Visualizing assumptions and results in network meta-analysis: The network graphs package

Anna Chaimani
Department of Hygiene and Epidemiology
University of Ioannina School of Medicine
Ioannina, Greece
achaiman@cc.uoi.gr

Georgia Salanti
Department of Hygiene and Epidemiology
University of Ioannina School of Medicine
Ioannina, Greece
gsalanti@cc.uoi.gr

Abstract. Network meta-analysis has been established in recent years as a particularly useful evidence synthesis tool. However, it is still challenging to develop understandable and concise ways to present data, assumptions, and results from network meta-analysis to inform decision making and evaluate the credibility of the results. In this article, we provide a suite of commands with graphical tools to facilitate the understanding of data, the evaluation of assumptions, and the interpretation of findings from network meta-analysis.

Keywords: st0411, clusterank, ifplot, intervalplot, mdsrank, netfunnel, netleague, netweight, networkplot, sucra, network graphs, network meta-analysis, mixed-treatment comparison, multiple treatments, ranking, inconsistency, graphical tools

1   Introduction

Network meta-analysis integrates direct and indirect evidence in a collection of trials providing information on the relative effects of three or more competing interventions for the same outcome. Despite the increasing popularity of network meta-analysis (Lee 2014; Nikolakopoulou et al. 2014), it is still the subject of skepticism, particularly among researchers with a less technical background. This phenomenon can be partly explained by the lack of flexible tools implemented in a user-friendly software environment providing comprehensive ways to understand and present the different steps of a network meta-analysis.

Graphical illustrations have been extensively used in statistical analyses because they offer a concise way to describe complex data structures, assumptions, and outputs (Anscombe 1973). Many different plots are available for the presentation of standard pairwise meta-analysis (Anzures-Cabrera and Higgins 2010; Bax et al. 2009; Chaimani, Mavridis, and Salanti 2014), but their adaptation into network meta-analysis, when feasible, needs to account for the presence of multiple treatment comparisons. For instance,

forest plots in large networks cannot easily accommodate study-level data and multiple summary estimates because of space limitations. Funnel plots are meaningless without an appropriate "adjustment" because they include studies that evaluate different pairs of interventions (Chaimani et al. 2013a; Chaimani and Salanti 2012).

Summary statistics from network meta-analysis are estimated under the assumption of consistency, which implies that all different sources of evidence are in statistical agreement (Caldwell, Ades, and Higgins 2005; Salanti 2012). Several sophisticated methods have been developed for the assessment of inconsistency in a network of interventions (Dias et al. 2010; Higgins et al. 2012; Lu and Ades 2006). However, less advanced methods, such as the "loop-specific approach" (Bucher et al. 1997), can be more easily conceived and interpreted by clinicians, especially through graphical depictions, and could be considered in conjunction with other approaches.

Transparent and concise presentation of findings might be challenging for networks that include a large number of treatments (for example, more than 10). Reporting of results can be supplemented by graphs that show the relative ranking of treatments (Salanti, Ades, and Ioannidis 2011) and allow for information on two outcomes (Chaimani et al. 2013a), which can be useful for decision makers.

In Stata, simple indirect comparisons (using information from a single loop) can be performed using the `indirect` command (Miladinovic et al. 2014), and full network meta-analysis can be performed using the `mvmeta` (White 2011; White et al. 2012) and `network`[1] (White 2015) packages. We have previously presented commands that can be used in conjunction with `mvmeta` to check assumptions and produce results (Chaimani et al. 2013a). In this article, we extend and elaborate on our previous work, and we present a suite of nine commands that produce graphs that describe the evidence base, evaluate the assumptions, and present the results obtained from a network meta-analysis.

2　Example datasets

We illustrate the use of the nine commands using three published networks. The first network includes 27 studies comparing the effectiveness (change in glycated hemoglobin A1c [HbA1c]) and tolerability (weight gain) of six noninsulin antidiabetic drugs and placebo (Phung et al. 2010). The second network comprises 23 trials that evaluate four antiplatelet regimens and placebo for the prevention of serious vascular events after transient ischemic attack or stroke (Thijs, Lemmens, and Fieuws 2008). The last network consists of 22 trials comparing the safety of five antihypertensive treatments and placebo with respect to the incidence of diabetes (Elliott and Meyer 2007).

---

1. `network` calls `mvmeta` and is equivalent.

3 The network graphs package

3.1 Notation and setting up the data

Input variables

There are two groups of commands in the `network graphs` package. The first group (`networkplot`, `netweight`, `ifplot`, and `netfunnel`) requires as input variables with study-level data that are stored in a dataset. The second group of commands (`intervalplot`, `netleague`, `sucra`, `mdsrank`, and `clusterank`) takes as input summary data produced by the `mvmeta` or `network` package that run network meta-analysis (see White [2015; 2011] and White et al. [2012] for a description of input data for `mvmeta` and `network`).

The input variables for the first group of commands require a format that assumes one treatment comparison per row. For a two-arm trial, this amounts to inputting the treatments being compared, the effect sizes, and other characteristics for the single treatment comparison observed in that study. For multiarm trials, all comparisons enter the data as different rows, with the study code (*id*) denoting which observations belong to the same study.

More specifically, the input variables for the `networkplot`, `netweight`, `ifplot`, and `netfunnel` commands include the following:

id: The variable specifying the ID numbers of the studies, with repetitions for multiarm studies. Each k-arm trial contributes $\{k \times (k-1)\}/2$ rows to the dataset.

t1 and *t2*: The variables containing the codes of two treatments being compared in every observation of the dataset. These variables can be numeric or string.

ES: The variable with the effect sizes for every treatment comparison defined by *t1* and *t2*. For ratio measures (odds ratio, risk ratio, hazard ratio, etc.), the effect sizes in *ES* should be in logarithmic scale (for example, log odds-ratio or log risk-ratio). In all example datasets, we have estimated *ES* as *t1* versus *t2*.

seES: The variable containing the standard errors of the effect sizes in *ES*.

The second group of commands—`intervalplot`, `sucra`, `netleague`, `mdsrank`, and `clusterank`—are commands used to present and interpret the output of network meta-analysis; hence, they need as input estimates that can be produced by a network meta-analysis routine (such as `network` or WinBUGS [Lunn et al. 2000] codes). For example:

LCI, *UCI*, *LPI*, and *UPI* (inputs for `intervalplot`) are variables that contain the lower and upper limits of the confidence and predictive intervals for the effect sizes of all possible pairwise comparisons estimated in network meta-analysis.

outcome1 and *outcome2* (inputs for `clusterank`) are variables containing the estimated values of a ranking measure for two different outcomes.

Options

In this section, we describe options that are available in multiple commands in the `network graphs` package. Other options that are unique to each command are described in the respective sections below and in the help files.

`mvmetaresults` specifies to derive the input directly from `mvmeta` or `network` (the default). This option is available in the commands `intervalplot`, `netleague`, and `sucra`, which also allow for different inputs (not derived from `mvmeta`).

`nomvmeta` specifies not to derive the input from `mvmeta` or `network`. This option is available in the commands `intervalplot`, `netleague`, and `sucra`.

`labels(`*string*`)` specifies names for competing treatments in the network, which are displayed in the output results and the corresponding plots. The treatments should be given in numerical or alphabetical order separated by a space. For example, specifying the option `labels(Placebo Aspirin "Pla+Asp" ...)` assigns the name `Placebo` to the treatment coded as 1 or A, the name `Aspirin` to the treatment coded 2 or B, and so on. For the commands `intervalplot`, `netleague`, and `sucra` when the option `nomvmeta` has not been specified, the first treatment in `labels()` should be the treatment assumed to be the reference when running `mvmeta` or `network`, and the following treatments should be given in numerical or alphabetical order. The `labels()` option is available for the commands `networkplot`, `ifplot`, `intervalplot`, `netleague`, `sucra`, and `mdsrank`.

`eform` specifies to display the results on the exponential scale. This option is available for the commands `ifplot`, `intervalplot`, and `netleague`.

`keep` specifies to store the results as additional variables at the end of the dataset. This option is available for the commands `ifplot` and `intervalplot`.

`title(`*string*`)` specifies a title for the produced graph. This option is available for the commands `networkplot`, `netweight`, `intervalplot`, and `sucra`.

`aspect(#)` specifies the aspect ratio for the region of the graph. This option is available for the commands `networkplot` and `netweight`.

`notable` suppresses display of the output. This option is available for the commands `netweight`, `ifplot`, `intervalplot`, and `sucra`.

`noplot` suppresses display of the produced graph. This option is available for the commands `netweight`, `ifplot`, `intervalplot`, `sucra`, and `mdsrank`.

`scatteroptions(`*string*`)` specifies standard options allowed for scatterplots. This option is available for the commands `netfunnel`, `mdsrank`, and `clusterank`.

`cilevel(`*integer*`)` specifies the level of statistical significance for the estimated confidence intervals. This option is available for the `ifplot` and `intervalplot` commands.

xtitle(*string*) and xlabel(*string*) specify a title and the values, respectively, that are displayed for the horizontal axis. This option is available for the commands `netfunnel` and `intervalplot`.

Dialog boxes

The dialog boxes are an alternative way to run a command within Stata; they might be preferable for users who are not familiar with Stata language. Dialog boxes have been developed for all commands of the `network graphs` package and can be accessed by typing `db` *commandname* in the Stata Command window.

3.2   The networkplot command

Description of networkplot

The `networkplot` command plots a network of interventions using nodes and edges. Nodes represent the competing treatments, and edges represent the available direct comparisons between pairs of treatments. `networkplot` allows for weighting and coloring options for both nodes and edges according to prespecified characteristics. The use of weighting and coloring schemes can reveal important differences in the characteristics of treatments or comparisons. Sometimes, differences in comparisons can be an indication of potential violation of the assumption underlying network meta-analysis (Jansen and Naci 2013; Salanti 2012).

Syntax for networkplot

networkplot *varlist* [*if*] [*in*] [, <u>lab</u>els(*string*) <u>title</u>(*string*) <u>aspect</u>(*#*)
    <u>nowe</u>ight <u>nodeweight</u>(*weightvar* sum|mean) <u>edgeweight</u>(*weightvar* sum|mean)
    <u>nodecolor</u>(*string*) <u>edgecolor</u>(*edge_color*|by *groupvar* *pool_method*)
    <u>bylevels</u>(*#*) <u>bycolors</u>(*string*) <u>edgepattern</u>(*string*) <u>plotregion</u>(*string*)
    <u>edgescale</u>(*#*) <u>nodes</u>cale(*#*)]

    In *varlist*, the variables *t1* and *t2* (see section 3.1) should be specified.

Options for networkplot

labels(*string*), title(*string*), aspect(*#*); see section 3.1.

noweight specifies that all nodes and edges are of equal size and thickness (when `nodeweight()` and `edgeweight()` have not been specified). By default, both nodes and edges are weighted according to the number of studies involved in each treatment and comparison, respectively.

nodeweight(*weightvar* sum|mean) specifies a variable according to which nodes are weighted. sum specifies to use the sum of the values in *weightvar* to weight the nodes, while mean specifies to use the mean of the values.

edgeweight(*weightvar* sum|mean) specifies a variable according to which edges are weighted. sum specifies to use the sum of the values in *weightvar* to weight the edges, while mean specifies to use the mean of the values.

nodecolor(*string*) specifies the color for all nodes. The default is nodecolor(blue).

edgecolor(*edge_color*|by *groupvar pool_method*) specifies the color for edges.

   edge_color specifies the color for all edges. The default color is black.

   by specifies that edges be colored according to the bias level for each comparison.

   groupvar is a variable that contains bias scores for each observation (that is, study data such as effect size). It can be a numeric variable coded as 1 = low risk, 2 = unclear risk, and 3 = high risk, or it can be a string variable with values l[ow], u[nclear], and h[igh]. By default, a three-level *groupvar* is assumed, allowing for green, yellow, and red lines representing comparisons at low, unclear, and high risk of bias, respectively. More or fewer than three levels with user-specified colors are also allowed if the options bylevels() and bycolors() have been specified.

   pool_method specifies the method by which to estimate the risk of bias level for the summary estimate of each comparison. It can be one of the following:

   mode specifies to use the most prevalent bias level within each comparison as the comparison-specific bias level (the default).

   mean specifies to use the average bias level of each comparison as the comparison-specific bias level.

   wmean *weightvar* specifies to use the weighted (according to the *weightvar*) average bias level of each comparison as the comparison-specific bias level.

   max specifies to use the maximum bias level observed within each comparison as the comparison-specific bias level.

   min specifies to use the minimum bias level observed within each comparison as the comparison-specific bias level.

bylevels(#) specifies the number of levels for the *groupvar* in edgecolor() when there are more than three bias levels. In this case, the *groupvar* variable can only be numeric. bycolors() is required with bylevels().

bycolors(*string*) specifies the colors for each bias level of the *groupvar* in the respective order. bylevels() is required with bycolors().

edgepattern(*string*) specifies the pattern for all edges.

plotregion(*string*) specifies options for the region of the network plot.

edgescale(#) specifies a real number that is used to scale all edges.

nodescale(#) specifies a real number that is used to scale all nodes.

Example of networkplot using the antidiabetics network

The dataset of the antidiabetics network contains information from 24 study comparisons. For each observation, negative values of the standardized mean difference (*ES*) favor the first treatment (*t1*) in the respective comparison over the second treatment (*t2*). The treatments have been coded as numbers [1 = placebo, 2 = sulfonylurea, 3 = dipeptidyl peptidase-4 (DPP-4) inhibitor, 4 = thiazolidinedione, 5 = glucagon-like peptide-1 (GLP-1) analog, 6 = alpha-glucosidase inhibitors (AGI), 7 = glinine]. For each study, the variable blinding contains the level of risk of bias with respect to double-blinding. Note that the fifth study is a three-arm study and so is represented by three rows.

```
. use antidiabetics_efficacy_wide
. list in 1/7, clean noobs
     id   t1   t2           ES         seES   blinding
      1    3    1   -.8701839     .1102315          1
      2    3    2    .1362578     .0427965          1
      3    3    1   -.6923013     .1341008          2
      4    3    1   -.3768538     .1208566          2
      5    2    1   -.7710395     .1149981          1
      5    5    1   -.7710395     .1149981          1
      5    5    2            0     .0909091          1
```

Using the networkplot command, we produce the plot of the network (see figure 1). This gives information on the network structure, the number of studies evaluating each intervention, the precision of the direct estimate for each pairwise comparison, and the average bias level for every comparison with respect to blinding.

```
. generate invvarES = 1/(seES^2)
. networkplot t1 t2, edgeweight(invvarES) edgecolor(by blinding mean)
> edgescale(1.2) aspect(0.8) labels(Placebo Sulfonylurea "DPP-4 inhibitor"
> Thiazolidinedione "GLP-1 analog" AGI Glinine)
```

The resulting graph, shown in figure 1, implies that the comparison DPP-4 inhibitor versus sulfonylurea provides the most precise direct estimate in the network (variance 0.001) and suggests that there is only one comparison (DPP-4 inhibitor versus thiazolidinedione) at high risk of bias.

Figure 2 shows how the above command can be executed via the networkplot dialog box, accessed by typing db networkplot in the Command window.
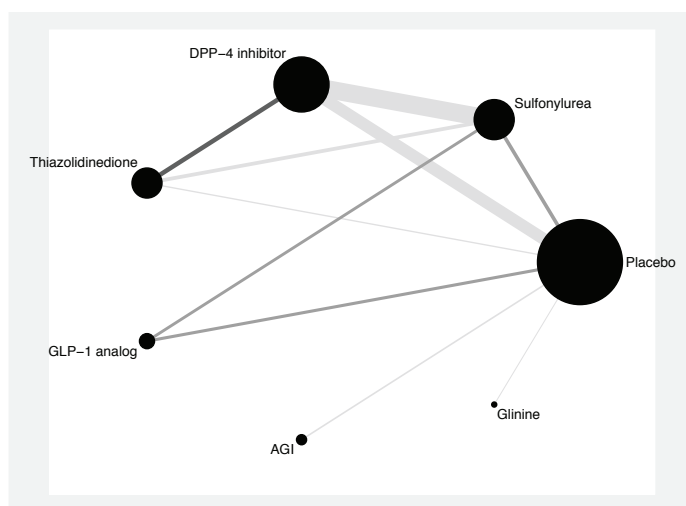
Figure 1. Plot of the antidiabetics network. The size of the nodes is proportional to the number of studies evaluating each intervention, and the thickness of the edges is proportional to the precision (the inverse of the variance) of each direct comparison. The color of the edges represents the average bias level of the corresponding comparisons with respect to blinding.
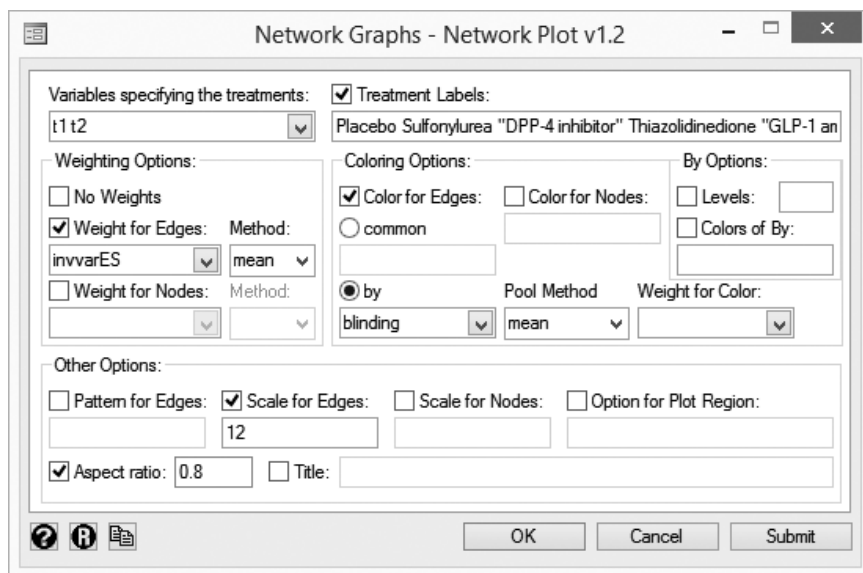

Figure 2. Example of using the `networkplot` dialog box for the antidiabetics network

3.3   The netweight command

Contributions of direct comparisons to the network

When performing a network meta-analysis, each direct comparison contributes to the estimation of each network meta-analytic summary effect by a different weight (Krahn, Binder, and König 2013; Lu et al. 2011). Identifying comparisons with large or small contributions is of great interest and enhances the understanding of the evidence flow. The contributions of the different pieces of evidence within a network have also been used in the evaluation of the quality of evidence from network meta-analysis (Salanti et al. 2014).

These contributions are obtained as complicated functions of a) the structure of the network and b) the variances of each pairwise direct summary effect. We obtain the contributions by fitting the network meta-analysis model in two stages:

1. Estimation of the direct relative effects using standard pairwise meta-analysis.

2. Synthesis of the direct relative effects assuming consistency to estimate the network relative effects.

This approach results in the estimation of vector $\widehat{\boldsymbol{\mu}}$ that contains the network estimates for the basic parameters (that is, a subset of the available comparisons sufficient to estimate all the comparisons via consistency) (Krahn, Binder, and König 2013; Lu et al. 2011). Defining with $\mathbf{X}$ the design matrix expressing the linear relationships between the available direct comparisons and the basic parameters, the vector $\widehat{\boldsymbol{\mu}}$ is

$$\widehat{\boldsymbol{\mu}} = \left\{ \mathbf{X}^{'} \left(\widehat{\mathbf{v}}^{\mathbf{D}} \right)^{-1} \mathbf{X} \right\} \mathbf{X}^{'} (\widehat{\mathbf{v}}^{\mathbf{D}})^{-1} \widehat{\boldsymbol{\mu}}^{\mathbf{D}}$$

where $\widehat{\boldsymbol{\mu}}^{\mathbf{D}}$ and $\widehat{\mathbf{v}}^{\mathbf{D}}$ are the vectors including the available direct estimates and their variances, respectively. The matrix $\mathbf{H} = \{\mathbf{X}^{'}(\widehat{\mathbf{v}}^{\mathbf{D}})^{-1}\mathbf{X}\}\mathbf{X}^{'}(\widehat{\mathbf{v}}^{\mathbf{D}})^{-1}$ maps the direct estimates to the network estimates (often called "the hat matrix") and contains the weight of each direct comparison in the estimation of every basic parameter. The matrix $\mathbf{H}$ can be extended to incorporate the weights of the direct comparisons to every possible network summary effect via the consistency equations. Then, its elements h_{ij} ($i = 1, \ldots, N$ and $j = 1, \ldots, N^{'}$ with N being the number of all possible pairwise comparisons and $N^{'}$ being the number of all available direct comparisons) can be expressed as the percentage contribution of the column-defining direct comparison to the row-defining network estimate [which is $|h_{ij}|/(|h_{i1}| + \cdots + |h_{iN^{'}}|)$]. The percentage contribution of each direct comparison j to the entire network is $(\sum_{i=1}^{N} |h_{ij}|)/(\sum_{j=1}^{N^{'}} \sum_{i=1}^{N} |h_{ij}|)$.

Description of netweight

The `netweight` command calculates all direct pairwise summary effect sizes with their variances, creates the design matrix, and estimates the percentage contribution of each

direct comparison to the network summary estimates and in the entire network. Then, it produces the contribution plot that uses weighted squares to represent the respective contributions. The command can also combine the estimated contributions with a particular trial-level characteristic (for example, the risk of bias of the studies) and produce a bar graph showing the percentage of information in each network estimate that corresponds to the different levels of the characteristic.

Currently, `netweight` does not account for the correlation in direct effect sizes from multiarm trials.

Syntax for netweight

`netweight` _varlist_ $\big[$_if_$\big]$ $\big[$_in_$\big]$ $\big[$, <u>title</u>(_string_) <u>aspect</u>(#) <u>notable</u> <u>noplot</u>

    `random` `fixed` <u>tau2</u>(#) <u>bargraph</u>(by _groupvar pool_method_) `order`

    `bylevels`(#) `bycolors`(_string_) <u>scale</u>(_string_) <u>novalues</u> <u>nostudies</u>

    <u>color</u>(_string_) <u>symbol</u>(_string_) <u>noymatrix</u> <u>novmatrix</u> <u>noxmatrix</u> <u>nohmatrix</u> $\big]$

In _varlist_, the variables _ES_, _seES_, _t1_, and _t2_ (see section 3.1) should be specified.

Options for netweight

`title`(_string_), `aspect`(#), `notable`, `noplot`; see section 3.1.

`random` specifies that the random-effects model be used for the estimation of all direct pairwise summary effects (the default).

`fixed` specifies that the fixed-effect model be used for the estimation of all direct pairwise summary effects.

`tau2`(#) specifies a real nonnegative number to use as the heterogeneity variance common for all comparisons.

`bargraph`(by _groupvar pool_method_) specifies that a bar graph be drawn instead of the contribution plot. The bars are colored according to the bias level (see the `edgecolor()` option of the `networkplot` command), and their length is proportional to the percentage contribution of each direct comparison to the network estimates (Salanti et al. 2014).

    For a description of _groupvar_ and _pool_method_, see the `edgecolor()` option for the `networkplot` command, in section 3.2 above.

`order` specifies to order the direct comparisons (on the horizontal axis) presented in the bar graph according to their bias level. This option is ignored unless `bargraph()` is specified.

`bylevels`(#) specifies the number of levels for the _groupvar_ in `bargraph()` when there are more than three bias levels. In this case, the _groupvar_ variable can only be numeric. `bycolors()` is required with `bylevels()`.

bycolors(*string*) specifies the colors for each bias level of the *groupvar* in the respective order. bylevels() is required with bycolors().

scale(*string*) specifies a real number that is used to scale the weighted squares in the contribution plot.

novalues suppresses the display of the percentage contributions in the contribution plot.

nostudies suppresses the display of the number of included studies in each comparison in the contribution plot.

color(*string*) specifies the color of the weighted squares in the contribution plot.

symbol(*string*) specifies an alternative symbol for the weighted squares in the contribution plot.

noymatrix suppresses the display of the vector containing the direct relative effects.

novmatrix suppresses the display of the matrix containing the variances of the direct relative effects.

noxmatrix suppresses the display of the design matrix.

nohmatrix suppresses the display of the hat matrix that maps the direct estimates into the network estimates.

Example for netweight using the antiplatelet regimens network

The dataset of the antiplatelet regimens network has 29 observations. For each pairwise comparison, a log odds-ratio (*ES*) smaller than 0 suggests that the first treatment in that particular comparison (*t1*) is more effective than the second treatment (*t2*).

For this network, we have no information on the risk of bias of the included studies. To illustrate the use of netweight, we considered that the risk of bias might be associated with the year of study publication (represented by the variable year), and we classified the trials into three categories: studies published before 1990 (considered at high risk of bias), studies between 1990 and 2000 (at unclear risk), and studies after 2000 (at low risk).

The data are in the same format as described above for the antidiabetics network. The treatments have been coded as numbers (1 = placebo, 2 = thienopyridines+aspirin, 3 = aspirin, 4 = dipyridamole + aspirin, 5 = thienopyridines).

```
. use antiplatelet.dta, clear
. list in 1/5, clean noobs
    id   t1   t2          ES        seES   year        rob
     1    4    3   -.0622905   .2781967   1983       high
     1    3    1    -.450053   .2575157   1983       high
     1    4    1   -.5123435   .2592596   1983       high
     2    5    3    .2051014   .1378609   2003        low
     3    5    3   -.1010332   .0706103   1996    unclear
```

We use the `netweight` command to produce the contribution plot of the network shown in figure 3:

```
. netweight ES seES t1 t2, aspect(0.9) notable

Note: Effect sizes of 29 observations were reversed (stored in matrix e(R))
```

Direct comparisons in the network

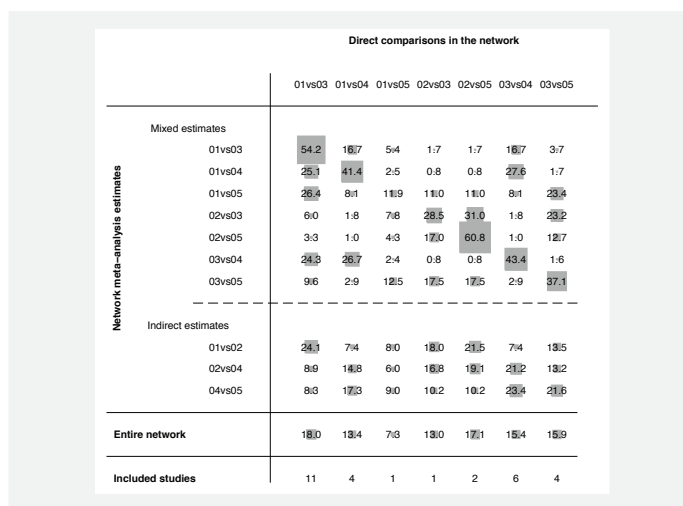| Network meta-analysis estimates		01vs03	01vs04	01vs05	02vs03	02vs05	03vs04	03vs05
Mixed estimates								
	01vs03	54.2	16.7	5.4	1.7	1.7	16.7	3.7
	01vs04	25.1	41.4	2.5	0.8	0.8	27.6	1.7
	01vs05	26.4	8.1	11.9	11.0	11.0	8.1	23.4
	02vs03	6.0	1.8	7.8	28.5	31.0	1.8	23.2
	02vs05	3.3	1.0	4.3	17.0	60.8	1.0	12.7
	03vs04	24.3	26.7	2.4	0.8	0.8	43.4	1.6
	03vs05	9.6	2.9	12.5	17.5	17.5	2.9	37.1
Indirect estimates								
	01vs02	24.1	7.4	8.0	18.0	21.5	7.4	13.5
	02vs04	8.9	14.8	6.0	16.8	19.1	21.2	13.2
	04vs05	8.3	17.3	9.0	10.2	10.2	23.4	21.6
Entire network		18.0	13.4	7.3	13.0	17.1	15.4	15.9
Included studies		11	4	1	1	2	6	4

Figure 3. Contribution plot for the antiplatelet regimens network. The size of the squares is proportional to the percentage contribution of the column-defining direct comparison to the row-defining network estimate.

The contribution plot of the network suggests that the comparison of placebo (treatment 1) versus aspirin (treatment 3) has the largest contribution in the entire network (18%).

Figure 4 shows how the above command can be executed via the `netweight` dialog box, accessed by typing `db netweight` in the Command window.

Figure 4. Example of using the `netweight` dialog box for the antiplatelet regimens network

Then, we produce a bar graph, shown in figure 5, showing how much information comes from high, unclear, and low risk of bias studies for each network estimate:

```
. netweight ES seES t1 t2, bargraph(by rob mean) notable

Note: Effect sizes of 29 observations were reversed (stored in matrix e(R))
```
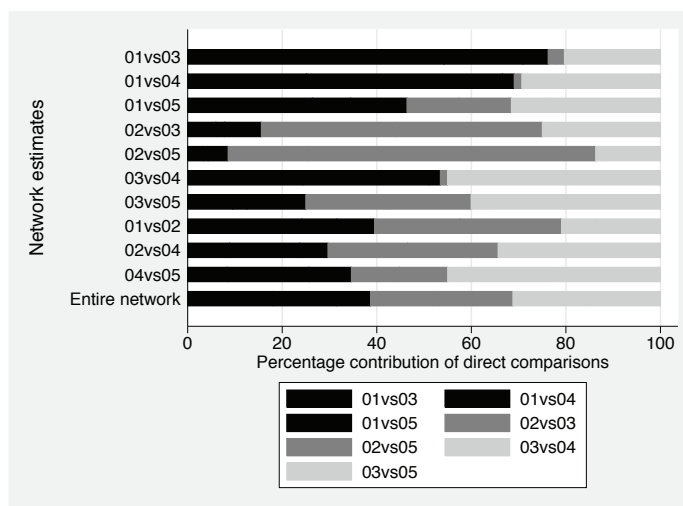

Figure 5. Bar graph showing the risk of bias for each network estimate and the entire network with respect to trial publication year for the antiplatelet regimens network. For each direct estimate, the average bias level of the included studies has been assumed as the comparison-specific level of bias.

The graph suggests that the estimated summary effects for the comparisons placebo versus aspirin, placebo versus dipyridamole + aspirin, and aspirin versus dipyridamole + aspirin derive most of their information from older studies.

Figure 6 shows how the above command can be executed via the `netweight` dialog box, accessed by typing `db netweight` in the Command window.
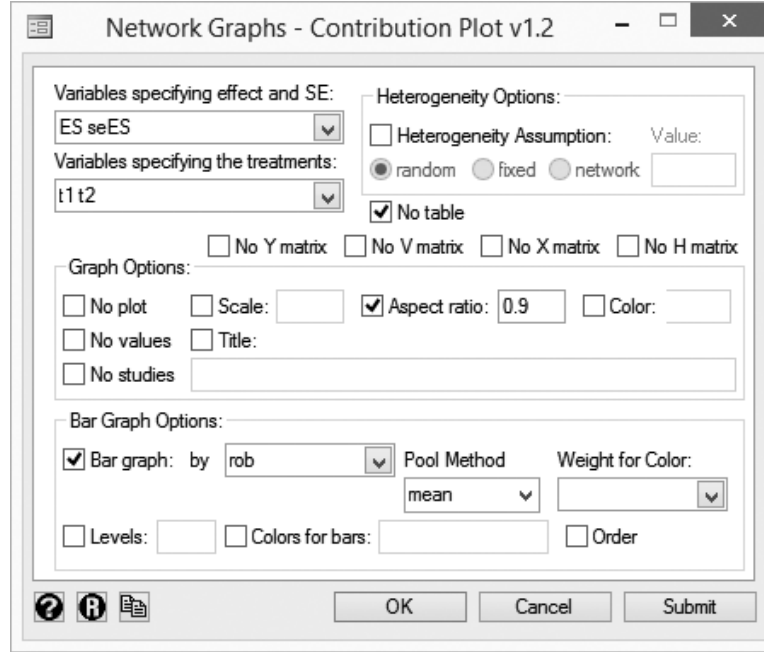

Figure 6. Another example of using the `netweight` dialog box for the antiplatelet regimens network

3.4   The ifplot command

Loop-specific approach for inconsistency

The "loop-specific approach" evaluates inconsistency separately in every closed loop of a network of interventions (Bucher et al. 1997). More specifically, in a network with L total number of loops, the inconsistency factor within each loop l ($l = 1, \ldots, L$) is estimated as

$$\widehat{w}_l^{XY} = |\widehat{\mu}_{XY}^D - \widehat{\mu}_{XY}^{I_l}|$$

where $\widehat{\mu}_{XY}^D$ and $\widehat{\mu}_{XY}^{I_l}$ are the estimated direct and indirect (using information only from studies in the l loop) summary effects for the XY comparison. The variance of $\widehat{w}_l^{XY}$ is

$$\mathrm{var}\left(\widehat{w}_l^{XY}\right) = \widehat{v}_{XY}^D + \widehat{v}_{XY}^{I_l}$$

where $\widehat{v}_{XY}^D$ and $\widehat{v}_{XY}^{I_l}$ are the variances of the direct and indirect estimates. Note that there is only one direct estimate for each XY comparison, whereas different loops may

give different indirect estimates (that is, $\widehat{\mu}_{XY}^{I_1} \neq \cdots \neq \widehat{\mu}_{XY}^{I_L}$). The choice of the comparison according to which inconsistency is estimated within each loop does not affect the results; that is, in a loop XYZ, it is $\widehat{w}_l^{XY} = \widehat{w}_l^{XZ} = \widehat{w}_l^{YZ} = \widehat{w}_l$.

Loops in which the lower confidence interval limit of the inconsistency factor does not reach the zero line are considered to present statistically significant inconsistency. However, the absence of statistically significant inconsistency is not evidence against the presence of inconsistency because of the multiple and correlated tests that are undertaken and the low power of the method (Song et al. 2012; Veroniki et al. 2014; Veroniki et al. 2013).

Description of ifplot

The `ifplot` command identifies all triangular and quadratic loops in a network of interventions and estimates the respective inconsistency factors and their uncertainties. Then, it produces the inconsistency plot that presents for each loop the estimated inconsistency factor and its confidence interval (truncated to 0). The command allows for different assumptions for the between-studies variance (that is, loop-specific, comparison-specific, or network-specific) and different estimators (for example, method of moments or restricted maximum likelihood).

Multiarm trials

The loop-specific approach does not account for the correlation in the effect sizes induced by multiarm trials. This is expected to emphasize the consistency in the loops because `ifplot` treats as independent the comparisons from the same trial, which are consistent by definition. The `ifplot` command slightly mitigates this by dropping one of the direct comparisons from the multiarm trials when it appears in a particular loop. Among the $\{k \times (k-1)\}/2$ comparisons belonging to the same trial that can be excluded from a loop, `ifplot` chooses the comparison with the largest number of studies within the loop. Inconsistency is not identifiable in loops formed only by multiarm trials, and hence such loops are excluded.

Syntax for ifplot

ifplot *varlist* $\big[\,$*if*$\,\big]$ $\big[\,$*in*$\,\big]$ $\big[\,$, <u>lab</u>els(*string*) eform keep <u>notable</u> <u>noplot</u>

    <u>ci</u>level(*integer*) <u>tau2</u>(loop $|$ comparison $|$ #) mm reml eb random fixed

    <u>summary</u> <u>details</u> <u>plotoptions</u>(*string*) <u>xlab</u>el(#,#,...,#) <u>separate</u> $\big]$

In *varlist*, the variables *ES*, *seES*, *t1*, *t2*, and *id* (see section 3.1) should be specified.

Options for ifplot

`labels(`*string*`)`, `eform`, `keep`, `notable`, `noplot`, `cilevel(`*integer*`)`; see section 3.1.

`tau2(loop | comparison | #)` specifies the assumption for the heterogeneity variance.

   `loop` specifies a common heterogeneity for all comparisons within each loop but different heterogeneities across loops (the default).

   `comparison` specifies different comparison-specific heterogeneities for each loop.

   `#` specifies a real nonnegative number for the network-specific heterogeneity variance common for all loops and comparisons.

`mm` specifies that the method of moments (DerSimonian and Laird 1986) (the default) be used for the estimation of the loop-specific heterogeneity (when `tau2(loop)` has been specified). When `tau2(comparison)` has been specified, this is the only possible method for estimating heterogeneity.

`reml` specifies that the restricted maximum likelihood method (Viechtbauer 2005) be used for the estimation of the loop-specific heterogeneity (when `tau2(loop)` has been specified).

`eb` specifies that the empirical Bayes method (Morris 1983) be used for the estimation of the loop-specific heterogeneity (when `tau2(loop)` has been specified).

`random` specifies that the random-effects model (the default) be used for all comparisons (when `tau2(comparison)` has been specified). When `tau2(loop)` or `tau2(#)` has been specified, this is the only possible model.

`fixed` specifies that the fixed-effect model be used for all comparisons (for use when `tau2(comparison)` has been specified). `fixed` is equal to `tau2(0)`.

`summary` specifies to display all direct and indirect summary estimates for every loop.

`details` specifies to display all comparisons that are dropped when there are multiarm studies in a loop.

`plotoptions(`*string*`)` specifies standard options of `metan` that handle the appearance of the inconsistency plot.

`xlabel(#,#,...,#)` specifies the values to display on the horizontal axis. The input values must be comma-delimited.

`separate` specifies to display inconsistency factors separately for loops with and without evidence of statistical inconsistency in the inconsistency plot and the output results.

Example for ifplot using the antiplatelet regimens network

The data for this network are described in section 3.3.

We use the `ifplot` command to assess the presence of inconsistency in every closed loop of the network. The resulting inconsistency plot is shown in figure 7.

```
. ifplot ES seES t1 t2 id, eform plotopt(texts(180)) xlabel(1,1.3,1.8) notab
* 3 triangular loops found
```


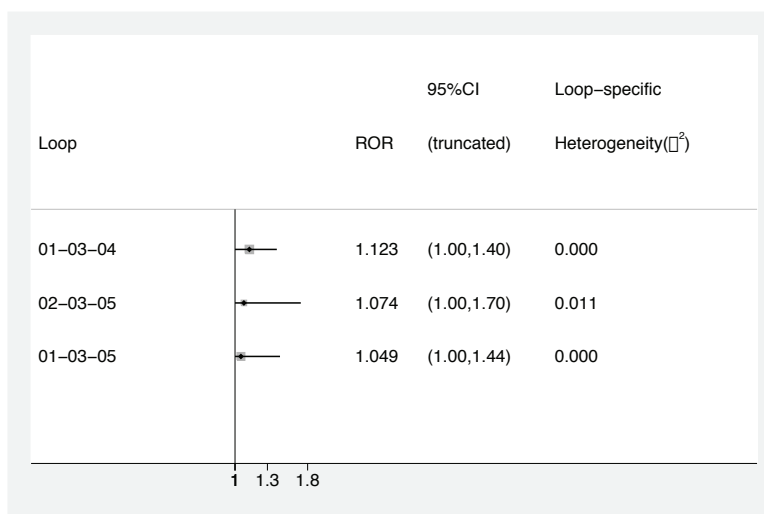| Loop		ROR	95%CI (truncated)	Loop–specific Heterogeneity(τ^2)
01–03–04		1.123	(1.00,1.40)	0.000
02–03–05		1.074	(1.00,1.70)	0.011
01–03–05		1.049	(1.00,1.44)	0.000

1 1.3 1.8

Figure 7. Inconsistency plot for the antiplatelet regimens network showing for each loop the ratio of odds ratios between direct and indirect estimates. The within-loop heterogeneities have been estimated using the method of moments estimator.

Using the loop-specific approach, none of the three loops in the network was found to present statistically significant inconsistency, although large values are included in the confidence intervals of the inconsistency factors.

Figure 8 shows how the above command can be executed via the `ifplot` dialog box, accessed by typing `db ifplot` in the Command window.
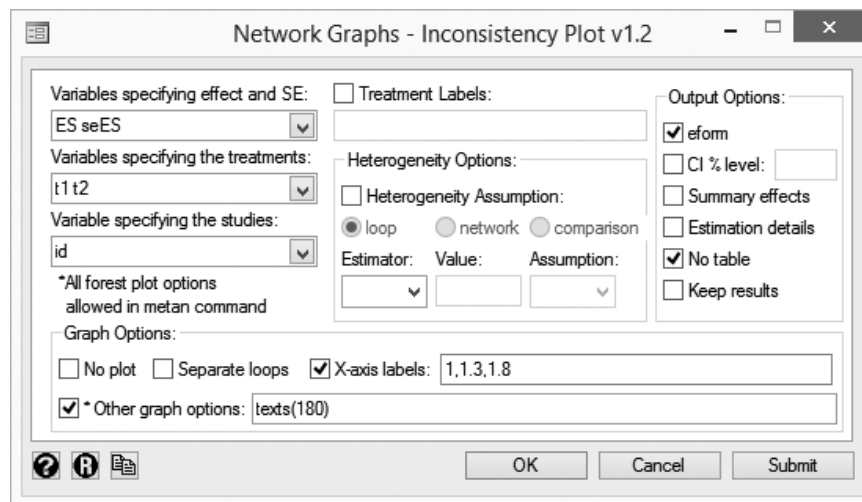

Figure 8. Example of using the `ifplot` dialog box for the antiplatelet regimens network

3.5   The netfunnel command

Comparison-adjusted funnel plot

Differences in the relative effects between small and large trials in a network of interventions often challenge the interpretation of the pairwise summary effects and need exploration (Chaimani et al. 2013b; Moreno et al. 2011). Extending the funnel plot from pairwise to network meta-analysis must account for different treatment comparisons being included: each comparison has its own summary effect. Hence, all the studies in the network do not have a common reference line of symmetry. In the comparison-adjusted funnel plot, the horizontal axis shows the difference of each i-study's estimate y_{iXY} from the direct summary effect for the respective comparison $(y_{iXY} - \widehat{\mu}_{XY}^{D})$, while the vertical axis presents a measure of dispersion of y_{iXY}. In the absence of small-study effects, all studies are expected to lie symmetrically around the 0 line of the comparison-adjusted funnel plot (Chaimani et al. 2013a; Chaimani and Salanti 2012).

Obtaining meaningful conclusions from this graph requires the definition of all comparisons across studies in a consistent direction, such as active intervention versus inactive, newer treatment versus older (Salanti et al. 2010), or sponsored versus nonsponsored.

Description of netfunnel

The `netfunnel` command plots a comparison-adjusted funnel plot for assessing small-study effects within a network of interventions. The command can plot observations using a different color for each comparison. The default direction for all comparisons in the network is in alphabetical or numerical order based on the codes of the treatments that appear in the dataset (for example, A versus B, B versus C, A versus C, or 1 versus 2, 2 versus 3, 1 versus 3). As explained above, investigators should order the treatments in a meaningful way rather than using the default ordering; for example, one might order the codes of the treatments from oldest to newest.

Syntax for netfunnel

netfunnel *varlist* $\begin{bmatrix} if \end{bmatrix}$ $\begin{bmatrix} in \end{bmatrix}$ $\begin{bmatrix} ,$ <u>scatt</u>eroptions(*string*) <u>xtitle</u>(*string*)

   <u>xlab</u>el(*string*) fixed random bycomparison <u>add</u>plot(*string*) <u>ytitle</u>(*string*)

   <u>ylab</u>el(*string*) noci <u>noalphabetical</u> $\end{bmatrix}$

   In *varlist*, the variables *ES*, *seES*, *t1*, and *t2* (see section 3.1) should be specified. The order of *t1* and *t2* specifies the direction of the comparisons in the plot; specifying *t1* first means that the effect sizes have been estimated as *t1* versus *t2*, whereas specifying *t1* second means that the direction of the effect sizes is *t2* versus *t1*. This affects the appearance of the comparison-adjusted funnel plot only when the option `noalphabetical` has not been specified; however, the interpretation of the graph always depends on the direction of the relative effects.

Options for netfunnel

`scatteroptions(`*string*`)`, `xtitle(`*string*`)`, `xlabel(`*string*`)`; see section 3.1.

`fixed` specifies that the fixed-effect model be used to estimate the direct summary effects (the default).

`random` specifies that the random-effects model be used to estimate the direct summary effects.

`bycomparison` specifies that different colors be used for different pairwise comparisons.

`addplot(`*string*`)` requests the addition of other `twoway` graphs specified in *string*.

`ytitle(`*string*`)` and `ylabel(`*string*`)` specify a title and the values, respectively, that are displayed for the vertical axis.

`noci` specifies to not include the pseudo 95% confidence interval lines for the difference $(y_{iXY} - \widehat{\mu}_{XY}^{D})$ in the plot.

`noalphabetical` specifies to include the comparisons in the plot as specified by the user (that is, *t1* versus *t2* for all studies) and not in alphabetical order (that is, A versus B, B versus C, and so on; the default).

Example for netfunnel using the antidiabetics network

The data for this network are described in section 3.2.

We use the `netfunnel` command to assess whether small and large trials tend to give different efficacy results. We focus on the comparisons of all active treatments against placebo, which might be more prone to small-study effects. As pointed out in section 3.1, all effect sizes (*ES*) in the dataset have been estimated as *t1* versus *t2* (placebo [1] appears only in *t2*); hence, all comparisons including placebo have been estimated as active treatment versus placebo. The resulting plot in figure 9 appears symmetric, implying the absence of small-study effects in the network.

```
. use antidiabetics_efficacy_wide.dta, clear
. netfunnel ES seES t1 t2 if t2==1, noalphabetical ylabel(0 0.1 0.2 0.3)

Comparisons in the plot:
   1.      7 vs 1
   2.      6 vs 1
   3.      5 vs 1
   4.      4 vs 1
   5.      3 vs 1
   6.      2 vs 1
```
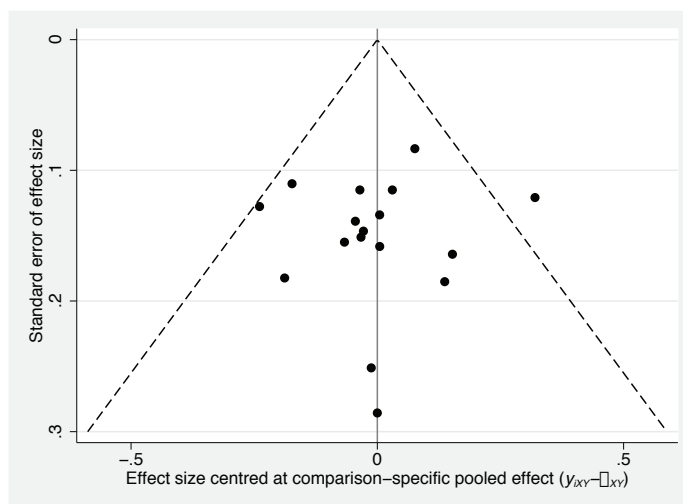

Figure 9. Comparison-adjusted funnel plot of the placebo-controlled antidiabetic trials

The comparison-adjusted funnel plot of figure 9 appears symmetric, implying the absence of small-study effects in the network.

Figure 10 shows how the above command can be executed via the `netfunnel` dialog box, accessed by typing `db netfunnel` in the Command window.
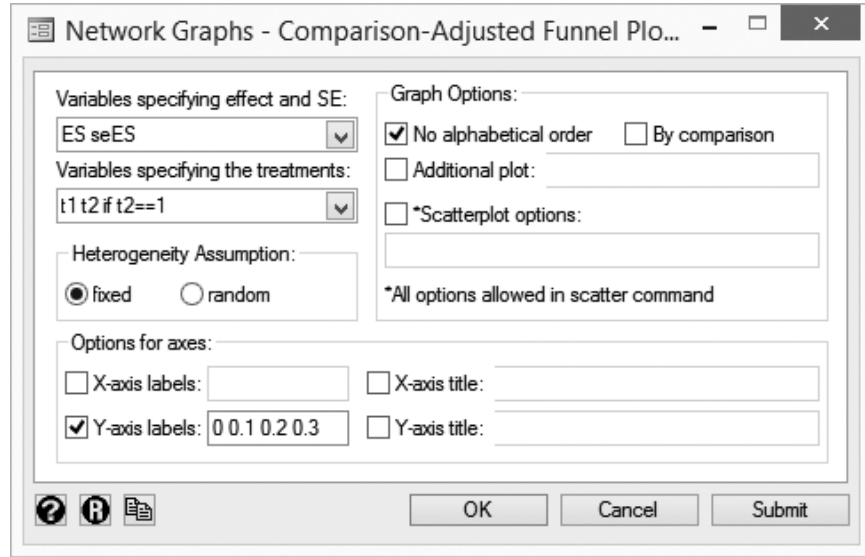

Figure 10. Example of using the `netfunnel` dialog box for the antidiabetics network

3.6   The intervalplot command

Predictive intervals in meta-analysis

The extent of uncertainty in the estimated treatment effects in meta-analysis is reflected not only by the confidence intervals but also by the predictive intervals that incorporate the extent of heterogeneity (Higgins, Thompson, and Spiegelhalter 2009; Riley, Higgins, and Deeks 2011). The predictive interval is the interval within which the relative treatment effect of a future study is expected to lie. For the relative effect of treatments X versus Y, we estimate the predictive interval as

$$\widehat{\mu}_{XY} \pm t^{\alpha}_{\text{d.f.}} \sqrt{\widehat{\tau}^2 + \widehat{v}_{XY}}$$

where $\widehat{\tau}^2$ is the estimated heterogeneity variance. A common choice for the degrees of freedom (d.f.) of the t distribution in a standard pairwise meta-analysis is d.f. $= S - 2$ (with S being the total number of studies). For the case of network meta-analysis, this can be modified into d.f. $= S - N^{'} - 1$ (with $N^{'}$ being the number of available direct comparisons) (Cooper, Hedges, and Valentine 2009).

Description of intervalplot

The `intervalplot` command plots the estimated effect sizes and their uncertainties for all pairwise comparisons in a network meta-analysis. More specifically, `intervalplot` produces a forest plot where the horizontal lines representing the confidence intervals are extended to simultaneously show the predictive intervals. The treatment effects and their uncertainties can be estimated within Stata using the `mvmeta` (or `network`) package or by using other software.

Syntax for intervalplot

`intervalplot` [*varlist*] [*if*] [*in*] [, <u>mvmeta</u>results <u>nomv</u>meta
    <u>lab</u>els(*treatment_labels* | comparison) eform keep <u>title</u>(*string*) <u>notable</u>
    noplot <u>xtitle</u>(*string*) <u>xlab</u>el(*string*) predictions separate
    <u>ref</u>erence(*reference_treatment*) <u>noval</u>ues null(*#*) <u>nulloptions</u>(*line_options*)
    <u>fcicolor</u>(*string*) <u>scicolor</u>(*string*) fcipattern(*string*) scipattern(*string*)
    symbol(*string*) <u>range</u>(*string*) labtitle(*string*) <u>valuetitle</u>(*string*)
    symbolsize(*string*) <u>textsize</u>(*string*) <u>lwidth</u>(*string*) <u>margin</u>(*# # # #*)]

The network meta-analysis summary effects and their uncertainties are required as input for this command. These can be provided in two different ways:

1. By running `intervalplot` directly after performing network meta-analysis with the `mvmeta` or `network` command. In this case, the option `mvmetaresults` (the default) may be specified, and the *varlist* should be omitted.

2. By including the network meta-analysis estimates for all comparisons we want to plot and their lower/upper confidence limits as variables in a dataset (for example, variables *ES*, *LCI*, and *UCI*; see section 3.1). Optionally, we can include the variables *LPI* and *UPI*, the predictive lower and upper limits. In this case, the option `nomvmeta` is required.

Options for intervalplot

`mvmetaresults`, `nomvmeta`, `labels`(*treatment_labels* | comparison), `eform`, `keep`,
    `title`(*string*), `notable`, `noplot`, `xtitle`(*string*), `xlabel`(*string*); see section 3.1.

`predictions` specifies that the predictive intervals be added in the plot (when the option `nomvmeta` has not been specified). In the default plot, only the confidence intervals are displayed.

`separate` specifies that results in the plot be classified according to the comparator treatment (possible when `labels()` has been specified).

`reference(`*reference_treatment*`)` specifies that only the relative effects of each treatment versus the *reference_treatment* be displayed (when `labels()` has been specified). Note that the reference treatment specified in `intervalplot` does not necessarily need to be the reference treatment of the analysis.

`novalues` suppresses the display of the numerical estimates and their uncertainties in the plot.

`null(`*#*`)` specifies the value for the line of no effect.

`nulloptions(`*line_options*`)` specifies options for the line of no effect.

`fcicolor(`*string*`)`, `scicolor(`*string*`)`, `fcipattern(`*string*`)`, `scipattern(`*string*`)`, and `symbol(`*string*`)` specify the color, the pattern, and the symbol for the confidence and predictive intervals, respectively (when specified). The defaults are `fcicolor(black)` and `scicolor(cranberry)`, `fcipattern(solid)` and `scipattern(dash)`, and `fcisymbol(diamond)` and `symbol(circle)`.

`range(`*string*`)` specifies the range for the horizontal axis.

`labtitle(`*string*`)` and `valuestitle(`*string*`)` specify titles for the treatment comparisons and the values of the confidence and predictive intervals, respectively.

`symbolsize(`*string*`)` and `textsize(`*string*`)` specify the sizes for the symbols of the effect sizes and the text, respectively. The default is `symbolsize(small)`.

`lwidth(`*string*`)` specifies the width for the horizontal lines representing the confidence and predictive intervals.

`margin(`*# # # #*`)` specifies the margins for the region of the plot.

Example for intervalplot using the antihypertensives network

The dataset for this network consists of 48 observations equal to the total number of study arms. The treatments (t) have been coded as numbers (1 = placebo, 2 = beta-blockers, 3 = diuretics, 4 = calcium-channel blockers, 5 = angiotensin-converting-enzyme inhibitors, 6 = angiotensin-receptor blockers), and the data are in the appropriate format for `network` (see White [2015; 2011] and White et al. [2012] for a detailed description), where `r` is the number of events and `n` is the total participants in each arm.

```
. use antihypertensives.dta, clear
. list in 1/5, clean noobs
    id     r      n    t
     1    45    410    5
     1    70    405    2
     1    32    202    4
     2   119   4096    5
     2   154   3954    4
```

Using the `network setup` command (White 2015), we prepared the data in a format suitable to run network meta-analysis in Stata.[2]

Then, we performed network meta-analysis via the command `network meta` to obtain the network estimates of the relative treatment effects. We then run the command `intervalplot` to estimate the predictive intervals and see the result in figure 11.

```
. network setup r n, studyvar(id) trtvar(t) numcodes ref(1)
  (output omitted)
. network meta c
  (output omitted)
. intervalplot, eform pred null(1) labels(Placebo BB Diuretics CCB ACE ARB)
> separate margin(10 40 5 5) notable

 The intervalplot command assumes that the saved results from mvmeta or network
> meta commands have been derived from the current dataset
```
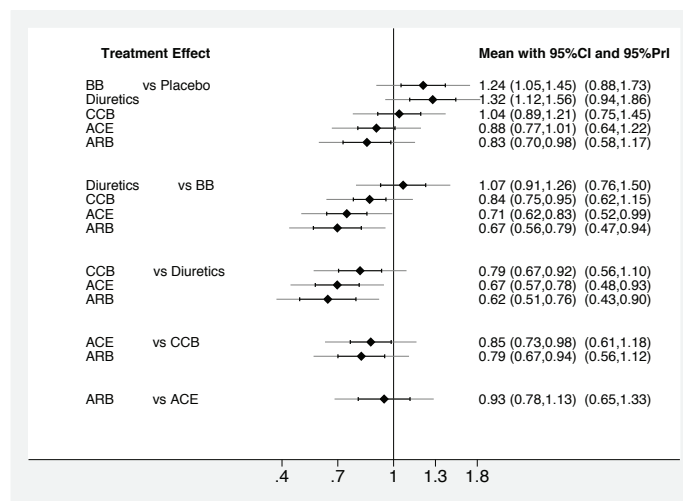

Figure 11. Predictive intervals plot for the antihypertensives network. The graph presents the network estimates for all pairwise comparisons. Black horizontal lines represent the confidence intervals, and gray lines represent the predictive intervals.

Conclusions about some comparisons seem to be substantially affected by the estimated (common) heterogeneity in the network; although their confidence intervals suggest an association, the respective predictive interval crosses the line of no effect and suggests that future studies might favor either treatment (for example, diuretics versus placebo or calcium-channel blockers versus diuretics).

---

2. Note that when only study-level data (that is, effect size and standard error for each study) are available, the `network import` and `network convert` augment commands should be used instead of `network setup`.

Figure 12 shows how the above command can be executed via the `intervalplot` dialog box, accessed by typing `db intervalplot` in the Command window.

Figure 12. Example of using the `intervalplot` dialog box for the antihypertensives network

3.7 The netleague command

Description of netleague

The `netleague` command creates a "league table" showing in the off-diagonal cells the relative treatment effects for all possible pairwise comparisons estimated in a network meta-analysis (Cipriani et al. 2009). The diagonal cells include the names of the competing treatments in the network, which can be sorted according to a prespecified order.

Syntax for netleague

netleague $\big[$ *varlist* $\big]$ $\big[$, <u>mvmeta</u>results <u>nomvm</u>eta <u>lab</u>els(*string*) eform

sort(*string*) <u>export</u>(*string*) nokeep$\big]$

The network meta-analysis summary effects and their uncertainties can be provided as input in two different ways:

1. By running `netleague` directly after performing network meta-analysis with the `mvmeta` or `network` command. In this case, the option `mvmetaresults` (the default) may be specified, and the *varlist* should be omitted.

2. By including these summary estimates as variables in the dataset; the variables *ES*, *seES*, *t1*, and *t2* (see section 3.1) should be specified in *varlist*. In this case, the option `nomvmeta` is required.

Options for netleague

`mvmetaresults`, `nomvmeta`, `labels(`*string*`)`, `eform`; see section 3.1.

`sort(`*string*`)` specifies the order for treatments from top to bottom in the league table. When the option `nomvmeta` has not been specified, `sort()` requires the option `labels()`. When the option `nomvmeta` has been specified, the names of treatments should be given as displayed in the dataset in variables *t1* and *t2*. By default, the treatments are ordered alphabetically from bottom to top.

`export(`*string*`)` specifies the path of an Excel file where the league table is exported.

`nokeep` specifies not to store the league table at the end of the dataset.

Example for netleague using the antihypertensives network

The data for this network are described in section 3.6.

Using the `network setup` and `network meta` commands, we perform network meta-analysis to obtain the network estimates of the relative treatment effects (see section 3.6). We then directly run the `netleague` command to produce the league table of the network. We order the treatments according to their relative rankings based on the surface under the cumulative ranking curves (SUCRA) percentages.

```
. netleague, labels(Placebo BB Diuretics CCB ACE ARB)
> sort(ARB ACE Placebo CCB BB Diuretics) eform

 Warning: The existing dataset is stored as a temporary file
 Warning: To save any changes applied at this temporary file in a specific
> directory you need to use the ´Save as´ menu

 The league table has been stored at the end of the dataset
```

The league table for the first four treatments of the network is

```
. list _ARB _ACE _Placebo _CCB in 1/4, clean noobs noheader
                  ARB    1.07 (0.89,1.29)   1.21 (1.02,1.44)   1.26 (1.07,1.49)
    0.93 (0.78,1.13)                 ACE    1.13 (0.99,1.29)   1.18 (1.02,1.36)
    0.83 (0.70,0.98)   0.88 (0.77,1.01)            Placebo    1.04 (0.89,1.21)
    0.79 (0.67,0.94)   0.85 (0.73,0.98)   0.96 (0.82,1.12)                 CCB
```

Figure 13 shows how the above command can be executed via the `netleague` dialog box, accessed by typing `db netleague` in the Command window.
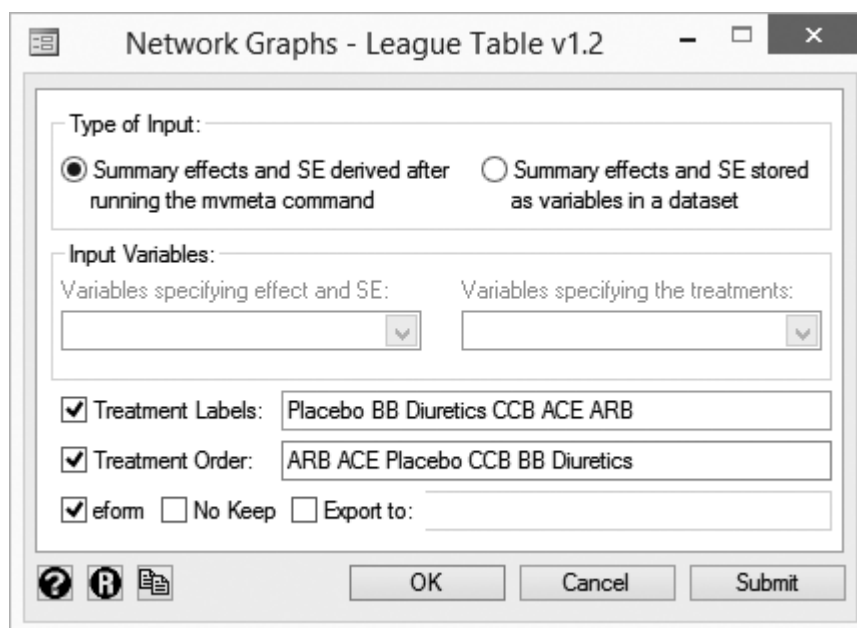

Figure 13. Example of using the `netleague` dialog box for the antihypertensives network

Note that the user can choose between the upper and lower triangle of the table, which provide the network estimates in the opposite direction (that is, row- versus column-defining treatment or column- versus row-defining treatment).

3.8   The sucra command

Ranking probabilities for competing treatments

When performing a network meta-analysis, it is common to estimate the ranking probabilities p_{tr} for each treatment t being at order r. Then, the competing treatments can be classified using the cumulative probabilities p_{tr}^{cum} that treatment t is ranked among the first r places. Two relative ranking measures that account for the uncertainty in treatment order are as follows:

1. The SUCRA that expresses the percentage of effectiveness or safety each treatment has compared with an "ideal" treatment always ranked first without uncertainty (Salanti, Ades, and Ioannidis 2011) [given by $(\sum_{r=1}^{T-1} p_{tr}^{\text{cum}})/(T-1)$, with T being the total number of treatments].

2. The mean rank, which is the mean of the distribution of the ranking probabilities [equal to $\sum_{r=1}^{T}(p_{tr} \times r)$].

Description of sucra

The `sucra` command gives the SUCRA percentages and mean ranks, and produces rankograms (line plots of the probabilities versus ranks) and cumulative ranking plots (line plots of the cumulative probabilities versus ranks) for all treatments in a network of interventions.

Syntax for sucra

`sucra` [*varlist*] [, <u>mvmeta</u>results <u>nomv</u>meta <u>label</u>s(*string*) <u>title</u>(*string*)

  <u>notable</u> <u>noplot</u> <u>compare</u>(*varlist*) <u>stats</u>(*string*) <u>rprobabilities</u>(*string*)

  <u>reverse</u> <u>rankograms</u> <u>names</u>(*string*) <u>lcolol</u>(*string*) <u>lpattern</u>(*string*)]

The input in *varlist* is the ranking probabilities for all treatments and ranks from a network meta-analysis. This can be provided in three different ways:

1. By running `sucra` after performing network meta-analysis with the `mvmeta` command (where the option `pbest(min|max, zero all reps() gen())` has been added; see the help file for `mvmeta`) or with `network rank min|max, zero all reps() gen()` (see the help file for `network`). These commands add in the data $t \times r$ new variables, each one corresponding to the probability of the rth rank for each treatment. These new variables are typically named `prob`*r*_*t* (where `prob` is the prefix we have specified in `gen()` of `pbest()`, r is the rank, and t is the treatment) and automatically appear at the end of the dataset. If `sucra` is run after `mvmeta` or `network`, the option `mvmetaresults` may be specified (the default), and all the variables containing the probability for a treatment being at a particular rank should be specified in *varlist*.

2. By providing the columns of the treatment-by-ranking probabilities matrix as variables in the dataset. In this case, all variables containing the ranking probabilities for each treatment should be specified in *varlist*, and the option `nomvmeta` is required.

3. By specifying the path of a `.txt` file where the ranking probabilities are stored (for example, after running network meta-analysis in WinBUGS). See table 1 for an example of this `.txt` file. The path of the file should be specified in option `stats()`, and the variable representing the ranking probabilities (for example, `prob`) should

be specified in option `rprobabilities()`; the *varlist* should be omitted. The command assumes that each node $\text{prob}[t, r]$ $(t, r = 1, \ldots, 6)$ represents the probability of treatment t being at order r, and the opposite (that is, $\text{prob}[r, t]$) when the option `reverse` has been specified. The option `nomvmeta` is also required.

Table 1. Example of a `.txt` file including the ranking probabilities for the antihypertensives network as estimated from WinBUGS

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
prob[1,1]	0.00659	0.08091	1.825E−4	0.0	0.0	0.0	1001	300000
prob[1,2]	0.071	0.2568	7.242E−4	0.0	0.0	1.0	1001	300000
prob[1,3]	0.6391	0.4803	0.001344	0.0	1.0	1.0	1001	300000
prob[1,4]	0.2769	0.4475	0.001472	0.0	0.0	1.0	1001	300000
prob[1,5]	0.00617	0.07831	1.631E−4	0.0	0.0	0.0	1001	300000
prob[1,6]	2.333E−4	0.01527	2.808E−5	0.0	0.0	0.0	1001	300000
prob[2,1]	0.0	0.0	1.054E−13	0.0	0.0	0.0	1001	300000
prob[2,2]	7.0E−5	0.008366	1.512E−5	0.0	0.0	0.0	1001	300000
prob[2,3]	8.867E−4	0.02976	5.734E−5	0.0	0.0	0.0	1001	300000
prob[2,4]	0.01184	0.1082	2.45E−4	0.0	0.0	0.0	1001	300000
prob[2,5]	0.7448	0.436	0.001502	0.0	1.0	1.0	1001	300000
prob[2,6]	0.2424	0.4285	0.001486	0.0	0.0	1.0	1001	300000
prob[3,1]	3.333E−6	0.001826	3.339E−6	0.0	0.0	0.0	1001	300000
prob[3,2]	3.333E−5	0.005773	1.05E−5	0.0	0.0	0.0	1001	300000
prob[3,3]	3.333E−4	0.01825	3.458E−5	0.0	0.0	0.0	1001	300000
prob[3,4]	0.00566	0.07502	1.505E−4	0.0	0.0	0.0	1001	300000
prob[3,5]	0.2372	0.4254	0.001465	0.0	0.0	1.0	1001	300000
prob[3,6]	0.7567	0.4291	0.001489	0.0	1.0	1.0	1001	300000
prob[4,1]	0.001077	0.03279	6.232E−5	0.0	0.0	0.0	1001	300000
prob[4,2]	0.01836	0.1343	3.052E−4	0.0	0.0	0.0	1001	300000
prob[4,3]	0.2728	0.4454	0.001423	0.0	0.0	1.0	1001	300000
prob[4,4]	0.6957	0.4601	0.001464	0.0	1.0	1.0	1001	300000
prob[4,5]	0.01135	0.1059	2.342E−4	0.0	0.0	0.0	1001	300000
prob[4,6]	6.167E−4	0.02483	4.823E−5	0.0	0.0	0.0	1001	300000
prob[5,1]	0.2761	0.4471	0.001272	0.0	0.0	1.0	1001	300000
prob[5,2]	0.6638	0.4724	0.001367	0.0	1.0	1.0	1001	300000
prob[5,3]	0.05498	0.2279	5.915E−4	0.0	0.0	1.0	1001	300000
prob[5,4]	0.005007	0.07058	1.447E−4	0.0	0.0	0.0	1001	300000
prob[5,5]	6.333E−5	0.007958	1.44E−5	0.0	0.0	0.0	1001	300000
prob[5,6]	3.333E−6	0.001826	3.339E−6	0.0	0.0	0.0	1001	300000
prob[6,1]	0.7162	0.4508	0.001309	0.0	1.0	1.0	1001	300000
prob[6,2]	0.2467	0.4311	0.001153	0.0	0.0	1.0	1001	300000
prob[6,3]	0.03186	0.1756	4.231E−4	0.0	0.0	1.0	1001	300000
prob[6,4]	0.004833	0.06935	1.409E−4	0.0	0.0	0.0	1001	300000
prob[6,5]	3.6E−4	0.01897	3.588E−5	0.0	0.0	0.0	1001	300000
prob[6,6]	3.333E−5	0.005773	1.15E−5	0.0	0.0	0.0	1001	300000

Options for sucra

`mvmetaresults`, `nomvmeta`, `labels(`*string*`)`, `title(`*string*`)`, `notable`, `noplot`; see section 3.1.

`compare(`*varlist*`)` specifies a second set of variables containing ranking probabilities. These can be, for example, the ranking probabilities for the same treatments but for different outcomes. `compare()` will add a second ranking plot to the existing ranking plot for each treatment.

`stats(`*string*`)` specifies the path of the file that has ranking probabilities.

`rprobabilities(`*string*`)` specifies the variable representing the ranking probabilities.

`reverse` specifies that the node probabilities are $\text{prob}(r, t)$ instead of $\text{prob}(t, r)$ with t being treatment and r being order (when option `stats()` has also been specified).

`rankograms` specifies that rankograms be drawn instead of cumulative ranking probability plots.

`names(`*string*`)` specifies a label name for the first (specified in *varlist*) and second (specified in `compare(`*varlist*`)`) set of ranking probabilities. An example for relative ranking results from different outcomes would be `names("Effectiveness" "Acceptability")`. These label names are displayed in the output results and ranking plots.

`lcolol(`*string*`)` and `lpattern(`*string*`)` specify the color and the pattern, respectively, of the lines in the ranking plots for the first and second (separated with a space) set of ranking probabilities. The default colors are `lcolol(black cranberry)`, and the default patterns are `lpattern(solid dash)`.

Example for sucra using the antihypertensives network

The data for this network are described in section 3.6.

Using the `network rank` command after `network setup` and `network meta` (see section 3.5), we performed network meta-analysis to obtain the ranking probabilities for all competing treatments in the network. These have been stored at the end of the dataset in the following format:

```
. network rank min, zero all reps(10000) gen(prob)
  (output omitted)
. list probmin_zero-probmin__y_3 prob2_zero-prob2__y_3 in 1, clean table noobs
> abbreviate(14)

   probmin_zero  probmin__y_2  probmin__y_3  prob2_zero  prob2__y_2  prob2__y_3
            .17             0             0    4.540009           0           0
```

Each $\mathtt{prob}r\_t$ represents the probability of treatment t (with $t = \mathtt{zero}, \mathtt{\_y\_2}, \mathtt{\_y\_3}$) being at order r (with $r = \mathtt{min}, 2, 3$). Note that $\mathtt{min}$ represents first rank (because the outcome is harmful), $\mathtt{zero}$ represents the reference treatment, and $\mathtt{\_y\_2}$ and $\mathtt{\_y\_3}$ are treatments coded as 2 and 3. For example, according to the above results, the reference treatment has a 4.54% probability of being second, that is, the variable $\mathtt{prob2\_zero}$.

Then, we run the $\mathtt{sucra}$ command to produce the rankograms:

```
. sucra prob*, labels(Placebo BB Diuretics CCB ACE ARB) rankog
Treatment Relative Ranking of Model 1
```

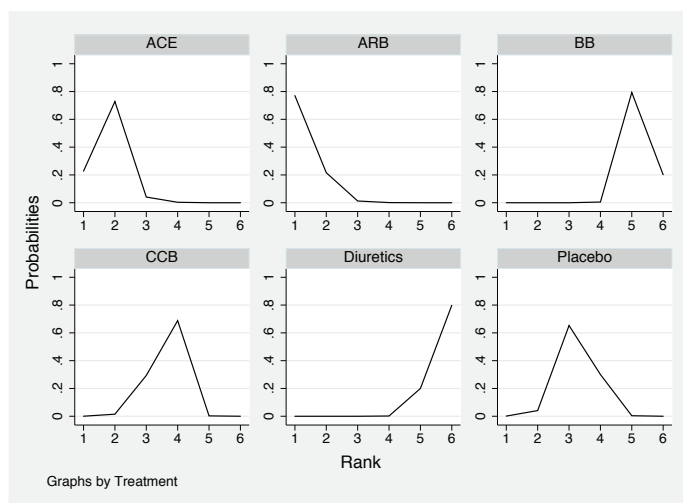| Treatment	SUCRA	PrBest	MeanRank
Placebo	54.9	0.2	3.3
BB	16.2	0.0	5.2
Diuretics	4.0	0.0	5.8
CCB	46.4	0.1	3.7
ACE	83.8	24.0	1.8
ARB	94.8	75.8	1.3

Figure 14. Rankograms for the antihypertensives network showing the probability for every treatment being at a particular order

Figure 15 shows how the above command can be executed via the `sucra` dialog box, accessed by typing `db sucra` in the Command window.
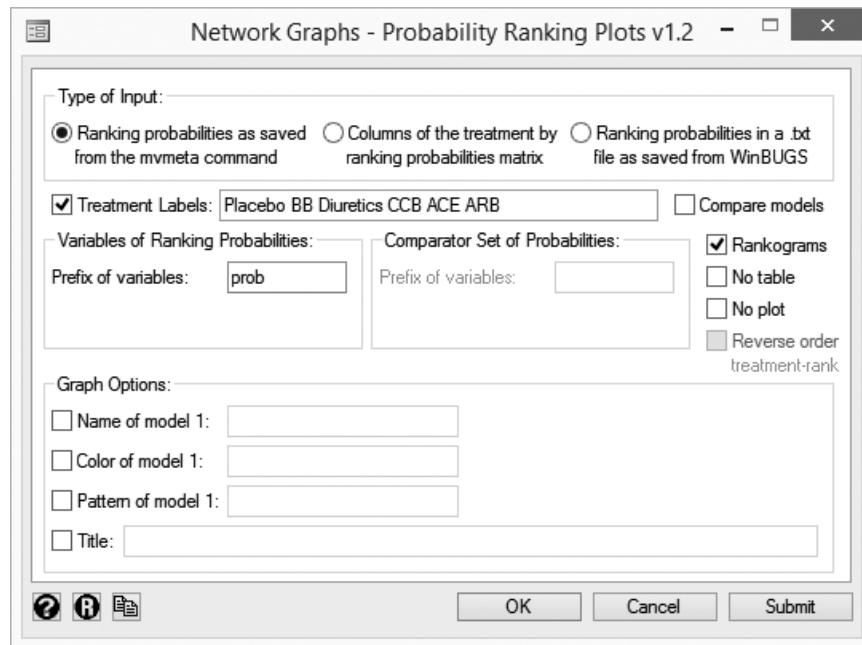

Figure 15. Example of using the `sucra` dialog box for the antihypertensives network

We again run `network rank` to estimate the predictive ranking probabilities (adding the option `predict` and choosing names `pred_prob` to denote the predictive probabilities stored in the dataset). Using the `sucra` command, we compare the cumulative ranking plots based on the estimated and predictive ranking probabilities.

The resulting curves (shown in figure 16) imply that the incorporation of the heterogeneity in the predictive ranking probabilities does not materially affect the results for the relative ranking of treatments.

```
. network rank min, zero all reps(10000) gen(pred_prob) predict
  (output omitted)
. sucra prob*, labels(Placebo BB Diuretics CCB ACE ARB) compare(pred_prob*)
> names("Estimated probabilities" "Predictive probabilities")
Treatment Relative Ranking of Estimated probabilities
```

Treatment	SUCRA	PrBest	MeanRank
Placebo	54.9	0.2	3.3
BB	16.2	0.0	5.2
Diuretics	4.0	0.0	5.8
CCB	46.4	0.1	3.7
ACE	83.8	24.0	1.8
ARB	94.8	75.8	1.3

```
Treatment Relative Ranking of Predictive probabilities
```

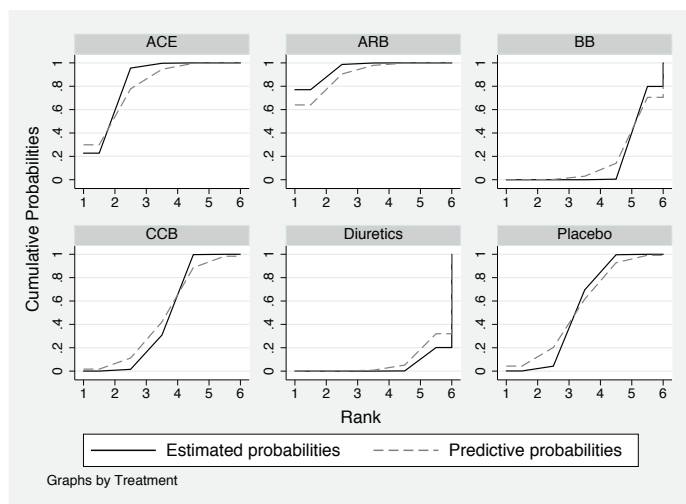| Treatment	SUCRA	PrBest	MeanRank
Placebo	56.0	3.7	3.2
BB	17.2	0.1	5.1
Diuretics	7.7	0.0	5.6
CCB	48.3	1.7	3.6
ACE	80.1	29.8	2.0
ARB	90.6	64.7	1.5

Figure 16. Cumulative probability curves for the antihypertensives network showing the estimated and predictive probabilities for each treatment being up to a specific rank

Figure 17 shows how the above command can be executed via the `sucra` dialog box, accessed by typing `db sucra` in the Command window.
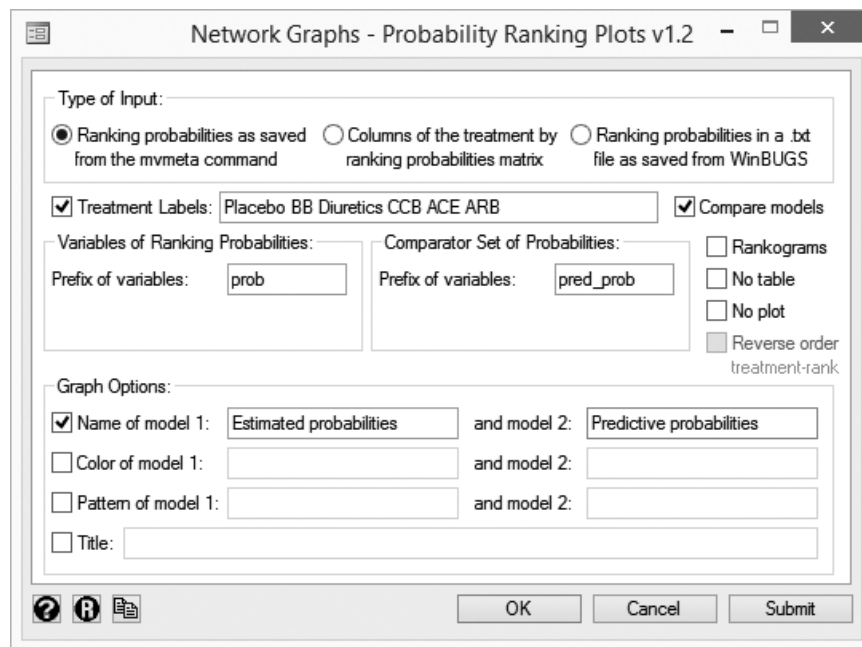

Figure 17. Another example of using the `sucra` dialog box for the antihypertensives network

3.9 The mdsrank command

Multidimensional scaling approach for ranking

A different approach to estimate the relative ranking is to use multidimensional scaling (MDS) techniques. To apply this method, the network estimates for all possible comparisons are treated as proximity data aiming to reveal their latent structure. In this way, the absolute value $|\widehat{\mu}_{XY}|$ defines the dissimilarity between the two treatments (X, Y) with $|\widehat{\mu}_{XX}| = 0$. Weighting the absolute effects sizes by their inverse standard errors or inverse variances ensures that the assumption of a common distribution between the elements of the matrix is plausible. Assuming that the rank of the treatments is the only dimension underlying the outcome, the purpose of the technique would be to reduce the $T \times T$ matrix into a $T \times 1$ vector. This vector involves the set of distances being as close as possible to the observed dissimilarities (that is, relative effects) and would represent the relative ranking of the treatments.

Description of mdsrank

The `mdsrank` command creates the squared matrix containing the pairwise relative effect sizes and plots the resulting values of the unique dimension for each treatment.

Syntax for mdsrank

mdsrank *varlist* [*if*] [*in*] [, <u>lab</u>els(*string*) <u>nopl</u>ot <u>scatt</u>eroptions(*string*)
   best(min|max)]

In *varlist*, the summary effect sizes from network meta-analysis and their uncertainties (variables *ES* and *seES*; see section 3.1), as well as the treatment comparisons they refer to (variables *t1* and *t2*; see section 3.1), should be specified.

Options for mdsrank

labels(*string*), noplot, scatteroptions(*string*); see section 3.1.

best(min|max) specifies whether larger-dimension scores correspond to a more favorable outcome with the treatment. The default is best(min).

Example for mdsrank using the antihypertensives network

The data for this network are described in section 3.6.

Using the `intervalplot` command (after `network meta`; see section 3.5), we obtain the estimated relative effects via the option `keep`.

```
. intervalplot, labels(Placebo BB Diuretics CCB ACE ARB) noplot notab keep
 The intervalplot command assumes that the saved results from mvmeta or network
> meta commands have been derived from the current dataset
```

These results have been stored at the end of the dataset. For example, the first five comparisons in the data give the results for all active treatments versus placebo.

```
. split _Comparison, par(" vs ") gen(t)
variables created as string:
t1  t2

. list _Comparison _Effect_Size _Standard_Error t1 t2 in 1/5, noobs clean
             _Comparison    _Effect~e    _Stand~r          t1          t2
             BB vs Placebo    .2121609    .0810291          BB     Placebo
      Diuretics vs Placebo    .2806841    .0842885   Diuretics     Placebo
            CCB vs Placebo    .0404351    .0786069         CCB     Placebo
            ACE vs Placebo   -.1233237    .0679924         ACE     Placebo
            ARB vs Placebo    -.191084    .0874827         ARB     Placebo
```

Then, we use the `mdsrank` command to estimate the relative ranking of treatments (shown in figure 18) using the MDS method:

```
. mdsrank _Effect_Size _Standard_Error t1 t2 if _Effect_Size!=., best(max)

 Warning: The existing dataset is stored as a temporary file
 Warning: To save any changes applied at this temporary file in a specific
> directory you need to use the ´Save as´ menu
```

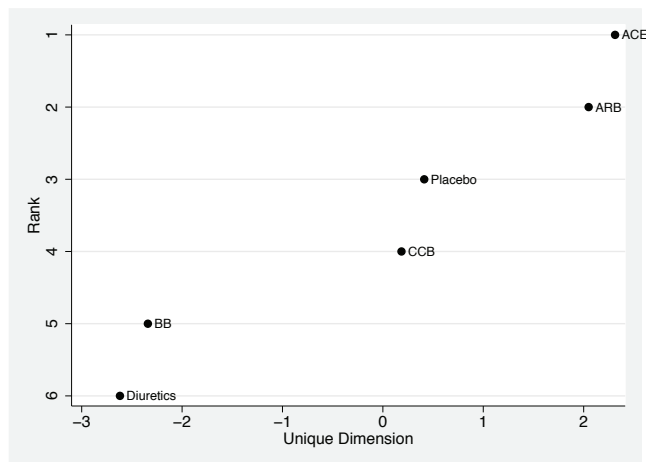| Treatment	Dim1	Rank
ACE	2.31	1
ARB	2.05	2
Placebo	0.41	3
CCB	0.19	4
BB	-2.34	5
Diuretics	-2.62	6

Figure 18. Relative ranking of treatments for the diabetes network based on the MDS approach. Larger values of the dimension correspond to higher ranks.

The relative ranking obtained from this approach is similar to that derived from the ranking probabilities, but they disagree in the order of the two best treatments.

Figure 19 shows how the above command can be executed via the `mdsrank` dialog box, accessed by typing `db mdsrank` in the Command window.
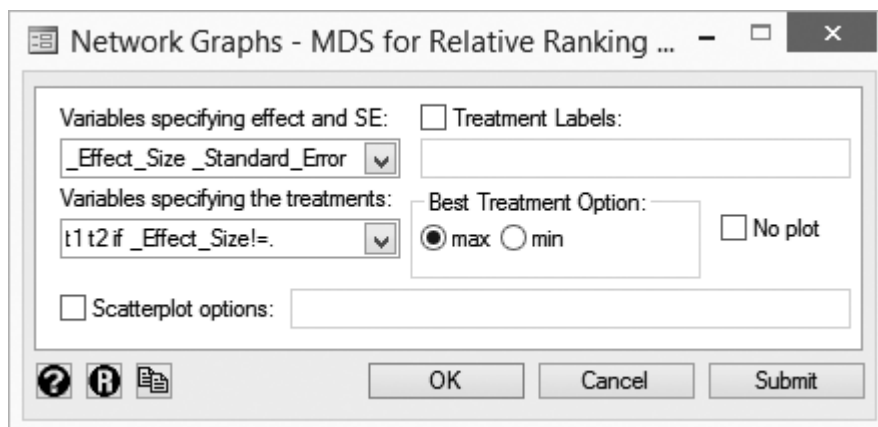

Figure 19. Example of using the `mdsrank` dialog box for the antihypertensives network

3.10   The clusterank command

Cluster analysis for more than one outcome

When performing a network meta-analysis, the competing treatments can be ranked according to their performance on one or more outcomes (for example, effectiveness and safety). However, the relative ranking for each outcome might be different, and this makes the choice of the "best" treatment challenging. A possible way to make inferences based on results for two outcomes is by using a two-dimensional plot and constructing groups of treatments with similar performance on both outcomes (Chaimani et al. 2013a). To form meaningful groups of treatments, hierarchical clustering methods have been used (Kaufman and Rousseeuw 2005).

Description of clusterank

The `clusterank` command performs hierarchical cluster analysis to group the competing treatments into meaningful groups. It requires the values of a ranking measure (for example, SUCRA percentages or MDS dimension) for two outcomes.

Optimal cluster analysis

The command chooses the appropriate metric (Euclidean, squared Euclidean, absolute-value distance, etc.) and linkage method (single, average, weighted, complete, ward, centroid, median) based on the cophenetic correlation coefficient, which measures how

faithfully the output dendrogram represents the dissimilarities between observations (Handl, Knowles, and Kell 2005). The optimal level of dendrogram and the optimal number of clusters are chosen using an internal cluster validation measure, called clustering gain (Jung et al. 2003). This measure has been designed to have a maximum value when intracluster similarity is maximized and intercluster similarity is minimized.

Syntax for clusterank

clusterank *varlist* [, <u>scatt</u>eroptions(*string*) best(min|max)

    <u>meth</u>od(*linkage_method distance_metric*) <u>clus</u>ters(*integer*) <u>dend</u>rogram]

In *varlist*, the variables *outcome1* and *outcome2* (a ranking measure—for example, mean rank—for each outcome; see section 3.1) and *t* (optional, the names of the treatments; see section 3.1) should be specified.

Options for clusterank

scatteroptions(*string*); see section 3.1.

best(min|max) specifies whether larger or smaller values of the ranking measure correspond to better outcome with the treatment (applies to both outcomes). The default is best(max).

method(*linkage_method distance_metric*) specifies which linkage method and distance metric to use. By default, the method of hierarchical clustering is decided according to the cophenetic correlation coefficient (Handl, Knowles, and Kell 2005). For other options, see the help file for cluster (type help cluster in Stata).

clusters(*integer*) specifies the number of clusters used to group the treatments. By default, the optimal number of clusters is decided according to the "clustering gain" (Jung et al. 2003).

dendrogram specifies that the dendrogram of the hierarchical analysis be displayed instead of the clustered ranking plot.

Example for clusterank using the antidiabetics network

The data for this network are described in section 3.2.

Both datasets for efficacy and tolerability are in the appropriate format for network (where y is the mean change score in each study arm, sd is the respective standard deviation, and n is the number of total participants). After running network meta-analysis using the network meta command, we obtain the SUCRA (via the sucra command; see section 3.8) percentages for efficacy (*outcome1*) and tolerability (*outcome2*) to present the relative ranking of the seven interventions separately for each outcome.

Relative ranking for efficacy:

```
. use antidiabetics_efficacy_long.dta, clear
. network setup y sd n, studyvar(id) trtvar(t) ref(1) numcodes
  (output omitted)
. network meta c
  (output omitted)
. network rank min, zero all reps(10000) gen(eff_prob)
  (output omitted)
. sucra eff_prob*, labels(Placebo Sulfonylurea "DPP-4 inhibitor"
> Thiazolidinedione "GLP-1 analog" AGI Glinine) noplot
Treatment Relative Ranking of Model 1
```

Treatment	SUCRA	PrBest	MeanRank
Placebo	1.0	0.0	6.9
Sulfonylurea	56.3	2.8	3.6
DPP-4 inhibitor	46.4	1.3	4.2
Thiazolidinedione	64.3	7.7	3.1
GLP-1 analog	66.4	21.1	3.0
AGI	41.9	10.2	4.5
Glinine	73.8	57.0	2.6

Relative ranking for tolerability:

```
. use antidiabetics_tolerability.dta, clear
. network setup y sd n, studyvar(id) trtvar(t) ref(1) numcodes
  (output omitted)
. network meta c
  (output omitted)
. network rank min, zero all reps(10000) gen(tol_prob)
  (output omitted)
. sucra tol_prob*, labels(Placebo Sulfonylurea "DPP-4 inhibitor"
> Thiazolidinedione "GLP-1 analog" AGI Glinine) noplot
Treatment Relative Ranking of Model 1
```

Treatment	SUCRA	PrBest	MeanRank
Placebo	59.9	0.0	3.4
Sulfonylurea	24.0	0.0	5.6
DPP-4 inhibitor	57.4	0.0	3.6
Thiazolidinedione	2.8	0.0	6.8
GLP-1 analog	91.8	51.0	1.5
AGI	91.8	49.0	1.6
Glinine	23.3	0.0	5.6

We store the SUCRA percentages in a dataset and then run the `clusterank` command, with results shown in figure 20.

```
. use antidiabeticssucras.dta, clear
. list, clean noobs
                     t    outcome1    outcome2
               Placebo          1        59.9
           Sulfonylurea       56.3         24
        DPP-4 inhibitor       46.4       57.4
      Thiazolidinedione       64.3        2.8
           GLP-1 analog       66.4       91.8
                   AGI       41.9       90.7
               Glinine       73.8       23.3
. clusterank outcome1 outcome2 t

Best linkage method: averagelinkage
Best distance metric: Canberra

Cophenetic Correlation Coefficient c = 0.91

** Maximum value of clustering gain = 2722.18
** Optimal number of clusters =  4
```
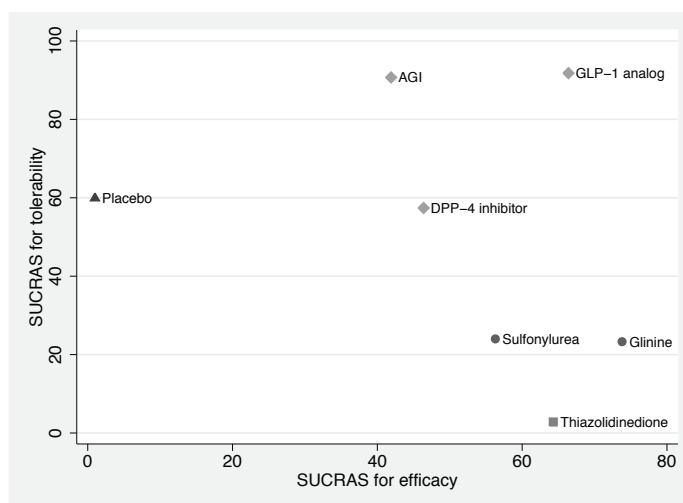

Figure 20. Clustered ranking plot for the antidiabetics network presenting jointly the relative ranking of treatments (based on the SUCRA percentages) for efficacy and tolerability. Treatments lying in the upper right corner are considered to perform well for both outcomes. Different plotting symbols represent different clusters of treatments.

Figure 20 shows three treatments performing similarly well on both outcomes that form the best group of treatments.

Figure 21 shows how the above command can be executed via the `clusterank` dialog box, accessed by typing `db clusterank` in the Command window.
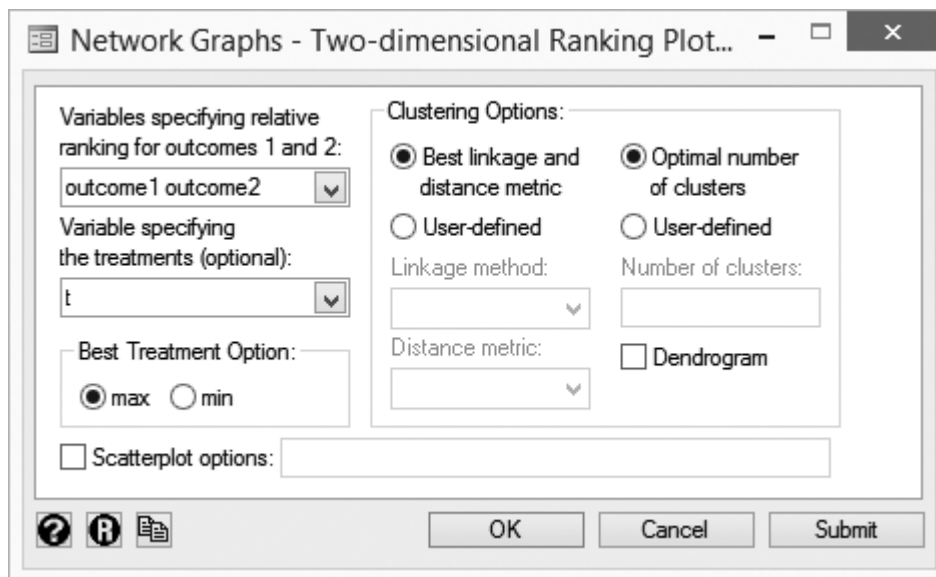

Figure 21. Example of using of the dialog box for `clusterank` for the antidiabetics network

4 Discussion

In this article, we introduced a package of nine commands that can be used to understand and present graphically the different steps in a network meta-analysis. Using these commands in conjunction with `mvmeta` (White 2011) or `network` (White 2015) can simplify the technicalities of the network meta-analysis procedure for dichotomous or continuous data; the commands might be less convenient for other types of data (for example, time-to-event data).

The suggested graphs aim to summarize in a comprehensive way the most important findings from this complex statistical tool and facilitate their interpretation. The commands' usefulness might be limited under specific conditions; for example, inconsistency plots cannot be applied in networks without closed loops, and comparison-adjusted funnel plots are not informative for networks with only one or two studies within each pairwise comparison.

5 Acknowledgments

6 References

Anscombe, F. J. 1973. Graphs in statistical analysis. *American Statistician* 27: 17–21.

Anzures-Cabrera, J., and J. P. T. Higgins. 2010. Graphical displays for meta-analysis: An overview with suggestions for practice. *Research Synthesis Methods* 1: 66–80.

Bax, L., N. Ikeda, N. Fukui, Y. Yaju, H. Tsuruta, and K. G. Moons. 2009. More than numbers: the power of graphs in meta-analysis. *American Journal of Epidemiology* 169: 249–255.

Bucher, H. C., G. H. Guyatt, L. E. Griffith, and S. D. Walter. 1997. The results of direct and indirect treatment comparisons in meta-analysis of randomized controlled trials. *Journal of Clinical Epidemiology* 50: 683–691.

Caldwell, D. M., A. E. Ades, and J. P. Higgins. 2005. Simultaneous comparison of multiple treatments: Combining direct and indirect evidence. *British Medical Journal* 331: 897–900.

Chaimani, A., J. P. T. Higgins, D. Mavridis, P. Spyridonos, and G. Salanti. 2013a. Graphical tools for network meta-analysis in Stata. *PLOS ONE* 8: e76654.

Chaimani, A., D. Mavridis, and G. Salanti. 2014. A hands-on practical tutorial on performing meta-analysis with Stata. *Evidence-Based Mental Health* 17: 111–116.

Chaimani, A., and G. Salanti. 2012. Using network meta-analysis to evaluate the existence of small-study effects in a network of interventions. *Research Synthesis Methods* 3: 161–176.

Chaimani, A., H. S. Vasiliadis, N. Pandis, C. H. Schmid, N. J. Welton, and G. Salanti. 2013b. Effects of study precision and risk of bias in networks of interventions: A network meta-epidemiological study. *International Journal of Epidemiology* 42: 1120–1131.

Cipriani, A., T. A. Furukawa, G. Salanti, J. R. Geddes, J. P. Higgins, R. Churchill, N. Watanabe, A. Nakagawa, I. M. Omori, H. McGuire, M. Tansella, and C. Barbui. 2009. Comparative efficacy and acceptability of 12 new-generation antidepressants: A multiple-treatments meta-analysis. *Lancet* 373: 746–758.

Cooper, H., L. V. Hedges, and J. C. Valentine. 2009. *The Handbook of Research Synthesis and Meta-Analysis.* 2nd ed. New York: Russell Sage Foundation.

DerSimonian, R., and N. Laird. 1986. Meta-analysis in clinical trials. *Controlled Clinical Trials* 7: 177–188.

Dias, S., N. J. Welton, D. M. Caldwell, and A. E. Ades. 2010. Checking consistency in mixed treatment comparison meta-analysis. *Statistics in Medicine* 29: 932–944.

Elliott, W. J., and P. M. Meyer. 2007. Incident diabetes in clinical trials of antihypertensive drugs: A network meta-analysis. *Lancet* 369: 201–207.

Handl, J., J. Knowles, and D. B. Kell. 2005. Computational cluster validation in post-genomic data analysis. *Bioinformatics* 21: 3201–3212.

Higgins, J. P. T., D. Jackson, J. K. Barrett, G. Lu, A. E. Ades, and I. R. White. 2012. Consistency and inconsistency in network meta-analysis: Concepts and models for multi-arm studies. *Research Synthesis Methods* 3: 98–110.

Higgins, J. P. T., S. G. Thompson, and D. J. Spiegelhalter. 2009. A re-evaluation of random-effects meta-analysis. *Journal of the Royal Statistical Society, Series A* 172: 137–159.

Jansen, J. P., and H. Naci. 2013. Is network meta-analysis as valid as standard pairwise meta-analysis? It all depends on the distribution of effect modifiers. *BMC Medicine* 11: 159.

Jung, Y., H. Park, D.-Z. Du, and B. L. Drake. 2003. A decision criterion for the optimal number of clusters in hierarchical clustering. *Journal of Global Optimization* 25: 91–111.

Kaufman, L., and P. J. Rousseeuw. 2005. *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ: Wiley.

Krahn, U., H. Binder, and J. König. 2013. A graphical tool for locating inconsistency in network meta-analyses. *BMC Medical Research Methodology* 13: 35.

Lee, A. W. 2014. Review of mixed treatment comparisons in published systematic reviews shows marked increase since 2009. *Journal of Clinical Epidemiology* 67: 138–143.

Lu, G., and A. E. Ades. 2006. Assessing evidence inconsistency in mixed treatment comparisons. *Journal of the American Statistical Association* 101: 447–459.

Lu, G., N. J. Welton, J. P. T. Higgins, I. R. White, and A. E. Ades. 2011. Linear inference for mixed treatment comparison meta-analysis: A two-stage approach. *Research Synthesis Methods* 2: 43–60.

Lunn, D. J., A. Thomas, N. Best, and D. Spiegelhalter. 2000. WinBUGS—A Bayesian modelling framework: Concepts, structure and extensibility. *Statistics and Computing* 10: 325–337.

Miladinovic, B., A. Chaimani, I. Hozo, and B. Djulbegovic. 2014. Indirect treatment comparison. *Stata Journal* 14: 76–86.

Moreno, S. G., A. J. Sutton, A. E. Ades, N. J. Cooper, and K. R. Abrams. 2011. Adjusting for publication biases across similar interventions performed well when compared with gold standard data. *Journal of Clinical Epidemiology* 64: 1230–1241.

Morris, C. N. 1983. Parametric empirical Bayes inference: Theory and applications. *Journal of the American Statistical Association* 78: 47–55.

Nikolakopoulou, A., A. Chaimani, A. A. Veroniki, H. S. Vasiliadis, C. H. Schmid, and G. Salanti. 2014. Characteristics of networks of interventions: A description of a database of 186 published networks. *PLOS ONE* 9: e86754.

Phung, O. J., J. M. Scholle, M. Talwar, and C. I. Coleman. 2010. Effect of noninsulin antidiabetic drugs added to metformin therapy on glycemic control, weight gain, and hypoglycemia in type 2 diabetes. *Journal of the American Medical Association* 303: 1410–1418.

Riley, R. D., J. P. T. Higgins, and J. J. Deeks. 2011. Interpretation of random effects meta-analyses. *British Medical Journal* 342: d549.

Salanti, G. 2012. Indirect and mixed-treatment comparison, network, or multiple-treatments meta-analysis: Many names, many benefits, many concerns for the next generation evidence synthesis tool. *Research Synthesis Methods* 3: 80–97.

Salanti, G., A. E. Ades, and J. P. Ioannidis. 2011. Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: An overview and tutorial. *Journal of Clinical Epidemiology* 64: 163–171.

Salanti, G., S. Dias, N. J. Welton, A. E. Ades, V. Golfinopoulos, M. Kyrgiou, D. Mauri, and J. P. Ioannidis. 2010. Evaluating novel agent effects in multiple-treatments meta-regression. *Statistics in Medicine* 29: 2369–2383.

Salanti, G., C. D. Giovane, A. Chaimani, D. M. Caldwell, and J. P. T. Higgins. 2014. Evaluating the quality of evidence from a network meta-analysis. *PLOS ONE* 9: e99682.

Song, F., A. Clark, M. O. Bachmann, and J. Maas. 2012. Simulation evaluation of statistical properties of methods for indirect and mixed treatment comparisons. *BMC Medical Research Methodology* 12: 138.

Thijs, V., R. Lemmens, and S. Fieuws. 2008. Network meta-analysis: Simultaneous meta-analysis of common antiplatelet regimens after transient ischaemic attack or stroke. *European Heart Journal* 29: 1086–1092.

Veroniki, A. A., D. Mavridis, J. P. T. Higgins, and G. Salanti. 2014. Characteristics of a loop of evidence that affect detection and estimation of inconsistency: A simulation study. *BMC Medical Research Methodology* 14: 106.

Veroniki, A. A., H. S. Vasiliadis, J. P. Higgins, and G. Salanti. 2013. Evaluation of inconsistency in networks of interventions. *International Journal of Epidemiology* 42: 332–345.

Viechtbauer, W. 2005. Bias and efficiency of meta-analytic variance estimators in the random-effects model. *Journal of Educational and Behavioral Statistics* 30: 261–293.

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

———. 2015. Network meta-analysis. *Stata Journal* 15: 951–984.

White, I. R., J. K. Barrett, D. Jackson, and J. P. T. Higgins. 2012. Consistency and inconsistency in network meta-analysis: Model estimation using multivariate meta-regression. *Research Synthesis Methods* 3: 111–125.

About the authors

Anna Chaimani is a postdoctoral research associate at the University of Ioannina School of Medicine in Ioannina, Greece. Her research interests focus on methodology for pairwise and network meta-analysis.

Georgia Salanti is an associate professor at the University of Ioannina School of Medicine in Ioannina, Greece. Her research interests focus on statistical models for evidence synthesis.