Shortest path or random walks? A framework for path weights in network meta-analysis

Gerta Rücker[1]

[1] Institute of Medical Biometry and Statistics, Faculty of Medicine and Medical Center – University of Freiburg, Germany
e-mail: gerta.ruecker@uniklinik-freiburg.de

Recent Advances in Meta-Analysis: Methods and Software, 23. August 2023

This is joint work with

Adriani Nikolakopoulou, Freiburg, Germany

Theodoros Papakonstantinou, Freiburg, Germany

Guido Schwarzer, Freiburg, Germany

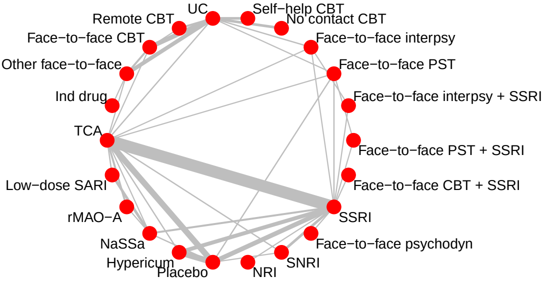
Tobias Galla, Palma de Mallorca, Spain

Annabel Davies, Bristol, UK

Outline

Network meta-analysis (NMA)

- A generalization of pairwise meta-analysis to more than two treatments
- Under certain assumptions of transitivity, allows estimating treatment effects for all comparisons in the network, even if not directly observed
- Bayesian and frequentist methods available
- Example: Depression in primary care [Linde et al., 2016]



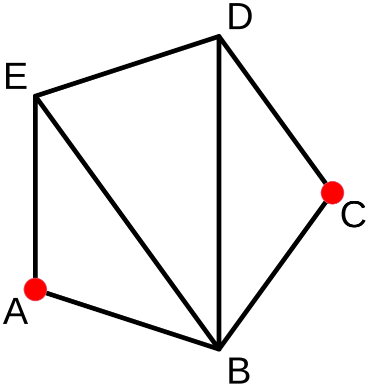
Percentage contributions of comparisons to network estimates

- Motivating question: How large is the contribution, measured as a percentage, of each observed direct comparison to a network estimate A:C?
- Percentage contributions implemented in web application CINeMA (Confidence in Network Meta-Analysis) [Nikolakopoulou et al., 2020, Papakonstantinou et al., 2018, Institute of Social and Preventive Medicine, 2017]
- Alternative approach, based on random walks [Davies et al., 2022]
- Both implemented in function `netcontrib()` of R package **netmeta**, methods `shortestpath` and `randomwalk` [Balduzzi et al., 2023]
- Both approaches are considering paths, based on the hat matrix
- **Today: Focus on paths and path weights!**

A small fictitious network

A network with 5 treatments and 7 direct comparisons

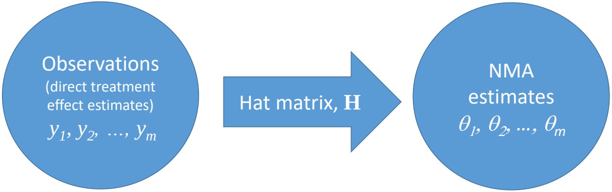
- All standard errors are assumed to be 1
- Focus on comparison A:C



Frequentist approach to NMA

- Estimation via weighted least squares
- Vector $\mathbf{y}$ of observed relative effects (i.e., mean differences) can be projected on network estimates via hat matrix $\mathbf{H}$
- The full hat matrix $\mathbf{H}$ of the aggregate model maps $\mathbf{y}$ to the vector of estimated NMA effects $\hat{\theta}^{nma}$:

$$\hat{\theta}^{nma} = \mathbf{H}\mathbf{y}$$



Frequentist approach to NMA

- Estimation via weighted least squares
- Vector $\mathbf{y}$ of observed relative effects (i.e., mean differences) can be projected on network estimates via hat matrix $\mathbf{H}$
- The full hat matrix $\mathbf{H}$ of the aggregate model maps $\mathbf{y}$ to the vector of estimated NMA effects $\hat{\theta}^{nma}$:

$$\hat{\theta}^{nma} = \mathbf{H}\mathbf{y}$$

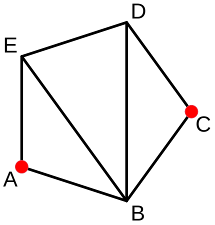
- For example, for comparison A:C we have

$$\hat{\theta}^{nma}_{A:C} = \mathbf{h}^{\top}_{A:C}\mathbf{y}$$

where $\mathbf{h}^{\top}_{A:C}$ is the A:C row of the hat matrix $\mathbf{H}$:

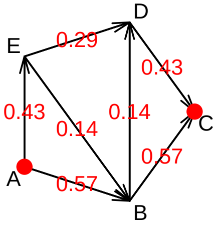
$$\hat{\theta}^{nma}_{A:C} = 0.57y_{AB} + 0.43y_{AE} + 0.57y_{BC} + 0.14y_{BD} - 0.14y_{BE} - 0.43y_{CD} - 0.29y_{DE}$$

Hat matrix for example



$$\mathbf{H} = \begin{pmatrix} AB & AC & AD & AE & BC & BD & BE & CD & CE & DE \\ 0.62 & 0 & 0 & 0.38 & -0.05 & -0.10 & -0.24 & -0.05 & 0 & -0.14 \\ 0.57 & 0 & 0 & 0.43 & 0.57 & 0.14 & -0.14 & -0.43 & 0 & -0.29 \\ 0.52 & 0 & 0 & 0.48 & 0.19 & 0.38 & -0.05 & 0.19 & 0 & -0.43 \\ 0.38 & 0 & 0 & 0.62 & 0.05 & 0.10 & 0.24 & 0.05 & 0 & 0.14 \\ -0.05 & 0 & 0 & 0.05 & 0.62 & 0.24 & 0.10 & -0.38 & 0 & -0.14 \\ -0.10 & 0 & 0 & 0.10 & 0.24 & 0.48 & 0.19 & 0.24 & 0 & -0.29 \\ -0.24 & 0 & 0 & 0.24 & 0.10 & 0.19 & 0.48 & 0.10 & 0 & 0.29 \\ -0.05 & 0 & 0 & 0.05 & -0.38 & 0.24 & 0.10 & 0.62 & 0 & -0.14 \\ -0.19 & 0 & 0 & 0.19 & -0.52 & -0.05 & 0.38 & 0.48 & 0 & 0.43 \\ -0.14 & 0 & 0 & 0.14 & -0.14 & -0.29 & 0.29 & -0.14 & 0 & 0.57 \end{pmatrix}$$

Idea: Interpret hat matrix row A:C as flow from A to C

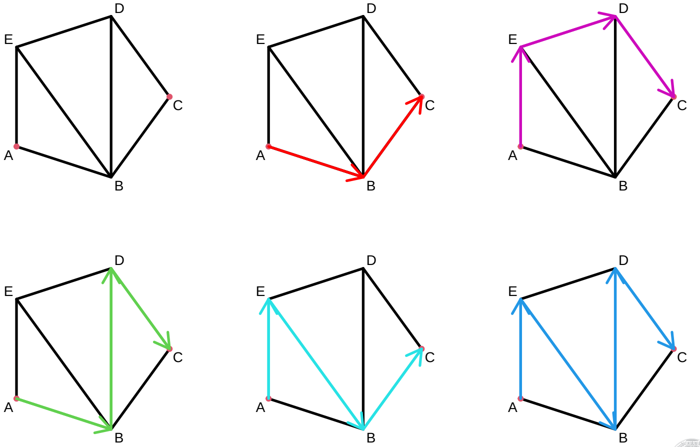


Signs indicate the direction [König et al., 2013]

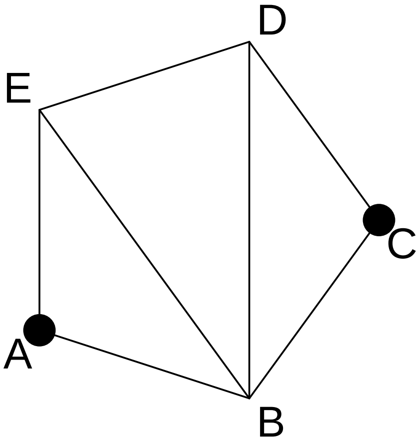
$$
\mathbf{H} = \left(\begin{array}{cccccccccc}
AB & AC & AD & AE & BC & BD & BE & CD & CE & DE \\
0.62 & 0 & 0 & 0.38 & -0.05 & -0.10 & -0.24 & -0.05 & 0 & -0.14 \\
0.57 & 0 & 0 & 0.43 & 0.57 & 0.14 & -0.14 & -0.43 & 0 & -0.29 \\
0.52 & 0 & 0 & 0.48 & 0.19 & 0.38 & -0.05 & 0.19 & 0 & -0.43 \\
0.38 & 0 & 0 & 0.62 & 0.05 & 0.10 & 0.24 & 0.05 & 0 & 0.14 \\
-0.05 & 0 & 0 & 0.05 & 0.62 & 0.24 & 0.10 & -0.38 & 0 & -0.14 \\
-0.10 & 0 & 0 & 0.10 & 0.24 & 0.48 & 0.19 & 0.24 & 0 & -0.29 \\
-0.24 & 0 & 0 & 0.24 & 0.10 & 0.19 & 0.48 & 0.10 & 0 & 0.29 \\
-0.05 & 0 & 0 & 0.05 & -0.38 & 0.24 & 0.10 & 0.62 & 0 & -0.14 \\
-0.19 & 0 & 0 & 0.19 & -0.52 & -0.05 & 0.38 & 0.48 & 0 & 0.43 \\
-0.14 & 0 & 0 & 0.14 & -0.14 & -0.29 & 0.29 & -0.14 & 0 & 0.57
\end{array} \right)
$$

Step 1: Look for all directed paths from A to C

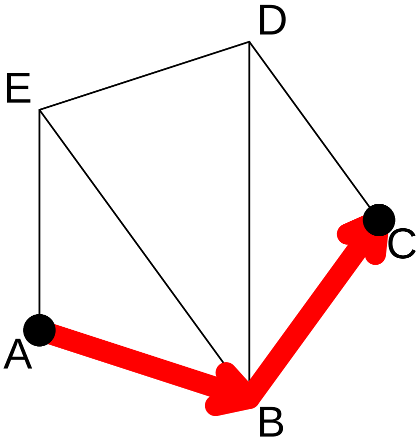
Note: We consider only paths that are compatible with the hat matrix row with respect to sign/direction!



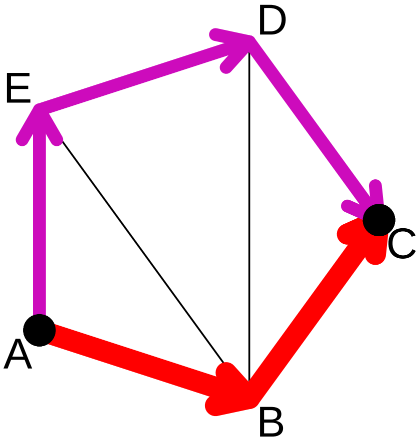
Idea: Partition the flow into streams across each of the directed paths [Papakonstantinou et al., 2018]



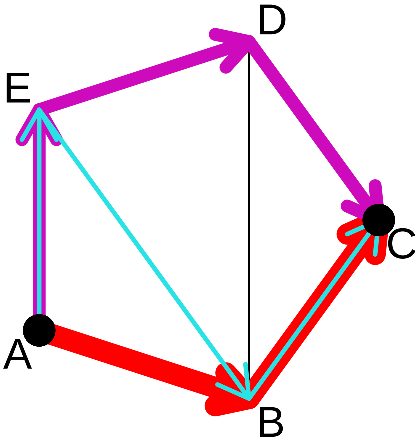
Idea: Partition the flow into streams across each of the directed paths [Papakonstantinou et al., 2018]



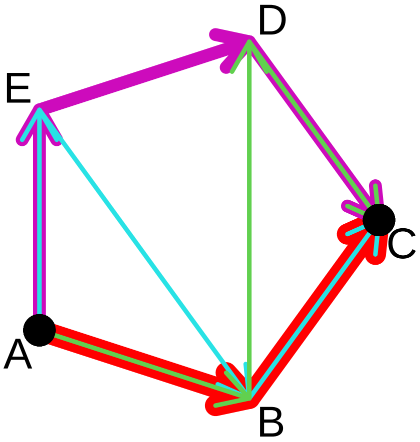
Idea: Partition the flow into streams across each of the directed paths [Papakonstantinou et al., 2018]



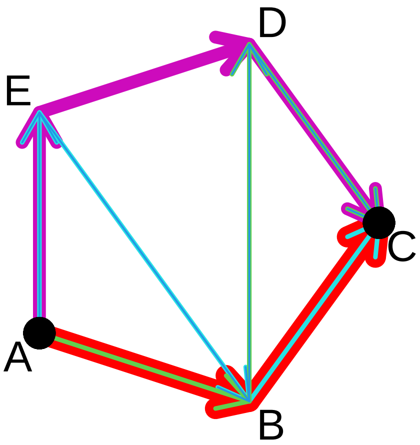
Idea: Partition the flow into streams across each of the directed paths [Papakonstantinou et al., 2018]



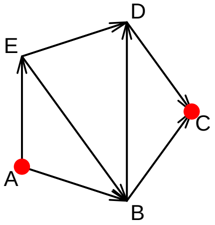
Idea: Partition the flow into streams across each of the directed paths [Papakonstantinou et al., 2018]



Idea: Partition the flow into streams across each of the directed paths [Papakonstantinou et al., 2018]



Step 2: Construct path-design matrix for comparison A:C



All paths from A to C that are compatible with the hat matrix row:

$A \to B \to C$
$A \to B \to D \to C$
$A \to E \to B \to C$
$A \to E \to B \to D \to C$
$A \to E \to D \to C$

$$\mathbf{Z}_{A:C} = \begin{pmatrix} A:B & A:C & A:D & A:E & B:C & B:D & B:E & C:D & C:E & D:E \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & -1 \end{pmatrix}$$

Step 3: Obtain path weights by solving a linear equation

Aim: Construct weights for each of the P (here $P = 5$) paths that contribute to the flow from A to C

- Looking for a vector of path weights ϕ of length P such that path p obtains weight $\phi_p (p = 1, \dots, P)$. Remember:

$$\hat{\theta}_{A:C}^{nma} = \mathbf{h}_{A:C}^{\top} \mathbf{y}$$

- Request

$$\mathbf{h}_{A:C}^{\top} \mathbf{y} = \phi^{\top} \mathbf{Z}_{A:C} \mathbf{y}$$

for all $\mathbf{y}$.

- We are looking for a solution ϕ of the linear equation system

$$\mathbf{h}_{A:C}^{\top} = \phi^{\top} \mathbf{Z}_{A:C}$$

Step 3: Obtain path weights by solving a linear equation

- Linear equation system for comparison A:C

$$\mathbf{h}_{A:C}^{\top} = \phi^{\top}\mathbf{Z}_{A:C}$$

- In general, the path-design matrix $\mathbf{Z}_{A:C}$ has no inverse
- `shortestpath` and `randomwalk` provide two solutions, often different
- Another solution is `pseudoinverse`:

$$\phi^{\top} = \mathbf{h}_{A:C}^{\top}\mathbf{Z}_{A:C}^{+}$$

($\mathbf{Z}_{A:C}^{+}$ Moore-Penrose pseudoinverse of $\mathbf{Z}_{A:C}$ [Albert, 1972])

Step 3: Obtain path weights by solving a linear equation

- General solution of the equation system [Albert, 1972, Theorem (3.12)]

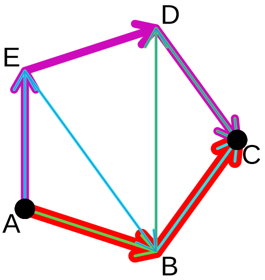
$$\phi^\top = \mathbf{h}_{A:C}^\top \mathbf{Z}_{A:C}^+ + \mathbf{x}^\top (\mathbf{I} - \mathbf{Z}_{A:C}\mathbf{Z}_{A:C}^+)$$

  where $\mathbf{I}$ is the $P \times P$ identity matrix and $\mathbf{x} \in \mathbb{R}^P$ an arbitrary vector of length P
- Unique if and only if $\mathbf{Z}_{A:C}\mathbf{Z}_{A:C}^+ = \mathbf{I}$
- `pseudoinverse` is the special case for $\mathbf{x} = \mathbf{0}$
- The sum of entries in each solution ϕ is 1 [Rücker et al., 2023]
- In general, there are infinitely many solutions!

Consider some particular solutions

- `shortestpath` [Papakonstantinou et al., 2018] - Starting from the shortest path from A to C, collect streams along further paths until the full flow is exhausted
- `randomwalk` [Davies et al., 2022] - Find all paths from A to C and use a random walk algorithm to define path weights and edge weights
- `pseudoinverse` [Rücker et al., 2023] - Find the least squares solution of the equation system - minimizes the L2 norm (sum of squares, ϕ^2)
- `cccp` - Use R package **cccp** [Pfaff, 2020] to find another solution that minimizes the L1 norm (sum of $|\phi|$ coefficients)
- `otherpath` - Similar to `shortestpath`, but start with another path

Path weights for comparison A:C for these solutions



Method	A-B-C	A-B-D-C	A-E-B-C	A-E-B-D-C	A-E-D-C
`shortestpath`	0.571	0	0	0.143	0.286
`randomwalk`	0.457	0.114	0.114	0.029	0.286
`cccp`	0.471	0.101	0.101	0.042	0.286
`pseudoinverse`	0.393	0.179	0.179	-0.036	0.286
`otherpath`	0.429	0.143	0.143	0	0.286

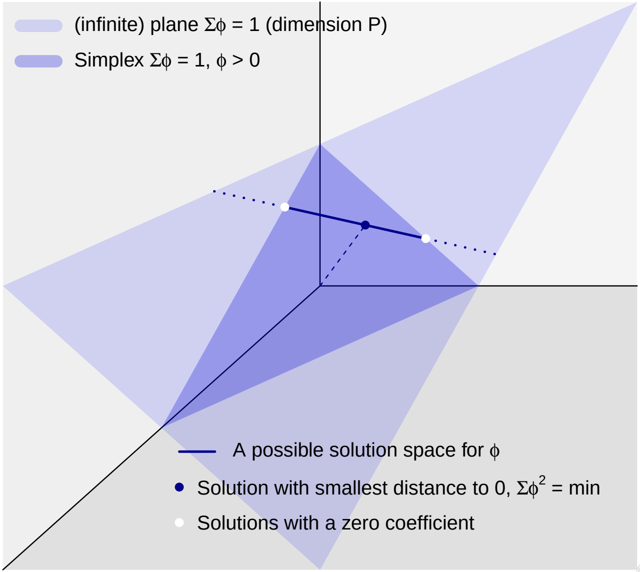
Shortest path order: ABC, AEDC, AEBDC; other path order: AEDC, AEBC, ABDC, ABC

UNIVERSITÄTS KLINIKUM FREIBURG

Properties of solutions for finding path weights

Method	Minimizes norm	Variance	Negative weights possible	Needs all-paths search
`shortestpath`	L1	large	no	no
`randomwalk`	L1	medium	no	yes
`cccp`	L1	medium	no	yes
`pseudoinverse`	L2	small	yes	yes

Geometric visualization of solutions



Summary I

- Contributions are not uniquely defined!
- Question: Which is the best solution, and what are the criteria?
- `pseudoinverse` minimizes L2 norm (Euklidean distance to origin) and thus the variance between path weights
- `shortestpath` , `randomwalk` and `cccp` minimize L1 norm (sum of absolute path weights)
- `shortestpath` often shows the largest variance between weights
- Sometimes another path order ”needs fewer paths”, i.e., attributes zero weight to more paths
  - Alternatively, start with the ”widest road” instead of the shortest?
- For further discussion, see [Rücker et al., 2023] (just submitted and on arXiv)

Summary II

- Path weights used to define edge weights (”contributions”)
- Implemented in function `netcontrib()`, argument `method` of R package **netmeta**
- Most similar: `randomwalk` and `cccp`
- Most different: `shortestpath` and `pseudoinverse`
- Run time: `shortestpath` is fastest (doesn’t need all-paths search, which can be VERY slow for large networks)
  - For example, `shortestpath` ignores paths $A \to B \to D \to C$ and $A \to E \to B \to C$

⇒ Current recommendation (and default in CINeMA and R package **netmeta**): `shortestpath`

Outlook

- So far, path weights mainly used to define edge weights ("contributions")
- Path weights may lead to a new definition of "path inconsistency" (current work in Freiburg)
- Interpreting hat matrix elements as flows may have applications also in more general regression theory

References I

Institute of Social and Preventive Medicine (2017).
CINeMA: Confidence in network meta-analysis.
University of Bern.
https://cinema.ispm.unibe.ch/.

Albert, A. E. (1972).
Regression and the Moore-Penrose Pseudoinverse.
Mathematics in Science and Engineering. Academic Press, New York.
ISBN: 0-12-048450-1.

Balduzzi, S., Rücker, G., Nikolakopoulou, A., Papakonstantinou, T., Salanti, G., Efthimiou, O., and Schwarzer, G. (2023).
netmeta: An R package for network meta-analysis using frequentist methods.
Journal of Statistical Software, 106(2):1–40.
10.18637/jss.v106.i02.

Davies, A. L., Papakonstantinou, T., Nikolakopoulou, A., Rücker, G., and Galla, T. (2022).
Network meta-analysis and random walks.
Statistics in Medicine, 41(12):2091–2114.

König, J., Krahn, U., and Binder, H. (2013).
Visualizing the flow of evidence in network meta-analysis and characterizing mixed treatment comparisons.
Statistics in Medicine, 32(30):5414–5429.

Linde, K., Rücker, G., Schneider, A., and Kriston, L. (2016).
Questionable assumptions hampered interpretation of a network meta-analysis of primary care depression treatments.
Journal of Clinical Epidemiology, 71:86–96.
doi:10.1016/j.jclinepi.2015.10.010.

References II

Nikolakopoulou, A., Higgins, J. P. T., Papakonstantinou, T., Chaimani, A., Del Giovane, C., Egger, M., and Salanti, G. (2020).
CINeMA: An approach for assessing confidence in the results of a network meta-analysis.
PLOS Medicine, 17(4):1–19.

Papakonstantinou, T., Nikolakopoulou, A., Rücker, G., Chaimani, A., Schwarzer, G., Egger, M., and Salanti, G. (2018).
Estimating the contribution of studies in network meta-analysis: paths, flows and streams.
F1000Research, 7(610):610.
doi:10.12688/f1000research.14770.3.

Pfaff, B. (2020).
cccp: Cone Constrained Convex Problems.
R package version 0.2-7, https://CRAN.R-project.org/package=cccp.

Rücker, G. (2012).
Network meta-analysis, electrical networks and graph theory.
Research Synthesis Methods, 3(4):312–324.

Rücker, G., Papakonstantinou, T., Nikolakopoulou, A., Schwarzer, G., Galla, T., and Davies, A. (2023).
Shortest path or random walks? A framework for path weights in network meta-analysis.
http://arxiv.org/abs/2308.05841.

Rücker, G. and Schwarzer, G. (2014).
Reduce dimension or reduce weights? Comparing two approaches to multi-arm studies in network meta-analysis.
Statistics in Medicine, 33:4353–4369.
DOI: 10.1002/sim.6236.

How to obtain edge weights from path weights

- $|\mathbf{Z}_{A:C}|$ matrix derived from $\mathbf{Z}_{A:C}$ by taking elementwise absolute values
- λ vector of path lengths, given by the row sums of $|\mathbf{Z}_{A:C}|$
- Transform the entries of ϕ to weights $\mathbf{w}$ by setting

$$\mathbf{w} = (\phi/\lambda)^\top |\mathbf{Z}_{A:C}|$$

  where the division ϕ/λ is made elementwise
- The sum of weights is 1:

$$\mathbf{w}^\top \mathbf{1} = (\phi/\lambda)^\top |\mathbf{Z}_{A:C}| \mathbf{1} = (\phi/\lambda)^\top \lambda = \phi^\top \mathbf{1} = 1$$

Appendix: Frequentist approach

- Estimation via weighted least squares
- Vector $\mathbf{y}$ of observed relative effects (i.e., mean differences) can be projected on network estimates via hat matrix $\mathbf{H}$
- $\mathbf{H}$ is defined as

$$\mathbf{H} = \mathbf{B}(\mathbf{B}^\top\mathbf{W}\mathbf{B})^+\mathbf{B}^\top\mathbf{W}$$

  where
  - $\mathbf{B}$ is the edge-incidence matrix of the full aggregate network (dimension $m \times n$),
  - $\mathbf{W}$ is a diagonal matrix of inverse variance weights (dimension $m \times m$), assumed to be appropriately adjusted for multi-arm studies [Rücker and Schwarzer, 2014]
  - $(\mathbf{B}^\top\mathbf{W}\mathbf{B})^+$ is the Moore-Penrose pseudoinverse of the Laplacian matrix $\mathbf{L} = \mathbf{B}^\top\mathbf{W}\mathbf{B}$ [Rücker, 2012]
- $m \times m$ full hat matrix $\mathbf{H}$ of the aggregate model maps $\mathbf{y}$ to the vector of estimated NMA effects $\hat{\theta}^{nma}$:

$$\hat{\theta}^{nma} = \mathbf{H}\mathbf{y}$$