Appendices

A brief guide to threshold analysis in R

Threshold analysis can be performed in R using the *nmathresh* package, which is installed from CRAN and loaded using the commands

```
install.packages("nmathresh")
library(nmathresh)
```

The function `nma_thresh` performs the threshold analysis. The basic syntax (for a fixed effect NMA model) is:

```
nma_thresh(mean.dk = d, lhood = V.data, post = V.est, X = X)
```

where:

`d` is a vector containing the NMA estimates for each treatment compared with treatment 1

`V.data` is the covariance matrix for the observed data (either study or contrast level). For a study level analysis, then this will be a (block) diagonal matrix with the squared standard errors on the diagonals (and for relative effect data, covariances between estimates from trials with more than two arms). For a contrast level analysis, the covariance matrix needs to be reconstructed using the `recon_vcov` function:

```
recon_vcov(post = V.est, X = X)
```

`V.est` is the covariance matrix of the NMA estimates for each treatment compared with treatment 1. This may be the posterior covariance matrix of the estimates from a Bayesian analysis, or the covariance matrix of the estimates from a frequentist analysis.

`X` is the design matrix showing the comparisons made in the data, either at study or contrast level. For example, suppose we had 2 studies comparing treatments 3 vs. 1, one study 4 vs. 2, and one study 3 vs. 2, each reporting study-level relative effects. The columns of the design matrix represent the basic treatment parameters d_{12}, d_{13}, and d_{14}. For a study-level threshold analysis, the design matrix would be:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}$$

For a contrast-level threshold analysis, the design matrix would be

$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}$$

Further options provided to `nma_thresh` specify the type of decision rule, or are required for threshold analysis of random effects NMA. For further details see the help page `?nma_thresh`.

`opt.max` specifies whether higher or lower relative effects are better (the default is `TRUE`, higher effects are better).

`mcid` specifies a minimally clinically important difference, to be used with the option `mcid.type` as follows.

When `mcid.type = "decision"` this specifies a decision rule to recommend the set of treatments better than treatment 1 by `mcid` or more, and within `mcid` of the most effective (as in the headaches example). When `mcid.type = "change"` threshold analysis is carried out for the greatest efficacy decision rule, but where the decision only changes if a treatment exceeds `mcid` compared to the existing optimal treatment. The default is `"decision"`, but `mcid.type` is only used when `mcid` is not 0.

trt.rnk specifies the treatment rank to derive thresholds for. For example, to derive thresholds for when the second-placed treatment would change, specify trt.rank = 2. To derive thresholds for worst-placed treatment (perhaps to investigate the robustness of a "do not do" decision), specify trt.rank equal to the number of treatments. This option cannot be used when mcid.type = "decision" and mcid is greater than 0. The default is to derive thresholds for the best treatment (trt.rnk = 1).

To specify threshold analysis for a random effects NMA, the following options are required:

nmatype = "random" for random effects NMA (the default is "fixed")

delta.design is the design matrix for the random effects terms, specifying which study data points have random effects. By default this is the identity matrix, so every data point has a random effect (suitable for relative effect data), but for arm-based data the reference arm in each study should be given a 0 on the corresponding diagonal entry.

mu.design is the design matrix for any additional parameters, for example study baselines for arm-based data, or covariates in a meta-regression model. The default value is NULL (no extra parameters).

For random effects NMA, V.est should be the covariance matrix of the treatment parameters, random effects terms, and any additional parameters.

The outputs of nma_thresh are the thresholds and new treatment recommendations (as given in Supplementary Tables 1, 2, and 3). We summarise these graphically alongside the data by plotting the invariant intervals formed by adding positive and negative thresholds to the point estimate (as in Figures 2, 5, and 6). The function thresh_forest performs these calculations and produces the figures. The basic function call looks like

```
thresh_forest(thresh, y, CI.lo, CI.hi, label, data)
```

where thresh is a threshold object produced by nma_thresh. y, CI.lo, CI.hi, and label are either vectors or columns in the data frame data representing the study or contrast estimates for which thresholds were derived, along with confidence interval limits and labels to display. For details on the further options available to customise the output, see the help page ?thresh_forest.

Further details of the package and walkthroughs of example analyses can be found using the help files and the vignette:

```
vignette("Examples", package = "nmathresh")
```

The supplementary material contains detailed code for each example presented in this paper.