Research Synthesis Methods (2026), **0**: 1–20
doi:10.1017/rsm.2026.10123

Research Synthesis Methods

RESEARCH ARTICLE

Network meta-analysis and diffusion

Gerta Rücker[1], Annabel Louisa Davies[2] and Guido Schwarzer[1]

[1]Institute of Medical Biometry and Statistics, Medical Center – University of Freiburg, Germany
[2]Bristol Medical School, University of Bristol, UK

Corresponding author: Gerta Rücker; Email: gertaruecker@aol.com

Received: 17 April 2026 UTC; **Revised:** 11 August 2026 UTC; **Accepted:** 14 August 2026 UTC

Keywords: covariance; diffusion; geometric series; hat matrix; network meta-analysis; random walk

Abstract

We show that the covariance matrix of the treatment effect estimates in a network meta-analysis can be obtained without matrix inversion using a geometric series of diffusion matrices. This property extends to the hat matrix and provides a connection between parameter estimation in regression analysis and random walks on the network graph. We also provide a number of visualization tools implemented in R.

Highlights

What is already known?

Network meta-analysis (NMA) is a powerful tool in evidence-based medicine.

What is new?

The covariance matrix of the treatment effect estimates in an NMA can be obtained using a geometric series of diffusion matrices.

Potential impact for RSM readers

Our result provides new insights into the connection between NMA, diffusion, and random walks on networks.

1. Introduction

Network meta-analysis (NMA) is a powerful tool in evidence-based medicine. Users may choose between frequentist and Bayesian methods, and software for both is widely available. Both contrast-based and arm-based methods have been implemented.[1]

Contrast-based frequentist methods for NMA can be viewed from two perspectives: the regression perspective and the graph-theoretical perspective. Although formally equivalent and leading to exactly the same results, both perspectives provide different insights.[2] In this article, we show that two characteristic objects of weighted least squares regression, the variance–covariance matrix and the hat matrix, can be obtained using diffusion, a concept from graph theory related to random walks.

The article is structured as follows. After introducing three examples in Section 2, we present our concepts in Section 3. In Section 4, we suggest physical models to understand the connection between the different concepts. We apply the methods to the examples (Section 5), using the R package

This article was awarded Open Data and Open Materials badges for transparent practices. See the Data availability statement for details.

© The Author(s), 2026. Published by Cambridge University Press on behalf of The Society for Research Synthesis Methodology. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

netmeta [3,4] and a number of additional R functions for visualization, described in Appendix E and found online on Zenodo, see the "Data availability statement" section. The article ends with a discussion (Section 6).

2. Examples

We use a small toy example to illustrate the methods and apply them to two real-data NMAs.

2.1. Example 1

The toy example with five treatments and seven two-arm studies is shown in the left panel of Figure 1. All variances are assumed to be 1. The observed treatment effects for the comparisons A:B, A:E, B:C, B:D, B:E, C:D, and D:E are also assumed to be equal to 1.

2.2. Example 2

The second example is an NMA of 41 trials (including seven four-arm studies and three three-arm studies, making 82 pairwise comparisons in total) investigating the safety of six inhaled medications, including placebo, in patients with chronic obstructive pulmonary disease. The outcome was mortality, measured as binary, with the odds ratio as the effect measure.[5] The data set is available from the R packages *netmeta* and *metabook*.[4,6] The network graph is shown in the middle panel of Figure 1.

2.3. Example 3

The third example, shown in the right panel of Figure 1, is a network of 102 trials (including three three-arm trials) comparing seven drug and non-drug interventions to prevent pain on propofol injection, with hand vein as reference.[7] The binary outcome was pain, with relative risk (RR) as a measure of effect. The data set is available from the R package *metabook*.[6]

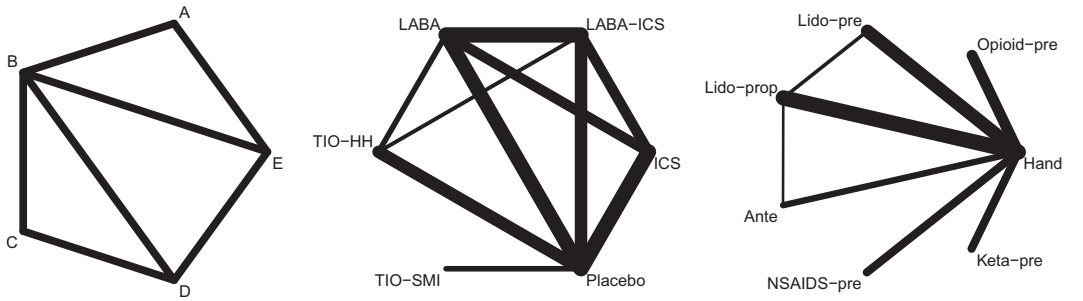


Figure 1. *Example network graphs. Left panel: A fictitious network with five treatments and seven studies. Middle panel: Network graph of the Dong 2013 data. ICS, inhaled corticosteroid; LABA, long-acting β2 agonist; TIO-HH, tiotropium dry powder delivered via HandiHaler; TIO-SMI, tiotropium solution delivered via Respimat Soft Mist Inhaler. Right panel: Network graph of the Jalota 2011 data. Ante, Antecubital vein; Hand, Hand vein; Lido-prop, Lidocaine–propofol admixture; Lido-pre, Lidocaine pretreatment; Keta-pre, pretreatment with ketamine; NSAIDS-pre, pretreatment with non-steroidal anti-inflammatory drugs; Opioid-pre, Pretreatment with opioids.*

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

3. Methods

Let us assume a connected network and let $\mathbf{X}$ be the design matrix of an NMA where the rows represent studies or comparisons (maybe aggregated over studies) and the columns belong to n treatments, represented as nodes in the network graph. Each row of $\mathbf{X}$ contains 1 in the column corresponding to one of the treatments in that study (or comparison), -1 in the column corresponding to the other treatment, and zeros elsewhere. Let $\mathbf{W}$ be the diagonal matrix of weights, after adjusting weights for multi-arm trials, where necessary.[8] For the common-effects model, we use inverse variance weights, and for a random-effects model, we choose an appropriate weight matrix based on one of the proposed estimators for the heterogeneity variance.[9,10] Comparisons for which no study provides information can (but need not) be represented in $\mathbf{W}$ with weight zero. The dimension of $\mathbf{W}$ must agree with the number of rows in $\mathbf{X}$.

Example 1

For our toy example, the weight matrix is equal to $\mathbf{W} = \mathbf{I}$, the 7×7 identity matrix, and the design matrix is

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

where the columns correspond to the five treatments (nodes) A, B, C, D, and E and the rows to the seven comparisons listed in Section 2.1 and shown in Figure 1, left panel. Alternatively, we may use the full design matrix, including the non-existent edges AC, AD, and CE and for $\mathbf{W}$ the 10×10 identity matrix with zeros at the diagonal positions of the non-existent edges. For real data, the study-based design matrix will be used, such that the rows correspond to studies and studies that compare the same treatments contribute identical rows.

We model the treatment effects in a contrast-based way. Let Y denote the vector of true treatment contrasts. The NMA model is

$$\mathrm{E}(Y|\mathbf{X}) = \mathbf{X}\beta$$

with a parameter vector β of length n, weight matrix $\mathbf{W}$, and $\mathrm{Var}(Y|\mathbf{X}) = \mathbf{W}^{-1}$. Based on this model, β is estimated by

$$\hat{\beta} = \mathbf{L}^{+}\mathbf{X}^{\top}\mathbf{W}\mathbf{y},$$

where $\mathbf{L} = \mathbf{X}^{\top}\mathbf{W}\mathbf{X}$ is the Laplacian matrix and $^{+}$ denotes the Moore–Penrose pseudoinverse. We point out that, in the absence of a baseline treatment, $\hat{\beta}$ is only determined up to an additive constant that cancels for $\mathbf{X}\hat{\beta}$.[2,8,11].

We now describe two matrices, the covariance matrix and the hat matrix of an NMA. The *covariance matrix* of the NMA-based treatment effect estimates is obtained as

$$\mathbf{C} = \mathbf{X}\,(\mathbf{X}^{\top}\mathbf{W}\mathbf{X})^{+}\mathbf{X}^{\top} = \mathbf{X}\,\mathbf{L}^{+}\mathbf{X}^{\top}$$

and the *hat matrix* $\mathbf{H}$ can be written as

$$\mathbf{H} = \mathbf{C}\mathbf{W} = \mathbf{X}\,(\mathbf{X}^{\top}\mathbf{W}\mathbf{X})^{+}\mathbf{X}^{\top}\mathbf{W} = \mathbf{X}\,\mathbf{L}^{+}\mathbf{X}^{\top}\mathbf{W}.$$

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

H and **C** are quadratic matrices of the same dimension as **W**. If **W** = **I**, the identity, we have **H** = **C**. Consistent contrasts are estimated by

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{H}\mathbf{y}.$$

The hat matrix is a projection matrix that maps the vector **y** of the observed pairwise contrasts from the studies (i.e., the data) to their consistent NMA counterparts, $\hat{\mathbf{y}}$. The length of **y** (which is the same as that of $\hat{\mathbf{y}}$) determines the number of rows in the design matrix **X**.

Each diagonal element of **L** corresponds to a treatment (node) in the network and is equal to the sum of the weights in the edges connected to that node, known as the weighted degree of the node. Therefore, we use the diagonal of **L** to define a vector **d** = diag(**L**), and a diagonal matrix **D** of weighted degrees. The diagonal elements of **D** are equal to **d** and the off-diagonal elements are equal to 0. The weighted adjacency matrix of the network is then given by

$$\mathbf{A} = \mathbf{D} - \mathbf{L}.$$

For the unweighted case (**W** = **I**), **A** is the usual adjacency matrix.

Now we define a transition matrix, also called a diffusion matrix, by

$$\mathbf{T} = \mathbf{A}\mathbf{D}^{-1} = (\mathbf{D} - \mathbf{L})\mathbf{D}^{-1} = \mathbf{I} - \mathbf{L}\mathbf{D}^{-1}. \tag{1}$$

This matrix describes a random diffusion process on the network. The concept of diffusion, related to that of random walks, plays a role in graph theory and its applications, particularly physics, see Spielman[12] and Subsection 10.1 in Spielman[13], and we applied it to NMA in an earlier paper.[14] Consider a random walker moving between nodes of the network; each element $\mathbf{T}_{ij}$ of the diffusion matrix describes the probability of a walker who is currently at node *j*, transitioning to node *i* in the next step. According to Equation (1), this transition probability is proportional to the weight associated with the edge connecting nodes *i* and *j* (normalized by the weighted degree of node *j*). Therefore, the walker is more likely to move along the edges with a larger weight. The movement of the walker is related to the flow of information in the network and the contribution of edges to the network-based treatment effect estimates for each comparison.

Example 1

For the toy example, we have

$$\mathbf{D} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ -1 & -1 & 0 & -1 & 3 \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} 0 & \frac{1}{4} & 0 & 0 & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{4} & 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & 0 & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{3} & 0 \end{pmatrix}.$$

The columns of **T** sum up to 1 and describe for each column (node) a multinomial distribution with probabilities for moving from the given node to each of its neighbors. For an unweighted network, non-zero probabilities within the same column are equal; for a general NMA, they become different. These probabilities provide the basis for the infinite diffusion process we describe in the next section.

3.1. A geometric sequence of diffusion matrices

For the following, we first assume that the network graph contains at least one loop of odd length, in other words, it is non-bipartite. We consider the geometric sequence of the powers of the diffusion

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

matrix, $(\mathbf{T}^i)_{i=0,1,2,\ldots}$. Under the assumption that this sequence is converging, its limit is the matrix

$$\mathbf{T}^\infty = \mathbf{d}_0\,\mathbf{1}^\top,$$

where $\mathbf{1}$ is the vector of ones of length n and $\mathbf{d}_0$ is the standardized vector of degree proportions, $\mathbf{d}_0 = \mathbf{d}/||\mathbf{d}||$, where $||\mathbf{d}||$ is the sum of the weighted degrees over all nodes. $\mathbf{T}^\infty$ has n equal columns that are all equal to $\mathbf{d}_0$ (see p. 10.1 in Spielman[13]).

If $\mathbf{p} = (p_1, \ldots, p_n)$ is a starting vector of probabilities $0 \le p_i \le 1, i = 1, \ldots, n$ and $\sum p_i = 1$, we have under the assumption of convergence

$$\mathbf{T}^\infty\mathbf{p} = \mathbf{d}_0\,\mathbf{1}^\top\mathbf{p} = \mathbf{d}_0 \sum p_i = \mathbf{d}_0, \tag{2}$$

justifying the notion of $\mathbf{T}$ as a diffusion matrix: starting from any distribution of a constant mass over the nodes, iterating $\mathbf{T}$ leads to a stable limit distribution of the mass such that each node obtains a mass proportional to its weighted degree.

Example 1

We apply the diffusion process to the toy example (Figure 1, left panel). Figure 2 shows the first six steps of the diffusion process when starting in node A (step 0). In the first step, the mass is equally divided between nodes B and E. In the following steps, the mass is dispersed across all available nodes. The limit distribution is reached with a precision of seven decimal places after 34 steps (see the R script available on Zenodo, as given in the "Data availability statement" section). It is given by

$$\mathbf{T}^\infty = \frac{1}{14}\begin{pmatrix} 2 & 2 & 2 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 \end{pmatrix}.$$

We note that we did not prove the convergence of the sequence $(\mathbf{T}^i)_{i=0,1,2,\ldots}$ and the existence of $\mathbf{T}^\infty$. In fact, for bipartite network graphs, this sequence does not converge but oscillates. We will treat this case in Section 3.2.

3.2. Bipartite networks

A bipartite graph is a graph that does not contain odd-numbered loops (cycles). As a result, the nodes can be split into two sets such that nodes from each set are exclusively connected to nodes from the other set and vice versa. Examples are networks without any loops, also called trees, such as star-shaped networks or networks containing only even-numbered loops, all of which occur frequently in NMA.

For bipartite networks, the geometric sequence of diffusion matrices, as defined above, does not converge. Instead, the diffusion matrices oscillate between the two sets of nodes and thus do not lead to a stable limit distribution. Oscillation is a well-known feature of bipartite graphs.[15,16] Fortunately, there is a pragmatic solution: We can define a so-called *lazy walk*, which means that in each step half of the mass remains in the current node and the other half continues to diffuse according to the edge weights. Following Spielman,[12,13] the transition matrix is then

$$\tilde{\mathbf{T}} = (\mathbf{T} + \mathbf{I})/2$$

(see p. 10.1 in Spielman[13]), and the limit of the sequence $\tilde{\mathbf{T}}^i$ again becomes

$$\mathbf{T}^\infty = \mathbf{d}_0\,\mathbf{1}^\top.$$

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

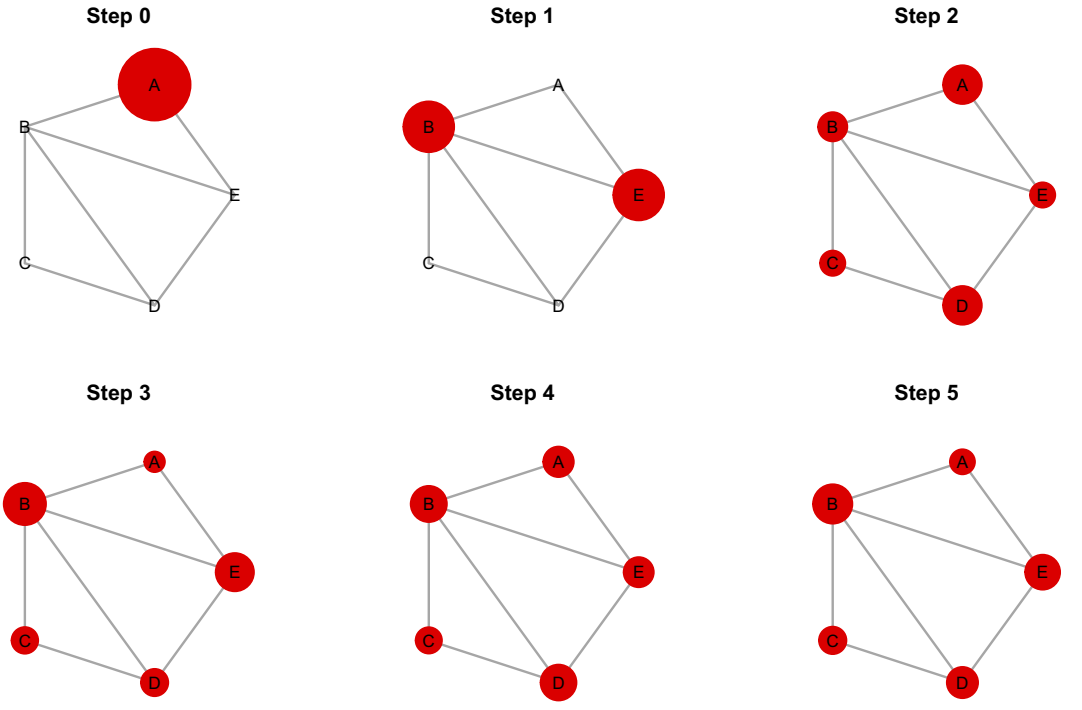


Figure 2. *Diffusion process for Example 1, starting in node A.*

As this is the more general version of diffusion, we use a lazy walk in most of the R functions described in Section E. By contrast to the lazy walk, we refer to the walk defined by $\mathbf{T}$ in Equation (1) as the *simple walk*.

3.3. A geometric series of diffusion matrices

If the sequence $\tilde{\mathbf{T}}^i \rightarrow \mathbf{T}^\infty \neq \mathbf{0}$ converges, the geometric series of the partial sums $\sum_{i=0}^{N} \tilde{\mathbf{T}}^i$ $(i = 0, 1, 2, \ldots, N)$ diverges. However, we can state the following theorem.

Theorem
For both bipartite and non-bipartite networks, the covariance matrix can be written using a geometric series of diffusion matrices:

$$\mathbf{C} = \frac{1}{2}\mathbf{X}\mathbf{D}^{-1} \sum_{i=0}^{\infty} \left(\tilde{\mathbf{T}}^i \mathbf{X}^\top\right). \tag{3}$$

For a proof of Equation (3), see Appendix A.

Corollary
For the hat matrix, we have

$$\mathbf{H} = \frac{1}{2}\mathbf{X}\mathbf{D}^{-1} \sum_{i=0}^{\infty} \left(\tilde{\mathbf{T}}^i \mathbf{X}^\top\right) \mathbf{W}. \tag{4}$$

This follows immediately from (3).

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

We note that although the sum of diffusion matrices $\sum_{i=0}^{\infty} \tilde{\mathbf{T}}^i$ diverges, Equations (3) and (4) first evaluate the column differences (via multiplication by $\mathbf{X}^\top$), which causes the sum to converge.

Remark
Equation (3) can also be written using $\mathbf{T}^\infty$:

$$\mathbf{C} = \frac{1}{2}\mathbf{X}\mathbf{D}^{-1}\sum_{i=0}^{\infty}\left(\tilde{\mathbf{T}} - \mathbf{T}^\infty\right)^i \mathbf{X}^\top. \tag{5}$$

In contrast to $\sum_{i=0}^{\infty} \mathbf{T}^i$ and $\sum_{i=0}^{\infty} \tilde{\mathbf{T}}^i$, the geometric series $\sum_{i=0}^{\infty}\left(\tilde{\mathbf{T}} - \mathbf{T}^\infty\right)^i$ is convergent; for details, see Appendix C.

Alternative: Diffusion with an absorbing node
As an alternative to the unlimited walk as described above, we can choose any treatment as the baseline, which means reducing the design matrix $\mathbf{X}$ by the respective column, giving $\mathbf{X}_r$, and reducing the other matrices by the respective column and row, giving $\mathbf{D}_r$ and $\mathbf{T}_r$ with dimension $n-1$. We call node r the *reference node*. As described in Appendix B, the reference node becomes absorbing, which means that all walks eventually end up there, and the geometric series $\sum_{k=0}^{\infty} \mathbf{T}_r^k$ becomes convergent. We then have

$$\mathbf{C} = \mathbf{X}_r\mathbf{D}_r^{-1}\sum_{i=0}^{\infty}\mathbf{T}_r^i\mathbf{X}_r^\top, \tag{6}$$

which holds in general, also for bipartite graphs.

Using a simplified version of the proof in Appendix A, it is possible to show that, for the special case of a non-bipartite graph, the covariance and hat matrices can also be found by replacing the reduced matrices $\mathbf{X}_r$, $\mathbf{D}_r$, and $\mathbf{T}_r$ in Equation (6) with their unreduced counterparts.

3.3.1. Finding NMA estimates iteratively
Equations (3)–(6) mean that we can obtain the covariance matrix $\mathbf{C}$ and the hat matrix $\mathbf{H}$ iteratively, without computing an inverse or pseudoinverse matrix. Given a vector of observed relative treatment effects $\mathbf{y}$, we can estimate the true effects by the consistent solution $\hat{\mathbf{y}}$, where

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{y} = \frac{1}{2}\mathbf{X}\mathbf{D}^{-1}\sum_{i=0}^{\infty}\left(\tilde{\mathbf{T}}^i\mathbf{X}^\top\right)\mathbf{W}\mathbf{y}.$$

This suggests approximating $\hat{\mathbf{y}}$ by a series of vectors

$$\hat{\mathbf{y}}_N = \frac{1}{2}\mathbf{X}\mathbf{D}^{-1}\sum_{i=0}^{N}\left(\tilde{\mathbf{T}}^i\mathbf{X}^\top\right)\mathbf{W}\mathbf{y},$$

which perform the sum up to some finite value N. The goodness of the approximation can be measured by the weighted sum of squared differences

$$(\hat{\mathbf{y}}_N - \hat{\mathbf{y}})^\top\mathbf{W}(\hat{\mathbf{y}}_N - \hat{\mathbf{y}}),$$

starting with $Q = (\mathbf{y} - \hat{\mathbf{y}})^\top\mathbf{W}(\mathbf{y} - \hat{\mathbf{y}})$. The examples (see the R script) often show a remarkably good approximation already in the first iteration steps. To understand why, we look at the very first step, $N = 0$. For $N = 0$, we have $\tilde{\mathbf{T}}^0 = \mathbf{I}$ and

$$\hat{\mathbf{y}}_0 = \frac{1}{2}\mathbf{X}\mathbf{D}^{-1}\sum_{i=0}^{0}\left(\tilde{\mathbf{T}}^i\mathbf{X}^\top\right)\mathbf{W}\mathbf{y} = \frac{1}{2}\mathbf{X}\mathbf{D}^{-1}\mathbf{X}^\top\mathbf{W}\mathbf{y}.$$

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

Due to the structure of $\mathbf{X}$, in this first step, the observations $\mathbf{y}$ are mapped onto a set of $n-1$ consistent relative effects, $\hat{\mathbf{y}}_0$ (see Section 3.1.2 in Rücker and Schwarzer[2]). Particularly, entries of $\mathbf{y}$ from different studies that represent the same comparison are mapped to the same value in $\hat{\mathbf{y}}_0$. As N increases, these consistent effects move closer to the set of effects that minimize the weighted sum of squared differences.

4. Interpretation

For any transition matrix $\mathbf{T}$ (also $\tilde{\mathbf{T}}$, $\mathbf{T}_r$), each element (e.g., $\mathbf{T}_{ij}$) describes the probability of a transition from node j to node i. Element (i, j) of $\mathbf{T}^2$ is given by

$$\mathbf{T}^2_{ij} = \sum_{k=1}^{n} \mathbf{T}_{ik}\mathbf{T}_{kj},$$

which describes the probability of arriving at node i after two steps when starting in node j. An analogous interpretation holds for $\mathbf{T}^3_{ij}, \mathbf{T}^4_{ij}, \ldots$ and so on.

4.1. Illustration 1: Length of stay

Consider a walker who starts in node A (step 0), switches to node B in step 1, then comes back to node A in step 2, and leaves A again in step 3. Until now, this walker has spent a total of two steps in node A. Now consider many such walkers on the network, who have a probability p_i of starting in node i. The expected total time a walker spends in node i after N steps is calculated by counting the number of walkers in node i at step k, then summing these counts up to a total number of steps N, and dividing the result by the total number of walkers. Therefore, the vector of expected times spent in each node is given by

$$\sum_{k=0}^{N} \mathbf{T}^k\mathbf{p},$$

where $\mathbf{p} = (p_1, \ldots, p_n)^\top$ is the starting distribution of the walk. For example, setting $\mathbf{p}$ equal to the first unit vector gives the first column of $\sum_{k=0}^{N} \mathbf{T}^k$, which contains the expected length of stay in each node up to step N for those who start their walk in node 1. The larger N becomes, the less this time depends on the starting point. However, unlike the stationary distribution of the walk in Equation (2), which describes the *fraction* of time spent in each node (and converges to the weighted degree of the node for large N), the *total* time until step N retains the dependence on the starting point. We also note that, because we are dealing with discrete time, the expected time spent in a node is the same as the expected number of visits to that node.

4.2. Illustration 2: Walkers and drinks

To provide a perhaps more comprehensible interpretation of times, we now mark the network nodes, for example, by placing colored flags, such as yellow, orange, red, green, and so on. Large numbers of walkers, wearing T-shirts in the same colors, are placed at "their own" node. There are also n drinks in various colors available, such as lemon juice, orange juice, red currant juice, kiwi juice, etc. We place n bottles, each with one of the different drinks, at each node, as shown in Figure 3 for Example 1 ($n = 5$). As before, we are working with discrete time. Before starting their walk, each walker takes a small, standardized sip of "their own" drink at their starting node; for example, "yellow" walkers take a sip of lemon juice. Then, all walkers simultaneously start stepwise dispersing through the network, taking each branch with probability proportional to its weight and taking a standardized sip of "their own"

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

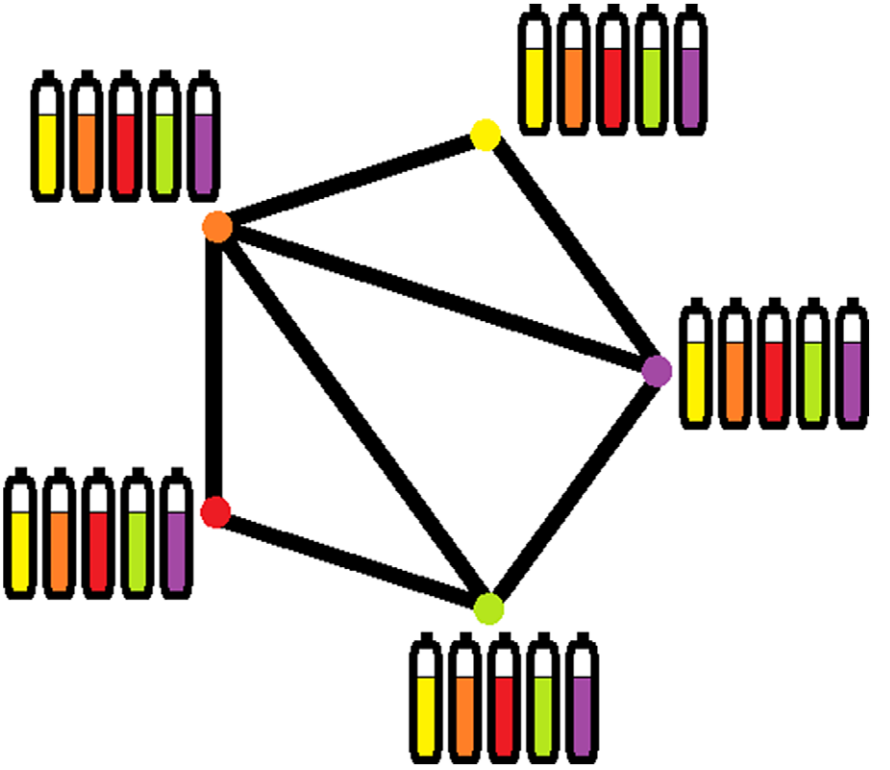


Figure 3. *A network with five colored nodes and corresponding drinks. Walkers are spreading through the network in steps, taking a sip of their own drink at every station they visit.*

drink at every station they reach. At any time, all walkers have had the same juice intake. However, nodes with a larger (weighted) degree have been visited by more walkers, and the bottles at those nodes contain less juice.

4.2.1. Scaling—Reducing by degree

After a large number of N steps, the remaining juice in each bottle is now further reduced according to the node's (weighted) degree. For example, if the degree is 2, half of every bottle is poured away; if the degree is 3, two-thirds of each bottle are removed. This action, which we call *scaling*, corresponds to multiplying the geometric series $\sum_{k=0}^{N} \mathbf{T}^k$ by $\mathbf{D}^{-1}$ on the left.

4.2.2. Column differences

Multiplication with $\mathbf{X}^\top$ on the right means comparing the remaining volume of juice between pairs of bottles of different colors at the same node, for each node in the network. These differences are only due to the different starting positions of the walkers.

4.2.3. Row differences

Multiplication with $\mathbf{X}$ on the left means comparing the differences above between nodes. The resulting "differences of differences" in volume then correspond to the covariances in NMA (see below). We note that, because we are calculating differences of differences, it is not necessary that the size of bottles at the beginning is equal for each node. The only requirement is that all bottles must start with enough liquid to facilitate the total number of walkers removing liquid over N steps.

The walkers-and-drinks model suggests an interpretation of the NMA covariance matrix in terms of the long random walk. We measure the difference between the amount of A-juice and B-juice,

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

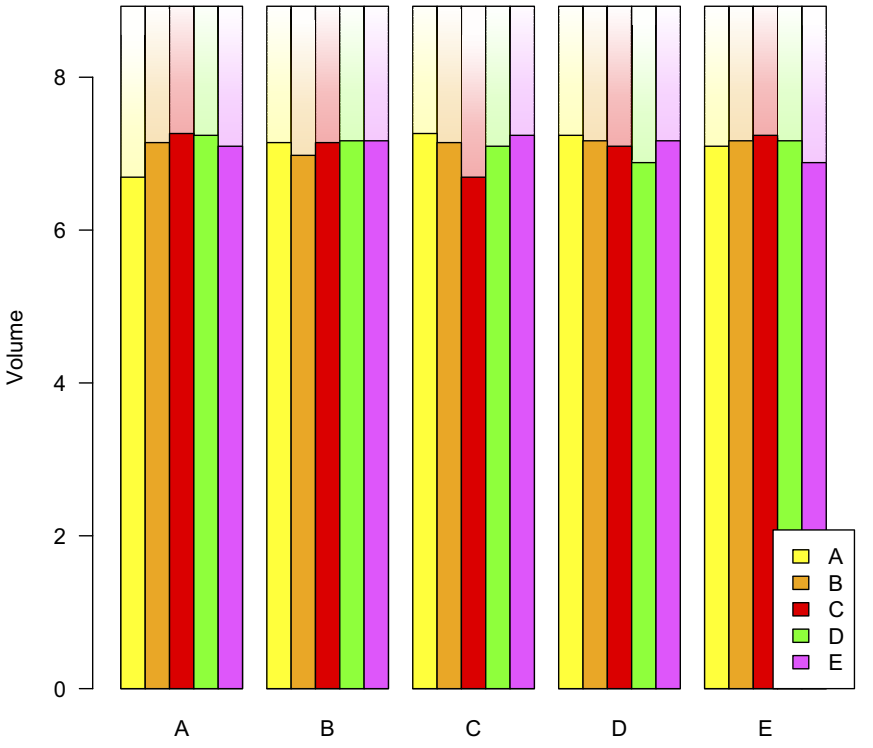


Figure 4. *Walkers-and-drinks model for Example 1. The bars represent bottles with juice left after 50 steps and scaling. Each block of bars belongs to a node, each bar in a particular color belongs to the walkers from a particular node. Yellow = A, orange = B, red = C, green = D, and violet = E.*

remaining at A, and the difference between the amount of A-juice and B-juice, remaining at B. We write the difference of differences as

$$(A_A - B_A) - (A_B - B_B) = (A_A + B_B) - (A_B + B_A) = A_A + B_B - 2\,A_B,$$

where we have used the fact that, due to symmetry, $A_B = B_A$ (see Appendix D). This may be interpreted as the difference between the fluid intake "at home" (i.e., A in A and B in B) and intake "at one's neighbor's" (A in B or B in A). We have implemented the diffusion process in a number of R functions described in Appendix E.

5. Results

5.1. Example 1

We have shown the diffusion process for the toy example already in Section 3.1 (Figure 2). Figure 4 illustrates the walkers-and-drinks interpretation. The bars represent the remaining juice in the bottles after 50 walking steps and pouring away part of the juice according to the node's degree. The differences of differences between the nodes correspond to the covariances in NMA. For example, the yellow and the red bar in B have the same height, 7.14, but in A the red bar is higher, 7.26, than the yellow bar, 6.69, with a difference of 0.57. This is the covariance matrix entry for (A:B, A:C).

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

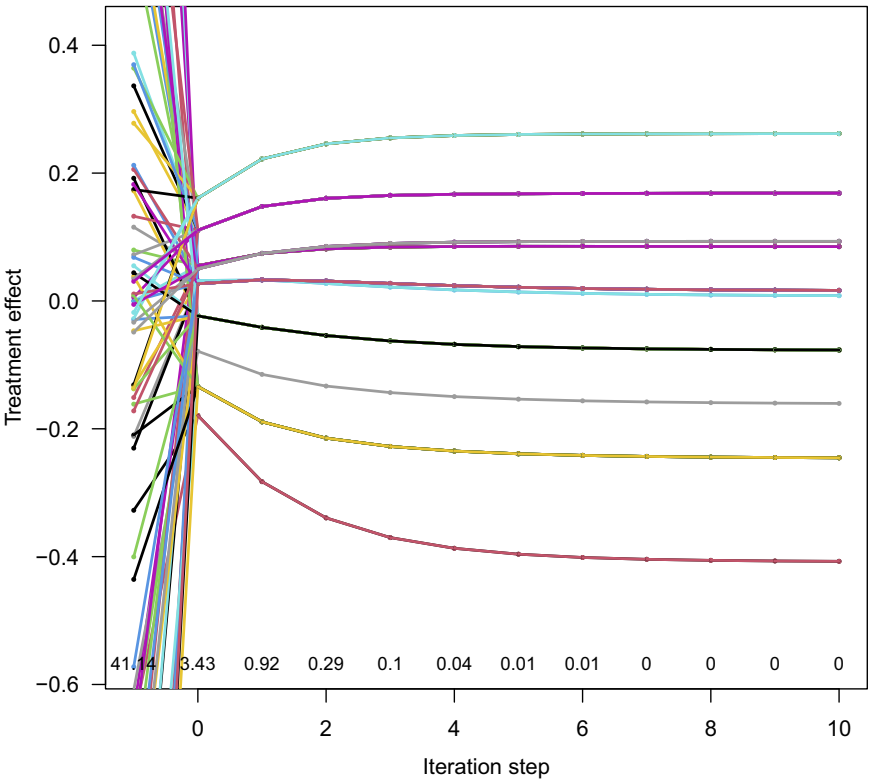


Figure 5. *Iterative estimation of the treatment effects for Example 2. The starting points of the lines (to the left of step 0) are the observed treatment effects from 82 comparisons. Each number at the bottom represents the weighted sum of squared distances of the estimates from the final NMA estimates.*

5.2. Example 2

Figure 5 shows how the estimated treatment effects of the second example (41 studies with 82 direct comparisons) evolve with each iteration of the geometric series that approximates the hat matrix, here using the lazy walk method. For the comparisons with direct evidence, the treatment effects converge quickly from their direct estimates to their network estimates (note that two pairs of estimates are very similar and cannot be easily distinguished in the figure). As noted in Section 3.3.1, the 82 direct observations are mapped to only 10 consistent estimates already in iteration step 0.

Figure 6 illustrates the walkers-and-drinks interpretation for this data. The most noticeable feature of this graph is the short blue bar at its corresponding node, TIO-SMI. As we see from the network graph (Figure 1, middle panel), this node is a spur, connected only to placebo and weakly, via only two trials. The height difference between the blue bar (TIO-SMI) and every other is small at all other nodes but is large at node TIO-SMI. This causes the difference in differences to be particularly large for all comparisons involving TIO-SMI. This makes sense as, by Equation (3), these differences in differences equal the variances of these comparisons, which will be large because they rely heavily on the two studies comparing TIO-SMI and placebo.

5.3. Example 3

As Figure 1 (right panel) shows, the network graph is almost star-shaped, with weak connections between treatments Ante and Lido-prop (one three-arm study) and between Lido-pre and Lido-prop (two three-arm studies). This graph is not bipartite, but could be seen as "almost bipartite," as

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

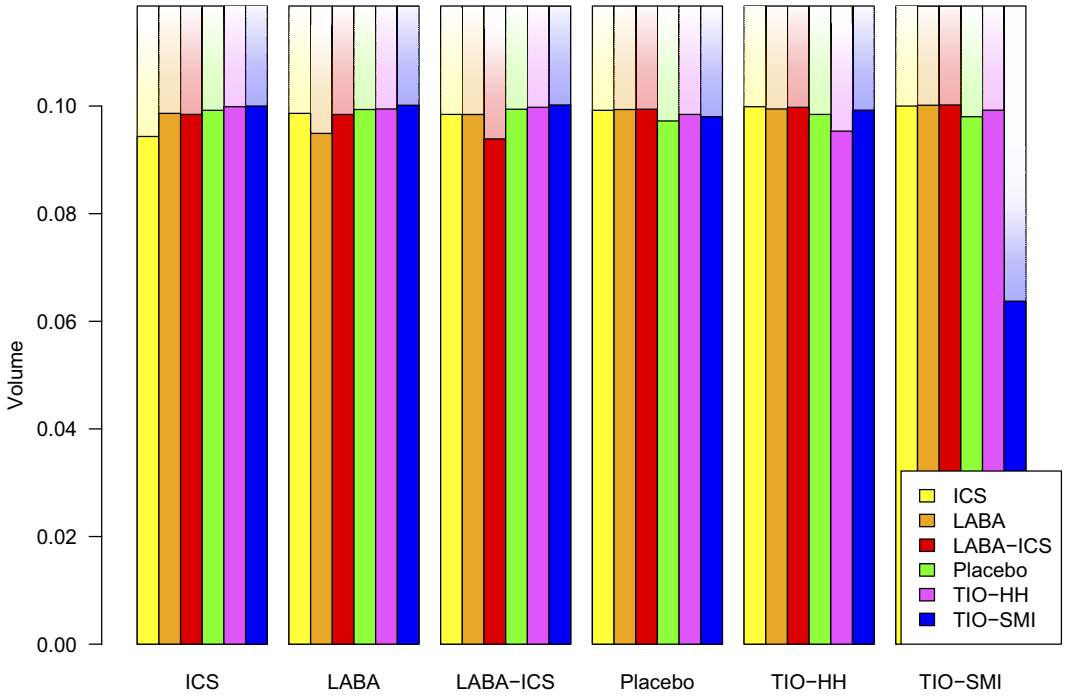


Figure 6. *Walkers-and-drinks model for Example 2. The bars represent bottles with juice left after 50 steps and scaling. Each block of bars belongs to a node, each bar in a particular color belongs to the walkers from a particular node. Yellow = ICS, orange = LABA, red = LABA-ICS, green = Placebo, violet = TIO-HH, and blue = TIO-SMI.*

demonstrated by Figure 7, which shows 3,000 steps of simple diffusion in the form of a stacked probability plot, starting with equal mass at each node and ending with the proportional weighted degrees. We see oscillatory behavior for about 2,000 steps.

Figure 8 shows four versions of the iteration for the relative treatment effect estimates. In Figure 8 (top left), we see that the simple walk (Equation (1)) oscillates very strongly, it needs about 3,000 steps to converge (see also Figure 7). By contrast, the lazy walk (Equation (4), top right) converges after only 10 steps. The bottom plots show the convergence of absorbing walks (Equation (6)) with two different reference treatments. The bottom left plot shows oscillatory behavior for the first 10 steps of the absorbing walk with reference treatment Lidocaine–propofol admixture. The bottom right plot shows fast convergence for the absorbing walk with reference treatment Hand vein. This example shows that oscillation may also occur for non-bipartite graphs and demonstrates that absorbing walks tend to converge most quickly if a central node (i.e., a node with maximum weighted degree) is chosen as reference.

6. Discussion

In this article, we have shown that the covariance matrix of the network estimates in an NMA can be written as a geometric series of diffusion matrices, framed by the design matrix and its transpose. A similar representation exists for the hat matrix.

This work was partly motivated by a method called network deconvolution that was introduced in the causal inference literature.[17,18] These authors assume that similarities between variables, or even directed causal effects, can be described by some matrix, such as a correlation matrix. The powers of this matrix, suitably scaled, are summed up to provide a matrix of "total effects" called the "transitive

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

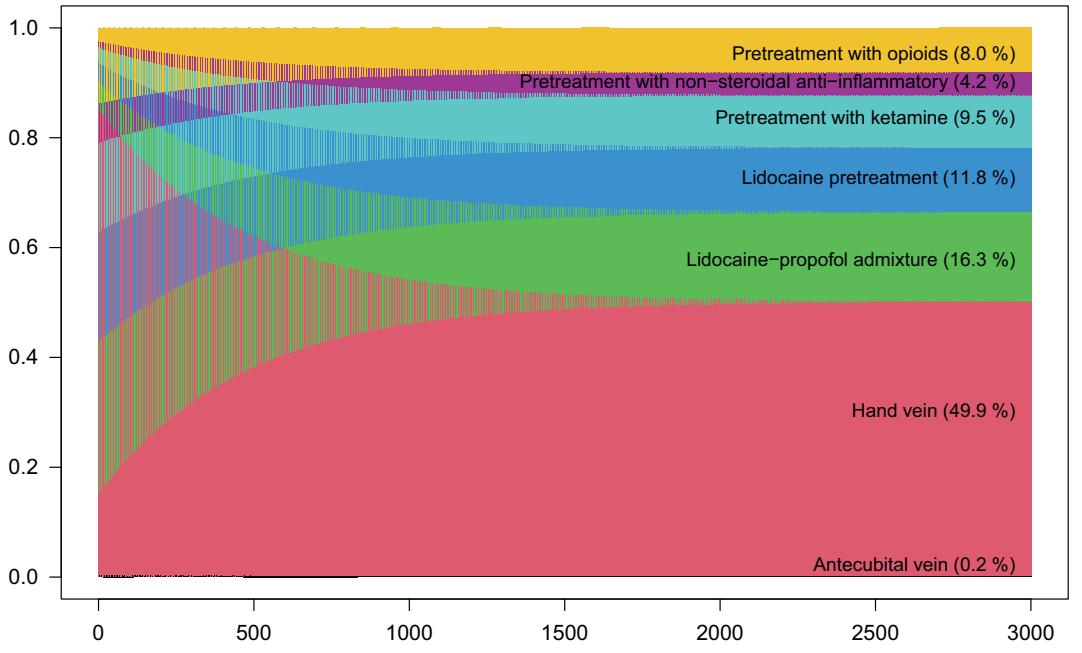


Figure 7. *Simple diffusion for Example 3, starting with uniform distribution. The numbers are the percentage weighted degrees.*

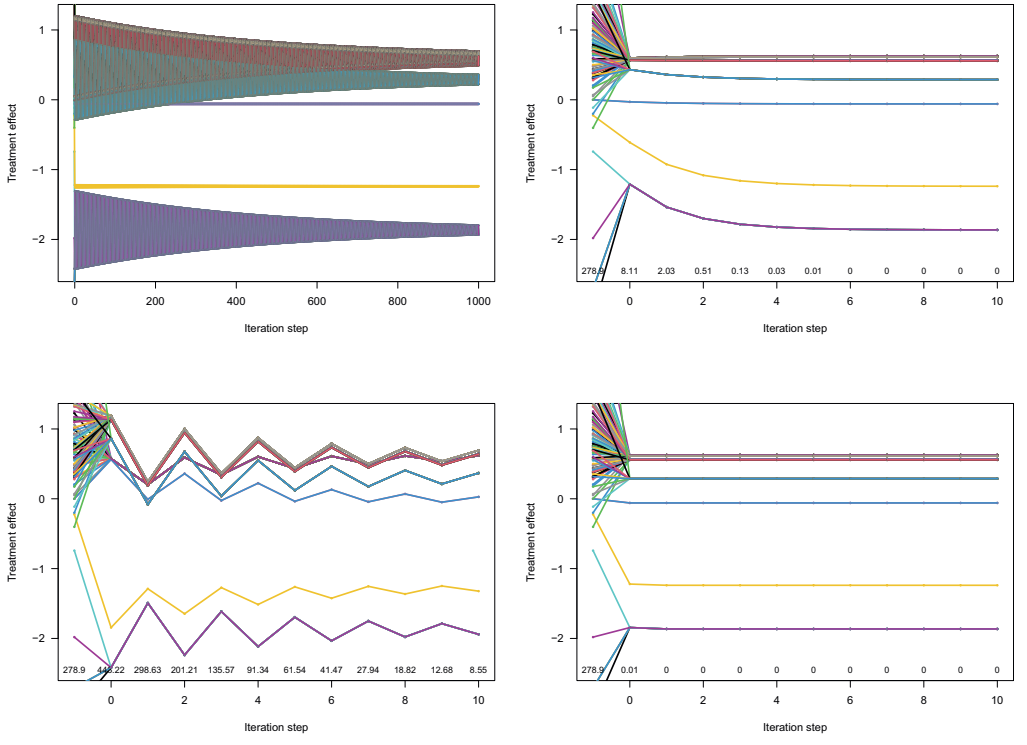


Figure 8. *Iterative estimation of the treatment effects for Example 3. The starting points of the lines are the observed treatment effects from 108 comparisons. Top left: Simple method, 1,000 steps; Top right: Lazy walk, 10 steps. Bottom left: Absorbing walk, reference is Lidocaine–propofol admixture, 10 steps. Bottom right: Absorbing walk, reference is Hand vein, 10 steps. Each number at the bottom represents the weighted sum of squared distances of the estimates from the final NMA estimates.*

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

closure" that can be observed. This process is also known as network convolution. The inverse process of identifying the starting matrix, given its transitive closure, is called network deconvolution. In our context of NMA, this would correspond to finding the diffusion matrix from a given covariance matrix, which would be a different task. Apart from that, it seems questionable whether multiplying correlation matrices makes sense, given that correlation is not transitive.

Matrix power series have many applications, for example, in physics (random walks on electrical networks[19]), chemistry (symmetry recognition in chemical compounds[20]), or biology (Krylov subspaces[21,22]). A geometric series of matrices is also called a Neumann series.[23] Finite Neumann series are used in applications to approximate the inverse of a matrix by iteration; for an example, see Wu et al.[24]

The iterative calculation of the covariance matrix naturally allows a similar iteration for the hat matrix and thus an iterative method for finding the network estimates that, in most of our examples, is fast converging. For the version with an absorbing node, the speed of convergence depends on the choice of the reference node. We experienced the best convergence when choosing a central node (e.g., placebo as a frequently used control treatment, or hand vein for Example 3). This is plausible, as a central node rapidly absorbs all the mass. Thus, the reduced transition matrix, described in Appendix B, converges quickly to the null matrix, causing the geometric series to reach its limit.

Example 3 demonstrates that for nearly star-shaped networks, with a distinguished central node, the version with the central node chosen as absorbing is clearly superior. Although the algorithm tends to be slow for complex networks, it seems generally to be numerically stable. Of note, even complex treatment networks are relatively small compared to large graphs in other application areas.

More generally, the speed of convergence is driven by the features of the network on the one hand and the chosen diffusion method on the other hand. Network features are the density of the network and its extent of "bipartiteness" (leading to oscillation, except for the lazy walk method). Density (or the degree of connectedness) can be measured, for example, by the second-smallest eigenvalue of the Laplacian matrix, also called the Fiedler value (which needs to be standardized for weighted networks). Similarly, bipartiteness can be measured by the magnitude of the smallest eigenvalue of the lazy diffusion matrix (if it equals 0, the graph is bipartite; see Subsection 1.5.4 in Spielman[13]). With respect to the method, the method with an absorbing node seems generally preferable, provided that a central node (ideally the one with maximum weighted degree) is chosen as the absorbing node.

At the comparison-aggregated level, $\mathbf{H}_{AB,AB}$ represents the proportion of direct evidence from edge AB for comparison A:B.[25–28] With study-level input, the individual diagonal elements represent study-specific leverages; their sum over all study contrasts corresponding to edge AB gives the corresponding edge-level quantity.[2,26,29]

Another interpretation of the hat matrix comes from the area of electrical circuits. Davies et al. showed that entry $\mathbf{H}_{AB,CD}$ can be interpreted as the current flowing through edge CD when a battery is attached across A and B.[14] Based on an analogy between electrical networks and random walks, this is equal to the expected net number of times a walker who starts a random walk from node A crosses the edge CD before eventually arriving at node B. Our illustration with times adds another interpretation: If we choose a reference node, say B, this node becomes absorbing, all walks eventually end in B, and the expected times in all other nodes become finite (see Appendix B). It turns out that $\mathbf{H}_{AB,CD}$ corresponds to the scaled difference in the expected total length of stay in A between walkers who start in C and those who start in D, all walkers eventually arriving in B.

In fact, our work highlights the potential for even further connections between NMA and the study of complex networks in other disciplines. For example, the effective resistance of a graph takes the same form as the variance in NMA.[8] In the random walk literature, there is an established interpretation for effective resistance in terms of "escape probability," that is, the probability that a walker who starts in node A, reaches node B before they return to A.[19] More recently, Estrada interpreted effective resistance as a diffusion distance.[30] By defining a diffusion process on a graph, he shows that the effective resistance between two nodes is equivalent to a difference in mass concentrations between the

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

nodes, relative to their initial conditions. This result is closely connected with our own and warrants further investigation.

Bipartite networks, which occur frequently in NMA, have a special structure that must be taken into account. A bipartite network graph must not be confused with a different application of bipartite graphs that occurs when an NMA is represented as a hypergraph, as suggested recently.[31,32] A hypergraph is a generalized graph, where generalized "hyperedges" (in the context of NMA, studies) can join more than two vertices (here treatments, thus including multiarm studies). Each NMA hypergraph can be interpreted as a bipartite graph that consists of two sets of vertices, one representing the treatments, the other representing the studies; both sets are joined by edges where a treatment was investigated in a study.

In the present article, we concentrated on the common-effects model. As mentioned, all considerations can be extended to the random-effects model by choosing an appropriate weight matrix. We note, however, that this does not address uncertainty in the estimation of the random-effects heterogeneity variance. As an example, we provide the analog to Figure 8 for the random-effects model in Appendix F.

We are aware that the phenomena described in this article, interesting though they may be, have no direct consequences for NMA application practice, as all essential NMA calculations are easily conducted by standard software. Nevertheless, we believe that our findings reveal a nice connection between regression analysis for NMA and the concepts of diffusion and random walks in network analysis, and they may stimulate further theoretical developments. For example, matrix inverses sometimes make theoretical proofs difficult, and being able to express these as geometric series may help with this sort of work.

Author contributions. Conceptualization: G.R.; Data curation: G.R.; Formal analysis: G.R.; Methodology: G.R., A.L.D.; Project administration: G.R.; Software: G.R., G.S.; Visualization: G.R., G.S.; Writing—original draft: G.R., A.L.D., G.S.; Writing—review and editing: G.R., A.L.D., G.S.

Competing interest statement. The authors declare that no competing interests exist.

Data availability statement. The R script and datasets supporting this work are openly available on Zenodo at https://zenodo.org/records/23134102.

Funding statement. A.L.D. is funded by the Engineering and Physical Sciences Research Council (EP/Y007905/1).

References

[1] Ades AE, Welton NJ, Dias S, Phillippo DM, Caldwell DM. Twenty years of network meta-analysis: continuing controversies and recent developments. *Res Synth Methods*. 2024;15: 702–727.
[2] Rücker G, Schwarzer G. Reduce dimension or reduce weights? Comparing two approaches to multi-arm studies in network meta-analysis. *Stat Med*. 2014;33: 4353–4369.
[3] Balduzzi S, Rücker G, Nikolakopoulou A, et al. netmeta: an R package for network meta-analysis using frequentist methods. *J Stat Softw*. 2023;106: 1–40.
[4] Rücker G, Krahn U, König J, et al. netmeta: network meta-analysis using frequentist methods. R package version 3.6-1. 2026. https://cran.r-project.org/web/packages/netmeta/.
[5] Dong YH, Lin HH, Shau WY, Wu YC, Chang CH, Lai MS. Comparative safety of inhaled medications in patients with chronic obstructive pulmonary disease: systematic review and mixed treatment comparison meta-analysis of randomised controlled trials. *Thorax*. 2013;68: 48–56.
[6] Schwarzer G. metabook: data sets and code for "meta-analysis with R". R package version 0.2-0. 2026. https://doi.org/10.32614/CRAN.package.metabook.
[7] Jalota L, Kalira V, George E, et al. Prevention of pain on injection of propofol: systematic review and meta-analysis. *BMJ*. 2011;342: d1110. https://doi.org/10.1136/bmj.d1110
[8] Rücker G. Network meta-analysis, electrical networks and graph theory. *Res Synth Methods*. 2012;3: 312–324.
[9] Veroniki AA, Jackson D, Bender R, et al. Methods to calculate uncertainty in the estimated overall effect size from a random-effects meta-analysis. *Res Synth Methods*. 2019;10: 23–43.
[10] Langan D, Higgins JPT, Jackson D, et al. A comparison of heterogeneity variance estimators in simulated random-effects meta-analyses. *Res Synth Methods*. 2019;10: 83–98.
[11] Albert AE. *Regression and the Moore-Penrose Pseudoinverse*, Mathematics in Science and Engineering. Academic Press; 1972.

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

[12] Spielman D. Spectral graph theory. In: Naumann U, Schenk O, eds. *Combinatorial Scientific Computing*, Computational Science. Chapman and Hall/CRC; 2012: 495–524. Chap. 18.
[13] Spielman DA. Spectral and algebraic graph theory. Tech. Rep. New Haven, Connecticut, US: Yale University, 2025. http://cs-www.cs.yale.edu/homes/spielman/sagt/.
[14] Davies AL, Papakonstantinou T, Nikolakopoulou A, Rücker G, Galla T. Network meta-analysis and random walks. *Stat Med*. 2022;41: 2091–2114.
[15] Bonchev D, Kier LB. Topological atomic indices and the electronic charges in alkanes. *J Math Chem*. 1992;9: 75–85.
[16] Rücker C, Rücker G. Mathematical relation between extended connectivity and eigenvector coefficients. *J Chem Inf Comput Sci*. 1994;34: 534–538.
[17] Feizi S, Marbach D, Médard M, Kellis M. Network deconvolution as a general method to distinguish direct dependencies in networks. *Nat Biotechnol*. 2013;31: 726–733.
[18] Lin Z, Pan I, Pan W. On network deconvolution for undirected graphs. *Biometrics*. 2024;80: ujae112.
[19] Doyle PG, Snell JL. *Random Walks and Electric Networks*, The Carus Mathematical Monographs. The American Mathematical Society; 1984. https://doi.org/10.5948/upo9781614440222.
[20] Rücker G, Rücker C. Computer perception of constitutional (topological) symmetry: TOPSYM, a fast algorithm for partitioning atoms and pairwise relations among atoms into equivalence classes. *J Chem Inf Comput Sci*. 1990;30: 187–191.
[21] Frost HR. Network analysis using Krylov subspace trajectories. *Complex Network Applications*. 2025;1187: 312–323.
[22] Frost HR. Analysis of biological networks using Krylov subspace trajectories. 2026, bioRxiv:2026.03.29.715092.
[23] Werner D. *Functional Analysis*. Springer; 2018. (eBook). https://doi.org/10.1007/978-3-662-55407-4.
[24] Wu M, Yin B, Vosoughi A, Studer C, Cavallaro JR, Dick C. Approximate matrix inversion for high-throughput data detection in the large-scale MIMO uplink. *IEEE International Symposium on Circuits and Systems (ISCAS2013)*. IEEE; 2013: 2155–2158. https://doi.org/10.1109/ISCAS.2013.6572301.
[25] Dias S, Welton NJ, Caldwell DM, Ades AE. Checking consistency in mixed treatment comparison meta-analysis. *Stat Med*. 2010;29: 932–944.
[26] König J, Krahn U, Binder H. Visualizing the flow of evidence in network meta-analysis and characterizing mixed treatment comparisons. *Stat Med*. 2013;32: 5414–5429.
[27] Papakonstantinou T, Nikolakopoulou A, Rücker G, et al. Estimating the contribution of studies in network meta-analysis: paths, flows and streams. *F1000Res*. 2018;7. https://doi.org/10.12688/f1000research.14770.3:610.
[28] Rücker G, Papakonstantinou T, Nikolakopoulou A, Schwarzer G, Galla T, Davies A. Shortest path or random walks? A framework for path weights in network meta-analysis. *Stat Med*. 2024;43: 4287–4304.
[29] Rücker G, Nikolakopoulou A, Papakonstantinou T, Salanti G, Riley RD, Schwarzer G. The statistical importance of a study for a network meta-analysis estimate. *BMC Med Res Methodol*. 2020;20. https://doi.org/10.1186/s12874-020-01075-y:190.
[30] Estrada E. The resistance distance is a diffusion distance on a graph. *Mathematics*. 2025;13(15): 2380. https://doi.org/10.3390/math13152380.
[31] Lumley T. Network meta-analysis: looping back. *Res Synth Methods*. 2024;15: 728–730.
[32] Davies AL. The bipartite structure of treatment-trial networks reveals the flow of information in network meta-analysis. *J Royal Stat Soc Series A Stat Soc*. Published online February 18, 2026; qnag012. https://doi.org/10.1093/jrsssa/qnag012.
[33] Norris JR. *Markov Chains*, Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press; 2007.

Appendix

A. Proof of Theorem (3)

We assume a connected network and positive weighted degrees for all included nodes. Due to $\mathbf{T} = \mathbf{I} - \mathbf{L}\mathbf{D}^{-1}$, we have $\mathbf{L}\mathbf{D}^{-1} = \mathbf{I} - \mathbf{T}$. Further, it holds $\mathbf{X} = \mathbf{X}\mathbf{L}^{+}\mathbf{L}$ (see Appendix A.2 in Rücker and Schwarzer[2]). Using these two relations, we start from the right-hand side of (3) by stating that

$$\frac{1}{2}\mathbf{X}\mathbf{D}^{-1}\sum_{k=0}^{\infty}\left(\tilde{\mathbf{T}}^{k}\mathbf{X}^{\top}\right) = \lim_{N\to\infty}\left(\frac{1}{2}\mathbf{X}\mathbf{D}^{-1}\sum_{k=0}^{N}\left(\tilde{\mathbf{T}}^{k}\mathbf{X}^{\top}\right)\right)$$

and continue with evaluating the finite sum on the right-hand side for a large N:

$$\begin{aligned}\frac{1}{2}\mathbf{X}\mathbf{D}^{-1}\sum_{k=0}^{N}\left(\tilde{\mathbf{T}}^{k}\mathbf{X}^{\top}\right) &= \frac{1}{2}\mathbf{X}\,\mathbf{L}^{+}\,(\mathbf{I}-\mathbf{T})\sum_{k=0}^{N}\left(\tilde{\mathbf{T}}^{k}\mathbf{X}^{\top}\right)\\ &= \frac{1}{2}\mathbf{X}\,\mathbf{L}^{+}\,(\mathbf{I}-\mathbf{T})\sum_{k=0}^{N}((\mathbf{T}+\mathbf{I})/2)^{k}\mathbf{X}^{\top}\end{aligned}$$

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

$$
\begin{aligned}
&= \frac{1}{2}\mathbf{X}\,\mathbf{L}^{+}\left(\sum_{k=0}^{N}((\mathbf{T}+\mathbf{I})/2)^{k} - \sum_{k=0}^{N}\mathbf{T}((\mathbf{T}+\mathbf{I})/2)^{k}\right)\mathbf{X}^{\top}\\
&= \frac{1}{2}\mathbf{X}\,\mathbf{L}^{+}\left(\sum_{k=0}^{N}((\mathbf{T}+\mathbf{I})/2)^{k} - \sum_{k=0}^{N}(2(\mathbf{T}+\mathbf{I})/2-\mathbf{I})((\mathbf{T}+\mathbf{I})/2)^{k}\right)\mathbf{X}^{\top}\\
&= \frac{1}{2}\mathbf{X}\,\mathbf{L}^{+}\left(\sum_{k=0}^{N}((\mathbf{T}+\mathbf{I})/2)^{k} - 2\sum_{k=0}^{N}((\mathbf{T}+\mathbf{I})/2)^{k+1} + \sum_{k=0}^{N}((\mathbf{T}+\mathbf{I})/2)^{k}\right)\mathbf{X}^{\top}\\
&= \frac{1}{2}\mathbf{X}\,\mathbf{L}^{+}\,2\left(\sum_{k=0}^{N}((\mathbf{T}+\mathbf{I})/2)^{k} - \sum_{k=0}^{N}((\mathbf{T}+\mathbf{I})/2)^{k+1}\right)\mathbf{X}^{\top}\\
&= \mathbf{X}\,\mathbf{L}^{+}\left(\sum_{k=0}^{N}((\mathbf{T}+\mathbf{I})/2)^{k} - \sum_{k=1}^{N+1}((\mathbf{T}+\mathbf{I})/2)^{k}\right)\mathbf{X}^{\top}\\
&= \mathbf{X}\,\mathbf{L}^{+}\left(((\mathbf{T}+\mathbf{I})/2)^{0} - ((\mathbf{T}+\mathbf{I})/2)^{N+1}\right)\mathbf{X}^{\top}\\
&= \mathbf{X}\,\mathbf{L}^{+}\left(\mathbf{I} - ((\mathbf{T}+\mathbf{I})/2)^{N+1}\right)\mathbf{X}^{\top}\\
&= \mathbf{X}\,\mathbf{L}^{+}\mathbf{X}^{\top} - \mathbf{X}\,\mathbf{L}^{+}\,\tilde{\mathbf{T}}^{N+1}\mathbf{X}^{\top}.
\end{aligned}
$$

For the limit, we have because of $\tilde{\mathbf{T}}^{N} \to \mathbf{T}^{\infty}$ and $\mathbf{T}^{\infty}\mathbf{X}^{\top} = \mathbf{0}$:

$$
\begin{aligned}
\lim_{N\to\infty}\left(\frac{1}{2}\mathbf{X}\mathbf{D}^{-1}\sum_{k=0}^{N}\left(\tilde{\mathbf{T}}^{k}\mathbf{X}^{\top}\right)\right) &= \mathbf{X}\,\mathbf{L}^{+}\mathbf{X}^{\top} - \lim_{N\to\infty}\left(\mathbf{X}\,\mathbf{L}^{+}\,\mathbf{T}^{\infty}\mathbf{X}^{\top}\right)\\
&= \mathbf{X}\,\mathbf{L}^{+}\mathbf{X}^{\top} = \mathbf{C},
\end{aligned}
$$

which proves the theorem.

B. Version with reference node

The idea also works, it even simplifies, if we choose an arbitrary baseline treatment, the reference node, and accordingly reduce the design matrix by the column that belongs to the reference node. All other matrices are reduced analogously, which means deleting the row and column that correspond to the reference node. This has interesting consequences. Let us denote the reduced versions by the index r. First, the Laplacian matrix $\mathbf{L}_r$ becomes regular, such that $\mathbf{L}_r^{-1}$ exists. Second, with respect to diffusion, the reference node becomes absorbing: each walk eventually ends up in this node. This also holds for bipartite networks. For $k \to \infty$, we have $\mathbf{T}_r^{k} \to \mathbf{0}$ and $\sum_{k=0}^{\infty}\mathbf{T}_r^{k}$ is a finite matrix. The same holds for the lazy walk diffusion matrix $\tilde{\mathbf{T}}_r = (\mathbf{T}_r + \mathbf{I}_r)/2$. For the covariance matrix, we have

$$
\begin{aligned}
\mathbf{C} &= \mathbf{X}_r\mathbf{L}_r^{-1}\mathbf{X}_r^{\top}\\
&= \mathbf{X}_r(\mathbf{D}_r - \mathbf{A}_r)^{-1}\mathbf{X}_r^{\top}\\
&= \mathbf{X}_r\left((\mathbf{I}_r - \mathbf{T}_r)\mathbf{D}_r\right)^{-1}\mathbf{X}_r^{\top}\\
&= \mathbf{X}_r\mathbf{D}_r^{-1}(\mathbf{I}_r - \mathbf{T}_r)^{-1}\mathbf{X}_r^{\top}\\
&= \frac{1}{2}\mathbf{X}_r\mathbf{D}_r^{-1}(\mathbf{I}_r - (\mathbf{T}_r + \mathbf{I}_r)/2)^{-1}\mathbf{X}_r^{\top}\\
&= \frac{1}{2}\mathbf{X}_r\mathbf{D}_r^{-1}(\mathbf{I}_r - \tilde{\mathbf{T}}_r)^{-1}\mathbf{X}_r^{\top}\\
&= \frac{1}{2}\mathbf{X}_r\mathbf{D}_r^{-1}\sum_{k=0}^{\infty}\tilde{\mathbf{T}}_r^{k}\mathbf{X}_r^{\top}.
\end{aligned}
$$

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

It turns out that for the version with a reference node, bipartite graphs do not need special attention: the results for the walk described by $\mathbf{T}_r$ and for the lazy walk $\tilde{\mathbf{T}}_r$ agree.

C. The behavior of $\sum_{i=0}^{\infty}(\mathbf{T}-\mathbf{T}^{\infty})^i$

We show Equation (5). For $i > 0$, we have

$$(\mathbf{T}-\mathbf{T}^{\infty})^i = \mathbf{T}^i - \mathbf{T}^{\infty},$$

which can be shown by complete induction. Because of $\mathbf{T}^{\infty}\mathbf{X}^{\top} = \mathbf{0}$, we may write

$$\begin{aligned}\mathbf{C} &= \frac{1}{2}\mathbf{X}\mathbf{D}^{-1}\sum_{i=0}^{\infty}\left(\tilde{\mathbf{T}}^i\mathbf{X}^{\top}\right) - \mathbf{X}\mathbf{D}^{-1}\mathbf{T}^{\infty}\mathbf{X}^{\top}\\ &= \frac{1}{2}\mathbf{X}\mathbf{D}^{-1}\sum_{i=0}^{\infty}\left(\tilde{\mathbf{T}}^i - \mathbf{T}^{\infty}\right)\mathbf{X}^{\top}\\ &= \frac{1}{2}\mathbf{X}\mathbf{D}^{-1}\sum_{i=0}^{\infty}\left(\tilde{\mathbf{T}} - \mathbf{T}^{\infty}\right)^i\mathbf{X}^{\top}.\end{aligned}$$

Although it has been shown for the average length of stay that $\frac{1}{N}\sum_{i=0}^{N}\mathbf{T}^i \to \mathbf{T}^{\infty}$,[33] the following simple example shows that series $\sum_{i=0}^{N}(\mathbf{T}-\mathbf{T}^{\infty})^i$ and $\sum_{i=0}^{N}(\tilde{\mathbf{T}}-\mathbf{T}^{\infty})^i$ generally do not approach the null matrix for $N \to \infty$.

Example

We consider a simple triangle with all standard errors equal to 1. Then

$$\mathbf{T} = \frac{1}{2}\begin{pmatrix}0&1&1\\1&0&1\\1&1&0\end{pmatrix},\quad \mathbf{T}^{\infty} = \frac{1}{3}\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix},\quad \mathbf{T}-\mathbf{T}^{\infty} = \frac{1}{6}\begin{pmatrix}-2&1&1\\1&-2&1\\1&1&-2\end{pmatrix}.$$

Because the spectral radius of matrix $\mathbf{T}-\mathbf{T}^{\infty}$ is less than 1, the matrix $\mathbf{I}-(\mathbf{T}-\mathbf{T}^{\infty})$ is nonsingular and the corresponding series gives

$$\sum_{i=0}^{\infty}(\mathbf{T}-\mathbf{T}^{\infty})^i = (\mathbf{I}-(\mathbf{T}-\mathbf{T}^{\infty}))^{-1} = \left(\frac{1}{6}\begin{pmatrix}8&-1&-1\\-1&8&-1\\-1&-1&8\end{pmatrix}\right)^{-1} = \frac{1}{9}\begin{pmatrix}7&1&1\\1&7&1\\1&1&7\end{pmatrix},$$

which is not the null matrix. A similar consideration holds for $\tilde{\mathbf{T}}$.

D. Symmetry

We note that the matrices $\mathbf{D}^{-1}\sum_{i=0}^{N}\mathbf{T}^i$ and $\mathbf{D}^{-1}\sum_{i=0}^{N}\tilde{\mathbf{T}}^i$ are symmetric. To see this, we need that $\mathbf{D}^{-1}\mathbf{T}$ and $\mathbf{D}^{-1}\tilde{\mathbf{T}}$ are symmetric:

$$\begin{aligned}\mathbf{D}^{-1}\mathbf{T} &= \mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1}\\ \mathbf{D}^{-1}\tilde{\mathbf{T}} &= \mathbf{D}^{-1}(\mathbf{A}\mathbf{D}^{-1}+\mathbf{I})/2 = (\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1}+\mathbf{D}^{-1})/2,\end{aligned}$$

which follows from the symmetry of $\mathbf{D}$ and $\mathbf{A}$. The same holds for the powers of $\mathbf{T}$ and $\tilde{\mathbf{T}}$ and their sums and the covariance matrix (3).

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

E. Implementation in R

The supporting information to this article on Zenodo contains three R scripts to reproduce the results (see the "Data availability statement" section). Script `functions.R` contains a number of functions, `data.R` provides the data sets, and `analysis.R` runs the functions on the examples. We note that `data.R` and `analysis.R` provide additional data and analyses that are not shown in this article.

1. Function `Power()` calculates matrix powers. Its arguments are a square matrix $\mathbf{M}$ and a natural number k (the power). The value of the function consists of the k'th power of $\mathbf{M}$, $\mathbf{M}^k$, and the partial sum $\sum_{i=0}^{k} \mathbf{M}^i$ of the geometric series.
2. Function `diffusion()` can be used to visualize the exchange of mass between the nodes of a network, given a *netmeta* object, a starting distribution (i.e., a vector), and a power N. There are two plotting options, guided by a logical argument `proportions`. The function provides either a series of network graphs (`proportions = FALSE`, default) showing how the mass distribution over the nodes develops with $i = 1, \ldots, N$, or a colored stacked probability plot to show this in one figure (`proportions = TRUE`). An argument `diffusion.type` allows users to choose between a simple walk (`"simple"`, default), the lazy walk (`"lazy"`), or a walk with an absorbing node (`"absorbing"`). Argument `ref` can be used to specify a reference node.

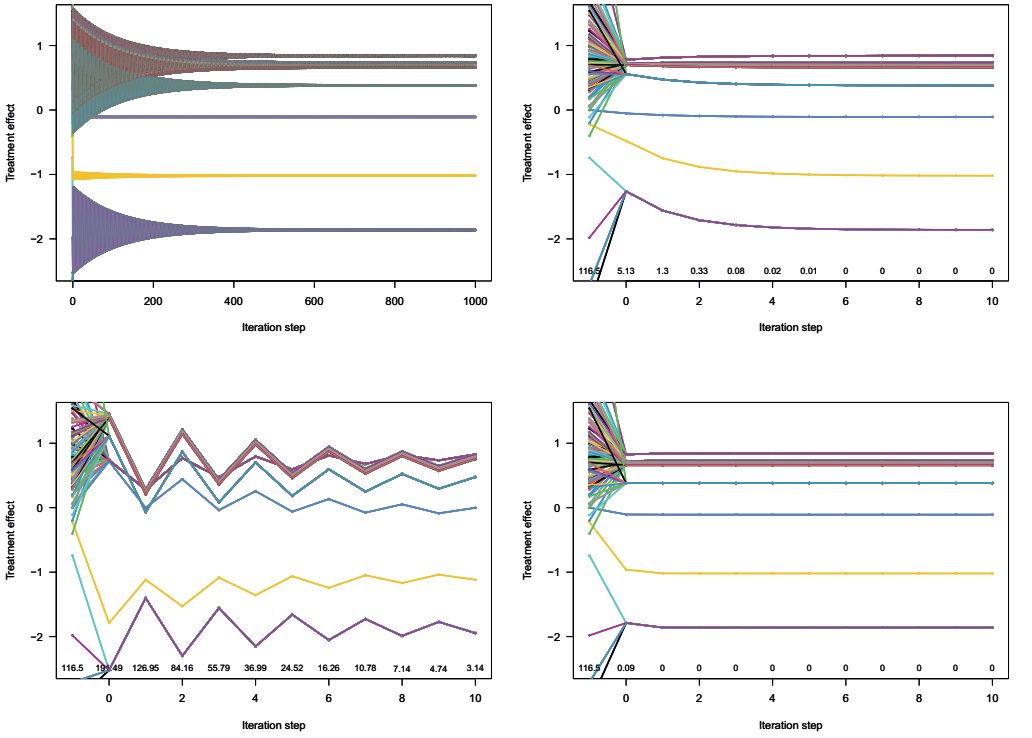


Figure E1. *Iterative estimation of the treatment effects for Example 3 based on the random-effects model. The starting points of the lines are the observed treatment effects from 108 comparisons. Top left: Simple method, 1,000 steps; Top right: Lazy walk, 10 steps. Bottom left: Absorbing walk, reference is Lidocaine–propofol admixture, 10 steps. Bottom right: Absorbing walk, reference is Hand vein, 10 steps. Each number at the bottom represents the weighted sum of squared distances of the estimates from the final NMA estimates.*

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.

3. Function `Hat()` serves to calculate the hat matrix using a matrix power series via Equations (3)–(6). Its arguments are a *netmeta* object, a power N, and a logical argument `ref`, `FALSE` by default. Argument `ref` can be used to specify a reference node.
4. Function `draw.TE()` is based on `Hat()`. For a given *netmeta* object and a power N, it allows visualizing the iterative course from the given vector of treatment effects `TE` to the network estimates `TE.nma.common` and produces a series of essentially decreasing Q values. Users can choose between the various diffusion types (argument `diffusion.type`) and specify a reference node (argument `ref`).
5. Function `draw.hat()`, likewise based on `Hat()`, helps visualizing the iterative course of the hat matrix diagonal (argument `hat`, default is `TRUE`), or the same for the treatment effect variances. As with function `draw.TE()`, there is a choice between diffusion types (argument `diffusion.type`) and a reference node can be specified (argument `ref`).
6. Function `walk()` calculates the steps described in the walkers-and-drinks model for a given *netmeta* object and visualizes them as bar charts.

F. Using the random-effects model for Example 3

Figure E1 shows the iterative estimation of the treatment effects for Example 3 based on the random-effects model in analogy to Figure 8.

Cite this article: Rücker G, Davies AL, Schwarzer G. Network meta-analysis and diffusion. *Research Synthesis Methods*. 2026;0: 1–20. https://doi.org/10.1017/rsm.2026.10123

Downloaded from https://www.cambridge.org/core. 11 Nov 2026 at 11:31:05, subject to the Cambridge Core terms of use.