in network meta-analysis: concepts and models for multi-arm studies. Res. Synth. Methods **3**(2), 98–110 (2012). doi:10.1002/jrsm.1044
14. Hu, Y.: Algorithms for visualizing large networks. In: U. Naumann, O. Schenk (eds.) Combinatorial Scientific Computing, pp. 525–549. Chapman and Hall/CRC Computational Science, Boca Raton, London, New York (2012). ISBN:9781439827352
15. Jackson, D., White, I.R., Riley, R.D.: Quantifying the impact of between-study heterogeneity in multivariate meta-analyses. Stat. Med. **31**(29), 3805–3820 (2012)
16. Jackson, D., White, I.R., Riley, R.D.: A matrix-based method of moments for fitting the multivariate random effects model for meta-analysis and meta-regression. Biom. J. **55**(2), 231–245 (2013). doi:10.1002/bimj.201200152
17. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. Inf. Process. Lett. **31**(1), 7–15 (1989)
18. König, J., Krahn, U., Binder, H.: Visualizing the flow of evidence in network meta-analysis and characterizing mixed treatment comparisons. Stat. Med. **32**, 5414–5429 (2013). doi:10.1002/sim.6001
19. Krahn, U., Binder, H., König, J.: A graphical tool for locating inconsistency in network meta-analyses. BMC Med. Res. Methodol. **13**(1), 35 (2013)
20. Krahn, U., Binder, H., König, J.: Visualizing inconsistency in network meta-analysis by independent path decomposition. BMC Med. Res. Methodol. **14**(1), 131 (2014)
21. Lee, A.W.: Review of mixed treatment comparisons in published systematic reviews shows marked increase since 2009. J. Clin. Epidemiol. **67**(2), 138–43 (2014). doi:10.1016/j.jclinepi.2013.07.014
22. Lu, G., Welton, N.J., Higgins, J.P.T., White, I.R., Ades, A.E.: Linear inference for mixed treatment comparison meta-analysis: a two-stage approach. Res. Synth. Methods **2**, 43–60 (2011). doi:10.1002/jrsm.34
23. Michailidis, G., de Leeuw, J.: Data visualization through graph drawing. Comput. Stat. **16**(3), 435–450 (2001)
24. Nietert, P.J., Wahlquist, A.E., Herbert, T.L.: Characteristics of recent biostatistical methods adopted by researchers publishing in general/internal medicine journals. Stat. Med. **32**(1–10) (2013). doi:10.1002/sim.5311
25. Paterson, L.: Circuits and efficiency in incomplete block designs. Biometrika **70**(1), 215–225 (1983)
26. Rao, C., Mitra, S.K.: Generalized Inverse of Matrices and its Applications. Wiley, New York, London, Sydney, Toronto (1971). ISBN:0-471-70821-6
27. Rücker, G.: Network meta-analysis, electrical networks and graph theory. Res. Synth. Methods **3**, 312–324 (2012)
28. Rücker, G., Schwarzer, G.: Reduce dimension or reduce weights? Comparing two approaches to multi-armed studies in network meta-analysis. Stat. Med. **33**(25), 4353–4369 (2014). doi:10.1002/sim.6236
29. Rücker, G., Schwarzer, G.: Automated drawing of network plots in network meta-analysis. Res. Syn. Meth. (2015). doi:10.1002/jrsm.1143
30. Rücker, G., Schwarzer, G., Krahn, U., König, J.: netmeta: network meta-analysis with R (2014). www.cran.R-project.org/package=netmeta. R package version 0.6-0
31. Salanti, G.: Indirect and mixed-treatment comparison, network, or multiple-treatments meta-analysis: many names, many benefits, many concerns for the next generation evidence synthesis tool. Res. Synth. Methods, 80–97 (2012). doi:10.1002/jrsm.1037
32. Salanti, G., Higgins, J.P., Ades, A.E., Ioannidis, J.P.: Evaluation of networks of randomized trials. Stat. Methods Med. Res. **17**(3), 279–301 (2008)

33. Senn, S., Gavini, F., Magrez, D., Scheen, A.: Issues in performing a network meta-analysis. Stat. Methods Med. Res. **22**, 169–89 (2013)
34. Spielman, D.: Spectral graph theory. In: U. Naumann, O. Schenk (eds.) Combinatorial Scientific Computing. Chapman and Hall/CRC Computational Science, Boca Raton (2012). ISBN:9781439827352
35. Veroniki, A.A., Vasiliadis, H.S., Higgins, J.P., Salanti, G.: Evaluation of inconsistency in networks of interventions. Int. J. Epidemiol. **42**(1), 332–345 (2013). doi:10.1093/ije/dys222
36. Yates, F.: The recovery of inter-block information in balanced incomplete block designs. Ann. Eugen. **10**(4), 317–325 (1940)