Package 'rnmamod'

March 24, 2024

Type Package

Title Bayesian Network Meta-Analysis with Missing Participants

Version 0.4.0

Date 2024-03-24

Maintainer Loukia Spineli <Spineli.Loukia@mh-hannover.de>

Description A comprehensive suite of functions to perform and visualise pairwise and network meta-analysis with aggregate binary or continuous missing participant outcome data. The package covers core Bayesian one-stage models implemented in a systematic review with multiple interventions, including fixed-effect and random-effects network meta-analysis, meta-regression, evaluation of the consistency assumption via the node-splitting approach and the unrelated mean effects model, and sensitivity analysis. Missing participant outcome data are addressed in all models of the package. The robustness to primary analysis results can also be investigated using a novel intuitive index. Methods to evaluate the transitivity assumption quantitatively are provided. The package also offers a rich, user-friendly visualisation toolkit that aids in appraising and interpreting the results thoroughly and preparing the manuscript for journal submission. The visualisation tools comprise the network plot, forest plots, panel of diagnostic plots, heatmaps on the extent of missing participant outcome data in the network, league heatmaps on estimation and prediction, rankograms, Bland-Altman plot, leverage plot, deviance scatterplot, heatmap of robustness, barplot of Kullback-Leibler divergence, heatmap of comparison dissimilarities and dendrogram of comparison clustering. The package also allows the user to export the results to an Excel file at the working directory.

License GPL (>= 3)

URL https://CRAN.R-project.org/package=rnmamod,
https://github.com/LoukiaSpin/rnmamod

BugReports https://github.com/LoukiaSpin/rnmamod/issues

Depends R (>= 4.0.0)

Imports cluster, dendextend, fdrtool, gemtc, ggfittext, ggplot2, ggpubr, ggrepel, heatmaply, igraph, knitr, MASS, mcmcplots, R2jags, reshape2, scales, stringr, writexl

Suggests rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.1

NeedsCompilation no

Author Loukia Spineli [aut, cre],
Chrysostomos Kalyvas [ctb],
Katerina Papadimitropoulou [ctb]

Repository CRAN

Date/Publication 2024-03-24 21:40:02 UTC

R topics documented:

`rnmamod-package` *rnmamod: Bayesian Network Meta-analysis with Missing Participants*

Description

An R package for performing Bayesian network meta-analysis while handling missing participant outcome data properly, assessing the robustness of the primary analysis results, and exploring the transitivity assumption.

Details

R-package **rnmamod** is built upon the WinBUGS program code found in the series of tutorial papers on evidence synthesis methods for decision making (Dias et al., 2013a; Dias et al., 2013b; Dias et al., 2013c) and Dias et al. (2010) that introduces the node-splitting approach. All models comprise Bayesian hierarchical models for one-stage network meta-analysis and they are implemented in JAGS through the R-package **R2jags**.

rnmamod comprises a suite of core models implemented in a systematic review with multiple interventions:

- fixed-effect and random-effects network meta-analysis (`run_model`) based on Dias et al. (2013c);
- fixed-effect and random-effects network meta-regression (`run_metareg`) based on Cooper et al. (2009), and Dias et al. (2013b);
- fixed-effect and random-effects separate pairwise meta-analyses for comparisons with at least two trials (`run_series_meta`);
- local evaluation of the consistency assumption using the fixed-effect or random-effects node-splitting approach (`run_nodesplit`) based on Dias et al. (2010), and van Valkenhoef et al. (2016);
- global evaluation of the consistency assumption using the fixed-effect or random-effects unrelated mean effects model (`run_ume`) based on Dias et al. (2013a) and Spineli (2021);
- comprehensive sensitivity analysis for the impact of aggregate binary and continuous missing participant outcome data (`run_sensitivity`) based on Spineli et al. (2021a);
- hierarchical baseline model for the selected reference intervention (`baseline_model`) based in Dias et al. (2013d).

rnmamod also includes a rich suite of visualisation tools to aid in the interpretation of the results and preparation of the manuscript for submission:

- network plot and description of the evidence base (`netplot` and `describe_network`, respectively) following the PRISMA statement for systematic reviews with network meta-analysis (Hutton et al., 2015);
- illustration of the R-hat (Gelman and Rubin, 1992) and MCMC error for all monitored nodes and creation of an HTML file with a panel of diagnostic plots for each monitored parameter (`mcmc_diagnostics`);
- heatmap on the proportion of missing participants across the network (`heatmap_missing_network`) and across the intervention arms of each trial in the dataset (`heatmap_missing_dataset`);
- league heatmap with the estimated and predicted summary effects of all possible pairwise comparisons in the network and integrated SUCRA (Salanti et al., 2011) or P-scores (Ruecker and Schwarzer, 2015) (`league_heatmap` and `league_heatmap_pred`, respectively) after performing network meta-analysis or network meta-regression;
- league table for relative and absolute effects for all pairwise comparisons and interventions when conducting network meta-analysis anew via the package (`league_table_absolute`) or using the results of a published systematic review with network meta-analysis (`league_table_absolute_user`);
- forest plot with the trial-specific and summary absolute risks when employing the hierarchical baseline model for the selected reference intervention (`baseline_model`) as described in Dias et al. (2013d);

- rankograms with integrated SUCRA values for each intervention in the network (`rankosucra_plot`) after performing network meta-analysis (Salanti et al., 2011);
- forest plot with the estimated and predicted summary effects of all comparisons with a selected intervention (`forestplot`) as obtained from the network meta-analysis model, and a forest plot with the corresponding SUCRA values (Salanti et al., 2011);
- tabulation of the estimated regression coefficient(s), the estimated and predicted summary effects, measures of model fit and estimated between-trial standard deviation before and after adjusting for a trial-specific covariate (`metareg_plot`), and visualisation of the summary effects and SUCRA values from both models (`forestplot_metareg`, and `scatterplot_sucra`, respectively–both found in `metareg_plot`);
- tabulation of the estimated direct and indirect effects of the split nodes and corresponding inconsistency factors, measures of model fit and estimated between-trial standard deviation after each split node, and visualisation of these results (`nodesplit_plot`);
- tabulation of the estimated summary effects of all comparisons observed in the network, measures of model fit and estimated between-trial standard deviation under the unrelated mean effects model and network meta-analysis, as well as visualisation of the summary effects from both models (`intervalplot_panel_ume`) and the goodness of fit of each model using a series of complementary plots (`scatterplots_dev` (Dias et al., 2013a), `bland_altman_plot` (Bland and Altman, 1999), and `leverage_plot` (Dias et al., 2010)–all found in `ume_plot`);
- tabulation of the estimated summary effects and corresponding between-trial standard deviation for comparisons with at least two trials under pairwise and network meta-analysis, as well as visualisation of these results (`series_meta_plot`);
- calculation and visualisation of the robustness index for all possible comparisons in the network (`robustness_index`, `robustness_index_user` and `heatmap_robustness`) (Spineli et al., 2021a);
- enhanced balloon plot with the summary effects and between-trial standard deviation for a selected pairwise comparison under several scenarios about the missingness parameter (`balloon_plot`) (Spineli et al., 2021a);
- barplot with the Kullback-Leibler divergence measure from each informative scenario to the missing-at-random assumption about the missingness parameter for a selected pairwise comparison (`kld_barplot`) (Spineli et al., 2021a).

rnmamod also assists the researcher in assessing the transitivity assumption quantitatively based on trial dissimilarities for various trial-level aggregate participant and methodological characteristics calculated using the Gower's dissimilarity coefficient (`gower_distance` and `comp_clustering`) (Gower, 1971). Results on the clustered comparisons based on hierarchical agglomerative clustering are illustrated using a dendrogram with integrated heatmap (`dendro_heatmap`). The distribution of the characteristics is presented using violin plots with integrated box plots and dots, and stacked bar plots across the observed treatment comparisons (`distr_characteristics`). Missing data in the characteristics across the trials and observed comparisons are visualised using bar plots and tile plot (`miss_characteristics`).

Missing participant outcome data are addressed in all models of the package after extending the code to incorporate the pattern-mixture model (Spineli et al., 2021b; Spineli, 2019).

Type `citation("rnmamod")` on how to cite **rnmamod**.

To report possible bugs and errors, send an email to Loukia Spineli (`<Spineli.Loukia@mh-hannove.de>`).

The development version of **rnmamod** is available on GitHub under the GPL-3.0 License.

Author(s)

Loukia M. Spineli

References

Bland JM, Altman DG. Measuring agreement in method comparison studies. *Stat Methods Med Res* 1999;**8**(2):135–60. doi: 10.1177/096228029900800204

Cooper NJ, Sutton AJ, Morris D, Ades AE, Welton NJ. Addressing between-study heterogeneity and inconsistency in mixed treatment comparisons: Application to stroke prevention treatments in individuals with non-rheumatic atrial fibrillation. *Stat Med* 2009;**28**(14):1861–81. doi: 10.1002/sim.3594

Dias S, Welton NJ, Sutton AJ, Caldwell DM, Lu G, Ades AE. Evidence synthesis for decision making 4: inconsistency in networks of evidence based on randomized controlled trials. *Med Decis Making* 2013a;**33**(5):641–56. doi: 10.1177/0272989X12455847

Dias S, Sutton AJ, Welton NJ, Ades AE. Evidence synthesis for decision making 3: heterogeneity–subgroups, meta-regression, bias, and bias-adjustment. *Med Decis Making* 2013b;**33**(5):618–40. doi: 10.1177/0272989X13485157

Dias S, Sutton AJ, Ades AE, Welton NJ. Evidence synthesis for decision making 2: a generalized linear modeling framework for pairwise and network meta-analysis of randomized controlled trials. *Med Decis Making* 2013c;**33**(5):607–17. doi: 10.1177/0272989X12458724

Dias S, Welton NJ, Sutton AJ, Ades AE. Evidence synthesis for decision making 5: the baseline natural history model. *Med Decis Making* 2013d;**33**(5):657–70. doi: 10.1177/0272989X13485155

Dias S, Welton NJ, Caldwell DM, Ades AE. Checking consistency in mixed treatment comparison meta-analysis. *Stat Med* 2010;**29**(7-8):932–44. doi: 10.1002/sim.3767

Gelman, A, Rubin, DB. Inference from iterative simulation using multiple sequences. *Stat Sci* 1992;**7**(4):457–72. doi: 10.1214/ss/1177011136

Gower JC. A General Coefficient of Similarity and Some of Its Properties. *Biometrics* 1971;**27**(4):857–71. http://dx.doi.org/10.2307/2528823

Hutton B, Salanti G, Caldwell DM, Chaimani A, Schmid CH, Cameron C, et al. The PRISMA extension statement for reporting of systematic reviews incorporating network meta-analyses of health care interventions: checklist and explanations. *Ann Intern Med* 2015;**162**(11):777–84. doi: 10.7326/M14-2385

Ruecker G, Schwarzer G. Ranking treatments in frequentist network meta-analysis works without resampling methods. *BMC Med Res Methodol* 2015;**15**:58. doi: 10.1186/s12874-015-0060-8

Salanti G, Ades AE, Ioannidis JP. Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: an overview and tutorial. *J Clin Epidemiol* 2011;**64**(2):163–71. doi: 10.1016/j.jclinepi.2010.03.016

Spineli LM. A revised framework to evaluate the consistency assumption globally in a network of interventions. *Med Decis Making* 2021. doi: 10.1177/0272989X211068005

Spineli LM, Kalyvas C, Papadimitropoulou K. Quantifying the robustness of primary analysis results: A case study on missing outcome data in pairwise and network meta-analysis. *Res Synth Methods* 2021a;**12**(4):475–90. doi: 10.1002/jrsm.1478

Spineli LM, Kalyvas C, Papadimitropoulou K. Continuous(ly) missing outcome data in network meta-analysis: a one-stage pattern-mixture model approach. *Stat Methods Med Res* 2021b;**30**(4):958–75. doi: 10.1177/0962280220983544

Spineli LM. An empirical comparison of Bayesian modelling strategies for missing binary outcome data in network meta-analysis. *BMC Med Res Methodol* 2019;**19**(1):86. doi: 10.1186/s12874-019-0731-y

van Valkenhoef G, Dias S, Ades AE, Welton NJ. Automated generation of node-splitting models for assessment of inconsistency in network meta-analysis. *Res Synth Methods* 2016;**7**(1):80–93. doi: 10.1002/jrsm.1167

`balloon_plot` *Enhanced balloon plot*

Description

Creates the enhanced balloon plot for the summary effect size and between-trial standard deviation, *tau*, under different scenarios about the missingness parameter for a pair of interventions. `balloon_plot` uses the scenarios considered in `run_sensitivity`.

Usage

```
balloon_plot(sens, compar, drug_names)
```

Arguments

`sens`	An object of S3 class `run_sensitivity`. See 'Value' in `run_sensitivity`.
`compar`	A character vector with two elements indicating the pairwise comparison of interest. The first element refers to the 'experimental' and the second element to the 'control' intervention of the comparison.
`drug_names`	A vector of labels with the name of the interventions in the order they appear in the argument `data` of `run_model`.

Details

For the `plot_effect_size` of the selected pairwise comparison, the different colours and sizes of the bubbles reflect the posterior standard deviation and the posterior mean, respectively. A colour key appears below the plot. The size of the bubble is proportional to the corresponding posterior mean. Crossed bubbles indicate scenarios with conclusive evidence (the 95% credible interval excludes the null value), and filled bubbles indicate scenarios with inconclusive evidence (the 95% credible interval includes the null value). The missing-at-random assumption (primary analysis) is labeled in a white frame. Both axes illustrate the scenarios as specified in the argument `mean_scenarios` of the `run_sensitivity`: the x-axis refers to the 'experimental' intervention, and the y-axis refers to the 'control' intervention.

The same enhanced balloon plot is created for *tau* (`plot_tau`). However, filled bubbles indicate low statistical heterogeneity (the posterior median of *tau* is lower than the median of the prior distribution for the heterogeneity parameter), and crossed bubbles indicate considerable statistical heterogeneity (the posterior median of *tau* exceeds the median of the prior distribution for the heterogeneity parameter).

balloon_plot can be used only when missing participant outcome data have been extracted for at least one trial. Otherwise, the execution of the function will be stopped and an error message will be printed on the R console.

Value

balloon_plot returns two enhanced balloon plots for one comparison (see 'Details'):

plot_effect_size
: The enhanced balloon plot for the summary effect size (according to the argument measure inherited by run_sensitivity) for one pairwise comparison.

plot_tau
: The enhanced balloon plot for *tau*. When the fixed-effect model has been performed in run_sensitivity, the function will not return the plot_tau.

Author(s)

Loukia M. Spineli

References

Spineli LM, Kalyvas C, Papadimitropoulou K. Quantifying the robustness of primary analysis results: A case study on missing outcome data in pairwise and network meta-analysis. *Res Synth Methods* 2021;**12**(4):475–490. doi: https://doi.org/10.1002/jrsm.1478

See Also

run_model, run_sensitivity

Examples

```
data("pma.taylor2004")

# Read results from 'run_sensitivity' (using the default arguments)
res_sens <- readRDS(system.file('extdata/res_sens_taylor.rds',
                    package = 'rnmamod'))

# The names of the interventions in the order they appear in the dataset
interv_names <- c("placebo", "inositol")

# Create the enhanced balloon plot for 'inositol versus placebo'
balloon_plot(sens = res_sens,
             compar = c("inositol", "placebo"),
             drug_names = interv_names)
```

baseline_model *The baseline model for binary outcome*

Description

To process the elements in the argument base_risk of the run_model function. It also runs the hierarchical baseline model, separately from the relative effects model as described in Dias et al. (2018) and Dias et al. (2013b). The output is to be passed to run_model and run_metareg to obtain the (unadjusted and adjusted, respectively) absolute risks for each intervention in the dataset.

Usage

```
baseline_model(base_risk, n_chains, n_iter, n_burnin, n_thin)
```

Arguments

base_risk
: A scalar, a vector of length three with elements sorted in ascending order, or a matrix with two columns and number of rows equal to the number of relevant trials. In the case of a scalar or vector, the elements should be in the interval (0, 1). For the matrix, the first column refers to the number of events and the second column to the sample size of the trials comprising the dataset for the baseline model. See 'Details' in run_model. This argument is only relevant for a binary outcome.

n_chains
: Positive integer specifying the number of chains for the MCMC sampling; an argument of the jags function of the R-package R2jags. The default argument is 2.

n_iter
: Positive integer specifying the number of Markov chains for the MCMC sampling; an argument of the jags function of the R-package R2jags. The default argument is 10000.

n_burnin
: Positive integer specifying the number of iterations to discard at the beginning of the MCMC sampling; an argument of the jags function of the R-package R2jags. The default argument is 1000.

n_thin
: Positive integer specifying the thinning rate for the MCMC sampling; an argument of the jags function of the R-package R2jags. The default argument is 1.

Details

If base_risk is a matrix, baseline_model creates the hierarchical baseline model in the JAGS dialect of the BUGS language. The output of this function (see 'Value') constitutes the posterior mean and precision of the predicted logit of an event for the selected reference intervention and it is plugged in the WinBUGS code for the relative effects model (Dias et al., 2013a) via the prepare_model function. Following (Dias et al., 2013a), a uniform prior distribution is assigned on the between-trial standard deviation with upper and lower limit equal to 0 and 5, respectively.

When base_risk is a matrix, the function also returns a forest plot with the estimated trial-specific probability of an event and 95% credible intervals (the random effects) alongside the corresponding observed probability of an event for the selected reference intervention. A grey rectangular illustrates the summary mean and 95% credible interval of the random effects.

When base_risk is a matrix (predicted baseline), the model is updated until convergence using the autojags function of the R-package R2jags with 2 updates and number of iterations and thinning equal to n_iter and n_thin, respectively.

Value

When base_risk is scalar (fixed baseline), the function returns the user-defined baseline for the selected reference intervention in the logit scale. When base_risk is a vector (random baseline), the function returns a vector with the calculated logit of an event for the selected reference intervention and its precision. Finally, when base_risk is a matrix (predicted baseline), the function returns the following elements:

ref_base A vector with the posterior mean and precision of the **predicted** logit of an event for the selected reference intervention. This vector is be passed to run_model and run_metareg.

figure A forest plot on the trial-specific observed and estimated baseline risk. See 'Details'.

table_baseline A table with the posterior and predictive distribution of the summary baseline mean and the posterior distribution of the between-trial standard deviation in baseline. All results are in the logit scale.

Author(s)

Loukia M. Spineli

References

Dias S, Ades AE, Welton NJ, Jansen JP, Sutton AJ. Network Meta-Analysis for Decision Making. Chichester (UK): Wiley; 2018.

Dias S, Sutton AJ, Ades AE, Welton NJ. Evidence synthesis for decision making 2: a generalized linear modeling framework for pairwise and network meta-analysis of randomized controlled trials. *Med Decis Making* 2013a;**33**(5):607–17. doi: 10.1177/0272989X12458724

Dias S, Welton NJ, Sutton AJ, Ades AE. Evidence synthesis for decision making 5: the baseline natural history model. *Med Decis Making* 2013b;**33**(5):657–70. doi: 10.1177/0272989X13485155

See Also

prepare_model, autojags, jags, run_metareg, run_model

bland_altman_plot *The Bland-Altman plot*

Description

This function facilitates creating the Bland-Altman plot on the posterior mean deviance contribution for two models using only three arguments.

Usage

```
bland_altman_plot(model1, model2, colour)
```

Arguments

model1	A vector with the numeric values of the target model (for instance, the consistency model).
model2	A vector with the numeric values of the reference model (for instance, the unrelated mean effects model).
colour	A string to define the colour of the data points in the plot.

Details

bland_altman_plot is integrated in ume_plot to create the Bland-Altman plot on the posterior mean of deviance under the consistency model (via run_model) and the unrelated mean effects model (via run_ume).

A uniform scattering of the data points within the 95% limits of agreement and average bias close to 0 indicate that the compared models have a good agreement. Data points positioned above or below the 95% limits of agreement correspond to trials that contribute to the poor fit of the consistency model or unrelated mean effects model, respectively.

bland_altman_plot can be used to compare the following models regarding deviance contribution:

- the consistency model (via run_model) with the unrelated effect means model (via run_ume);
- the network meta-analysis model (via run_model) with the network meta-analysis model (via run_metareg).

Value

Bland-Altman plot on the posterior mean deviance contribution of the individual data points under model 1 and model 2. Each data point corresponds to a trial-arm indicated by a pair of numbers. The first number refers to the position of the trial in the dataset, and the second arm refers to the corresponding trial-arm (see 'Arguments' and 'Value' in data_preparation). The plot also displays the average bias and the 95% limits of agreement with horizontal solid black lines.

Author(s)

Loukia M. Spineli

References

Bland JM, Altman DG. Measuring agreement in method comparison studies. *Stat Methods Med Res* 1999;**8**:135–60. doi: 10.1177/096228029900800204

See Also

data_preparation, run_metareg, run_model, run_ume, ume_plot

---

comp_clustering *End-user-ready results for comparison dissimilarity and hierarchical clustering (Comparisons' comparability for transitivity evaluation)*

Description

comp_clustering hosts a toolkit of functions that facilitates conducting, visualising and evaluating hierarchical agglomerative of observed comparisons of interventions for a specific network and set of characteristics that act as effect modifiers. It also calculates the non-statistical heterogeneity within-comparisons and between-comparisons using the dissimilarities among all trials of the network.

Usage

```
comp_clustering(
  input,
  drug_names,
  threshold,
  informative = TRUE,
  ranged_values = FALSE,
  optimal_clusters,
  get_plots = "none",
  label_size = 4,
  title_size = 14,
  axis_title_size = 14,
  axis_text_size = 14,
  axis_x_text_angle = 0,
  legend_text_size = 13,
  str_wrap_width = 10
)
```

Arguments

input A data-frame in the long arm-based format. Two-arm trials occupy one row in the data-frame. Multi-arm trials occupy as many rows as the number of possible comparisons among the interventions. The first three columns refer to the trial name, first and second arm of the comparison, respectively. The remaining columns refer to summary characteristics. See 'Details' for the specification of the columns.

`drug_names` A vector of labels with the name of the interventions in the order they have been defined in the argument `input`.

`threshold` A positive scalar to indicate the cut-off of low dissimilarity of two comparisons. The value must be low.

`informative` Logical with `TRUE` for evaluating only the comparison dissimilarity and `FALSE` for performing hierarchical agglomerative clustering, thus, allowing the user to define the number of clusters via the argument `optimal_clusters`. The default argument is `TRUE`.

`ranged_values` Whether to use a colour scale when creating the heatmap of within-comparison and between-comparison dissimilarities (`TRUE`) or colour the cells with green and orange, when below or exceeding the specified `threshold`. Relevant only when `informative = TRUE`. The default argument is `FALSE`.

`optimal_clusters`

A positive integer for the optimal number of clusters, ideally, decided after inspecting the profile plot with average silhouette widths for a range of clusters, and the dendrogram. The user **must** define the value. It takes values from two to the number of trials minus one.

`get_plots` Logical with values `TRUE` for returning all plots and `FALSE` for concealing the plots. The default argument is `FALSE`.

`label_size` A positive integer for the font size of labels in the violin plot for the study dissimilarities per comparison and comparison between comparisons. `label_size` determines the size argument found in the geom's aesthetic properties in the R-package ggplot2.

`title_size` A positive integer for the font size of legend title in the stacked barplot on the percentage studies of each comparison found in the clusters. `title_size` determines the title argument found in the theme's properties in the R-package ggplot2.

`axis_title_size`

A positive integer for the font size of axis title in the violin plot for the study dissimilarities per comparison and comparison between comparisons, and the barplot of percentage trials per comparison and cluster. `axis_title_size` determines the axis.title argument found in the theme's properties in the R-package ggplot2.

`axis_text_size` A positive integer for the font size of axis text in the violin plot for the study dissimilarities per comparison and comparison between comparisons, the heatmap of comparison dissimilarity, and the barplot of percentage trials per comparison and cluster. `axis_text_size` determines the axis.text argument found in the theme's properties in the R-package ggplot2.

`axis_x_text_angle`

A positive integer for the angle of axis text in the violin plot for the study dissimilarities per comparison and comparison between comparisons. `axis_x_text_angle` determines the axis.text.x argument found in the theme's properties in the R-package ggplot2.

`legend_text_size`

A positive integer for the font size of legend text in the barplot of percentage trials per comparison and cluster. `legend_text_size` determines the legend.text argument found in the theme's properties in the R-package ggplot2.

str_wrap_width A positive integer for wrapping the axis labels in the the violin plot for the study dissimilarities per comparison between comparisons. str_wrap_width determines the str_wrap function of the R-package stringr.

Details

The correct type mode of columns in input must be ensured to use the function comp_clustering. The first three columns referring to the trial name, first and second arm of the comparison, respectively, must be **character**. The remaining columns referring to the characteristics must be **double** or **integer** depending on whether the corresponding characteristic refers to a quantitative or qualitative variable. The type mode of each column is assessed by comp_clustering using the base function typeof. Note that comp_clustering invites unordered and ordered variables; for the latter, add the argument ordered = TRUE in the base function **factor**().

The interventions should be sorted in an ascending order of their identifier number within the trials so that the first intervention column (second column in input) is the control arm for every pairwise comparison. This is important to ensure consistency in the intervention order within the comparisons obtained from the other related functions.

comp_clustering excludes from the dataset the following type of characteristics: (i) completely missing characteristics and (ii) characteristics with missing values in all but one studies for at least one non-single-stufy comparison. Then it proceeds with the clustering process.

The cophenetic correlation coefficient is calculated using the cophenetic function alongside the hclust function for selected linkage methods.

comp_clustering can be used only for a network with at least three comparisons. Otherwise, the execution of the function will be stopped and an error message will be printed on the R console.

Value

Initially, comp_clustering prints on the console the following messages: the number of observed comparisons (and number of single-study comparisons, if any); the number of dropped characteristics due to many missing data; the maximum value of the cophenetic correlation coefficient; and the optimal linkage method selected based on the cophenetic correlation coefficient. Then, the function returns the following list of elements:

Trials_diss_table
A lower off-diagonal matrix of 'dist' class with the Gower dissimilarities of all pairs of studies in the network.

Comparisons_diss_table
A lower off-diagonal matrix of 'dist' class with the within-comparison dissimilarities at the main diagonal and the between-comparison dissimilarities of all pairs of observed intervention comparisons at the off-diagonal elements.

Total_dissimilarity
A data-frame on the observed comparisons and comparisons between comparisons, alongside the corresponding within-comparison and between-comparisons dissimilarity. The data-frame has been sorted in decreasing within each dissimilarity 'type'.

Types_used A data-frame with type mode (i.e., double or integer) of each characteristic.

`Total_missing` The percentage of missing cases in the dataset, calculated as the ratio of total missing cases to the product of the number of studies with the number of characteristics.

`Cluster_comp` A data-frame on the studies and the cluster they belong (based on the argument `optimal_clusters`.

`Table_average_silhouette_width`
A data-frame with the average silhouette width for a range of 2 to P-1 trials, with P being the number trials.

`Table_cophenetic_coefficient`
A data-frame on the cophenetic correlation coefficient for eight linkage methods (Ward's two versions, single, complete, average, Mcquitty, median and centroid). The data-frame has been sorted in decreasing order of the cophenetic correlation coefficient.

`Optimal_link` The optimal linkage method (ward.D, ward.D2, single, complete, average, mcquitty, median, or centroid) based on the cophenetic correlation coefficient.

If `get_plots = FALSE` only the list of elements mentioned above is returned. If `get_plots = TRUE`, `comp_clustering` returns a series of plots in addition to the list of elements mentioned above:

`Within_comparison_dissimilarity`
A violin plot with integrated box plots and dots on the study dissimilarities per observed comparison (x-axis). Violins are sorted in descending order of the within-comparison dissimilarities (blue point).

`Between_comparison_dissimilarity`
A violin plot with integrated box plots and dots on the study dissimilarities per comparison between comparisons (x-axis). Violins are sorted in descending order of the between-comparison dissimilarities (blue point).

`Dissimilarity_heatmap`
A heatmap on within-comparison and between-comparison dissimilarities when (`informative = TRUE`). Diagonal elements refer to within-comparison dissimilarity, and off-diagonal elements refer to between-comparisons dissimilarity. Using a threshold of high similarity (specified using the argument `threshold`), cells equal or above this threshold are highlighted in orange; otherwise, in green. This heatmap aids in finding 'hot spots' of comparisons that may violate the plausibility of transitivity in the network. Single-study comparisons are indicated with white numbers.

`Profile_plot` A profile plot on the average silhouette width for a range of 2 to P-1 clusters, with P being the number of trials. The candidate optimal number of clusters is indicated with a red point directly on the line.

`Silhouette_width_plot`
A silhouette plot illustrating the silhouette width for each trial, with the trials sorted in decreasing order within the cluster they belong. This output is obtained by calling the `silhouette` function in the R-package cluster.

`Barplot_comparisons_cluster`
As stacked barplot on the percentage trials of each comparison found in the clusters (based on the argument `optimal_clusters`.

Author(s)

Loukia M. Spineli

References

Gower J. General Coefficient of Similarity and Some of Its Properties. *Biometrics* 1971;**27**(4):857–71. doi: 10.2307/2528823

Sokal R, Rohlf F. The Comparison of Dendrograms by Objective Methods. *Int Assoc Plant Taxon* 1962;**11**(2):33–40. doi: 10.2307/1217208

Handl J, Knowles J, Kell DB. Computational cluster validation in post-genomic data analysis. *Biometrics* 2005;**21**(15):3201–120. doi: 10.1093/bioinformatics/bti517

Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987;**20**:53–65.

See Also

cophenetic, hclust, internal_measures_plot, silhouette, str_wrap

Examples

```
# Fictional dataset
data_set <- data.frame(Trial_name = as.character(1:7),
                       arm1 = c("1", "1", "1", "1", "1", "2", "2"),
                       arm2 = c("2", "2", "2", "3", "3", "3", "3"),
                       sample = c(140, 145, 150, 40, 45, 75, 80),
                       age = c(18, 18, 18, 48, 48, 35, 35),
                       blinding = factor(c("yes", "yes", "yes", "no", "no", "no", "no")))

# Obtain comparison dissimilarities (informative = TRUE)
comp_clustering(input = data_set,
                drug_names = c("A", "B", "C"),
                threshold = 0.13,  # General research setting
                informative = TRUE,
                get_plots = TRUE)
```

data_preparation *Prepare the dataset in the proper format for R2jags*

Description

data_preparation prepares the dataset in the proper format for R2jags and returns a list of elements that run_model inherits via the argument data.

Usage

```
data_preparation(data, measure)
```

Arguments

`data`	A data-frame of the one-trial-per-row format with arm-level data. See 'Format' in `run_model`.
`measure`	Character string indicating the effect measure. For a binary outcome, the following can be considered: `"OR"`, `"RR"` or `"RD"` for the odds ratio, relative risk, and risk difference, respectively. For a continuous outcome, the following can be considered: `"MD"`, `"SMD"`, or `"ROM"` for mean difference, standardised mean difference and ratio of means, respectively.

Details

`data_preparation` prepares the data for the Bayesian analysis (See 'Format' in `run_model`). `data_preparation` creates the pseudo-data-frames `m_new`, `I`, and `m_pseudo` that have the same dimensions with the element `N`. `m_new` takes the zero value for the observed trial-arms with unreported missing participant outcome data (i.e., **m** equals `NA` for the corresponding trial-arms), the same value with **m** for the observed trial-arms with reported missing participant outcome data, and `NA` for the unobserved trial-arms. `I` is a dummy data-frame and takes the value one for the observed trial-arms with reported missing participant outcome data, the zero value for the observed trial-arms with unreported missing participant outcome data (i.e., `m_new` equals zero for the corresponding trial-arms), and `NA` for the unobserved trial-arms. Thus, `I` indicates whether missing participant outcome data have been collected for the observed trial-arms. If the user has not defined the element **m** in `data_preparation`, `m_new` and `I` take the zero value for all observed trial-arms to indicate that no missing participant outcome data have been collected for the analysed outcome. `I` and `m_new` are used from the following functions of the package: `run_model`, `run_metareg`, `prepare_model`, `run_nodesplit`, `prepare_nodesplit`, `run_ume`, `prepare_ume`, and `run_sensitivity`. Lastly, `m_pseudo` is a variant of `m_new`: it takes the value -1 for the observed trial-arms with unreported missing participant outcome data (i.e., **m** equals `NA` for the corresponding trial-arms), the same value with **m** for the observed trial-arms with reported missing participant outcome data, and `NA` for the unobserved trial-arms. It is used in function `heatmap_missing_network` to calculate and illustrate the percentage of missing participant outcome data across the observed comparisons and interventions of the network and the function `heatmap_missing_dataset` to illustrate the trial-arms with unreported missing participant outcome data. All pseudo-data-frames aim to retain the trials without information on missing participant outcome data.

Furthermore, `data_preparation` sorts the interventions across the arms of each trial in an ascending order and correspondingly the remaining elements in `data` (See 'Format' in `run_model`). `data_preparation` considers the first column in **t** as being the control arm for every trial. Thus, this sorting ensures that interventions with a lower identifier are consistently treated as the control arm in each trial. This case is relevant in non-star-shaped networks.

Value

A list of data-frames on the following elements to be passed to `run_model`:

`pseudo_m` A pseudo-data-frame with values -1 and **m** for the corresponding trial-arms with unreported and reported missing participant outcome data, respectively (see 'Details').

`m` The number of missing participant outcome data in each trial-arm (see 'Details').

`N` The number of randomised participants in each trial-arm.

`t` The intervention identifier in each trial-arm.

`I` A pseudo-data-frame that indicates whether missing participant outcome data have been reported or not for each observed trial-arm (see 'Details').

`measure` The effect measure for the analysed outcome.

`y0` The observed mean value of the outcome in each trial-arm, when the outcome is continuous.

`se0` The observed standard deviation of the outcome in each trial-arm, when the outcome is continuous.

`r` The number of observed events of the outcome in each trial-arm, when the outcome is binary.

Author(s)

Loukia M. Spineli

See Also

`heatmap_missing_dataset`, `heatmap_missing_network`, R2jags, `run_metareg`, `run_model`, `run_nodesplit`, `run_sensitivity`, `run_ume`, `prepare_model`, `prepare_nodesplit`, `prepare_ume`

`dendro_heatmap` *Dendrogram with amalgamated heatmap (Comparisons' comparability for transitivity evaluation)*

Description

`dendro_heatmap` creates a dendrogram alongside the heatmap of Gower dissimilarities among the trials in the network for a specific linkage method and number of clusters.

Usage

```
dendro_heatmap(input, label_size = 12, axis_text_size = 10)
```

Arguments

`input` An object of S3 class `comp_clustering`. See 'Value' in `comp_clustering`.

`label_size` A positive integer for the font size of the heatmap elements. `label_size` determines the size argument found in the geom's aesthetic properties in the R-package ggplot2.

`axis_text_size` A positive integer for the font size of row and column names of the heatmap. `axis_text_size` determines the axis.text argument found in the theme's properties in the R-package ggplot2.

Value

`dendro_heatmap` uses the `heatmaply` function of the R-package heatmaply to create a cluster heatmap for a selected linkage method and number of clusters. The function uses different colours to indicate the clusters directly on the dendrogram, specified using the R-package dendextend. The names of the leaves refer to the trials and corresponding pairwise comparison.

@details The function inherits the linkage method and number of optimal clusters by the `comp_clustering` function.

Remember: when using the `comp_clustering` function, inspect the average silhouette width for a wide range of clusters to decide on the optimal number of clusters.

Author(s)

Loukia M. Spineli

See Also

`comp_clustering`, `heatmaply`

Examples

```
# Fictional dataset
data_set <- data.frame(Trial_name = as.character(1:7),
                       arm1 = c("1", "1", "1", "1", "1", "2", "2"),
                       arm2 = c("2", "2", "2", "3", "3", "3", "3"),
                       sample = c(140, 145, 150, 40, 45, 75, 80),
                       age = c(18, 18, 18, 48, 48, 35, 35),
                       blinding = factor(c("yes", "yes", "yes", "no", "no", "no", "no")))

# Apply hierarchical clustering (informative = FALSE)
hier <- comp_clustering(input = data_set,
                        drug_names = c("A", "B", "C"),
                        threshold = 0.13,  # General research setting
                        informative = FALSE,
                        optimal_clusters = 3,
                        get_plots = TRUE)

# Create the dendrogram with integrated heatmap
dendro_heatmap(hier)
```

describe_network *A function to describe the evidence base*

Description

Calculates the necessary elements to describe the evidence base for an outcome across the network, the interventions, and observed comparisons.

Usage

```
describe_network(data, drug_names, measure, save_xls)
```

Arguments

`data` A data-frame of a one-trial-per-row format containing arm-level data of each trial. See 'Format' in `run_model`.

`drug_names` A vector of labels with the name of the interventions in the order they appear in the argument `data`.

`measure` Character string indicating the effect measure. For a binary outcome, the following can be considered: `"OR"`, `"RR"` or `"RD"` for the odds ratio, relative risk, and risk difference, respectively. For a continuous outcome, the following can be considered: `"MD"`, `"SMD"`, or `"ROM"` for mean difference, standardised mean difference and ratio of means, respectively.

`save_xls` Logical to indicate whether to export the tabulated results to an 'xlsx' file (via the `write_xlsx` function of the R-package writexl) at the working directory of the user. The default is `FALSE` (do not export).

Details

`describe_network` calls `data_preparation` to facilitate the calculations.

Furthermore, `describe_network` exports the data-frames to separate 'xlsx' files (via the `write_xlsx` function of the R-package writexl) at the working directory of the user.

Value

`describe_network` returns the following data-frames that describe the evidence base:

`network_description` The number of: interventions, possible comparisons, direct and indirect comparisons, number of trials in total, number of two-arm and multi-arm trials, number of randomised participants, and proportion of participants completing the trial (completers). When the outcome is binary, the number of trials with at least one zero event, and the number of trials with all zero events are also presented.

`table_interventions`
: For each intervention, the number of trials, number of randomised participants, and proportion of completers. When the outcome is binary, the data-frame presents also the corresponding proportion of total observed events, the minimum, median and maximum proportion of observed events across the corresponding trials.

`table_comparisons`
: Identical structure to `table_interventions` but for each observed comparison in the network.

Author(s)

Loukia M. Spineli

See Also

`data_preparation`, `run_model` `write_xlsx`

`distr_characteristics` *Visualising the distribution of characteristics (Comparisons' comparability for transitivity evaluation)*

Description

`distr_characteristics` uses violin and bar plots to visualise the distribution of each characteristic in the dataset either per comparison or cluster of comparisons.

Usage

```
distr_characteristics(
  input,
  drug_names,
  rename_char = NULL,
  cluster = NULL,
  label_size = 4,
  title_size = 14,
  axis_title_size = 14,
  axis_text_size = 14,
  axis_x_text_angle = 0,
  legend_text_size = 13
)
```

Arguments

`input`
: A data-frame in the long arm-based format. Two-arm trials occupy one row in the data-frame. Multi-arm trials occupy as many rows as the number of possible comparisons among the interventions. The first three columns refer to the trial

name, first and second arm of the comparison (their identifier number), respectively. The remaining columns refer to summary characteristics. See 'Details' for specifying the columns.

drug_names A vector of labels with the name of the interventions in the order they appear in the argument input.

rename_char A list of two elements: (i) a numeric vector with the position of the characteristics in input, and (ii) a character vector with the names of the characteristics, as they are wished to appear in the title of the plots. This argument is optional, in case the user wants to control the appearance of the titles.

cluster An object of S3 class comp_clustering that has information on the cluster of each comparison. See 'Value' in comp_clustering. If cluster is not provided, the function presents the distribution of characteristics per comparison; otherwise per cluster. In the latter, the function prints a table with the comparisons and the corresponding cluster.

label_size A positive integer for the font size of labels in the plots. label_size determines the size argument found in the geom's aesthetic properties in the R-package ggplot2).

title_size A positive integer for the font size of legend title in the plots. title_size determines the title argument found in the theme's properties in the R-package ggplot2.

axis_title_size

A positive integer for the font size of axis title in the plots. axis_title_size determines the axis.title argument found in the theme's properties in the R-package ggplot2.

axis_text_size A positive integer for the font size of axis text in the plots. axis_text_size determines the axis.text argument found in the theme's properties in the R-package ggplot2).

axis_x_text_angle

A positive integer for the angle of axis text in the plots. axis_text_angle determines the axis.text.x argument found in the theme's properties in the R-package ggplot2).

legend_text_size

A positive integer for the font size of legend text in the plots. legend_text_size determines the legend.text argument found in the theme's properties in the R-package ggplot2.

Details

The correct type mode of columns in input must be ensured to use the function distr_characteristics. The first three columns referring to the trial name, first and second arm of the comparison, respectively, must be **character**. The remaining columns referring to the characteristics must be **double** or **integer** depending on whether the corresponding characteristic refers to a quantitative or qualitative variable. The type mode of each column is assessed by distr_characteristics using the base function typeof.

The interventions should be sorted in an ascending order of their identifier number within the trials so that the first treatment column (second column in input) is the control arm for every pairwise comparison. This is important to ensure consistency in the order of interventions within the comparisons obtained from the other related functions.

Value

distr_characteristics returns a list of plots using the proper plot (violin or bar plot) for each characteristic. The size of the dots in the violin plot (with amalgamated box plots and dots) are proportional to the total sample size of the study: the large the sample size of the study, the larger the size of the corresponding point.

Author(s)

Loukia M. Spineli

See Also

comp_clustering

Examples

```
# Fictional dataset
set.seed(13022024)
data_set <- data.frame(Trial_name = as.character(1:(5 + 7 +2)),
                       arm1 = rep(c("1", "2"), c(5 + 7, 2)),
                       arm2 = rep(c("2", "3"), c(5, 7 + 2)),
                       sample = as.numeric(sample(50:300, 5 + 7 + 2)),
                       age = as.numeric(sample(18:50, 5 + 7 + 2)),
                       blinding = factor(rep(c("yes", "no", "yes"), c(5, 7, 2))))

distr_characteristics(input = data_set,
                      drug_names = c("A", "B", "C"))
```

forestplot *Comparator-specific forest plot for network meta-analysis*

Description

Provides a forest plot with the posterior median and 95% credible and prediction intervals for comparisons with the selected intervention (comparator) in the network, and a forest plot with the corresponding SUCRA values.

Usage

```
forestplot(full, compar, drug_names)
```

Arguments

`full` An object of S3 class `run_model`. See 'Value' in `run_model`.

`compar` A character to indicate the comparator intervention. It must be any name found in `drug_names`.

`drug_names` A vector of labels with the name of the interventions in the order they appear in the argument `data` of `run_model`.

Details

The y-axis of the forest plot on **effect sizes** displays the labels of the interventions in the network; the selected intervention that comprises the `compar` argument is annotated in the plot with the label 'Comparator intervention'. For each comparison with the selected intervention, the 95% credible and prediction intervals are displayed as overlapping lines in different colours. The corresponding numerical results are displayed above each line: 95% credible intervals are found in parentheses, and 95% predictive intervals are found in brackets. Odds ratios, relative risks, and ratio of means are reported in the original scale after exponentiation of the logarithmic scale.

The y-axis for the forest plot on **SUCRA** values displays the labels of the interventions in the network. The corresponding numerical results are displayed above each line. Three coloured rectangles appear in the forest plot: a red rectangle for SUCRA values up to 50%, a yellow rectangular for SUCRA values between 50% and 80%, and a green rectangle for SUCRA values over 80%. Interventions falling at the green area are considered as the highest ranked interventions, whilst interventions falling at the red area are considered as the lowest ranked interventions.

In both plots, the interventions are sorted in descending order of their SUCRA values.

`forestplot` can be used only for a network of interventions. Otherwise, the execution of the function will be stopped and an error message will be printed on the R console.

Value

A panel of two forest plots: (1) a forest plot on the effect estimates and predictions of comparisons with the selected intervention in the network, and (2) a forest plot on the posterior mean and 95% credible interval of SUCRA values of the interventions (Salanti et al., 2011).

Author(s)

Loukia M. Spineli

References

Salanti G, Ades AE, Ioannidis JP. Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: an overview and tutorial. *J Clin Epidemiol* 2011;**64**(2):163–71. doi: 10.1016/j.jclinepi.2010.03.016

See Also

`run_model`

Examples

```
data("nma.liu2013")

# Show the first six trials of the dataset (one-trial-per-row format)
head(nma.liu2013)

# Read results from 'run_model' (using the default arguments)
res <- readRDS(system.file('extdata/res_liu.rds', package = 'rnmamod'))

# The names of the interventions in the order they appear in the dataset
interv_names <- c("placebo", "pramipexole", "serotonin-norepinephrine
reuptake inhibitor", "serotonin reuptake inhibitor", "tricyclic
antidepressant", "pergolide")

# Create the forest plot
forestplot(full = res,
           compar = "placebo",
           drug_names = interv_names)
```

`forestplot_metareg` *Comparator-specific forest plot for network meta-regression*

Description

Provides a forest plot with the posterior median and 95% credible and prediction intervals for comparisons with the selected intervention (comparator) in the network under the network meta-analysis *and* network meta-regression for a specified level or value of the investigated covariate, and a forest plot with the corresponding SUCRA values.

Usage

```
forestplot_metareg(full, reg, compar, cov_value, drug_names)
```

Arguments

`full` An object of S3 class `run_model`. See 'Value' in `run_model`.

`reg` An object of S3 class `run_metareg`. See 'Value' in `run_metareg`.

`compar` A character to indicate the comparator intervention. It must be any name found in `drug_names`.

`cov_value` A list of two elements in the following order: a number for the covariate value of interest (see 'Arguments' in `run_metareg`), and a character to indicate the name of the covariate. See also 'Details'.

`drug_names` A vector of labels with the name of the interventions in the order they appear in the argument `data` of `run_model`. If `drug_names` is not defined, the order of the interventions as they appear in `data` is used, instead.

Details

The y-axis of the forest plot on **effect sizes** displays the labels of the interventions in the network; the selected intervention that comprises the `compar` argument is annotated in the plot with the label 'Comparator intervention'. For each comparison with the selected intervention, the 95% credible and prediction intervals are displayed as overlapping lines. Black lines refer to estimation under both analyses. Green and red lines refer to prediction under network meta-analysis and network meta-regression, respectively. The corresponding numerical results are displayed above each line: 95% credible intervals are found in parentheses, and 95% predictive intervals are found in brackets. Odds ratios, relative risks, and ratio of means are reported in the original scale after exponentiation of the logarithmic scale.

The y-axis for the forest plot on **SUCRA** values displays the labels of the interventions in the network. The corresponding numerical results are displayed above each line. Three coloured rectangles appear in the forest plot: a red rectangle for SUCRA values up to 50%, a yellow rectangular for SUCRA values between 50% and 80%, and a green rectangle for SUCRA values over 80%. Interventions falling at the green area are considered as the highest ranked interventions, whilst interventions falling at the red area are considered as the lowest ranked interventions.

In both plots, the interventions are sorted in the descending order of their SUCRA values based on the network meta-analysis.

To obtain the posterior distribution of SUCRAs under the network meta-regression for a specified level or value of the investigated covariate, a two-step procedured is followed. First, the posterior median and standard deviation of the basic parameters and corresponding beta coefficients are considered to calculate the meadin and standard deviation of the basic parameters at the selected level or value of the investigated covariate: d + beta * cov_value. Then, these calculated values are fed into the BUGS code to get the posterior distribution of SUCRA. The progress of the simulation appears on the R console.

`forestplot_metareg` is integrated in `metareg_plot`.

`forestplot_metareg` can be used only for a network of interventions. In the case of two interventions, the execution of the function will be stopped and an error message will be printed on the R console.

Value

A panel of two forest plots: (1) a forest plot on the effect estimates and predictions of comparisons with the selected intervention in the network under the network meta-analysis and network meta-regression for a specified level or value of the investigated covariate, and (2) a forest plot on the posterior mean and 95% credible interval of SUCRA values of the interventions (Salanti et al., 2011).

Author(s)

Loukia M. Spineli

References

Salanti G, Ades AE, Ioannidis JP. Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: an overview and tutorial. *J Clin Epidemiol* 2011;**64**(2):163–71. doi:10.1016/j.jclinepi.2010.03.016

See Also

metareg_plot, run_metareg, run_model

gower_distance *Gower's dissimilarity measure (Trials' comparability for transitivity evaluation)*

Description

gower_distance calculate the Gower's dissimilarity coefficient for all pairs of trials included in a network of interventions, considering several characteristics measured at trial level. It takes values from 0 to 1, with 0 implying complete similarity and 1 complete dissimilarity.

Usage

```
gower_distance(input)
```

Arguments

input	A data-frame in the long arm-based format. Two-arm trials occupy one row in the data-frame. Multi-arm trials occupy as many rows as the number of possible comparisons among the interventions. The first two columns refer to the trial name, and the pairwise comparison, respectively. The remaining columns refer to summary characteristics. See 'Details' for the specification of the columns.

Details

The correct type mode of columns in `input` must be ensured to use the function `gower_distance`. The first two columns referring to the trial name, and pairwise comparison, respectively, must be **character**. The remaining columns referring to the characteristics must be **double** or **integer** depending on whether the corresponding characteristic refers to a quantitative or qualitative variable. The type mode of each column is assessed by `gower_distance` using the base function `typeof`. Note that `gower_distance` invites unordered and ordered variables; for the latter, add the argument `ordered = TRUE` in the base function **factor()**.

`gower_distance` is integrated in the function comp_clustering.

Value

`gower_distance` returns the following list of elements:

Dissimilarity_table	A lower off-diagonal matrix of 'dist' class with the dissimilarities of all pairs of trials.
Types_used	A data-frame with type mode (i.e., double or integer) of each characteristic.
Total_missing	The percentage of missing cases in the comparison, calculated as the ratio of total missing cases to the product of the number of studies with the number of characteristics.

Author(s)

Loukia M. Spineli

References

Gower J. General Coefficient of Similarity and Some of Its Properties. *Biometrics* 1971;**27**(4):857–71. doi: 10.2307/2528823

See Also

`comp_clustering`

`heatmap_missing_dataset`

Heatmap of proportion of missing participants in the dataset

Description

Illustrates the proportion of missing participants and the associated risk of bias in each arm of every trial in the dataset.

Usage

```
heatmap_missing_dataset(data, trial_names, drug_names)
```

Arguments

`data`	A data-frame of a one-trial-per-row format containing arm-level data of each trial. See 'Format' in `run_model`.
`trial_names`	A vector of labels with the name of the trials in the order they appear in the argument `data`.
`drug_names`	A vector of labels with the name of the interventions in the order they appear in the argument `data`.

Details

When the number of missing participants has not been extracted for any arm of the trials, the execution of the function will be stopped and an error message will be printed on the R console.

When there are more than 80 trials, the value on the proportion of missing participants will not appear on the heatmap. This is because the number on the cells will not be distinguishable.

Value

A heatmap presenting the proportion of missing participants in each trial-arm of the dataset. The columns and the rows of the heatmap correspond to the interventions and trials, respectively. The 'five-and-twenty' rule of Sackett and colleagues (1997) is used to characterise the proportion of missing participants as being associated with low (up to 5%), moderate (more than 5% and up to 20%), and high risk of bias (more than 20%). Low, moderate, and high risk of bias due to missing participants are indicated using green, orange, and red colour, respectively. The function is also applicable for a pairwise meta-analysis. If missing participants have not been reported for a trial-arm, the corresponding cell is indicated in grey.

Author(s)

Loukia M. Spineli

References

Sackett DL, Richardson WS, Rosenberg WM, Haynes RB. Evidence-based medicine: how to practice and teach EBM. New York: Churchill Livingstone 1997. ISBN: 0-443-05686-2.

See Also

run_model

Examples

```
data("nma.schwingshackl2014")

# Return the first six trials of the dataset
head(nma.schwingshackl2014)

# The names of the interventions in the order they appear in the dataset
interv_names <- c("aerobic", "resistance", "combined training")

# Create the heatmap
heatmap_missing_dataset(data = nma.schwingshackl2014,
                        trial_names = nma.schwingshackl2014$study,
                        drug_names = interv_names)
```

---

heatmap_missing_network

Heatmap of proportion of missing participants in the network

Description

Illustrates the distribution of missing participants and the associated risk of bias for each intervention and observed comparison in the network.

Usage

```
heatmap_missing_network(data, drug_names)
```

Arguments

`data`	A data-frame of a one-trial-per-row format containing arm-level data of each trial. See 'Format' in `run_model`.
`drug_names`	A vector of labels with the name of the interventions in the order they appear in the argument `data`.

Value

A heatmap with the proportion of missing participants in each intervention and observed comparison in the network. Each cell annotates the median, minimum and maximum (the latter two in parenthesis) proportion of missing participants across the corresponding trials. The proportion of missing participants in each intervention and observed comparison are depicted in the main diagonal and lower off-diagonal with white and black colour, respectively. The pairwise comparisons are read from left to right.

The 'five-and-twenty' rule of Sackett and colleagues (1997) is used to characterise the **median** proportion of missing participants as being associated with low (up to 5%), moderate (more than 5% and up to 20%), and high risk of bias (more than 20%). Low, moderate, and high risk of bias associated with missing participants is indicated using green, orange, and red colour, respectively. If missing participants have not been reported for an intervention or comparison, the corresponding cell is indicated in grey.

The summary statistics (median, minimum and maximum) for each intervention (main diagonal; white font) result from calculating the proportion of missing participants in each arm of every trial and then summarising across the corresponding trial-arms. Similarly, the summary statistics for each observed comparison (lower off-diagonal; black font) result from calculating the proportion of total missing participants in each trial and then summarising across the corresponding trials.

`heatmap_missing_network` can be used only for a network of interventions. Otherwise, the execution of the function will be stopped and an error message will be printed on the R console. Likewise, when the number of missing participants has not been extracted for any arm of the trials.

Author(s)

Loukia M. Spineli

References

Sackett DL, Richardson WS, Rosenberg WM, Haynes RB. Evidence-based medicine: how to practice and teach EBM. New York: Churchill Livingstone 1997. ISBN: 0-443-05686-2.

See Also

`run_model`

Examples

```
data("nma.stowe2011")

# Return the first six trials of the dataset
head(nma.stowe2011)

# The names of the interventions in the order they appear in the dataset
interv_names <- c("PBO+LD", "DA+LD", "COMTI+LD", "MAOBI+LD")

# Create the heatmap
heatmap_missing_network(data = nma.stowe2011,
                        drug_names = interv_names)
```

heatmap_robustness *Heatmap of robustness*

Description

Facilitates the detection of comparisons that are associated with a lack of robustness in the context of a sensitivity analysis.

Usage

```
heatmap_robustness(robust, drug_names)
```

Arguments

robust	An object of S3 class robustness_index and robustness_index_user. See 'Value' in robustness_index and robustness_index_user.
drug_names	A vector of labels with the name of the interventions in the order they appear in the argument data of run_model. If drug_names is not defined, the order of the interventions as they appear in data is used, instead.

Details

The heatmap illustrates the robustness index for each possible pairwise comparison in the network. The pairwise comparisons are read from left to right. Comparisons highlighted with green or red colour imply robust or frail conclusions for the primary analysis, respectively. This corresponds to robustness index below or at least the selected threshold of robustness. heatmap_robustness inherits the threshold of robustness selected in the robustness_index or robustness_index_user function. The robustness index of each pairwise comparison also appears in the corresponding cell. When there is at least one comparison with frail conclusions, the primary analysis results may be questionable for the whole network (Spineli et al., 2021).

heatmap_robustness is *not* restricted to the sensitivity analysis concerning the impact of missing participant outcome data.

heatmap_robustness can be used only for a network of interventions. Otherwise, the execution of the function will be stopped and an error message will be printed on the R console.

Value

heatmap_robustness first prints on the R console a message on the threshold of robustness determined by the user in robustness_index and robustness_index_user. Then, it returns a lower triangular heatmap matrix with the robustness index value of all possible pairwise comparisons.

Author(s)

Loukia M. Spineli

References

Spineli LM, Kalyvas C, Papadimitropoulou K. Quantifying the robustness of primary analysis results: A case study on missing outcome data in pairwise and network meta-analysis. *Res Synth Methods* 2021;**12**(4):475–90. doi: 10.1002/jrsm.1478

See Also

robustness_index, robustness_index_user, run_model

Examples

```
data("nma.baker2009")

# Read results from 'run_sensitivity' (using the default arguments)
res_sens <- readRDS(system.file('extdata/res_sens_baker.rds',
                    package = 'rnmamod'))

# Calculate the robustness index
robust <- robustness_index(sens = res_sens,
                           threshold = 0.28)

# The names of the interventions in the order they appear in the dataset
interv_names <- c("placebo", "budesonide", "budesonide plus formoterol",
                  "fluticasone", "fluticasone plus salmeterol",
                  "formoterol", "salmeterol", "tiotropium")

# Create the heatmap of robustness
heatmap_robustness(robust = robust,
                   drug_names = interv_names)
```

heterogeneity_param_prior

Determine the prior distribution for the heterogeneity parameter

Description

Generates the prior distribution (weakly informative or empirically-based) for the heterogeneity parameter. run_model inherits heterogeneity_param_prior via the argument heter_prior.

Usage

```
heterogeneity_param_prior(measure, model, heter_prior)
```

Arguments

`measure` Character string indicating the effect measure. For a binary outcome, the following can be considered: `"OR"`, `"RR"` or `"RD"` for the odds ratio, relative risk, and risk difference, respectively. For a continuous outcome, the following can be considered: `"MD"`, `"SMD"`, or `"ROM"` for mean difference, standardised mean difference and ratio of means, respectively.

`model` Character string indicating the analysis model with values `"RE"`, or `"FE"` for the random-effects and fixed-effect model, respectively. The default argument is `"RE"`.

`heter_prior` A list of three elements with the following order: 1) a character string indicating the distribution with (currently available) values `"halfnormal"`, `"uniform"`, `"lognormal"`, or `"logt"`; 2) two numeric values that refer to the parameters of the selected distribution. For `"lognormal"`, and `"logt"` these numbers refer to the mean and precision, respectively. For `"halfnormal"`, these numbers refer to zero and the scale parameter (equal to 4 or 1 being the corresponding precision of the scale parameter 0.5 or 1). For `"uniform"`, these numbers refer to the minimum and maximum value of the distribution.

Details

The names of the (current) prior distributions follow the JAGS syntax. The mean and precision of `"lognormal"` and `"logt"` should align with the values proposed by Turner et al. (2015) and Rhodes et al. (2015) for the corresponding empirically-based prior distributions when `measure` is `"OR"` or `"SMD"`, respectively. The users may refer to Dias et al. (2013) to determine the minimum and maximum value of the uniform distribution, and to Friede et al. (2017) to determine the mean and precision of the half-normal distribution. When `model` is `"FE"`, `heterogeneity_param_prior` is ignored in `run_model`.

Value

A value to be passed to `run_model`.

Author(s)

Loukia M. Spineli

References

Dias S, Sutton AJ, Ades AE, Welton NJ. Evidence synthesis for decision making 2: a generalized linear modeling framework for pairwise and network meta-analysis of randomized controlled trials. *Med Decis Making* 2013;**33**(5):607–17. doi: 10.1177/0272989X12458724

Friede T, Roever C, Wandel S, Neuenschwander B. Meta-analysis of two studies in the presence of heterogeneity with applications in rare diseases. *Biom J* 2017;**59**(4):658–71. doi: 10.1002/bimj.201500236

Rhodes KM, Turner RM, Higgins JP. Predictive distributions were developed for the extent of heterogeneity in meta-analyses of continuous outcome data. *J Clin Epidemiol* 2015;**68**(1):52–60. doi: 10.1016/j.jclinepi.2014.08.012

Turner RM, Jackson D, Wei Y, Thompson SG, Higgins JP. Predictive distributions for between-study heterogeneity and simple methods for their application in Bayesian meta-analysis. *Stat Med* 2015;**34**(6):984–98. doi: 10.1002/sim.6381

See Also

`run_model`

`improved_ume` *Detect the frail comparisons in multi-arm trials*

Description

Detects the frail comparisons in multi-arm trials, that is, comparisons between non-baseline interventions not investigated in any two-arm trial in the network (Spineli, 2021). The 'original' model of Dias et al. (2013) omits the frail comparisons from the estimation process of the unrelated mean effects model. Consequently, their posterior distribution coincides with the prior distribution yielding implausible posterior standard deviations.

Usage

```
improved_ume(t, N, ns, na)
```

Arguments

`t`	A data-frame of the one-trial-per-row format containing the intervention identifier in each arm of every trial (see 'Details' below, and 'Format' in `run_model`).
`N`	A data-frame of the one-trial-per-row format containing the number of participants randomised to the assigned intervention in each arm of every trial (see 'Details' below, and 'Format' in `run_model`).
`ns`	A scale parameter on the number trials.
`na`	A vector of length equal to `ns` with the number of arms in each trial.

Details

`improved_ume` is integrated in `run_ume` and calls the output of `data_preparation` after sorting the rows so that multi-arm trials appear at the bottom of the dataset. When there are no multi-arm trials or no frail comparisons in the network, `improved_ume` returns only the element `obs_comp` (see, 'Value').

Value

The output of `improved_ume` is a list of elements that are inherited by `run_ume`:

`nbase_multi`	A scalar parameter on the number of frail comparisons.
`t1_bn`	A vector with numeric values referring to the first arm of each frail comparison.
`t2_bn`	A vector with numeric values referring to the second arm of each frail comparison.
`ref_base`	A scalar referring to the reference intervention for the subnetwork of interventions in frail comparisons.
`base`	A vector with numeric values referring to the baseline intervention of the multi-arm trials that contain the frail comparisons.
`obs_comp`	A data-frame that indicates how many two-arm and multi-arm trials have included each pairwise comparison observed in the network.

Author(s)

Loukia M. Spineli

References

Dias S, Welton NJ, Sutton AJ, Caldwell DM, Lu G, Ades AE. Evidence synthesis for decision making 4: inconsistency in networks of evidence based on randomized controlled trials. *Med Decis Making* 2013;**33**(5):641–56. doi: 10.1177/0272989X12455847

Spineli LM. A revised framework to evaluate the consistency assumption globally in a network of interventions. *Med Decis Making* 2021. doi: 10.1177/0272989X211068005

See Also

`data_preparation`, `run_model`, `run_ume`

`internal_measures_plot` *Internal measures for cluster validation (Comparisons' comparability for transitivity evaluation)*

Description

`internal_measures_plot` currently prepares the table with the results of the average silhouette width for a range of clusters, and visualises the results using a profile plot.

Usage

```
internal_measures_plot(
  input,
  optimal_link,
  label_size = 4,
  axis_title_size = 14,
  axis_text_size = 14
)
```

Arguments

input
: An object of 'dist' class. It is a lower off-diagonal matrix with the dissimilarities of all pairs of comparisons.

optimal_link
: A character string with values "ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median", or "centroid" for the optimal linkage method, corresponding to the highest cophenetic correlation coefficient value.

label_size
: A positive integer for the font size of labels in the profile plot with the average silhouette width per candidate cluster. label_size determines the size argument found in the geom's aesthetic properties in the R-package ggplot2.

axis_title_size
: A positive integer for the font size of axis title in the profile plot with the average silhouette width per candidate cluster. axis_title_size determines the axis.title argument found in the theme's properties in the R-package ggplot2.

axis_text_size
: A positive integer for the font size of axis text in the profile plot with the average silhouette width per candidate cluster. axis_text_size determines the axis.text argument found in the theme's properties in the R-package ggplot2.

Details

internal_measures_plot also calls the function comp_clustering to define the argument optimal_link to create the silhouette plot for the selected number of clusters.

internal_measures_plot calls the silhouette function in the R-package cluster to obtain the results on average silhouette for each candidate cluster.

internal_measures_plot is integrated in the function comp_clustering.

Value

internal_measures_plot currently returns the following list of elements:

Table_internal_measures
: A data-frame of the average silhouette width for a range of 2 to P-1 clusters, with P being the number of trials

Internal_measures_panel
: A profile plot on the average silhouette width for a range of 2 to P-1 clusters, with P being the number of trials The candidate optimal number of clusters is indicated with a red point directly on the line.

Author(s)

Loukia M. Spineli

References

Handl J, Knowles J, Kell DB. Computational cluster validation in post-genomic data analysis. *Biometrics* 2005;**21**(15):3201–120. doi: 10.1093/bioinformatics/bti517

Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987;**20**:53–65.

See Also

`comp_clustering`, `silhouette`

`intervalplot_panel_ume` *A panel of interval plots for the unrelated mean effects model*

Description

Creates a panel of interval plots on the summary effect sizes under the consistency model and the unrelated mean effects model. The number of interval plots equals the number of pairwise comparisons observed in the network.

Usage

```
intervalplot_panel_ume(full, ume, drug_names)
```

Arguments

`full`	An object of S3 class `run_model`. See 'Value' in `run_model`.
`ume`	An object of S3 class `run_ume`. See 'Value' in `run_ume`.
`drug_names`	A vector of labels with the name of the interventions in the order they appear in the argument `data` of `run_model`. If the argument `drug_names` is not defined, the order of the interventions as they appear in `data` is used, instead.

Details

`intervalplot_panel_ume` is integrated in the `ume_plot` function. The consistency model and the unrelated mean effects model are abbreviated in the y-axis as 'NMA model' and 'UME model', respectively. The intervals are highlighted with green, when the corresponding summary effect sizes do not cross the vertical line of no difference, and red otherwise. Grey panels refer to the frail comparisons as detected by the `improved_ume` function (see 'Details' in `improved_ume`).

For a binary outcome, when `measure` is "RR" (relative risk) or "RD" (risk difference) in `run_model`, `intervalplot_panel_ume` currently presents the results in the odds ratio scale.

Value

A panel of interval plots on the posterior mean and 95% credible interval of the summary effect size under the consistency model and the improved unrelated mean effects model (Spineli, 2021) of all pairwise comparisons observed in the network.

Author(s)

Loukia M. Spineli

References

Spineli LM. A revised framework to evaluate the consistency assumption globally in a network of interventions. *Med Decis Making* 2021. doi: 10.1177/0272989X211068005

See Also

`improved_ume` `run_model`, `run_ume`, `ume_plot`

`kld_barplot` *Barplot for the Kullback-Leibler divergence measure*

Description

Produces a barplot with the Kullback-Leibler divergence measure from each re-analysis to the primary analysis for a pairwise comparison. Currently, `kld_barplot` is used concerning the impact of missing participant outcome data.

Usage

```
kld_barplot(robust, compar, drug_names)
```

Arguments

`robust`	An object of S3 class `robustness_index`. See 'Value' in `robustness_index`.
`compar`	A character vector with two elements that indicates the pairwise comparison of interest. The first element refers to the 'experimental' intervention and the second element refers to the 'control' intervention of the comparison.
`drug_names`	A vector of labels with the name of the interventions in the order they appear in the argument `data` of `run_model`. If `drug_names` is not defined, the order of the interventions as they appear in `data` is used, instead.

Details

kld_barplot uses the scenarios inherited by robustness_index via the run_sensitivity function. The scenarios for the missingness parameter (see 'Details' in run_sensitivity) in the compared interventions are split to *Extreme*, *Sceptical*, and *Optimistic* following the classification of Spineli et al. (2021). In each class, bars will green, orange, and red colour refer to scenarios without distance, less distant, and more distant from the primary analysis (the missing-at-random assumption).

kld_barplot can be used only when missing participant outcome data have been extracted for at least one trial. Otherwise, the execution of the function will be stopped and an error message will be printed on the R console.

Value

kld_barplot returns a panel of barplots on the Kullback-Leibler divergence measure for each reanalysis.

Author(s)

Loukia M. Spineli

References

Kullback S, Leibler RA. On information and sufficiency. *Ann Math Stat* 1951;**22**(1):79–86. doi: 10.1214/aoms/1177729694

Spineli LM, Kalyvas C, Papadimitropoulou K. Quantifying the robustness of primary analysis results: A case study on missing outcome data in pairwise and network meta-analysis. *Res Synth Methods* 2021;**12**(4):475–90. doi: 10.1002/jrsm.1478

See Also

robustness_index, run_model, run_sensitivity

Examples

```
data("pma.taylor2004")

# Read results from 'run_sensitivity' (using the default arguments)
res_sens <- readRDS(system.file('extdata/res_sens_taylor.rds',
                    package = 'rnmamod'))

# Calculate the robustness index
robust <- robustness_index(sens = res_sens,
                           threshold = 0.17)

# The names of the interventions in the order they appear in the dataset
interv_names <- c("placebo", "inositol")

# Create the barplot for the comparison 'inositol versus placebo'
kld_barplot(robust = robust,
            compar = c("inositol", "placebo"),
            drug_names = interv_names)
```

league_heatmap *League heatmap for estimation*

Description

For one outcome, it creates a heatmap of the estimated effect measure for all possible comparisons of interventions in the network. For two outcomes, the heatmap illustrates these two outcomes for the same effect measure in the upper and lower off-diagonals for all possible comparisons of interventions in the network. The function can also be used to illustrate the results of two different models on the same outcome and effect measure. league_heatmap can be used for a random-effects or fixed-effect network meta-analysis, network meta-regression, and series of pairwise meta-analyses.

Usage

```
league_heatmap(
  full1,
  full2 = NULL,
  cov_value = NULL,
  drug_names1,
  drug_names2 = NULL,
  name1 = NULL,
  name2 = NULL,
  show = NULL
)
```

Arguments

full1 An object of S3 class run_model for network meta-analysis, or run_metareg for network meta-regression. See 'Value' in run_model and run_metareg.

full2 An object of S3 class run_model for network meta-analysis, run_metareg for network meta-regression, or run_series_meta for a series of pairwise meta-analyses. See 'Value' in run_model, run_metareg, and run_series_meta.

cov_value A list of two elements in the following order: a number for the covariate value of interest and a character for the name of the covariate. See also 'Details'.

drug_names1 A vector of labels with the name of the interventions in the order they appear in the argument data of run_model for full1.

drug_names2 A vector of labels with the name of the interventions in the order they appear in the argument data of run_model for full2. The elements must be a subset of drug_names1.

name1 The text for the title of the results that refer to the outcome or model under full1.

`name2`	The text for the title of the results that refer to the outcome or model under `full2`.
`show`	A vector of at least three character strings that refer to the names of the interventions *exactly* as defined in `drug_names1`. Then, the league table will be created for these interventions only. If `show` is not defined, the league table will present all interventions as defined in `drug_names1`.

Details

`heatmap_league` offers the following options to display **one** estimated effect measure for all (or some) pairwise comparisons:

- one outcome, with results in the lower triangle referring to comparisons in the opposite direction after converting negative values into positive values (in absolute or logarithmic scale), and vice versa. Comparisons between interventions should be read from left to right. Therefore, each cell refers to the corresponding row-defining intervention against the column-defining intervention. Results that indicate strong evidence in favour of the row-defining intervention (i.e. the respective 95% credible interval does not include the null value) are indicated in bold. A message is printed on the R console on how to read the heatmap;
- two outcomes for the same model, namely, network meta-analysis (via `run_model`) or network meta-regression (via `run_metareg`). When one of the outcomes includes more interventions, the argument `full1` should be considered for that outcome. Comparisons between interventions should be read as follows: for the upper diagonal, each cell refers to the corresponding row-defining intervention against the column-defining intervention, and for the lower diagonal, each cell refers to the corresponding column-defining intervention against the row-defining intervention. Results that indicate strong evidence (i.e. the respective 95% credible interval does not include the null value) are indicated in bold. A message is printed on the R console on how to read the heatmap;
- two models for the same outcome, namely, network meta-analysis versus network meta-regression, or network meta-analysis versus series of pairwise meta-analyses. The instructions to read the heatmap are in line with the previous point. A message is printed on the R console on how to read the heatmap.

For a beneficial outcome, red favours the first intervention of the comparison, and blue favours the second intervention. For a harmful outcome, blue favours the first intervention of the comparison, and red favours the second intervention. The larger the treatment effect, the darker the colour shade.

The function displays the effect measure as inherited by the argument `full1`. For binary outcome, it can display the odds ratio, relative risk, and risk difference. See 'Details' in `run_model` for the relative risk, and risk difference. For continuous outcome, it can display the mean difference, standardised mean difference, and ratio of means. Odds ratios, relative risk and ratio of means are reported in the original scale after exponentiation of the logarithmic scale.

The rows and columns of the heatmap display the names of interventions sorted by decreasing order from the best to the worst based on their SUCRA value (Salanti et al., 2011) for the outcome or model under the argument `full1`. The off-diagonals contain the posterior median and 95% credible interval of the effect measure (according to the argument `measure` as inherited in the argument `full1`) of the corresponding comparisons.

The main diagonal contains the posterior mean of SUCRA of the corresponding interventions when the arguments `full1` refers to the `run_model` function. When the arguments `full1` refers to the run_metareg function, the p-score (Ruecker and Schwarzer, 2015) is calculated for each intervention while taking into account the covariate value in the argument cov_value. P-score is the 'frequentist analogue to SUCRA' (Ruecker and Schwarzer, 2015).

In the case of network meta-regression, when the covariate is binary, specify in the second element of cov_value the name of the level for which the heatmap will be created.

league_heatmap can be used only for a network of interventions. In the case of two interventions, the execution of the function will be stopped and an error message will be printed on the R console.

Value

A heatmap of the league table showing the posterior median and 95% credible interval of the comparisons in the off-diagonals, and the posterior mean of the SUCRA values in the diagonal.

Author(s)

Loukia M. Spineli, Chrysostomos Kalyvas, Katerina Papadimitropoulou

References

Ruecker G, Schwarzer G. Ranking treatments in frequentist network meta-analysis works without resampling methods. *BMC Med Res Methodol* 2015;**15**:58. doi: 10.1186/s12874-015-0060-8

Salanti G, Ades AE, Ioannidis JP. Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: an overview and tutorial. *J Clin Epidemiol* 2011;**64**(2):163–71. doi: 10.1016/j.jclinepi.2010.03.016

See Also

run_metareg, run_model, run_series_meta

Examples

```
data("nma.liu2013")

# Read results from 'run_model' (using the default arguments)
res <- readRDS(system.file('extdata/res_liu.rds', package = 'rnmamod'))

# The names of the interventions in the order they appear in the dataset
interv_names <- c("placebo", "pramipexole", "serotonin-norepinephrine
                  reuptake inhibitor", "serotonin reuptake inhibitor",
                  "tricyclic antidepressant", "pergolide")

# Create the league heatmap
league_heatmap(full1 = res,
               drug_names1 = interv_names)
```

`league_heatmap_pred` *League heatmap for prediction*

Description

For one outcome, it creates a heatmap with the predicted effect measure for all possible comparisons of interventions in the network. For two outcomes, the heatmap illustrates these two outcomes for the same effect measure in the upper and lower off-diagonals for all possible comparisons of interventions in the network. `league_heatmap_pred` can be used *only* for a random-effects network meta-analysis and network meta-regression.

Usage

```
league_heatmap_pred(
  full1,
  full2 = NULL,
  cov_value = NULL,
  drug_names1,
  drug_names2 = NULL,
  name1 = NULL,
  name2 = NULL,
  show = NULL
)
```

Arguments

`full1` An object of S3 class `run_model` for network meta-analysis, or `run_metareg` for network meta-regression. See 'Value' in `run_model` and `run_metareg`.

`full2` An object of S3 class `run_model` for network meta-analysis, or `run_metareg` for network meta-regression. See 'Value' in `run_model` and `run_metareg`.

`cov_value` A list of two elements in the following order: a number for the covariate value of interest and a character for the name of the covariate. See also 'Details'.

`drug_names1` A vector of labels with the name of the interventions in the order they appear in the argument `data` of `run_model` for `full1`.

`drug_names2` A vector of labels with the name of the interventions in the order they appear in the argument `data` of `run_model` for `full2`. The elements must be a subset of `drug_names1`.

`name1` The text for the title of the results that refer to the outcome or model under `full1`.

`name2` The text for the title of the results that refer to the outcome or model under `full2`.

`show` A vector of at least three character strings that refer to the names of the interventions *exactly* as defined in `drug_names1`. Then, the league table will be created for these interventions only. If `show` is not defined, the league table will present all interventions as defined in `drug_names1`.

Details

heatmap_league offers the following options to display **one** estimated effect measure for all (or some) pairwise comparisons:

- one outcome, with results in the lower triangle referring to comparisons in the opposite direction after converting negative values into positive values (in absolute or logarithmic scale), and vice versa. Darker shades of red and green correspond to larger treatment effects in the upper and lower triangle, respectively, for a beneficial outcome, and vice versa for a harmful outcome. Comparisons between interventions should be read from left to right. Therefore, each cell refers to the corresponding row-defining intervention against the column-defining intervention. Results that indicate strong evidence in favour of the row-defining intervention (i.e. the respective 95% prediction interval does not include the null value) are indicated in bold. A message is printed on the R console on how to read the heatmap;
- two outcomes for the same model, namely, network meta-analysis (via run_model) or network meta-regression (via run_metareg). When one of the outcomes includes more interventions, the argument full1 should be considered for that outcome. Comparisons between interventions should be read as follows: for the upper diagonal, each cell refers to the corresponding row-defining intervention against the column-defining intervention, and for the lower diagonal, each cell refers to the corresponding column-defining intervention against the row-defining intervention. Results that indicate strong evidence (i.e. the respective 95% prediction interval does not include the null value) are indicated in bold. A message is printed on the R console on how to read the heatmap;
- two models for the same outcome, namely, network meta-analysis versus network meta-regression. The instructions to read the heatmap are in line with the previous point. A message is printed on the R console on how to read the heatmap.

The function displays the effect measure as inherited by the argument full1. For binary outcome, it can display the odds ratio, relative risk, and risk difference. See 'Details' in run_model for the relative risk, and risk difference. For continuous outcome, it can display the mean difference, standardised mean difference, and ratio of means. Odds ratios, relative risk and ratio of means are reported in the original scale after exponentiation of the logarithmic scale.

The rows and columns of the heatmap display the names of interventions which are sorted by decreasing order from the best to the worst based on their SUCRA value (Salanti et al., 2011) for the outcome or model under the argument full1. The off-diagonals contain the posterior median and 95% prediction interval of the effect measure (according to the argument measure as inherited in the argument full1) of the corresponding comparisons.

The main diagonal contains the SUCRA values of the corresponding interventions when the argument full1 refers to the run_model function. When the argument full1 refers to the run_metareg function, the p-score (Ruecker and Schwarzer, 2015) is calculated for each intervention while taking into account the covariate value in the argument cov_value. P-score is the 'frequentist analogue to SUCRA' (Ruecker and Schwarzer, 2015).

In the case of network meta-regression, when the covariate is binary, specify in the second element of cov_value the name of the level for which the heatmap will be created.

league_heatmap_pred can be used only for a network of interventions. In the case of two interventions, the execution of the function will be stopped and an error message will be printed on the R console. Similarly, when the function is executed for a fixed-effect network meta-analysis or network meta-regression.

Value

A league heatmap of the posterior median and 95% prediction interval of the effect measure (according to the argument `measure` defined in `run_model`) for all possible comparisons in the off-diagonals, and the posterior mean of the SUCRA values in the diagonal.

Author(s)

Loukia M. Spineli, Chrysostomos Kalyvas, Katerina Papadimitropoulou

References

Ruecker G, Schwarzer G. Ranking treatments in frequentist network meta-analysis works without resampling methods. *BMC Med Res Methodol* 2015;**15**:58. doi: 10.1186/s12874-015-0060-8

Salanti G, Ades AE, Ioannidis JP. Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: an overview and tutorial. *J Clin Epidemiol* 2011;**64**(2):163–71. doi: 10.1016/j.jclinepi.2010.03.016

See Also

`run_metareg`, `run_model`

Examples

```
data("nma.liu2013")

# Read results from 'run_model' (using the default arguments)
res <- readRDS(system.file('extdata/res_liu.rds', package = 'rnmamod'))

# The names of the interventions in the order they appear in the dataset
interv_names <- c("placebo", "pramipexole", "serotonin-norepinephrine
                  reuptake inhibitor", "serotonin reuptake inhibitor",
                  "tricyclic antidepressant", "pergolide")

# Create the league heatmap
league_heatmap_pred(full1 = res,
                    drug_names1 = interv_names)
```

`league_table_absolute` *League table for relative and absolute effects*

Description

Provides a league table of the estimated odds ratio, and risk difference per 1000 participants for all possible comparisons of interventions in the network. The main diagonal of the table presents the absolute risk for each intervention in the network. `league_table_absolute` can be used for a random-effects or fixed-effect network meta-analysis. This function should be used when the user has access to the raw trial-level data (one-trial-per-row format with arm-level data). `league_table_absolute` is applied for one binary outcome only.

Usage

```
league_table_absolute(full, drug_names, show = NULL)
```

Arguments

full	An object of S3 class run_model. See 'Value' in run_model.
drug_names	A vector of labels with the name of the interventions in the order they appear in the argument data of run_model.
show	A vector of at least three character strings that refer to the names of the interventions *exactly* as defined in drug_names. Then, the league table will be created for these interventions only. If show is not defined, the league table will present all interventions as defined in drug_names.

Details

The user must define the argument measure = "RD" in run_model; otherwise, the function will be stopped and an error message will be printed in the R console.

The rows and columns of the league table display the names of the interventions sorted by decreasing order from the best to the worst based on their SUCRA value (Salanti et al., 2011) for the odds ratio. The upper off-diagonals contain the posterior median and 95% credible interval of the odds ratio, the lower off-diagonals contain the posterior median and 95% credible interval of the risk difference (per 1000 participants), and the main diagonal comprises the posterior median and 95% credible interval of the absolute risks (per 1000 participants) of the corresponding interventions. The reference intervention of the network (which the baseline risk has been selected for) is indicated in the main diagonal with a black, thick frame.

Comparisons between interventions should be read from left to right. Results that indicate strong evidence in favor of the row-defining intervention (i.e. the respective 95% credible interval does not include the null value) are indicated in bold.

To obtain unique absolute risks for each intervention, the network meta-analysis model has been extended to incorporate the transitive risks framework, namely, an intervention has the same absolute risk regardless of the comparator intervention(s) in a trial (Spineli et al., 2017). The absolute risks are a function of the odds ratio (the **base-case** effect measure for a binary outcome) and the selected baseline risk for the reference intervention (Appendix in Dias et al., 2013). See 'Arguments' in run_model. We advocate using the odds ratio as an effect measure for its desired mathematical properties. Then, the risk difference can be obtained as a function of the absolute risks of the corresponding interventions in the comparison of interest.

league_table_absolute can be used only for a network of interventions. In the case of two interventions, the execution of the function will be stopped and an error message will be printed in the R console.

Value

A league table showing the posterior estimate and 95% credible interval of the odds ratio (upper off-diagonals), risk difference per 1000 participants (lower off-diagonals), and absolute risks per 1000 participants (main diagonal).

Author(s)

Loukia M. Spineli

References

Salanti G, Ades AE, Ioannidis JP. Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: an overview and tutorial. *J Clin Epidemiol* 2011;**64**(2):163–71. doi: 10.1016/j.jclinepi.2010.03.016

Spineli LM, Brignardello-Petersen R, Heen AF, Achille F, Brandt L, Guyatt GH, et al. Obtaining absolute effect estimates to facilitate shared decision making in the context of multiple-treatment comparisons. Abstracts of the Global Evidence Summit, Cape Town, South Africa. *Cochrane Database of Systematic Reviews* 2017;**9**(Suppl 1):1891.

See Also

`run_model`

`league_table_absolute_user` *League table for relative and absolute effects (user defined)*

Description

In line with `league_table_absolute`, provides a league table of the estimated odds ratio, and risk difference per 1000 participants for all possible comparisons of interventions in the network. The main diagonal of the table presents the absolute risk for each intervention in the network. `league_table_absolute_user` requires users to input the summary effect and 95% credible or confidence interval of the basic parameters in the reported effect measure. This function should be used when the user has access to the results of a published systematic review rather than the raw trial-level data. In the latter case, the user should consider the function `league_table_absolute`. `league_table_absolute_user` is applied for one binary outcome only.

Usage

```
league_table_absolute_user(
  data,
  measure,
  base_risk,
  drug_names,
  show = NULL,
  save_xls
)
```

Arguments

`data` A data-frame with the summary effects of comparisons with the reference intervention of the network, known as basic parameters. The data-frame has `T` rows (`T` is the number of interventions in the network) and four columns that contain the point estimate, the lower and upper bound of the 95% (confidence or credible) interval of the corresponding basic parameters, and a ranking measure to indicate the order of the interventions in the hierarchy from the best to the worst with possible choices a non-zero positive integer for the rank, the SUCRA value (Salanti et al., 2011) or p-score value (Ruecker and Schwarzer, 2015). The first row of the data-frame refers to the selected reference intervention and should include (1) the null value three times at the investigated effect measure (i.e. 1 for odds ratio and relative risk, and 0 for risk difference), and (2) the value of the ranking measure.

`measure` Character string indicating the effect measure of `data`. For a binary outcome, the following can be considered: `"OR"`, `"RR"` or `"RD"` for the odds ratio, relative risk, and risk difference, respectively.

`base_risk` A number in the interval (0, 1) that indicates the baseline risk for the selected reference intervention.

`drug_names` A vector of labels with the name of the interventions in the order they appear in the argument `data`. The first intervention should be the selected reference intervention.

`show` A vector of at least three character strings that refer to the names of the interventions *exactly* as defined in `drug_names`. Then, the league table will be created for these interventions only. If `show` is not defined, the league table will present all interventions as defined in `drug_names`.

`save_xls` Logical to indicate whether to export the tabulated results to an 'xlsx' file (via the `write_xlsx` function of the R-package writexl) to the working directory of the user. The default is `FALSE` (do not export).

Details

When the published results are reported in the relative risk scale (i.e., `measure = "RR"`), the function calculates odds ratios and risk differences (point estimate and 95% confidence interval) for all possible pairwise comparisons in the network based on the obtained absolute risks and the selected baseline risk. Likewise, when the published results are in the odds ratio or risk difference scale (i.e., `measure = "OR"` or `measure = "RD"`, respectively), the function calculates risk differences or odds ratios (point estimate and 95% confidence interval), respectively, for all possible pairwise comparisons in the network based on the obtained absolute risks and the selected baseline risk.

The rows and columns of the league table display the names of the interventions sorted by decreasing order from the best to the worst based on the ranking measure in the fourth column of the argument `data`. The upper off-diagonals contain the estimate and 95% confidence interval of the odds ratio, the lower off-diagonals contain the estimate and 95% confidence interval of the risk difference (per 1000 participants), and the main diagonal comprises the absolute risks and their 95% confidence interval (per 1000 participants) of the corresponding non-reference interventions. The reference intervention of the network (which the baseline risk has been selected for) is indicated in the main diagonal with a black, thick frame.

Comparisons between interventions should be read from left to right. Results that indicate strong evidence in favour of the row-defining intervention (i.e. the respective 95% confidence interval does not include the null value) are indicated in bold.

Furthermore, `league_table_absolute_user` exports `table_relative_absolute_effect`, a table with the relative and absolute effects of the basic parameters, as an 'xlsx' file (via the `write_xlsx` function) to the working directory of the user.

To obtain unique absolute risks for each intervention, we have considered the transitive risks framework, namely, an intervention has the same absolute risk regardless of the comparator intervention(s) in a trial (Spineli et al., 2017).

`league_table_absolute_user` can be used only for a network of interventions. In the case of two interventions, the execution of the function will be stopped and an error message will be printed in the R console.

Value

A league table showing the estimate and 95% confidence interval of the odds ratio (upper off-diagonals), risk difference per 1000 participants (lower off-diagonals), and absolute risks per 1000 participants (main diagonal).

Author(s)

Loukia M. Spineli

References

Ruecker G, Schwarzer G. Ranking treatments in frequentist network meta-analysis works without resampling methods. *BMC Med Res Methodol* 2015;**15**:58. doi: 10.1186/s12874-015-0060-8

Salanti G, Ades AE, Ioannidis JP. Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: an overview and tutorial. *J Clin Epidemiol* 2011;**64**(2):163–71. doi: 10.1016/j.jclinepi.2010.03.016

Spineli LM, Brignardello-Petersen R, Heen AF, Achille F, Brandt L, Guyatt GH, et al. Obtaining absolute effect estimates to facilitate shared decision making in the context of multiple-treatment comparisons. Abstracts of the Global Evidence Summit, Cape Town, South Africa. *Cochrane Database of Systematic Reviews* 2017;**9**(Suppl 1):1891.

See Also

`league_table_absolute`, `write_xlsx`

`leverage_plot` *Leverage plot*

Description

Plots the leverage against the square root of the posterior mean of residual deviance of the trial-arms under the model of interest.

Usage

```
leverage_plot(net, drug_names, title)
```

Arguments

`net`	An object of S3 class `run_metareg`, `run_model`, or `run_ume`. See 'Value' in `run_metareg`, `run_model`, or `run_ume`.
`drug_names`	A vector of labels with the name of the interventions in the order they appear in the argument `data` of `run_model`. If `drug_names` is not defined, the order of the interventions as they appear in `data` is used, instead.
`title`	A title to indicate the model (consistency model, network meta-regression or unrelated mean effects model).

Details

`leverage_plot` is integrated in the `ume_plot` function to create the leverage plot for the consistency model and the unrelated mean effects model. These plots appear side-by-side in the output of `ume_plot`. Dias et al. (2010) used leverage plots to investigate the fit of the consistency and inconsistency models–the latter through the node-splitting approach.

Value

A scatterplot of the leverage against the square root of the posterior mean of residual deviance of the trial-arms under the model of interest. The green, yellow, and red curves correspond to the parabola $x^2 + y = k$ with k = 1, 2, and 3, respectively. The data points correspond to trial-arms. Data points found outside the yellow parabola are linked with a pair of numbers. The first number refers to the position of the trial in the dataset, and the second number refers to the corresponding trial-arm (see 'Arguments' and 'Value' in `data_preparation`). These trial-arms contribute more than 1 to the deviance information criterion and, hence, the model's poor fit.

Author(s)

Loukia M. Spineli

References

Dias S, Welton NJ, Caldwell DM, Ades AE. Checking consistency in mixed treatment comparison meta-analysis. *Stat Med* 2010;**29**(7-8):932–44. doi: 10.1002/sim.3767

See Also

`data_preparation`, `run_metareg`, `run_model`, `run_ume`, `ume_plot` `mcmc_diagnostics` *Markov Chain Monte Carlo diagnostics*

Description

Evaluates whether convergence has been achieved for the monitored parameters of the Bayesian models. The Gelman-Rubin convergence diagnostic, the Markov Chain Monte Carl (MCMC) error and relevant diagnostic plots are applied.

Usage

```
mcmc_diagnostics(net, par = NULL)
```

Arguments

net	An object of S3 class run_metareg, run_model, run_nodesplit, run_sensitivity, run_series_meta, and run_ume. See 'Value' in the functions above.
par	A vector of at least one character string that refers to the monitored parameters in jagsfit which is an object of S3 class run_metareg, run_model, and run_ume. The selected parameters will be considered in the diagnostic plots (see 'Value'). This argument will be ignored for objects of S3 class run_nodesplit, run_sensitivity, and run_series_meta.

Details

For each monitored parameter, mcmc_diagnostics considers the R-hat and MCMC error and compares them with the thresholds 1.1 and 5% of the posterior standard deviation (the rule of thumb), respectively. Convergence is achieved for the monitored parameter, when the R-hat is below the corresponding threshold. Visual inspection of the trace plots and posterior density of the monitored parameters should also be considered when drawing conclusions about convergence.

Value

mcmc_diagnostics considers the following monitored parameters:

EM	The estimated summary effect measure.
EM_pred	The predicted summary effect measure.
delta	The estimated trial-specific effect measure.
tau	The between-trial standard deviation.
direct	The direct estimate of the split node (see 'Value' in run_nodesplit).
indirect	The indirect estimate of the split node (see 'Value' in run_nodesplit).
diff	The inconsistency factor of the split node (see 'Value' in run_nodesplit).
phi	The informative missingness parameter.
beta	The regression coefficient.

For each monitored parameter mentioned above, mcmc_diagnostics also returns a barplot on the ratio of MCMC error to the posterior standard deviation and a barplot on the Gelman-Rubin R diagnostic. Bars that correspond to a ratio less than 5% are indicated in green (the corresponding parameters have been estimated accurately); otherwise, the bars are indicated in red (inaccurate estimation). Furthermore, bars that correspond to an R value less than 1.10 are indicated in green (the corresponding parameters have been converged); otherwise, the bars are indicated in red (convergence is not achieved). mcmc_diagnostics returns histograms than barplots for EM when run_sensitivity is considered.

mcmc_diagnostics also uses the mcmcplot function of the R-package mcmcplots to create an HTML file with a panel of diagnostic plots (trace, density, and autocorrelation) for each monitored parameter.

Author(s)

Loukia M. Spineli

References

Gelman, A, Rubin, DB. Inference from iterative simulation using multiple sequences. *Stat Sci* 1992;**7**(4):457–72. doi: 10.1214/ss/1177011136

See Also

mcmcplot, run_metareg, run_model, run_nodesplit, run_sensitivity, run_series_meta, run_ume

Examples

```
data("nma.baker2009")

# Read results from 'run_nodesplit' (using the default arguments)
res <- readRDS(system.file('extdata/node_baker.rds', package = 'rnmamod'))

# Check convergence based on R-hat
mcmc_diagnostics(net = res,
                 par = c("tau", "EM[2,1]", "EM.pred[2,1]"))
```

metareg_plot *End-user-ready results for network meta-regression*

Description

Illustrates the effect estimates, predictions and regression coefficients of comparisons with a specified comparator intervention for a selected covariate value and also exports these results to an Excel file.

Usage

```
metareg_plot(full, reg, compar, cov_value, drug_names, save_xls)
```

Arguments

`full`	An object of S3 class `run_model`. See 'Value' in `run_model`.
`reg`	An object of S3 class `run_metareg`. See 'Value' in `run_metareg`.
`compar`	A character to indicate the comparator intervention. It must be any name found in `drug_names`.
`cov_value`	A list of two elements in the following order: a number for the covariate value of interest (see 'Arguments' in `run_metareg`), and a character to indicate the name of the covariate. See also 'Details'.
`drug_names`	A vector of labels with the name of the interventions in the order they appear in the argument `data` of `run_model`. If `drug_names` is not defined, the order of the interventions as they appear in `data` is used, instead.
`save_xls`	Logical to indicate whether to export the tabulated results to an 'xlsx' file (via the `write_xlsx` function of the R-package writexl) at the working directory of the user. The default is `FALSE` (do not export).

Details

The deviance information criterion (DIC) of the network meta-analysis model is compared with the DIC of the network meta-regression model. If the difference in DIC exceeds 5, the network meta-regression model is preferred; if the difference in DIC is less than -5, the network meta-analysis model is preferred; otherwise, there is little to choose between the compared models.

When the covariate is binary, specify in the second element of `cov_value` the name of the level for which the output will be created.

Furthermore, `metareg_plot` exports all tabulated results to separate 'xlsx' files (via the `write_xlsx` function of the R-package writexl) to the working directory of the user.

`metareg_plot` can be used only for a network of interventions. In the case of two interventions, the execution of the function will be stopped and an error message will be printed on the R console.

Value

`metareg_plot` prints on the R console a message on the most parsimonious model (if any) based on the DIC (in red text). Furthermore, the function returns the following list of elements:

`table_estimates`
: The posterior median, and 95% credible interval of the summary effect measure (according to the argument `measure` defined in `run_model`) for each comparison with the selected intervention under network meta-analysis and network meta-regression based on the specified `cov_value`.

`table_predictions`
: The posterior median, and 95% prediction interval of the summary effect measure (according to the argument `measure` defined in `run_model`) for each comparison with the selected intervention under network meta-analysis and network meta-regression based on the specified `cov_value`.

`table_model_assessment`
: The DIC, total residual deviance, number of effective parameters, and the posterior median and 95% credible interval of between-trial standard deviation (*tau*)

under each model (Spiegelhalter et al., 2002). When a fixed-effect model has been performed, `metareg_plot` does not return results on *tau*. For a binary outcome, the results refer to the odds ratio scale.

`table_regression_coeffients`
: The posterior median and 95% credible interval of the regression coefficient(s) (according to the argument `covar_assumption` defined in `run_metareg`). For a binary outcome, the results refer to the odds ratio scale.

`interval_plot`
: A forest plot on the estimated and predicted effect sizes of comparisons with the selected comparator intervention under network meta-analysis and network meta-regression based on the specified `cov_value` alongside a forest plot with the corresponding SUCRA values. See 'Details' and 'Value' in `forestplot_metareg`.

`sucra_scatterplot`
: A scatterplot of the SUCRA values from the network meta-analysis against the SUCRA values from the network meta-regression based on the specified `cov_value`. See 'Details' and 'Value' in `scatterplot_sucra`.

Author(s)

Loukia M. Spineli

References

Salanti G, Ades AE, Ioannidis JP. Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: an overview and tutorial. *J Clin Epidemiol* 2011;**64**(2):163–71. doi: 10.1016/j.jclinepi.2010.03.016

Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A. Bayesian measures of model complexity and fit. *J R Stat Soc B* 2002;**64**(4):583–616. doi: 10.1111/1467-9868.00353

See Also

`forestplot_metareg`, `run_metareg`, `run_model`, `scatterplot_sucra`, `write_xlsx`

Examples

```
data("nma.baker2009")


# Read results from 'run_model' (using the default arguments)
res <- readRDS(system.file('extdata/res_baker.rds', package = 'rnmamod'))

# Read results from 'run_metareg' (exchangeable structure)
reg <- readRDS(system.file('extdata/reg_baker.rds', package = 'rnmamod'))

# Publication year as the covariate
pub_year <- c(1996, 1998, 1999, 2000, 2000, 2001, rep(2002, 5), 2003, 2003,
              rep(2005, 4), 2006, 2006, 2007, 2007)

# The names of the interventions in the order they appear in the dataset
interv_names <- c("placebo", "budesonide", "budesonide plus formoterol",
                  "fluticasone", "fluticasone plus salmeterol",
                 "formoterol", "salmeterol", "tiotropium")

# Plot the results from both models for all comparisons with salmeterol and
# publication year 2000
metareg_plot(full = res,
             reg = reg,
             compar = "salmeterol",
             cov_value = list(2000, "publication year"),
             drug_names = interv_names)
```

missingness_param_prior *Define the mean value of the normal distribution of the missingness parameter*

Description

Generates the mean value of the normal distribution of the missingness parameter in the proper format depending on the assumed structure of the missingness parameter. run_model inherits missingness_param_prior through the argument mean_misspar (see 'Argument' in run_model).

Usage

```
missingness_param_prior(assumption, mean_misspar)
```

Arguments

assumption
: Character string indicating the structure of the informative missingness parameter. Set assumption equal to one of the following: "HIE-COMMON", "HIE-TRIAL", "HIE-ARM", "IDE-COMMON", "IDE-TRIAL", "IDE-ARM", "IND-CORR", or "IND-UNCORR". The default argument is "IDE-ARM". The abbreviations "IDE", "HIE", and "IND" stand for identical, hierarchical and independent, respectively. "CORR" and "UNCORR" stand for correlated and uncorrelated, respectively.

mean_misspar
: A numeric value or a vector of two numeric values for the mean of the normal distribution of the informative missingness parameter (see 'Details'). The default argument is 0 and corresponds to the missing-at-random assumption for assumption = "IDE-ARM".

Details

run_model considers the informative missingness odds ratio in the logarithmic scale for binary outcome data (Spineli, 2019a; Turner et al., 2015; White et al., 2008), the informative missingness difference of means when measure is "MD" or "SMD", and the informative missingness ratio of means in the logarithmic scale when measure is "ROM" (Spineli et al., 2021; Mavridis et al., 2015).

When assumption is trial-specific (i.e., "IDE-TRIAL" or "HIE-TRIAL"), or independent (i.e., "IND-CORR" or "IND-UNCORR"), only one numeric value can be assigned to mean_misspar because the same missingness scenario is applied to all trials and trial-arms of the dataset, respectively. When `assumption` is `"IDE-ARM"` or `"HIE-ARM"`, a maximum of two *different or identical* numeric values can be assigned as a vector to `mean_misspars`: the first value refers to the experimental arm, and the second value refers to the control arm of a trial. In the case of a network, the first value is considered for all non-reference interventions and the second value is considered for the reference intervention of the network (see 'Argument' `ref` in `run_model`). This is necessary to ensure transitivity in the assumptions for the missingness parameter across the comparisons in the network (Spineli, 2019b).

Currently, there are no empirically-based prior distributions for the informative missingness parameters. The users may refer to Mavridis et al. (2015) and Spineli (2019) to determine `mean_misspar` for an informative missingness parameter.

Value

A scalar or numeric vector to be passed to `run_model`.

Author(s)

Loukia M. Spineli

References

Mavridis D, White IR, Higgins JP, Cipriani A, Salanti G. Allowing for uncertainty due to missing continuous outcome data in pairwise and network meta-analysis. *Stat Med* 2015;**34**(5):721–41. doi: 10.1002/sim.6365

Spineli LM, Kalyvas C, Papadimitropoulou K. Continuous(ly) missing outcome data in network meta-analysis: a one-stage pattern-mixture model approach. *Stat Methods Med Res* 2021;**30**(4):958–75. doi: 10.1177/0962280220983544

Spineli LM. An empirical comparison of Bayesian modelling strategies for missing binary outcome data in network meta-analysis. *BMC Med Res Methodol* 2019a;**19**(1):86. doi: 10.1186/s12874-019-0731-y

Spineli LM. Modeling missing binary outcome data while preserving transitivity assumption yielded more credible network meta-analysis results. *J Clin Epidemiol* 2019b;**105**:19–26. doi: 10.1016/j.jclinepi.2018.09.002

Turner NL, Dias S, Ades AE, Welton NJ. A Bayesian framework to account for uncertainty due to missing binary outcome data in pairwise meta-analysis. *Stat Med* 2015;**34**(12):2062–80. doi: 10.1002/sim.6475

White IR, Higgins JP, Wood AM. Allowing for uncertainty due to missing data in meta-analysis–part 1: two-stage methods. *Stat Med* 2008;**27**(5):711–27. doi: 10.1002/sim.3008

See Also

`run_model`

miss_characteristics *Visualising missing data in characteristics (Comparisons' comparability for transitivity evaluation)*

Description

miss_characteristics hosts a set of visualisation tools to assess the size and pattern of missing characteristics values in the dataset.

Usage

```
miss_characteristics(
  input,
  drug_names,
  rename_char = NULL,
  label_size = 4,
  axis_title_size = 14,
  axis_text_size = 14,
  axis_x_text_angle = 0,
  legend_text_size = 14,
  legend_title_size = 14,
  strip_text_size = 14,
  strip_text_angle = 0
)
```

Arguments

input
A data-frame in the long arm-based format. Two-arm trials occupy one row in the data-frame. Multi-arm trials occupy as many rows as the number of possible comparisons among the interventions. The first two columns refer to the trial name, and the pairwise comparison, respectively. The remaining columns refer to summary characteristics. See 'Details' for the specification of the columns.

drug_names
A vector of labels with the name of the interventions in the order they appear in the argument input.

rename_char
A list of two elements: (i) a numeric vector with the position of the characteristics in input, and (ii) a character vector with the names of the characteristics, as they are wished to appear in the title of the plots. This argument is optional, in case the user wants to control the appearance of the titles.

label_size
A positive integer for the font size of labels in the plots. label_size determines the size argument found in the geom's aesthetic properties in the R-package ggplot2).

axis_title_size
A positive integer for the font size of axis titles in the plots. axis_title_size determines the axis.title argument found in the theme's properties in the R-package ggplot2).

`axis_text_size` A positive integer for the font size of axis text in the plots. `axis_text_size` determines the axis.text argument found in the theme's properties in the R-package ggplot2).

`axis_x_text_angle`
A positive integer for the angle of axis text in plots related to missing data. `axis_text_angle` determines the axis.text.x argument found in the theme's properties in the R-package ggplot2).

`legend_text_size`
A positive integer for the font size of legend text in the plots. `legend_text_size` determines the legend.text argument found in the theme's properties in the R-package ggplot2).

`legend_title_size`
A positive integer for the font size of legend title in the plots. `legend_title_size` determines the legend.title argument found in the theme's properties in the R-package ggplot2).

`strip_text_size`
A positive integer for the font size of strip text in the plots. `strip_text_size` determines the strip.text argument found in the theme's properties in the R-package ggplot2).

`strip_text_angle`
A positive integer for the angle of strip text in the plots. `strip_text_angle` determines the strip.text argument found in the theme's properties in the R-package ggplot2).

Details

The correct type mode of columns in `input` must be ensured to use the function `miss_characteristics`. The first two columns referring to the trial name, and pairwise comparison, respectively, must be **character**. The remaining columns referring to the characteristics must be **double** or **integer** depending on whether the corresponding characteristic refers to a quantitative or qualitative variable. The type mode of each column is assessed by `miss_characteristics` using the base function `typeof`.

The interventions should be sorted in an ascending order of their identifier number within the trials so that the first treatment column (second column in `input`) is the control arm for every pairwise comparison. This is important to ensure consistency in the order of interventions within the comparisons obtained from the other related functions.

Value

`miss_characteristics` returns the following list of elements:

`Barplot_missing_combined`
A panel of barplots on the percentage of missing and observed cases for each comparison and characteristic.

`Barplot_missing_characteristics`
A barplot on the percentage of missing and observed cases for each comparison.

`Tileplot_missing`
A plot that illustrates the position of missing cases for each trial, comparison and characteristic.

Author(s)

Loukia M. Spineli

Examples

```
# Fictional dataset
set.seed(13022024)
data_set <- data.frame(Trial_name = as.character(1:(5 + 7 +2)),
                       arm1 = rep(c("1", "2"), c(5 + 7, 2)),
                       arm2 = rep(c("2", "3"), c(5, 7 + 2)),
                       sample = as.numeric(sample(c(50:300), 5 + 7 + 2)),
                       age = as.numeric(sample(c(18:50), 5 + 7 + 2)),
                       blinding = factor(rep(c("yes", "no", "yes"), c(5, 7, 2))))
data_set$age[data_set$age < 30] <- NA

miss_characteristics(input = data_set,
                      drug_names = c("A", "B", "C"))
```

---

netplot *Network plot*

Description

Illustrates the network plot for one outcome.

Usage

```
netplot(
  data,
  drug_names,
  show_multi = FALSE,
  multi_frame = -16,
  alpha_multi_color = 0.1,
  layout = igraph::layout_in_circle,
  node_color = "tomato",
  node_frame_color = "black",
  node_frame_width = 1,
  node_shape = "circle",
  node_label_color = "black",
  node_label_font = 1,
  node_label_cex = 1,
  node_label_dist = 0,
  edge_color = "grey50",
  edge_arrow_size = 0.5,
  edge_lty = 1,
  edge_label = NULL,
  edge_label_color = "black",
  edge_label_font = 1,
  edge_label_cex = 2,
  edge_curved = 0,
  direction = FALSE,
  ...
)
```

Arguments

data
: A data-frame of a one-trial-per-row format containing arm-level data of each trial. See 'Format' in run_model.

drug_names
: A vector of labels with the name of the interventions (nodes) in the order they appear in the argument data.

show_multi
: Logical to indicate whether to colour the closed-loops informed by multi-arm trials. The default is show_multi = FALSE.

multi_frame
: A numeric scalar to determine the size of the border around the closed-loops formed by multi-arm trials. The default is -16. multi_frame determines the mark.expand argument found in the plot.igraph function of the R-package igraph).

alpha_multi_color
: A numerical scalar with range from 0 to 1 to determine the opacity of multi_frame coloured using the rainbow colour palette. The default is 0.1.

layout
: The layout specification. The default is layout = layout_in_circle to plot the nodes in a circular layout. For more information refer to the R-package igraph,

node_color
: A character or vector of characters (with length equal to the number of nodes) to indicate the colour of the nodes. The default is node_color = "tomato". node_color determines the vertex.color argument found in the plot.igraph function of the R-package igraph).

node_frame_color
: A character to indicate the colour of the frame around the nodes. The default is node_frame_color = "black". node_frame_color determines the vertex.frame.color argument found in the plot.igraph function of the R-package igraph).

node_frame_width
: A numerical scalar to indicate the width of the frame around the nodes. The default is 1. node_frame_width determines the vertex.frame.width argument found in the plot.igraph function of the R-package igraph).

node_shape
: A character to indicate the shape of the nodes. The default is node_shape = "circle". node_shape determines the vertex.shape argument found in the plot.igraph function of the R-package igraph).

node_label_color
: A character to indicate the color of the node labels. The default is node_label_color = "black". node_label_color determines the vertex.label.color argument found in the plot.igraph function of the R-package igraph).

`node_label_font`
A numerical scalar to indicate the font of the node labels. The default is 1. `node_label_font` determines the vertex.label.font argument found in the `plot.igraph` function of the R-package igraph).

`node_label_cex`
A numerical scalar to indicate the font size of the node labels. The default is 1. `node_label_cex` determines the vertex.label.cex argument found in the `plot.igraph` function of the R-package igraph).

`node_label_dist`
A numerical scale between 0 and 1 to indicate the position of the node labels relative to the node center. The default is 0, where the label is centered. `node_label_dist` determines the vertex.label.dist argument found in the `plot.igraph` function of the R-package igraph).

`edge_color`
A character or vector of characters (with length equal to the number of edges) to indicate the colour of the edges. The default is `edge_color = "grey50"`. `edge_color` determines the edge.color argument found in the `plot.igraph` function of the R-package igraph).

`edge_arrow_size`
A numerical scalar between 0 and 1 to indicate the arrow size. The default is 0.5. `edge_arrow_size` determines the edge.arrow.size argument found in the `plot.igraph` function of the R-package igraph). This argument work only when `direction = FALSE`.

`edge_lty`
A numerical scalar, discrete with values from 0 to 6 to indicate the line type of the edges. The default is 1 (solid). `edge_lty` determines the edge.lty argument found in the `plot.igraph` function of the R-package igraph).

`edge_label`
A vector of number of characters with length equal to the number of edges to present the edge label. The default is `edge_label = NULL` and refers to the number of studies investigating the corresponding comparisons. `edge_label` determines the edge.label argument found in the `plot.igraph` function of the R-package igraph).

`edge_label_color`
A character to indicate the color of the edge labels. The default is `edge_label_color = "black"`. `edge_label_color` determines the edge.label.color argument found in the `plot.igraph` function of the R-package igraph).

`edge_label_font`
A numerical scalar to indicate the font of the edge labels. The default is 1. `edge_label_font` determines the edge.label.font argument found in the `plot.igraph` function of the R-package igraph).

`edge_label_cex`
A numerical scalar to indicate the font size of the edge labels. The default is 2. `edge_label_cex` determines the edge.label.cex argument found in the `plot.igraph` function of the R-package igraph).

`edge_curved`
A numerical scalar with range from 0 to 1 that indicates the edge curvature. The default is 0 (no curvature). `edge_curved` determines the edge.curved argument found in the `plot.igraph` function of the R-package igraph).

`direction`
Logical to draw (TRUE) or not (FALSE) arrow for each edge according to each direction. The default is `direction = FALSE`. For more information refer to the R-package igraph).

... Further graphical arguments of the plot.igraph function of the R-package igraph

Details

The edge thickness is proportional to the number of trials investigating the corresponding comparison. The node size is weighted by the total sample size of the corresponding intervention.

The user can control many of the arguments found in the plot.igraph function of the R-package igraph), such as the colour of the nodes and edges, the node and edge label size, and so on.

Value

A network plot with coloured closed-loops informed by multi-arm trials. Each node indicates an intervention and each edge an observed pairwise comparison.

Author(s)

Loukia M. Spineli

See Also

plot.igraph, run_model

Examples

```
data("nma.bottomley2011")

# Return the first six trials of the dataset
head(nma.bottomley2011)

# The names of the interventions in the order they appear in the dataset
interv_names <- c("betamethasone dipropionate", "betamethasone valerate",
                  "calcipotriol", "calcipotriol plus polytar", "capasal",
                  "two-compound formulation gel", "placebo")

# Create the network plot
netplot(data = nma.bottomley2011,
        drug_names = interv_names,
        edge_label_cex = 1)
```

nma.baker2009 *Pharmacological interventions for chronic obstructive pulmonary disease*

Description

A dataset of 21 trials comparing seven pharmacologic interventions with each other and placebo in patients with chronic obstructive pulmonary disease (COPD). The exacerbation of COPD is the analysed binary outcome.

Usage

```
data(nma.baker2009)
```

Format

A data-frame with 21 rows of arm-based data and 17 columns (maximum number of 4 arms).

Details

The interventions have been coded as follows: 1, placebo; 2, budesonide; 3, budesonide plus formoterol; 4, fluticasone; 5, fluticasone plus salmeterol; 6, formoterol; 7, salmeterol; and 8, tiotropium.

Source

Baker WL, Baker EL, Coleman CI. Pharmacologic treatments for chronic obstructive pulmonary disease: a mixed-treatment comparison meta-analysis. *Pharmacotherapy* 2009;**29**(8):891–905. doi: 10.1592/phco.29.8.891

`nma.bottomley2011` *Pharmacological interventions for moderately severe scalp psoriasis*

Description

A dataset of 9 trials comparing six pharmacologic interventions with each other and placebo for moderately severe scalp psoriasis. The analysed binary outcome is the investigator's global assessment response at 4 weeks.

Usage

```
data(nma.bottomley2011)
```

Format

A data frame with 9 rows of arm-based data and 17 columns (maximum number of 4 arms).

Details

The interventions have been coded as follows: 1, betamethasone dipropionate; 2, betamethasone valerate; 3, calcipotriol; 4, calcipotriol plus polytar; 5, capasal; 6, two-compound formulation gel; and 7, placebo.

Source

Bottomley JM, Taylor RS, Ryttov J. The effectiveness of two-compound formulation calcipotriol and betamethasone dipropionate gel in the treatment of moderately severe scalp psoriasis: a systematic review of direct and indirect evidence. *Curr Med Res Opin* 2011;**27**(1):251–68. doi: 10.1185/03007995.2010.541022

nma.dogliotti2014 *Oral antithrombotics for stroke episode*

Description

A dataset of 16 trials comparing seven oral antithrombotics with each other and placebo in patients with atrial fibrillation. The analysed binary outcome is prevention of a stroke episode.

Usage

```
data(nma.dogliotti2014)
```

Format

A data frame with 16 rows of arm-based data and 13 columns (maximum number of 3 arms).

Details

The interventions have been coded as follows: 1, placebo; 2, aspirin; 3, aspirin plus clopidogrel; 4, dabigatran 110 mg; 5, dabigatran 150 mg; 6, rivaroxaban; 7, vitamin K antagonist; and 8, apixaban

Source

Dogliotti A, Paolasso E, Giugliano RP. Current and new oral antithrombotics in non-valvular atrial fibrillation: a network meta-analysis of 79808 patients. *Heart* 2014;**100**(5):396–405. doi: 10.1136/heartjnl-2013-304347

nma.liu2013 *Antidepressants in Parkinson's disease*

Description

A dataset of 11 trials comparing the effectiveness of five antidepressants and placebo in Parkinson's disease. The analysed binary outcome is the number of patients with a reduction of at least 50/ the baseline score.

Usage

```
data(nma.liu2013)
```

Format

A data frame with 11 rows of arm-based data and 13 columns (maximum number of 3 arms).

Details

The interventions have been coded as follows: 1, placebo; 2, pramipexole; 3, serotonin-norepinephrine reuptake inhibitor; 4, selective serotonin reuptake inhibitor; 5, tricyclic antidepressant; and 6, pergolide.

Source

Liu J, Dong J, Wang L, Yan P, Sun S. Comparative efficacy and acceptability of antidepressants in Parkinson's disease: a network meta-analysis. *PLoS One* 2013;**8**(10):e76651. doi: 10.1371/journal.pone.0076651

`nma.schwingshackl2014` *Training modalities for patients with type 2 diabetes*

Description

A dataset of 14 trials comparing three different training modalities (a triangle network) for patients with type 2 diabetes. The analysed continuous outcome is change from baseline in HbA1c levels.

Usage

```
data(nma.schwingshackl2014)
```

Format

A data frame with 14 rows of arm-based data and 16 columns (maximum number of 3 arms).

Details

The interventions have been coded as follows: 1, aerobic; 2, resistance; and 3, combined training;

Source

Schwingshackl L, Missbach B, Dias S, et al. M Impact of different training modalities on glycaemic control and blood lipids in patients with type 2 diabetes: a systematic review and network meta-analysis. *Diabetologia* 2014;**57**(9):1789–97. doi: 10.1007/s00125-014-3303-z

nma.stowe2011 *Antiparkinsonian interventions for later Parkinson's disease*

Description

A dataset of 29 trials comparing three antiparkinsonian interventions with placebo (a star-shaped network) in patients with later Parkinson's disease. The analysed continuous outcome is the change from baseline of patient off-time reduction.

Usage

```
data(nma.stowe2011)
```

Format

A data frame with 29 rows of arm-based data and 11 columns (maximum number of 2 arms).

Details

The interventions have been coded as follows: 1, placebo plus levodopa (PBO+LD); 2, dopamine agonist plus levodopa (DA+LD); 3, catechol-O-methyl transferase inhibitors plus levodopa (COMBI+LD); and 4, monoamine oxidase type B inhibitors plus levodopa (MAOBI+LD).

Source

Stowe R, Ives N, Clarke CE, et al. Meta-analysis of the comparative efficacy and safety of adjuvant treatment to levodopa in later Parkinson's disease. *Mov Disord* 2011;**26**(4):587–98. doi: 10.1002/mds.23517

nodesplit_plot *End-user-ready results for the node-splitting approach*

Description

nodesplit_plot hosts a toolkit of functions that facilitates the comparison of the consistency model (via run_model) with the node-splitting approach (via run_nodesplit) regarding the posterior summaries of the direct and indirect effects and inconsistency factor of the split nodes, the between-trial standard deviation and model assessment parameters (Spiegelhalter et al., 2002) after each split node in the network.

Usage

```
nodesplit_plot(full, node, drug_names, save_xls)
```

Arguments

`full`	An object of S3 class `run_model`. See 'Value' in `run_model`.
`node`	An object of S3 class `run_nodesplit`. See 'Value' in `run_nodesplit`.
`drug_names`	A vector of labels with the name of the interventions in the order they appear in the argument `data` of `run_model`. If `drug_names` is not defined, the order of the interventions as they appear in `data` is used, instead.
`save_xls`	Logical to indicate whether to export the tabulated results to an 'xlsx' file (via the `write_xlsx` function of the R-package writexl) at the working directory of the user. The default is `FALSE` (do not export).

Details

`intervalplot_inconsistency_factor` includes as many interval plots as the number of split nodes in the network. Each interval plot illustrates the posterior median and 95% credible interval of the direct and indirect effect of the split nodes and the corresponding inconsistency factor. The line that corresponds to the inconsistency factor is highlighted with green, when it does not cross the vertical line of no difference (between the direct and indirect effect), and red otherwise. If there are more than 30 split nodes, the function presents the interval plots on split nodes with conclusive inconsistency factor (green intervals) or those with an opposite sign in the direct and indirect effects.

`intervalplot_tau` is an interval plot on the median and 95% credible interval of *tau* after each split node. The lines that correspond to the split nodes are sorted in ascending order of the deviance information criterion (DIC) which appears at the top of each line. The estimated median and 95% credible intervals of *tau* under the consistency model appear in the interval plot as a solid and two dotted parallel blue lines, respectively. The different levels of heterogeneity appear as green, yellow, orange, and red rectangulars to indicate a low, reasonable, fairly high, and fairly extreme heterogeneity, respectively, following the classification of Spiegelhalter et al. (2004). When a fixed-effect model has been performed, `nodesplit_plot` does not return the `intervalplot_tau`.

`table_model_assessment` also includes the column *DIC-based better fit* that indicates the preferred model in terms of parsimony for each split node. Therefore, the DIC of the model after each split node is compared with the DIC of the consistency model (Dias et al., 2010). If the difference in DIC exceeds 5, the consistency model is preferred; if the difference in DIC is less than -5, the model after the split node is preferred; otherwise, there is little to choose between the compared models.

For a binary outcome, when `measure` is "RR" (relative risk) or "RD" (risk difference) in `run_model`, `nodesplit_plot` currently presents the results in the odds ratio scale. This is because, the odds ratio is used as the 'best-case' effect measure in `run_model`. Then, relative risk, and risk difference are obtained as a function of the odds ratio and the selected baseline risk (See 'Details' in `run_model`).

The split nodes have been automatically selected via the `mtc.nodesplit.comparisons` function of the R-package gemtc. See 'Details' in `run_nodesplit`.

Furthermore, `nodesplit_plot` exports both data-frames to separate 'xlsx' files (via the `write_xlsx` function of the R-package writexl) to the working directory of the user.

`nodesplit_plot` can be used only for a network of interventions and when there is at least one split node. Otherwise, the execution of the function will be stopped and an error message will be printed on the R console.

Value

nodesplit_plot returns the following list of elements:

table_effect_size
: A data-frame with the posterior median, posterior standard deviation and 95% credible interval of the direct and indirect effect and the inconsistency factor of each split node.

table_model_assessment
: A data-frame with the model assessment parameters (DIC, posterior mean of total residual deviance, and number of effective parameters), the posterior median, posterior standard deviation and 95% credible interval of *tau* under the consistency model and after each split node. See 'Details'.

intervalplot_inconsistency_factor
: A panel of interval plots on the direct and indirect effect of the split nodes and the corresponding inconsistency factor. See 'Details'.

intervalplot_tau
: An interval plot on *tau* after each split node. See 'Details'.

Author(s)

Loukia M. Spineli

References

Dias S, Welton NJ, Caldwell DM, Ades AE. Checking consistency in mixed treatment comparison meta-analysis. *Stat Med* 2010;**29**(7-8):932–44. doi: 10.1002/sim.3767

Spiegelhalter DJ, Abrams KR, Myles JP. Bayesian approaches to clinical trials and health-care evaluation. John Wiley and Sons, Chichester, 2004.

Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A. Bayesian measures of model complexity and fit. *J R Stat Soc B* 2002;**64**(4):583–616. doi: 10.1111/1467-9868.00353

See Also

mtc.nodesplit.comparisons, run_model, run_nodesplit, write_xlsx

Examples

```
data("nma.baker2009")

# Read results from 'run_model' (using the default arguments)
res <- readRDS(system.file('extdata/res_baker.rds', package = 'rnmamod'))

# Read results from 'run_nodesplit' (using the default arguments)
node <- readRDS(system.file('extdata/node_baker.rds', package = 'rnmamod'))

# The names of the interventions in the order they appear in the dataset
interv_names <- c("placebo", "budesonide", "budesonide plus formoterol",
                  "fluticasone", "fluticasone plus salmeterol",
                  "formoterol", "salmeterol", "tiotropium")
```

```
# Plot the results from both models
nodesplit_plot(full = res,
               node = node,
               drug_names = interv_names)
```

pma.hetrick2012 *Paroxetine versus placebo for depressive disorders*

Description

A pairwise meta-analysis of 4 trials comparing paroxetine with placebo for depressive disorders in children and adolescents. The analysed binary outcome is remission or response as defined in the trials.

Usage

```
data(pma.hetrick2012)
```

Format

A data frame with 4 rows of arm-based data and 9 columns.

Details

The interventions have been coded as follows: 1, placebo; 2, paroxetine.

Source

Hetrick SE, McKenzie JE, Cox GR, Simmons MB, Merry SN. Newer generation antidepressants for depressive disorders in children and adolescents. *Cochrane Database Syst Rev* 2012;**11**(11):CD004851. doi: 10.1002/14651858.CD004851.pub3

pma.taylor2004 *Inositol versus glucose for depressive episode*

Description

A pairwise meta-analysis of 4 trials comparing inositol with glucose (placebo) for depressive episode. The analysed continuous outcome is the resolution of a depressive episode using the Hamilton Depression Rating Scale.

Usage

```
data(pma.taylor2004)
```

Format

A data frame with 4 rows of arm-based data and 11 columns.

Details

The interventions have been coded as follows: 1, placebo; 2, inositol.

Source

Taylor MJ, Wilder H, Bhagwagar Z, Geddes J. Inositol for depressive disorders. *Cochrane Database Syst Rev* 2004;**2004**(2):CD004049. doi: 10.1002/14651858.CD004049.pub2

`prepare_model` *WinBUGS code for Bayesian pairwise or network meta-analysis and meta-regression*

Description

The WinBUGS code, as written by Dias et al. (2013) to run a one-stage Bayesian network meta-analysis, extended to incorporate the pattern-mixture model for binary or continuous missing participant outcome data (Spineli et al., 2021; Spineli, 2019). The model has been also extended to incorporate a trial-level covariate to apply meta-regression (Cooper et al., 2009). In the case of two interventions, the code boils down to a one-stage Bayesian pairwise meta-analysis with pattern-mixture model (Turner et al., 2015; Spineli et al, 2021).

Usage

```
prepare_model(measure, model, covar_assumption, assumption)
```

Arguments

`measure` Character string indicating the effect measure. For a binary outcome, the following can be considered: `"OR"`, `"RR"` or `"RD"` for the odds ratio, relative risk, and risk difference, respectively. For a continuous outcome, the following can be considered: `"MD"`, `"SMD"`, or `"ROM"` for mean difference, standardised mean difference and ratio of means, respectively.

`model` Character string indicating the analysis model with values `"RE"`, or `"FE"` for the random-effects and fixed-effect model, respectively. The default argument is `"RE"`.

`covar_assumption`
Character string indicating the structure of the intervention-by-covariate interaction, as described in Cooper et al., (2009). Set `covar_assumption` equal to one of the following, when meta-regression is performed: `"exchangeable"`, `"independent"`, and `"common"`. Assign `"NO"` to perform pairwise or network meta-analysis.

`assumption` Character string indicating the structure of the informative missingness parameter. Set `assumption` equal to one of the following: `"HIE-COMMON"`, `"HIE-TRIAL"`, `"HIE-ARM"`, `"IDE-COMMON"`, `"IDE-TRIAL"`, `"IDE-ARM"`, `"IND-CORR"`, or `"IND-UNCORR"`. The default argument is `"IDE-ARM"`. The abbreviations `"IDE"`, `"HIE"`, and `"IND"` stand for identical, hierarchical and independent, respectively. `"CORR"` and `"UNCORR"` stand for correlated and uncorrelated, respectively.

Details

`prepare_model` creates the model in the JAGS dialect of the BUGS language. The output of this function constitutes the argument `model.file` of the `jags` function (in the R-package R2jags) via the `textConnection` function.

Value

An R character vector object to be passed to `run_model` and `run_metareg` through the `textConnection` function as the argument `object`.

Author(s)

Loukia M. Spineli

References

Cooper NJ, Sutton AJ, Morris D, Ades AE, Welton NJ. Addressing between-study heterogeneity and inconsistency in mixed treatment comparisons: Application to stroke prevention treatments in individuals with non-rheumatic atrial fibrillation. *Stat Med* 2009;**28**(14):1861–81. doi: 10.1002/sim.3594

Dias S, Sutton AJ, Ades AE, Welton NJ. Evidence synthesis for decision making 2: a generalized linear modeling framework for pairwise and network meta-analysis of randomized controlled trials. *Med Decis Making* 2013;**33**(5):607–17. doi: 10.1177/0272989X12458724

Spineli LM, Kalyvas C, Papadimitropoulou K. Continuous(ly) missing outcome data in network meta-analysis: a one-stage pattern-mixture model approach. *Stat Methods Med Res* 2021;**30**(4):958–75. doi: 10.1177/0962280220983544

Spineli LM. An empirical comparison of Bayesian modelling strategies for missing binary outcome data in network meta-analysis. *BMC Med Res Methodol* 2019;**19**(1):86. doi: 10.1186/s12874-019-0731-y

Turner NL, Dias S, Ades AE, Welton NJ. A Bayesian framework to account for uncertainty due to missing binary outcome data in pairwise meta-analysis. *Stat Med* 2015;**34**(12):2062–80. doi: 10.1002/sim.6475

See Also

`run_metareg`, `run_model`, `jags`, `textConnection`

prepare_nodesplit *WinBUGS code for the node-splitting approach*

Description

The WinBUGS code, as written by Dias et al. (2010) to run a one-stage Bayesian node-splitting model, extended to incorporate the pattern-mixture model for binary or continuous missing participant outcome data (Spineli et al., 2021; Spineli, 2019).

Usage

```
prepare_nodesplit(measure, model, assumption)
```

Arguments

measure	Character string indicating the effect measure. For a binary outcome, the following can be considered: "OR", "RR" or "RD" for the odds ratio, relative risk, and risk difference, respectively. For a continuous outcome, the following can be considered: "MD", "SMD", or "ROM" for mean difference, standardised mean difference and ratio of means, respectively.
model	Character string indicating the analysis model with values "RE", or "FE" for the random-effects and fixed-effect model, respectively. The default argument is "RE".
assumption	Character string indicating the structure of the informative missingness parameter. Set assumption equal to one of the following: "HIE-COMMON", "HIE-TRIAL", "HIE-ARM", "IDE-COMMON", "IDE-TRIAL", "IDE-ARM", "IND-CORR", or "IND-UNCORR". The default argument is "IDE-ARM". The abbreviations "IDE", "HIE", and "IND" stand for identical, hierarchical and independent, respectively. "CORR" and "UNCORR" stand for correlated and uncorrelated, respectively.

Details

This functions creates the model in the JAGS dialect of the BUGS language. The output of this function constitutes the argument model.file of jags (in the R-package R2jags) via the textConnection function.

prepare_nodesplit inherits measure, model, and assumption from the run_model function. For a binary outcome, when measure is "RR" (relative risk) or "RD" (risk difference) in run_model, prepare_nodesplit currently considers the WinBUGS code for the odds ratio.

The split nodes have been automatically selected via the mtc.nodesplit.comparisons function of the R-package gemtc. See 'Details' in run_nodesplit.

Value

An R character vector object to be passed to run_nodesplit through the textConnection function as the argument object.

Author(s)

Loukia M. Spineli

References

Dias S, Welton NJ, Caldwell DM, Ades AE. Checking consistency in mixed treatment comparison meta-analysis. *Stat Med* 2010;**29**(7-8):932–44. doi: 10.1002/sim.3767

Spineli LM, Kalyvas C, Papadimitropoulou K. Continuous(ly) missing outcome data in network meta-analysis: a one-stage pattern-mixture model approach. *Stat Methods Med Res* 2021;**30**(4):958–75. doi: 10.1177/0962280220983544

Spineli LM. An empirical comparison of Bayesian modelling strategies for missing binary outcome data in network meta-analysis. *BMC Med Res Methodol* 2019;**19**(1):86. doi: 10.1186/s12874-019-0731-y

See Also

`jags`, `run_model`, `mtc.nodesplit.comparisons`, `run_nodesplit`, `textConnection`

`prepare_ume` *WinBUGS code for the unrelated mean effects model*

Description

The WinBUGS code, as proposed by Dias et al. (2013) to run a one-stage Bayesian unrelated mean effects model, refined (Spineli, 2021), and extended to incorporate the pattern-mixture model for binary or continuous missing participant outcome data (Spineli et al., 2021; Spineli, 2019).

Usage

```
prepare_ume(measure, model, assumption, connected)
```

Arguments

`measure`	Character string indicating the effect measure with values `"OR"`, `"MD"`, `"SMD"`, or `"ROM"` for the odds ratio, mean difference, standardised mean difference and ratio of means, respectively.
`model`	Character string indicating the analysis model with values `"RE"`, or `"FE"` for the random-effects and fixed-effect model, respectively. The default argument is `"RE"`.
`assumption`	Character string indicating the structure of the informative missingness parameter. Set `assumption` equal to one of the following: `"HIE-COMMON"`, `"HIE-TRIAL"`, `"HIE-ARM"`, `"IDE-COMMON"`, `"IDE-TRIAL"`, `"IDE-ARM"`, `"IND-CORR"`, or `"IND-UNCORR"`. The default argument is `"IDE-ARM"`. The abbreviations `"IDE"`, `"HIE"`, and `"IND"` stand for identical, hierarchical and independent, respectively. `"CORR"` and `"UNCORR"` stand for correlated and uncorrelated, respectively.
`connected`	An integer equal to one or larger that indicates the number of subnetworks.

Details

This functions creates the model in the JAGS dialect of the BUGS language. The output of this function constitutes the argument `model.file` of `jags` (in the R-package R2jags) via the `textConnection` function.

`prepare_ume` inherits `measure`, `model`, and `assumption` from the `run_model` function. For a binary outcome, when `measure` is "RR" (relative risk) or "RD" (risk difference) in `run_model`, `prepare_ume` currently considers the WinBUGS code for the odds ratio.

Value

An R character vector object to be passed to `run_ume` through the `textConnection` function as the argument `object`.

Author(s)

Loukia M. Spineli

References

Dias S, Welton NJ, Sutton AJ, Caldwell DM, Lu G, Ades AE. Evidence synthesis for decision making 4: inconsistency in networks of evidence based on randomized controlled trials. *Med Decis Making* 2013;**33**(5):641–56. doi: 10.1177/0272989X12455847

Spineli LM. A revised framework to evaluate the consistency assumption globally in a network of interventions. *Med Decis Making* 2021. doi: 10.1177/0272989X211068005

Spineli LM, Kalyvas C, Papadimitropoulou K. Continuous(ly) missing outcome data in network meta-analysis: a one-stage pattern-mixture model approach. *Stat Methods Med Res* 2021;**30**(4):958–75. doi: 10.1177/0962280220983544

Spineli LM. An empirical comparison of Bayesian modelling strategies for missing binary outcome data in network meta-analysis. *BMC Med Res Methodol* 2019;**19**(1):86. doi: 10.1186/s12874-019-0731-y

See Also

`jags`, `run_model`, `run_ume`, `textConnection`

`rankosucra_plot` *Rankograms and SUCRA curves*

Description

It returns a panel of rankograms with integrated SUCRA curves for each intervention in the network. The function can illustrate the results of a single or two outcomes simultaneously.

Usage

```
rankosucra_plot(
  full1,
  full2 = NULL,
  drug_names1,
  drug_names2 = NULL,
  name1 = NULL,
  name2 = NULL
)
```

Arguments

`full1` An object of S3 class `run_model` for network meta-analysis. See 'Value' in `run_model`.

`full2` An object of S3 class `run_model` for network meta-analysis of a second outcome. See 'Value' in `run_model`.

`drug_names1` A vector of labels with the name of the interventions in the order they appear in the argument `data` of `run_model` for `full1`.

`drug_names2` A vector of labels with the name of the interventions in the order they appear in the argument `data` of `run_model` for `full2`. The elements must be a subset of `drug_names1`.

`name1` The text for the title of the results that refer to the outcome under `full1`.

`name2` The text for the title of the results that refer to the outcome under `full2`.

Details

Interventions are sorted in the descending order of their SUCRA value. The SUCRA value expressed in percentage appears on the top left corner of each panel. In the case of two outcomes, the SUCRA values of the outcome under the argument `full1` are considered to sort the interventions from the best to the worst.

When a second outcome is also considered, different colours are used to draw the corresponding SUCRA curves and the rankograms: green for the outcome under `full1`, and red for the outcome under `full2`.

`rankosucra_plot` can be used only for a network of interventions. Otherwise, the execution of the function will be stopped and an error message will be printed on the R console.

Value

A panel of rankograms (yellowish bars) with integrated reddish SUCRA curves for each intervention in the network (Salanti et al., 2011). The x-axis of each panel refers to the ranking, and the y-axis refers to the ranking probability expressed in percentage.

Author(s)

Loukia M. Spineli, Chrysostomos Kalyvas, Katerina Papadimitropoulou

References

Salanti G, Ades AE, Ioannidis JP. Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: an overview and tutorial. *J Clin Epidemiol* 2011;**64**(2):163–71. doi: 10.1016/j.jclinepi.2010.03.016

See Also

`run_model`

Examples

```
data("nma.liu2013")

# Read results from 'run_model' (using the default arguments)
res <- readRDS(system.file('extdata/res_liu.rds', package = 'rnmamod'))

# The names of the interventions in the order they appear in the dataset
interv_names <- c("placebo", "pramipexole",
                  "serotonin-norepinephrine reuptake inhibitor",
                  "serotonin reuptake inhibitor",
                  "tricyclic antidepressant", "pergolide")

# Create the integrated rankograms and SUCRA curves
rankosucra_plot(full1 = res,
                drug_names1 = interv_names)
```

---

robustness_index *Robustness index*

Description

Calculates the robustness index, a novel index that quantifies the overall divergence of the sensitivity analysis results from the primary analysis results. The robustness index considers objective decision rules to infer the presence or lack of robustness of the primary analysis results when conducting a sensitivity analysis (Spineli et al., 2021).

Usage

```
robustness_index(sens, threshold)
```

Arguments

sens	An object of S3 class `run_sensitivity` when sensitivity analysis refers to different scenarios about the average missingness parameter. See ’Value’ in `run_sensitivity`. For a **general** sensitivity analysis, insert a list of at least two objects of S3 class `run_model` indicating different re-analyses: the first object (of class `run_model`) in the list should refer to the primary analysis.

threshold A number indicating the threshold of robustness, that is, the minimally allowed deviation between the primary analysis and re-analysis results. See 'Details' below.

Details

Thresholds of robustness have been proposed only for the odds ratio and standardised mean difference (Spineli et al., 2021). The user may consider the values 0.28 and 0.17 in the argument threshold for the odds ratio and standardised mean difference effect measures (the default values), respectively, or consider other plausible values. When the argument threshold has not been defined, robustness_index considers the default values 0.28 and 0.17 as threshold for robustness for binary and continuous outcome, respectively, regardless of the effect measure (the default thresholds may not be proper choices for other effect measures; hence, use these threshold with great caution in this case). Spineli et al. (2021) offers a discussion on specifying the threshold of robustness.

In the case of binary outcome, robustness_index considers the results in the odds ratio scale to calculate the robustness index. This is because, the odds ratio is used as the 'best-case' effect measure in run_model. Then, relative risk, and risk difference are functions of the odds ratio and the selected baseline risk (See 'Details' in run_model).

In the case of missing participant outcome data, the primary analysis is considered to be the middle of the numbers in the argument mean_scenarios of run_sensitivity (see 'Arguments' and 'Details' in run_sensitivity).

In robust, the value "robust" appears when robust_index is less than threshold; otherwise, the value "frail" appears.

In the case of missing participant outcome data, robustness_index can be used only when missing participant outcome data have been extracted for at least one trial. Otherwise, the execution of the function will be stopped and an error message will be printed in the R console.

Value

robustness_index prints on the R console a message in green text on the threshold of robustness determined by the user. Then, the function returns the following list of elements:

robust_index A numeric scalar or vector on the robustness index values. In the case of a pairwise meta-analysis, robust_index is scalar as only one summary effect size is obtained. In the case of network meta-analysis, robust_index is a vector with length equal to the number of possible pairwise comparisons; one robustness index per possible comparison.

robust A character or character vector (of same length with robust_index) on whether the primary analysis results are *robust* or *frail* to the different re-analyses.

kld A vector or matrix on the Kullback-Leibler divergence (KLD) measure in the summary effect size from a subsequent re-analysis to the primary analysis. In the case of a pairwise meta-analysis, kld is a vector with length equal to the number of total analyses (one KLD value is obtained per analysis). The number of total analyses equals the square of the number of scenarios indicated in the argument mean_scenarios of run_sensitivity, in the case of missing participant outcome data; otherwise, the length of the character vector in argument sens. In the case of network meta-analysis, robust_index is a matrix with number of rows equal to the number of total analyses and number of columns equal to the number of possible pairwise comparisons; one KLD value per analysis and possible comparison.

`threshold` The threshold used to be inherited by the `heatmap_robustness` function.

`scenarios` The scenarios considered to be inherited by the `kld_barplot` function.

Author(s)

Loukia M. Spineli

References

Kullback S, Leibler RA. On information and sufficiency. *Ann Math Stat* 1951;**22**(1):79–86. doi: 10.1214/aoms/1177729694

Spineli LM, Kalyvas C, Papadimitropoulou K. Quantifying the robustness of primary analysis results: A case study on missing outcome data in pairwise and network meta-analysis. *Res Synth Methods* 2021;**12**(4):475–90. doi: 10.1002/jrsm.1478

See Also

`heatmap_robustness`, `kld_barplot`, `run_model`, `run_sensitivity`

Examples

```
data("nma.baker2009")

# Read results from 'run_sensitivity' (using the default arguments)
res_sens <- readRDS(system.file('extdata/res_sens_baker.rds',
                    package = 'rnmamod'))

# Calculate the robustness index
robustness_index(sens = res_sens,
                 threshold = 0.28)
```

`robustness_index_user` *Robustness index when 'metafor' or 'netmeta' are used*

Description

Calculates the robustness index for a sensitivity analysis (Spineli et al., 2021) performed using the results of the analysis performed via the R-package netmeta or metafor. The user defines the input and the function returns the robustness index.

Usage

```
robustness_index_user(sens, pkg, attribute, threshold)
```

Arguments

sens
: A list of R objects of class netmeta, netmetabin (see netmeta) or rma, rma.glmm, rma.mh, rma.mv, rma.peto, and rma.uni (see metafor). The number of elements equals the number of analyses using the same dataset and the same R-package. The first element should refer to the primary analysis. Hence, the list should include at least two elements (see 'Details').

pkg
: Character string indicating the R-package with values "netmeta", or "metafor".

attribute
: This is relevant only for netmeta. A vector of at least two characters with values "TE.common" or "TE.random". See 'Values' in netmeta or netmetabin.

threshold
: A number indicating the threshold of robustness, that is, the minimally allowed deviation between the primary analysis (the first element in sens) and re-analysis results. See 'Details' below.

Details

Thresholds of robustness have been proposed only for the odds ratio and standardised mean difference (Spineli et al., 2021). The user may consider the values 0.28 and 0.17 in the argument threshold for the odds ratio and standardised mean difference effect measures (the default values), respectively, or consider other plausible values. When the argument threshold has not been defined, robustness_index considers the default values 0.28 and 0.17 as threshold for robustness for binary and continuous outcome, respectively, regardless of the effect measure (the default thresholds may not be proper choices for other effect measures; hence, use these threshold with great caution in this case). Spineli et al. (2021) offers a discussion on specifying the threshold of robustness.

When other effect measure is used (other than odds ratio or standardised mean difference) or the elements in sens refer to different effect measures, the execution of the function will be stopped and an error message will be printed in the R console.

In robust, the value "robust" appears when the calculated robust_index is less than threshold; otherwise, the value "frail" appears.

Value

robustness_index_user prints on the R console a message in red text on the threshold of robustness determined by the user. Then, the function returns the following list of elements:

robust_index
: A numeric scalar or vector on the robustness index values. In the case of a pairwise meta-analysis, robust_index is scalar as only one summary effect size is obtained. In the case of network meta-analysis, robust_index is a vector with length equal to the number of possible pairwise comparisons; one robustness index per pairwise comparison.

robust
: A character or character vector (of same length with robust_index) on whether the primary analysis results are *robust* or *frail* to the different re-analyses.

kld
: A vector or matrix on the Kullback-Leibler divergence (KLD) measure in the summary effect size from a subsequent re-analysis to the primary analysis. In the case of a pairwise meta-analysis, kld is a vector with length equal to the number of total analyses (one KLD value is obtained per analysis). The number of total analyses equals the length of sens. In the case of network meta-analysis, robust_index is a matrix with number of rows equal to the number of

total analyses and number of columns equal to the number of possible pairwise comparisons; one KLD value per analysis and possible comparison.

`attribute` The attributes considered.

`threshold` The threshold used to be inherited by the `heatmap_robustness` function. See 'Details'.

Author(s)

Loukia M. Spineli

References

Kullback S, Leibler RA. On information and sufficiency. *Ann Math Stat* 1951;**22**(1):79–86. doi: 10.1214/aoms/1177729694

Spineli LM, Kalyvas C, Papadimitropoulou K. Quantifying the robustness of primary analysis results: A case study on missing outcome data in pairwise and network meta-analysis. *Res Synth Methods* 2021;**12**(4):475–90. doi: 10.1002/jrsm.1478

See Also

`rma`, `rma.glmm`, `rma.mh`, `rma.mv`, `rma.peto`, `rma.uni`, `netmeta`, `netmetabin`, `heatmap_robustness`

Examples

```
## Not run:
library(netmeta)

data(Baker2009)

# Transform from arm-based to contrast-based format
p1 <- pairwise(treatment, exac, total, studlab = paste(study, year),
data = Baker2009, sm = "OR")

# Conduct standard network meta-analysis
net1 <- netmeta(p1, ref = "Placebo")

# Calculate the robustness index (random-effects versus fixed-effect)
robustness_index_user(sens = list(net1, net1),
                      pkg = "netmeta",
                      attribute = c("TE.random", "TE.common"),
                      threshold = 0.28)

## End(Not run)
```

run_metareg *Perform Bayesian pairwise or network meta-regression*

Description

Performs a one-stage pairwise or network meta-regression while addressing aggregate binary or continuous missing participant outcome data via the pattern-mixture model.

Usage

```
run_metareg(
  full,
  covariate,
  covar_assumption,
  n_chains,
  n_iter,
  n_burnin,
  n_thin,
  inits = NULL
)
```

Arguments

`full` An object of S3 class `run_model`. See 'Value' in `run_model`.

`covariate` A numeric vector or matrix for a trial-specific covariate that is a potential effect modifier. See 'Details'.

`covar_assumption` Character string indicating the structure of the intervention-by-covariate interaction, as described in Cooper et al. (2009). Set `covar_assumption` equal to `"exchangeable"`, `"independent"`, or `"common"`.

`n_chains` Positive integer specifying the number of chains for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 2.

`n_iter` Positive integer specifying the number of Markov chains for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 10000.

`n_burnin` Positive integer specifying the number of iterations to discard at the beginning of the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 1000.

`n_thin` Positive integer specifying the thinning rate for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 1.

`inits` A list with the initial values for the parameters; an argument of the `jags` function of the R-package R2jags. The default argument is `NULL`, and JAGS generates the initial values.

Details

run_metareg inherits the arguments data, measure, model, assumption, heter_prior, mean_misspar, var_misspar, D, ref, indic, and base_risk from run_model (now contained in the argument full). This prevents specifying a different Bayesian model from that considered in run_model. Therefore, the user needs first to apply run_model, and then use run_metareg (see 'Examples').

The model runs in JAGS and the progress of the simulation appears on the R console. The output of run_metareg is used as an S3 object by other functions of the package to be processed further and provide an end-user-ready output. The model is updated until convergence using the autojags function of the R-package R2jags with 2 updates and number of iterations and thinning equal to n_iter and n_thin, respectively.

The models described in Spineli et al. (2021), and Spineli (2019) have been extended to incorporate one *study-level covariate* variable following the assumptions of Cooper et al. (2009) for the structure of the intervention-by-covariate interaction. The covariate can be either a numeric vector or matrix with columns equal to the maximum number of arms in the dataset.

Value

A list of R2jags outputs on the summaries of the posterior distribution, and the Gelman-Rubin convergence diagnostic (Gelman et al., 1992) for the following monitored parameters for a fixed-effect pairwise meta-analysis:

EM The estimated summary effect measure (according to the argument measure defined in run_model).

beta_all The estimated regression coefficient for all possible pairwise comparisons according to the argument covar_assumption.

dev_o The deviance contribution of each trial-arm based on the observed outcome.

hat_par The fitted outcome at each trial-arm.

phi The informative missingness parameter.

For a fixed-effect network meta-analysis, the output additionally includes:

SUCRA The surface under the cumulative ranking (SUCRA) curve for each intervention.

effectiveness
The ranking probability of each intervention for every rank.

For a random-effects pairwise meta-analysis, the output additionally includes the following elements:

EM_pred The predicted summary effect measure (according to the argument measure defined in run_model).

delta The estimated trial-specific effect measure (according to the argument measure defined in run_model). For a multi-arm trial, we estimate *T-1* effects, where *T* is the number of interventions in the trial.

tau The between-trial standard deviation.

In network meta-analysis, EM and EM_pred refer to all possible pairwise comparisons of interventions in the network. Furthermore, tau is typically assumed to be common for all observed comparisons in the network. For a multi-arm trial, we estimate a total *T-1* of delta for comparisons with the baseline intervention of the trial (found in the first column of the element **t**), with T being the number of interventions in the trial.

Furthermore, the output includes the following elements:

`abs_risk` The adjusted absolute risks for each intervention. This appears only when `measure = "OR"`, `measure = "RR"`, or `measure = "RD"`.

`leverage_o` The leverage for the observed outcome at each trial-arm.

`sign_dev_o` The sign of the difference between observed and fitted outcome at each trial-arm.

`model_assessment` A data-frame on the measures of model assessment: deviance information criterion, number of effective parameters, and total residual deviance.

`jagsfit` An object of S3 class `jags` with the posterior results on all monitored parameters to be used in the `mcmc_diagnostics` function.

The `run_metareg` function also returns the arguments `data`, `measure`, `model`, `assumption`, `covariate`, `covar_assumption`, `n_chains`, `n_iter`, `n_burnin`, and `n_thin` to be inherited by other relevant functions of the package.

Author(s)

Loukia M. Spineli

References

Cooper NJ, Sutton AJ, Morris D, Ades AE, Welton NJ. Addressing between-study heterogeneity and inconsistency in mixed treatment comparisons: Application to stroke prevention treatments in individuals with non-rheumatic atrial fibrillation. *Stat Med* 2009;**28**(14):1861–81. doi: 10.1002/sim.3594

Gelman A, Rubin DB. Inference from iterative simulation using multiple sequences. *Stat Sci* 1992;**7**(4):457–72. doi: 10.1214/ss/1177011136

Spineli LM, Kalyvas C, Papadimitropoulou K. Continuous(ly) missing outcome data in network meta-analysis: a one-stage pattern-mixture model approach. *Stat Methods Med Res* 2021;**30**(4):958–75. doi: 10.1177/0962280220983544

Spineli LM. An empirical comparison of Bayesian modelling strategies for missing binary outcome data in network meta-analysis. *BMC Med Res Methodol* 2019;**19**(1):86. doi: 10.1186/s12874-019-0731-y

See Also

`autojags`, `jags`, `run_model`

Examples

```
data("nma.baker2009")

# Read results from 'run_model' (using the default arguments)
res <- readRDS(system.file('extdata/res_baker.rds', package = 'rnmamod'))
```

```
# Publication year
pub_year <- c(1996, 1998, 1999, 2000, 2000, 2001, rep(2002, 5), 2003, 2003,
              rep(2005, 4), 2006, 2006, 2007, 2007)


# Perform a random-effects network meta-regression (exchangeable structure)
# Note: Ideally, set 'n_iter' to 10000 and 'n_burnin' to 1000
run_metareg(full = res,
            covariate = pub_year,
            covar_assumption = "exchangeable",
            n_chains = 3,
            n_iter = 1000,
            n_burnin = 100,
            n_thin = 1)
```

run_model *Perform Bayesian pairwise or network meta-analysis*

Description

Performs a one-stage pairwise or network meta-analysis while addressing aggregate binary or continuous missing participant outcome data via the pattern-mixture model.

Usage

```
run_model(
  data,
  measure,
  model,
  assumption,
  heter_prior,
  mean_misspar,
  var_misspar,
  D,
  ref,
  base_risk,
  n_chains,
  n_iter,
  n_burnin,
  n_thin,
  inits = NULL
)
```

Arguments

data — A data-frame of the one-trial-per-row format with arm-level data. See ’Format’ for the specification of the columns.

measure — Character string indicating the effect measure. For a binary outcome, the following can be considered: "OR", "RR" or "RD" for the odds ratio, relative risk, and risk difference, respectively. For a continuous outcome, the following can be considered: "MD", "SMD", or "ROM" for mean difference, standardised mean difference and ratio of means, respectively.

model — Character string indicating the analysis model with values "RE", or "FE" for the random-effects and fixed-effect model, respectively. The default argument is "RE".

assumption — Character string indicating the structure of the informative missingness parameter. Set assumption equal to one of the following: "HIE-COMMON", "HIE-TRIAL", "HIE-ARM", "IDE-COMMON", "IDE-TRIAL", "IDE-ARM", "IND-CORR", or "IND-UNCORR". The default argument is "IDE-ARM". The abbreviations "IDE", "HIE", and "IND" stand for identical, hierarchical and independent, respectively. "CORR" and "UNCORR" stand for correlated and uncorrelated, respectively.

heter_prior — A list of three elements with the following order: 1) a character string indicating the distribution with (currently available) values "halfnormal", "uniform", "lognormal", or "logt"; 2) two numeric values that refer to the parameters of the selected distribution. For "lognormal", and "logt" these numbers refer to the mean and precision, respectively. For "halfnormal", these numbers refer to zero and the scale parameter (equal to 4 or 1 being the corresponding precision of the scale parameter 0.5 or 1). For "uniform", these numbers refer to the minimum and maximum value of the distribution. See ’Details’ in heterogeneity_param_prior.

mean_misspar — A scalar or numeric vector of two numeric values for the mean of the normal distribution of the informative missingness parameter (see ’Details’). The default argument is 0 and corresponds to the missing-at-random assumption. See also ’Details’ in missingness_param_prior.

var_misspar — A positive non-zero number for the variance of the normal distribution of the informative missingness parameter. When the measure is "OR", "MD", or "SMD" the default argument is 1. When the measure is "ROM" the default argument is 0.04.

D — A binary number for the direction of the outcome. Set D = 1 for beneficial outcome and D = 0 for harmful outcome.

ref — An integer specifying the reference intervention. The number should match the intervention identifier under element **t** in data (See ’Format’).

base_risk — A scalar, a vector of length three with elements sorted in ascending order, or a matrix with two columns and number of rows equal to the number of relevant trials. In the case of a scalar or vector, the elements should be in the interval (0, 1) (see ’Details’). If base_risk has not been defined, the function uses the median event risk for the reference intervention from the corresponding trials in data. This argument is only relevant for a binary outcome.

`n_chains` Positive integer specifying the number of chains for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 2.

`n_iter` Positive integer specifying the number of Markov chains for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 10000.

`n_burnin` Positive integer specifying the number of iterations to discard at the beginning of the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 1000.

`n_thin` Positive integer specifying the thinning rate for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 1.

`inits` A list with the initial values for the parameters; an argument of the `jags` function of the R-package R2jags. The default argument is `NULL`, and JAGS generates the initial values.

Format

The columns of the data-frame in the argument `data` refer to the following elements for a continuous outcome:

t An intervention identifier in each arm.

y The observed mean value of the outcome in each arm.

sd The observed standard deviation of the outcome in each arm.

m The number of missing participant outcome data in each arm.

n The number of randomised participants in each arm.

For a binary outcome, the columns of the data-frame in the argument `data` refer to the following elements:

t An intervention identifier in each arm.

r The observed number of events of the outcome in each arm.

m The number of missing participant outcome data in each arm.

n The number of randomised participants in each arm.

The number of rows in `data` equals the number of collected trials. Each element appears in `data` as many times as the maximum number of interventions compared in a trial of the dataset. In pairwise meta-analysis, the maximum number of arms is inherently two. The same holds for a network meta-analysis without multi-arm trials. In the case of network meta-analysis with multi-arm trials, the maximum number of arms exceeds two. See 'Examples' that illustrates the structure of data for a network with a maximum number of four arms. It is not a prerequisite of run_model that the multi-arm trials appear at the bottom of the dataset.

Details

The model runs in JAGS and the progress of the simulation appears on the R console. The output of run_model is used as an S3 object by other functions of the package to be processed further and provide an end-user-ready output.

The data_preparation function is called to prepare the data for the Bayesian analysis. data_preparation creates the pseudo-data-frames m_new, and I, that have the same dimensions with the element N. m_new takes the zero value for the observed trial-arms with unreported missing participant outcome data (i.e., m equals NA for the corresponding trial-arms), the same value with m for the observed trial-arms with reported missing participant outcome data, and NA for the unobserved trial-arms. I is a dummy pseudo-data-frame and takes the value one for the observed trial-arms with reported missing participant outcome data, the zero value for the observed trial-arms with unreported missing participant outcome data (i.e., m_new equals zero for the corresponding trial-arms), and NA for the unobserved trial-arms. Thus, I indicates whether missing participant outcome data have been collected for the observed trial-arms. If the user has not defined the element **m** in data, m_new and I take the zero value for all observed trial-arms to indicate that no missing participant outcome data have been collected for the analysed outcome. See 'Details' in data_preparation.

Furthermore, data_preparation sorts the interventions across the arms of each trial in an ascending order and correspondingly the remaining elements in data (see 'Format'). data_preparation considers the first column in **t** as being the control arm for every trial. Thus, this sorting ensures that interventions with a lower identifier are consistently treated as the control arm in each trial. This case is relevant in non-star-shaped networks.

The model is updated until convergence using the autojags function of the R-package R2jags with 2 updates and number of iterations and thinning equal to n_iter and n_thin, respectively.

To perform a Bayesian pairwise or network meta-analysis, the prepare_model function is called which contains the WinBUGS code as written by Dias et al. (2013a) for binomial and normal likelihood to analyse aggregate binary and continuous outcome data, respectively. prepare_model uses the consistency model (as described in Lu and Ades (2006)) to estimate all possible comparisons in the network. It also accounts for the multi-arm trials by assigning conditional univariate normal distributions on the underlying trial-specific effect size of comparisons with the baseline arm of the multi-arm trial (Dias et al., 2013a).

The code of Dias et al. (2013a) has been extended to incorporate the pattern-mixture model to adjust the underlying outcome in each arm of every trial for missing participant outcome data (Spineli et al., 2021; Spineli, 2019a; Turner et al., 2015). The assumptions about the missingness parameter are specified using the arguments mean_misspar and var_misspar. Specifically, run_model considers the informative missingness odds ratio in the logarithmic scale for binary outcome data (Spineli, 2019a; Turner et al., 2015; White et al., 2008), the informative missingness difference of means when measure is "MD" or "SMD", and the informative missingness ratio of means in the logarithmic scale when measure is "ROM" (Spineli et al., 2021; Mavridis et al., 2015).

When assumption is trial-specific (i.e., "IDE-TRIAL" or "HIE-TRIAL"), or independent (i.e., "IND-CORR" or "IND-UNCORR"), only one numeric value can be assigned to mean_misspar because the same missingness scenario is applied to all trials and trial-arms of the dataset, respectively. When assumption is "IDE-ARM" or "HIE-ARM", a maximum of two *different* or *identical* numeric values can be assigned as a vector to mean_misspars: the first value refers to the experimental arm, and the second value refers to the control arm of a trial. In the case of a network, the first value is considered for all non-reference interventions and the second value is considered for the reference intervention of the network (i.e., the intervention with identifier equal to ref). This is necessary to ensure transitivity in the assumptions for the missingness parameter across the network (Spineli, 2019b).

When there is at least one trial-arm with unreported missing participant outcome data (i.e., m equals NA for the corresponding trial-arms) or when missing participant outcome data have not been collected for the analysed outcome (i.e., m is missing in data), run_model assigns the assumption "IND-UNCORR" to assumption.

Currently, there are no empirically-based prior distributions for the informative missingness parameters. The user may refer to Spineli (2019), Turner et al. (2015), Mavridis et al. (2015), and White et al. (2008) to determine mean_misspar and select a proper value for var_misspar.

The scalar base_risk refers to a fixed baseline risk for the selected reference intervention (as specified with ref). When base_risk is a three-element vector, it refers to a random baseline risk and the elements should be sorted in ascending order as they refer to the lower bound, mean value, and upper bound of the 95% confidence interval for the baseline risk for the selected reference intervention. The baseline_model function is called to calculate the mean and variance of the approximately normal distribution of the logit of an event for ref using these three elements (Dias et al., 2018). When base_risk is a matrix, it refers to the predicted baseline risk with first column being the number of events, and second column being the sample size of the corresponding trials on the selected reference intervention. Then the baseline_model function is called that contains the WinBUGS code as written by Dias et al. (2013b) for the hierarchical baseline model. The posterior mean and precision of the predictive distribution of the logit of an event for the selected reference intervention are plugged in the WinBUGS code for the relative effects model (via the prepare_model function). The matrix base_risk should not comprise the trials in data that include the ref, unless justified (Dias et al., 2018).

To obtain unique absolute risks for each intervention, the network meta-analysis model has been extended to incorporate the transitive risks framework, namely, an intervention has the same absolute risk regardless of the comparator intervention(s) in a trial (Spineli et al., 2017). The absolute risks are a function of the odds ratio (the **base-case** effect measure for a binary outcome) and the selected baseline risk for the reference intervention (ref) (Appendix in Dias et al., 2013a). We advocate using the odds ratio as an effect measure for its desired mathematical properties. Then, the relative risk and risk difference can be obtained as a function of the absolute risks of the corresponding interventions in the comparison of interest. Hence, regardless of the selected measure for a binary outcome, run_model performs pairwise or network meta-analysis based on the odds ratio.

Value

A list of R2jags output on the summaries of the posterior distribution, and the Gelman-Rubin convergence diagnostic (Gelman et al., 1992) of the following monitored parameters for a fixed-effect pairwise meta-analysis:

EM The estimated summary effect measure (according to the argument measure).

EM_LOR The estimated summary odd ratio in the logarithmic scale when measure = "RR" or measure = "RD".

`dev_o` The deviance contribution of each trial-arm based on the observed outcome.

`hat_par` The fitted outcome at each trial-arm.

`phi` The informative missingness parameter.

For a fixed-effect network meta-analysis, the output additionally includes:

`SUCRA` The surface under the cumulative ranking curve for each intervention.

`SUCRA_LOR` The surface under the cumulative ranking curve for each intervention under the odds ratio effect measure when `measure = "RR"` or `measure = "RD"`.

`effectiveneness`
The ranking probability of each intervention for every rank.

For a random-effects pairwise meta-analysis, the output additionally includes the following elements:

`EM_pred` The predicted summary effect measure (according to the argument `measure`).

`EM_pred_LOR` The predicted summary odds ratio in the logarithmic scale when `measure = "RR"` or `measure = "RD"`.

`delta` The estimated trial-specific effect measure (according to the argument `measure`).

`tau` The between-trial standard deviation.

In network meta-analysis, `EM` and `EM_pred` refer to all possible pairwise comparisons of interventions in the network. Furthermore, `tau` is typically assumed to be common for all observed comparisons in the network. For a multi-arm trial, we estimate a total of *T-1* `delta` for comparisons with the baseline intervention of the trial (found in the first column of the element **t**), with *T* being the number of interventions in the trial.

Furthermore, the output includes the following elements:

`leverage_o` The leverage for the observed outcome at each trial-arm.

`sign_dev_o` The sign of the difference between observed and fitted outcome at each trial-arm.

`model_assessment`
A data-frame on the measures of model assessment: deviance information criterion, number of effective parameters, and total residual deviance.

`indic` The sign of basic parameters in relation to the reference intervention as specified in argument `reg`

`jagsfit` An object of S3 class `jags` with the posterior results on all monitored parameters to be used in the `mcmc_diagnostics` function.

The `run_model` function also returns the arguments `data`, `measure`, `model`, `assumption`, `heter_prior`, `mean_misspar`, `var_misspar`, `D`, `ref`, `base_risk`, `n_chains`, `n_iter`, `n_burnin`, and `n_thin` as specified by the user to be inherited by other functions of the package.

Author(s)

Loukia M. Spineli

References

Cooper NJ, Sutton AJ, Morris D, Ades AE, Welton NJ. Addressing between-study heterogeneity and inconsistency in mixed treatment comparisons: Application to stroke prevention treatments in individuals with non-rheumatic atrial fibrillation. *Stat Med* 2009;**28**(14):1861–81. doi: 10.1002/sim.3594

Dias S, Ades AE, Welton NJ, Jansen JP, Sutton AJ. Network Meta-Analysis for Decision Making. Chichester (UK): Wiley; 2018.

Dias S, Sutton AJ, Ades AE, Welton NJ. Evidence synthesis for decision making 2: a generalized linear modeling framework for pairwise and network meta-analysis of randomized controlled trials. *Med Decis Making* 2013a;**33**(5):607–17. doi: 10.1177/0272989X12458724

Dias S, Welton NJ, Sutton AJ, Ades AE. Evidence synthesis for decision making 5: the baseline natural history model. *Med Decis Making* 2013b;**33**(5):657–70. doi: 10.1177/0272989X13485155

Gelman A, Rubin DB. Inference from iterative simulation using multiple sequences. *Stat Sci* 1992;**7**(4):457–72. doi: 10.1214/ss/1177011136

Lu G, Ades AE. Assessing evidence inconsistency in mixed treatment comparisons. *J Am Stat Assoc* 2006;**101**:447–59. doi: 10.1198/016214505000001302

Mavridis D, White IR, Higgins JP, Cipriani A, Salanti G. Allowing for uncertainty due to missing continuous outcome data in pairwise and network meta-analysis. *Stat Med* 2015;**34**(5):721–41. doi: 10.1002/sim.6365

Spineli LM, Kalyvas C, Papadimitropoulou K. Continuous(ly) missing outcome data in network meta-analysis: a one-stage pattern-mixture model approach. *Stat Methods Med Res* 2021;**30**(4):958–75. doi: 10.1177/0962280220983544

Spineli LM. An empirical comparison of Bayesian modelling strategies for missing binary outcome data in network meta-analysis. *BMC Med Res Methodol* 2019a;**19**(1):86. doi: 10.1186/s12874-019-0731-y

Spineli LM. Modeling missing binary outcome data while preserving transitivity assumption yielded more credible network meta-analysis results. *J Clin Epidemiol* 2019b;**105**:19–26. doi: 10.1016/j.jclinepi.2018.09.002

Spineli LM, Brignardello-Petersen R, Heen AF, Achille F, Brandt L, Guyatt GH, et al. Obtaining absolute effect estimates to facilitate shared decision making in the context of multiple-treatment comparisons. Abstracts of the Global Evidence Summit, Cape Town, South Africa. *Cochrane Database of Systematic Reviews* 2017;**9**(Suppl 1):18911.

Turner NL, Dias S, Ades AE, Welton NJ. A Bayesian framework to account for uncertainty due to missing binary outcome data in pairwise meta-analysis. *Stat Med* 2015;**34**(12):2062–80. doi: 10.1002/sim.6475

White IR, Higgins JP, Wood AM. Allowing for uncertainty due to missing data in meta-analysis–part 1: two-stage methods. *Stat Med* 2008;**27**(5):711–27. doi: 10.1002/sim.3008

See Also

autojags, baseline_model, data_preparation, heterogeneity_param_prior, jags, missingness_param_prior, prepare_model

Examples

```
data("nma.baker2009")

# Show the first six trials of the dataset
head(nma.baker2009)


# Perform a random-effects network meta-analysis
# Note: Ideally, set 'n_iter' to 10000 and 'n_burnin' to 1000
run_model(data = nma.baker2009,
          measure = "OR",
          model = "RE",
          assumption = "IDE-ARM",
          heter_prior = list("halfnormal", 0, 1),
          mean_misspar = c(0, 0),
          var_misspar = 1,
          D = 0,
          ref = 1,
          n_chains = 3,
          n_iter = 1000,
          n_burnin = 100,
          n_thin = 1)
```

run_nodesplit *Perform the node-splitting approach*

Description

Performs the Bayesian node-splitting approach of Dias et al. (2010) extended to address aggregate binary and continuous missing participant outcome data via the pattern-mixture model (Spineli et al., 2021; Spineli, 2019). This model offers a local evaluation of the plausibility of the consistency assumption in the network (Dias et al., 2010).

Usage

```
run_nodesplit(full, n_chains, n_iter, n_burnin, n_thin, inits = NULL)
```

Arguments

`full` An object of S3 class `run_model`. See 'Value' in `run_model`.

`n_chains` Positive integer specifying the number of chains for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 2.

`n_iter` Positive integer specifying the number of Markov chains for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 10000.

n_burnin	Positive integer specifying the number of iterations to discard at the beginning of the MCMC sampling; an argument of the jags function of the R-package R2jags. The default argument is 1000.
n_thin	Positive integer specifying the thinning rate for the MCMC sampling; an argument of the jags function of the R-package R2jags. The default argument is 1.
inits	A list with the initial values for the parameters; an argument of the jags function of the R-package R2jags. The default argument is NULL, and JAGS generates the initial values.

Details

run_nodesplit inherits the arguments data, measure, model, assumption, heter_prior, mean_misspar, var_misspar, ref, and indic from run_model (now contained in the argument full). This prevents specifying a different Bayesian model from that considered in run_model. Therefore, the user needs first to apply run_model, and then use run_nodesplit (see 'Examples').

For a binary outcome, when measure is "RR" (relative risk) or "RD" (risk difference) in run_model, run_nodesplit currently performs node-splitting using the odds ratio as effect measure for being the **base-case** effect measure in run_model for a binary outcome (see also 'Details' in run_model).

To perform the Bayesian node-splitting approach, the prepare_nodesplit function is called which contains the WinBUGS code as written by Dias et al. (2010) for binomial and normal likelihood to analyse binary and continuous outcome data, respectively. prepare_nodesplit has been extended to incorporate the pattern-mixture model with informative missingness parameters for binary and continuous outcome data (see 'Details' in run_model).

run_nodesplit runs the Bayesian node-splitting approach in JAGS. The progress of the simulation appears on the R console. The number of times run_nodesplit is used appears on the R console as a text in red and it equals the number of split nodes (see 'Examples'). If there are no split nodes in the network, the execution of the function will be stopped and an error message will be printed on the R console. The model is updated until convergence using the autojags function of the R-package R2jags with 2 updates and number of iterations and thinning equal to n_iter and n_thin, respectively.

run_nodesplit uses the mtc.nodesplit.comparisons function of the R-package gemtc to obtain automatically the nodes to split based on the decision rule of van Valkenhoef et al. (2016). run_nodesplit uses the option (1) in van Valkenhoef et al. (2016) to parameterise multi-arm trials that contain the node-to-split. In contrast, mtc.nodesplit.comparisons uses the option (3) in van Valkenhoef et al. (2016). Option (1) keeps the baseline arm of the node-to-split in the corresponding multi-arms. Option (3) excludes both arms of the node-to-split from the corresponding multi-arm trials.

The output of run_nodesplit is not end-user-ready. The nodesplit_plot function inherits the output of run_nodesplit as an S3 object and processes it further to provide an end-user-ready output.

run_nodesplit can be used only for a network of interventions. In the case of two interventions, the execution of the function will be stopped and an error message will be printed on the R console.

Value

An R2jags output on the summaries of the posterior distribution, and the Gelman-Rubin convergence diagnostic of the following monitored parameters:

`direct` The summary effect measure (according to the argument `measure` defined in `run_model`) of each split node based on the corresponding trials.

`indirect` The indirect summary effect measure (according to the argument `measure` defined in `run_model`) of each split node based on the remaining network after removing (splitting) the corresponding node.

`diff` The inconsistency parameter for each split node defined as the difference between the direct and indirect effect of the corresponding split node.

`tau` The between-trial standard deviation after each split node, when the random-effects model has been specified.

Furthermore, the output includes the following element:

`model_assessment` A data-frame on the measures of model assessment after each split node: deviance information criterion, total residual deviance, and number of effective parameters.

Author(s)

Loukia M. Spineli

References

Dias S, Welton NJ, Caldwell DM, Ades AE. Checking consistency in mixed treatment comparison meta-analysis. *Stat Med* 2010;**29**(7-8):932–44. doi: 10.1002/sim.3767

Gelman A, Rubin DB. Inference from iterative simulation using multiple sequences. *Stat Sci* 1992;**7**(4):457–72. doi: 10.1214/ss/1177011136

Spineli LM, Kalyvas C, Papadimitropoulou K. Continuous(ly) missing outcome data in network meta-analysis: a one-stage pattern-mixture model approach. *Stat Methods Med Res* 2021;**30**(4):958–75. doi: 10.1177/0962280220983544

Spineli LM. An empirical comparison of Bayesian modelling strategies for missing binary outcome data in network meta-analysis. *BMC Med Res Methodol* 2019;**19**(1):86. doi: 10.1186/s12874-019-0731-y

van Valkenhoef G, Dias S, Ades AE, Welton NJ. Automated generation of node-splitting models for assessment of inconsistency in network meta-analysis. *Res Synth Methods* 2016;**7**(1):80–93. doi: 10.1002/jrsm.1167

See Also

`autojags`, `jags`, `mtc.nodesplit.comparisons`, `nodesplit_plot`, `prepare_nodesplit`, `run_model`

Examples

```
data("nma.baker2009")

# Read results from 'run_model' (using the default arguments)
res <- readRDS(system.file('extdata/res_baker.rds', package = 'rnmamod'))


# Run random-effects node-splitting approach
# Note: Ideally, set 'n_iter' to 10000 and 'n_burnin' to 1000
run_nodesplit(full = res,
              n_chains = 3,
              n_iter = 1000,
              n_burnin = 100,
              n_thin = 1)
```

---

run_sensitivity *Perform sensitivity analysis for missing participant outcome data*

Description

Performs a sensitivity analysis by applying pairwise meta-analysis or network meta-analysis for a series of different scenarios about the informative missingness parameter.

Usage

```
run_sensitivity(
  full,
  assumption,
  mean_scenarios,
  var_misspar,
  n_chains,
  n_iter,
  n_burnin,
  n_thin,
  inits = NULL
)
```

Arguments

`full`	An object of S3 class `run_model`. See 'Value' in `run_model`.
`assumption`	Character string indicating the structure of the informative missingness parameter. Set `assumption` equal to one of the following two: `"HIE-ARM"`, or `"IDE-ARM"` (see 'Details'). The default argument is `"IDE-ARM"`. The abbreviations `"IDE"`, and `"HIE"` stand for identical, and hierarchical, respectively.

`mean_scenarios` A vector with numeric values for the mean of the normal distribution of the informative missingness parameter (see 'Details'). The vector should have a length equal to 5 or larger. The missing-at-random (MAR) assumption should be the median of the vector, so that the same number of informative scenarios appear before and after the MAR. The default scenarios are c(-log(3), -log(2), log(0.9999), log(2), log(3)) and c(-2, -1, 0, 1, 2) for binary and continuous outcome data, respectively.

`var_misspar` A positive non-zero number for the variance of the normal distribution of the informative missingness parameter. When the `measure` (defined in `run_model`) is `"OR"`, `"MD"`, or `"SMD"` the default argument is 1. When the `measure` is `"ROM"`, the default argument is 0.04.

`n_chains` Integer specifying the number of chains for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 2.

`n_iter` Integer specifying the number of Markov chains for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 10000.

`n_burnin` Integer specifying the number of iterations to discard at the beginning of the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 1000.

`n_thin` Integer specifying the thinning rate for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 1.

`inits` A list with the initial values for the parameters; an argument of the `jags` function of the R-package R2jags. The default argument is `NULL`, and JAGS generates the initial values.

Details

The model runs in `JAGS` and the progress of the simulation appears on the R console. The number of times `run_sensitivity` is used appears on the R console as a text in red and it equals the **number of scenarios** defined as *the square of the length of the vector specified in* `mean_scenarios` (see 'Examples'). The output of `run_sensitivity` is used as an S3 object by other functions of the package to be processed further and provide an end-user-ready output.

In the case of pairwise meta-analysis, `EM` and `tau` are estimated as many times as the number of scenarios considered. In the case of network meta-analysis, each possible pairwise comparison is estimated as many times as the number of scenarios considered.

The informative missingness parameter is assumed to differ only across the interventions of the dataset. Therefore, the user can specify the informative missingness parameter to be arm-specific *and* identical (`assumption = "IDE-ARM"`), or arm-specific *and* hierarchical (`assumption = "HIE-ARM"`) (Spineli et al., 2021).

The length of the vector specified in argument `mean_scenarios` should be equal to or more than 5 (a positive odd integer) to allow for an adequate number of scenarios. It is important that the number corresponding to the MAR assumption is the middle of the numbers in the vector specified in argument `mean_scenarios`. The **MAR assumption** constitutes the **primary analysis**. Under the informative missingness difference of means parameter (relevant for the raw and standardised mean diffenre), the MAR assumption equals 0. Under the informative missingness odds ratio parameter (IMOR; relevant for the odds ratio) and the informative missingness ratio of means (IMRoM; relevant for the ratio of means) parameter, the MAR assumption equals 1; however, both parameters are analysed in the logarithmic scale. We advise using the value 0.999 rather than 1 in mean_scenarios for the IMOR and IMRoM parameters; otherwise, the execution of the function will be stopped and the error 'Invalid parent values' will be printed on the R console.

Currently, there are no empirically-based prior distributions for the informative missingness parameters. The users may refer to Spineli (2019), Mavridis et al. (2015), Turner et al. (2015), and White et al. (2008) to determine mean_scenarios for an informative missingness mechanism and select a proper value for var_misspar.

run_sensitivity inherits the arguments data, measure, model, heter_prior, D, indic, base_risk, and ref from run_model (now contained in the argument full). This prevents specifying a different Bayesian model from that considered in the primary analysis (via run_model)–an exception in the assumption argument as it is restricted to only two character strings. Therefore, the user needs first to apply run_model, and then use run_sensitivity (see 'Examples').

The run_sensitivity function also returns the arguments measure, scenarios, D, heter, n_chains, n_iter, n_burnin, and n_thin as specified by the user to be inherited by other relevant functions of the package.

The model is updated until convergence using the autojags function of the R-package R2jags with 2 updates and number of iterations and thinning equal to n_iter and n_thin, respectively.

run_sensitivity can be used only when missing participant outcome data have been extracted for at least one trial. Otherwise, the execution of the function will be stopped and an error message will be printed on the R console.

Value

A list of R2jags outputs on the summaries of the posterior distribution, and the Gelman-Rubin convergence diagnostic (Gelman et al., 1992) of the following monitored parameters for a random-effects pairwise meta-analysis:

EM The estimated summary effect measure (according to the argument measure defined in run_model).

EM_LOR The estimated summary odd ratio in the logarithmic scale when measure = "RR" or measure = "RD".

tau The between-trial standard deviation. This element does not appear in the case of a fixed-effect pairwise meta-analysis.

In a random-effects network meta-analysis, EM refer to all possible pairwise comparisons of interventions in the network. Furthermore, tau is typically assumed to be common for all observed comparisons in the network.

Author(s)

Loukia M. Spineli

References

Gelman, A, Rubin, DB. Inference from iterative simulation using multiple sequences. *Stat Sci* 1992;**7**(4):457–72. doi: 10.1214/ss/1177011136

Mavridis D, White IR, Higgins JP, Cipriani A, Salanti G. Allowing for uncertainty due to missing continuous outcome data in pairwise and network meta-analysis. *Stat Med* 2015;**34**(5):721–41. doi: 10.1002/sim.6365

Spineli LM, Kalyvas C, Papadimitropoulou K. Quantifying the robustness of primary analysis results: A case study on missing outcome data in pairwise and network meta-analysis. *Res Synth Methods* 2021;**12**(4):475–90. doi: 10.1002/jrsm.1478

Spineli LM. An empirical comparison of Bayesian modelling strategies for missing binary outcome data in network meta-analysis. *BMC Med Res Methodol* 2019;**19**(1):86. doi: 10.1186/s12874-019-0731-y

Turner NL, Dias S, Ades AE, Welton NJ. A Bayesian framework to account for uncertainty due to missing binary outcome data in pairwise meta-analysis. *Stat Med* 2015;**34**(12):2062–80. doi: 10.1002/sim.6475

White IR, Higgins JP, Wood AM. Allowing for uncertainty due to missing data in meta-analysis–part 1: two-stage methods. *Stat Med* 2008;**27**(5):711–27. doi: 10.1002/sim.3008

See Also

autojags, jags, run_model

Examples

```
data("pma.taylor2004")

# Read results from 'run_model' (using the default arguments)
res <- readRDS(system.file('extdata/res_taylor.rds', package = 'rnmamod'))


# Perform the sensitivity analysis (default arguments)
# Note: Ideally, set 'n_iter' to 10000 and 'n_burnin' to 1000
run_sensitivity(full = res,
                assumption = "IDE-ARM",
                var_misspar = 1,
                n_chains = 3,
                n_iter = 1000,
                n_burnin = 100,
                n_thin = 5)
```

run_series_meta *Perform a series of Bayesian pairwise meta-analyses*

Description

Performs a Bayesian pairwise meta-analysis for each pairwise comparison with at least two trials in the network.

Usage

```
run_series_meta(full, n_chains, n_iter, n_burnin, n_thin, inits = NULL)
```

Arguments

full	An object of S3 class run_model. See 'Value' in run_model.
n_chains	Integer specifying the number of chains for the MCMC sampling; an argument of the jags function of the R-package R2jags. The default argument is 2.
n_iter	Positive integer specifying the number of Markov chains for the MCMC sampling; an argument of the jags function of the R-package R2jags. The default argument is 10000.
n_burnin	Positive integer specifying the number of iterations to discard at the beginning of the MCMC sampling; an argument of the jags function of the R-package R2jags. The default argument is 1000.
n_thin	Positive integer specifying the thinning rate for the MCMC sampling; an argument of the jags function of the R-package R2jags. The default argument is 1.
inits	A list with the initial values for the parameters; an argument of the jags function of the R-package R2jags. The default argument is NULL, and JAGS generates the initial values.

Details

run_series_meta inherits the arguments data, measure, model, assumption, heter_prior, mean_misspar, and var_misspar from run_model (now contained in the argument full). This prevents specifying a different Bayesian model from that considered in run_model. Therefore, the user needs first to apply run_model, and then use run_series_meta (see 'Examples').

For a binary outcome, when measure is "RR" (relative risk) or "RD" (risk difference) in run_model, run_series_meta currently performs a series of pairwise meta-analysis using the odds ratio as effect measure for being the **base-case** effect measure in run_model for a binary outcome (see also 'Details' in run_model).

run_series_meta runs a series of Bayesian pairwise meta-analyses in JAGS. The progress of the simulation appears on the R console. The number of times the function is used is also printed on the console (in red) and is equal to the number of observed pairwise comparisons in the network (see 'Examples'). The model is updated until convergence using the autojags function of the R-package R2jags with 2 updates and number of iterations and thinning equal to n_iter and n_thin, respectively.

The output of run_series_meta is not end-user-ready. The series_meta_plot function inherits the output of run_series_meta as an S3 object and processes it further to provide an end-user-ready output.

run_series_meta can be used only for a network of interventions. In the case of two interventions, the execution of the function will be stopped and an error message will be printed on the R console.

Value

An R2jags output on the summaries of the posterior distribution, and the Gelman-Rubin convergence diagnostic (Gelman et al., 1992) of the following monitored parameters:

EM — The summary effect estimate (according to the argument measure defined in run_model) of each observed pairwise comparison with at least two trials in the network.

tau — The between-trial standard deviation for pairwise comparisons with at least two trials, when the random-effects model has been specified.

single — A binary vector that indicates the comparisons in EM with one trial.

Author(s)

Loukia M. Spineli

References

Gelman A, Rubin DB. Inference from iterative simulation using multiple sequences. *Stat Sci* 1992;**7**(4):457–72. doi: 10.1214/ss/1177011136

See Also

jags, run_model, series_meta_plot

Examples

```
data("nma.dogliotti2014")

# Show the first six trials of the dataset (one-trial-per-row format)
head(nma.dogliotti2014)

# Read results from 'run_model' (using the default arguments)
res <- readRDS(system.file('extdata/res_dogliotti.rds', package = 'rnmamod'))


# Run separate random-effects pairwise meta-analyses
# Note: Ideally, set 'n_iter' to 10000 and 'n_burnin' to 1000
run_series_meta(full = res,
                n_chains = 3,
                n_iter = 1000,
                n_burnin = 100,
                n_thin = 1)
```

run_ume *Perform the unrelated mean effects model*

Description

Performs the unrelated mean effects model of Dias et al. (2013) that has been refined (Spineli, 2021) and extended to address aggregate binary and continuous missing participant outcome data via the pattern-mixture model (Spineli et al. 2021; Spineli, 2019). This model offers a global evaluation of the plausibility of the consistency assumption in the network.

Usage

```
run_ume(full, n_iter, n_burnin, n_chains, n_thin, inits = NULL)
```

Arguments

`full` An object of S3 class `run_model`. See 'Value' in `run_model`.

`n_iter` Positive integer specifying the number of Markov chains for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 10000.

`n_burnin` Positive integer specifying the number of iterations to discard at the beginning of the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 1000.

`n_chains` Positive integer specifying the number of chains for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 2.

`n_thin` Positive integer specifying the thinning rate for the MCMC sampling; an argument of the `jags` function of the R-package R2jags. The default argument is 1.

`inits` A list with the initial values for the parameters; an argument of the `jags` function of the R-package R2jags. The default argument is `NULL`, and JAGS generates the initial values.

Details

`run_ume` inherits the arguments `data`, `measure`, `model`, `assumption`, `heter_prior`, `mean_misspar`, `var_misspar`, and `ref` from `run_model`. This prevents specifying a different Bayesian model from that considered in `run_model`.Therefore, the user needs first to apply `run_model`, and then use `run_ume` (see 'Examples').

The `run_ume` function also returns the arguments `data`, `model`, `measure`, `assumption`, `n_chains`, `n_iter`, `n_burnin`, and `n_thin` as specified by the user to be inherited by other relevant functions of the package.

Initially, `run_ume` calls the `improved_ume` function to identify the *frail comparisons*, that is, comparisons between non-baseline interventions in multi-arm trials not investigated in any two-arm or multi-arm trial of the network (Spineli, 2021). The 'original' model of Dias et al. (2013)

omits the frail comparisons from the estimation process. Consequently, the number of estimated summary effects is less than those obtained by performing separate pairwise meta-analyses (see run_series_meta).

For a binary outcome, when measure is "RR" (relative risk) or "RD" (risk difference) in run_model, run_ume currently considers the odds ratio as effect measure for being the **base-case** effect measure in run_model for a binary outcome (see also 'Details' in run_model).

run_ume calls the prepare_ume function which contains the WinBUGS code as written by Dias et al. (2013) for binomial and normal likelihood to analyse binary and continuous outcome data, respectively. prepare_ume has been extended to incorporate the pattern-mixture model with informative missingness parameters for binary and continuous outcome data (see 'Details' in run_model). prepare_ume has also been refined to account for the multi-arm trials by assigning conditional univariate normal distributions on the underlying trial-specific effect size of comparisons with the baseline arm of the multi-arm trial (Spineli, 2021).

run_ume runs Bayesian unrelated mean effects model in JAGS. The progress of the simulation appears on the R console. The model is updated until convergence using the autojags function of the R-package R2jags with 2 updates and number of iterations and thinning equal to n_iter and n_thin, respectively.

The output of run_ume is not end-user-ready. The ume_plot function uses the output of run_ume as an S3 object and processes it further to provide an end-user-ready output.

run_ume can be used only for a network of interventions. In the case of two interventions, the execution of the function will be stopped and an error message will be printed on the R console.

Value

An R2jags output on the summaries of the posterior distribution, and the Gelman-Rubin convergence diagnostic (Gelman et al., 1992) of the following monitored parameters:

EM The summary effect estimate (according to the argument measure defined in run_model) for each pairwise comparison observed in the network.

dev_o The deviance contribution of each trial-arm based on the observed outcome.

hat_par The fitted outcome at each trial-arm.

tau The between-trial standard deviation (assumed common across the observed pairwise comparisons) for the whole network, when a random-effects model has been specified.

m_tau The between-trial standard deviation (assumed common across the observed pairwise comparisons) for the subset of multi-arm trials, when a random-effects model has been specified.

The output also includes the following elements:

leverage_o The leverage for the observed outcome at each trial-arm.

sign_dev_o The sign of the difference between observed and fitted outcome at each trial-arm.

model_assessment

A data-frame on the measures of model assessment: deviance information criterion, number of effective parameters, and total residual deviance.

jagsfit — An object of S3 class jags with the posterior results on all monitored parameters to be used in the mcmc_diagnostics function.

Furthermore, run_ume returns a character vector with the pairwise comparisons observed in the network, obs_comp, and a character vector with comparisons between the non-baseline interventions observed in multi-arm trials only, frail_comp. Both vectors are used in ume_plot function.

Author(s)

Loukia M. Spineli

References

Dias S, Welton NJ, Sutton AJ, Caldwell DM, Lu G, Ades AE. Evidence synthesis for decision making 4: inconsistency in networks of evidence based on randomized controlled trials. *Med Decis Making* 2013;**33**(5):641–56. doi: 10.1177/0272989X12455847

Gelman A, Rubin DB. Inference from iterative simulation using multiple sequences. *Stat Sci* 1992;**7**(4):457–72. doi: 10.1214/ss/1177011136

Spineli LM. A Revised Framework to Evaluate the Consistency Assumption Globally in a Network of Interventions. *Med Decis Making* 2021. doi: 10.1177/0272989X211068005

Spineli LM, Kalyvas C, Papadimitropoulou K. Continuous(ly) missing outcome data in network meta-analysis: a one-stage pattern-mixture model approach. *Stat Methods Med Res* 2021;**30**(4):958–75. doi: 10.1177/0962280220983544

Spineli LM. An empirical comparison of Bayesian modelling strategies for missing binary outcome data in network meta-analysis. *BMC Med Res Methodol* 2019;**19**(1):86. doi: 10.1186/s12874-019-0731-y

See Also

autojags, jags, prepare_ume, run_model, run_series_meta, ume_plot

Examples

```
data("nma.liu2013")

# Read results from 'run_model' (using the default arguments)
res <- readRDS(system.file('extdata/res_liu.rds', package = 'rnmamod'))


# Run random-effects unrelated mean effects model
# Note: Ideally, set 'n_iter' to 10000 and 'n_burnin' to 1000
run_ume(full = res,
        n_chains = 3,
        n_iter = 1000,
        n_burnin = 100,
        n_thin = 1)
```

scatterplots_dev *Deviance scatterplots*

Description

Illustrates the posterior mean of deviance contribution of the individual data points under the unrelated mean effects model (via `run_ume`) against the posterior mean of deviance contribution under the consistency model (via `run_model`).

Usage

```
scatterplots_dev(full, ume, colour)
```

Arguments

`full`	A numeric vector with the posterior mean of deviance obtained using the consistency model (see 'Value' in `run_model`).
`ume`	A numeric vector with the posterior mean of deviance obtained using the unrelated mean effects model (see 'Value' in `run_ume`).
`colour`	A string to define the colour of the points in the plot.

Details

`scatterplots_dev` is integrated in the `ume_plot` function to compare the models regarding the posterior mean of deviance. This scatterplot has also been considered by Dias et al. (2013). When the majority of data points are scattered across the diagonal line, we may conclude that the compared models have a good agreement. Data points systematically scattered above or below the diagonal line may contribute more to the poor fit of the unrelated mean effects model and the consistency model, respectively.

Value

A scatterplot of the posterior mean deviance contribution of the individual data points from the unrelated mean effects model against those from the consistency model. Each data point corresponds to a trial-arm indicated by a pair of numbers. The first number refers to the position of the trial in the dataset, and the second number refers to the corresponding trial-arm (see 'Arguments' and 'Value' in `data_preparation`).

Author(s)

Loukia M. Spineli

References

Dias S, Welton NJ, Sutton AJ, Caldwell DM, Lu G, Ades AE. Evidence synthesis for decision making 4: inconsistency in networks of evidence based on randomized controlled trials. *Med Decis Making* 2013a;**33**(5):641–56. doi: 10.1177/0272989X12455847

See Also

data_preparation, run_model, run_ume, ume_plot

scatterplot_sucra *Scatterplot of SUCRA values*

Description

Creates a scatterplot of the SUCRA values from the network meta-analysis and the network meta-regression for a specified level or value of the investigated covariate.

Usage

```
scatterplot_sucra(full, reg, cov_value, drug_names)
```

Arguments

full	An object of S3 class run_model. See 'Value' in run_model.
reg	An object of S3 class run_metareg. See 'Value' in run_metareg.
cov_value	A list of two elements in the following order: 1) a number for the covariate value of interest (see 'Arguments' in run_metareg), and 2) a character to indicate the name of the covariate. See also 'Details'.
drug_names	A vector of labels with the name of the interventions in the order they appear in the argument data of run_model. If drug_names is not defined, the order of the interventions as they appear in data is used, instead.

Details

The names of the interventions appear above each point in the plot. Three coloured rectangles are drawn in the scatterplot: a red rectangle for SUCRA values up to 50%, a yellow rectangular for SUCRA values between 50% and 80%, and a green rectangle for SUCRA values over 80%. Interventions falling at the green area are considered as the highest ranked interventions, whilst interventions falling at the red area are considered as the lowest ranked interventions.

When the covariate is binary, specify in the second element of cov_value the name of the level for which the scatterplot will be created.

scatterplot_sucra is integrated in metareg_plot.

scatterplot_sucra can be used only for a network of interventions. Otherwise, the execution of the function will be stopped and an error message will be printed on the R console.

Value

A scatterplot of the SUCRA values under the network meta-analysis (y-axis) against the SUCRA values under the network meta-regression (x-axis) for a specified level or value of the investigated covariate.

Author(s)

Loukia M. Spineli

References

Salanti G, Ades AE, Ioannidis JP. Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: an overview and tutorial. *J Clin Epidemiol* 2011;**64**(2):163–71. doi: 10.1016/j.jclinepi.2010.03.016

See Also

`metareg_plot`, `run_metareg`, `run_model`

`series_meta_plot` *End-user-ready results for a series of pairwise meta-analyses*

Description

Facilitates the comparison of the consistency model (via `run_model`) with a series of pairwise meta-analyses (via `run_series_meta`) regarding the estimated summary effect sizes and between-trial standard deviation for comparisons with at least two trials.

Usage

```
series_meta_plot(full, meta, drug_names, save_xls)
```

Arguments

`full`	An object of S3 class `run_model`. See 'Value' in `run_model`.
`meta`	An object of S3 class `run_series_meta`. See 'Value' in `run_series_meta`.
`drug_names`	A vector of labels with the name of the interventions in the order they appear in the argument `data` of `run_model`. If `drug_names` is not defined, the order of the interventions as they appear in `data` is used, instead.
`save_xls`	Logical to indicate whether to export the tabulated results to an 'xlsx' file (via the `write_xlsx` function of the R-package writexl) at the working directory of the user. The default is `FALSE` (do not export).

Details

`series_meta_plot` can be used only for a network of interventions. Otherwise, the execution of the function will be stopped and an error message will be printed on the R console.

For a binary outcome, when `measure` is "RR" (relative risk) or "RD" (risk difference) in `run_model`, `series_meta_plot` currently presents the results in the odds ratio for being the **base-case** effect measure in `run_model` for a binary outcome (see also 'Details' in `run_model`).

The user can detect any inconsistencies in the estimated effects from the compared models and explore the gains in precision stemming from applying network meta-analysis. Furthermore, the user can investigate the plausibility of the common between-trial heterogeneity assumption which is typically considered in network meta-analysis.

Value

The R console prints the data-frame with the estimated summary effect sizes and between-trial standard deviation of comparisons under both models. The comparisons have at least two trials. In the case of a fixed-effect model, the data-frame is printed without the results on the between-trial standard deviation.

Furthermore, `series_meta_plot` exports the data-frame to an 'xlsx' file at the working directory of the user.

`series_meta_plot` returns a panel of two forest plots: (1) a forest plot on the posterior median and 95% credible interval of the summary effect size for the observed comparisons from network meta-analysis and the corresponding pairwise meta-analyses, and (2) a forest plot on the posterior median and 95% credible interval of the between-trial standard deviation for these observed comparisons. The estimated median and 95% credible intervals of the between-trial standard deviation from network meta-analysis appear in the forest plot as a solid and two dotted parallel blue lines, respectively. The different levels of heterogeneity appear as green, yellow, orange, and red rectangles to indicate a low, reasonable, fairly high, and fairly extreme heterogeneity, respectively, following the classification of Spiegelhalter et al. (2004). When a fixed-effect model has been fitted, only the forest plot on the estimated summary effect sizes is shown.

Author(s)

Loukia M. Spineli

References

Spiegelhalter DJ, Abrams KR, Myles JP. Bayesian approaches to clinical trials and health-care evaluation. John Wiley and Sons, Chichester, 2004.

See Also

`run_model`, `run_series_meta`, `write_xlsx`

Examples

```
data("nma.dogliotti2014")

# Read results from 'run_model' (using the default arguments)
res <- readRDS(system.file('extdata/res_dogliotti.rds', package = 'rnmamod'))

# Read results from 'run_series_meta' (using the default arguments)
meta <- readRDS(system.file('extdata/meta_dogliotti.rds',
                package = 'rnmamod'))

# The names of the interventions in the order they appear in the dataset
interv_names <- c("placebo", "aspirin", "aspirin plus clopidogrel",
                  "dabigatran 110 mg", "dabigatran 150 mg", "rivaroxaban",
                  "vitamin K antagonist", "apixaban")

# Plot the results from both models
series_meta_plot(full = res,
                 meta = meta,
                  drug_names = interv_names)
```

taylor_continuous *Pattern-mixture model with Taylor series for continuous outcome*

Description

Applies the pattern-mixture model under a specific assumption about the informative missingness parameter in trial-arms with **continuous** missing participant outcome data and uses the Taylor series to obtain the effect size and standard error for each trial (Mavridis et al., 2015).

Usage

```
taylor_continuous(data, measure, mean_value, var_value, rho)
```

Arguments

`data` A data-frame in the long arm-based format. Two-arm trials occupy one row in the data-frame. Multi-arm trials occupy as many rows as the number of possible comparisons among the interventions. See 'Format' for the specification of the columns.

`measure` Character string indicating the effect measure with values `"MD"`, `"SMD"`, or `"ROM"` for the mean difference, standardised mean difference, and ratio of means, respectively.

`mean_value` A numeric value for the mean of the normal distribution of the informative missingness parameter. The same value is considered for all trial-arms of the dataset. The default argument is 0 and corresponds to the missing-at-random assumption. For the informative missingness ratio of means, the mean value is defined in the logarithmic scale.

`var_value` A positive non-zero number for the variance of the normal distribution of the informative missingness parameter. When the `measure` is `"MD"`, or `"SMD"` the default argument is 1; when the `measure` is `"ROM"` the default argument is 0.04. The same value is considered for all trial-arms of the dataset.

`rho` A numeric value in the interval [-1, 1] that indicates the correlation coefficient between two informative missingness parameters in a trial. The same value is considered across all trials of the dataset. The default argument is 0 and corresponds to uncorrelated missingness parameters.

Format

The columns of the data-frame in the argument `data` refer to the following ordered elements for a continuous outcome:

id A unique identifier for each trial.
y1 The observed mean outcome in the first arm of the comparison.
y2 The observed mean outcome in the second arm of the comparison.

sd1 The observed standard deviation of the outcome in the first arm of the comparison.
sd2 The observed standard deviation of the outcome in the second arm of the comparison.
m1 The number of missing participants in the first arm of the comparison.
m2 The number of missing participants in the second arm of the comparison.
n1 The number randomised in the first arm of the comparison.
n2 The number randomised in the second arm of the comparison.
t1 An identifier for the intervention in the first arm of the comparison.
t2 An identifier for the intervention in the second arm of the comparison.

Details

The `taylor_continuous` function is integrated in the `unrelated_effects_plot` function.

Value

A data-frame that additionally includes the following elements:

`EM` The effect size adjusted for the missing participants and obtained using the Taylor series.

`se.EM` The standard error of the effect size adjusted for the missing participants and obtained using the Taylor series.

Author(s)

Loukia M. Spineli

References

Mavridis D, White IR, Higgins JP, Cipriani A, Salanti G. Allowing for uncertainty due to missing continuous outcome data in pairwise and network meta-analysis. *Stat Med* 2015;**34**(5):721–41. doi: 10.1002/sim.6365

See Also

`run_model`, `unrelated_effects_plot`

`taylor_imor` *Pattern-mixture model with Taylor series for a binary outcome*

Description

Applies the pattern-mixture model under a specific assumption about the informative missingness odds ratio in trial-arms with **binary** missing participant outcome data and uses the Taylor series to obtain the odds ratio (in the logarithmic scale) and standard error for each trial (White et al., 2008).

Usage

```
taylor_imor(data, mean_value, var_value, rho)
```

Arguments

`data`	A data-frame in the long arm-based format. Two-arm trials occupy one row in the data-frame. Multi-arm trials occupy as many rows as the number of possible comparisons among the interventions. See 'Format' for the specification of the columns.
`mean_value`	A numeric value for the mean of the normal distribution of the informative missingness odds ratio in the logarithmic scale. The same value is considered for all trial-arms of the dataset. The default argument is 0 and corresponds to the missing-at-random assumption.
`var_value`	A positive non-zero number for the variance of the normal distribution of the informative missingness odds ratio in the logarithmic scale. The default argument is 1.
`rho`	A numeric value in the interval [-1, 1] that indicates the correlation coefficient between two missingness parameters in a trial. The same value is considered across all trials of the dataset. The default argument is 0 and corresponds to uncorrelated missingness parameters.

Format

The columns of the data-frame in the argument `data` refer to the following ordered elements for a binary outcome:

id	A unique identifier for each trial.
r1	The observed number of events in the first arm of the comparison.
r2	The observed number of events in the second arm of the comparison.
m1	The number of missing participants in the first arm of the comparison.
m2	The number of missing participants in the second arm of the comparison.
n1	The number of participants randomised in the first arm of the comparison.
n2	The number of participants randomised in the second arm of the comparison.
t1	An identifier for the intervention in the first arm of the comparison.
t2	An identifier for the intervention in the second arm of the comparison.

Details

The `taylor_imor` function is integrated in the `unrelated_effects_plot` function.

Value

A data-frame that additionally includes the following elements:

`EM`	The odds ratio in the logarithmic scale (log OR) adjusted for missing participants and obtained using the Taylor series.

se.EM The standard error of the log OR adjusted for missing participants and obtained using the Taylor series.

Author(s)

Loukia M. Spineli

References

White IR, Higgins JP, Wood AM. Allowing for uncertainty due to missing data in meta-analysis–part 1: two-stage methods. *Stat Med* 2008;**27**(5):711–27. doi: 10.1002/sim.3008

See Also

run_model, unrelated_effects_plot

---

trans_quality *Rating the quality of transitivity evaluation*

Description

Classifies a systematic review with multiple interventions as having low, unclear or high quality regarding the transitivity evaluation.

Usage

```
trans_quality(
  plan_protocol,
  plan_methods,
  report_results,
  discuss_trans,
  proper_table
)
```

Arguments

plan_protocol Character string that indicates how the systematic review planned in the protocol to assess the transitivity assumption. The following values can be considered: "Both", "No", "No protocol", "Only direct methods", and "Only indirect methods". "Both" implies that at least one direct and one indirect method has been used.

plan_methods Character string that indicates whether the systematic review described in the methods section a strategy to assess the transitivity assumption. The following values can be considered: "Yes", and "No".

`report_results` Character string that indicates whether the systematic review reported in the results section the transitivity evaluation and which strategy was employed. The following values can be considered: `"Both"`, `"No"`, `"Only direct methods"`, and `"Only indirect methods"`. `"Both"` implies that at least one direct and one indirect method has been used.

`discuss_trans` Character string that indicates whether the systematic review discussed the transitivity assumption and which model parameters where considered. The following values can be considered: `"Both"`, `"No"`, `"Only treatment effects"`, `"Other parameter"`, and `"NMA not conducted"`. `"Both"` implies that the treatment effect and other model parameters were considered.

`proper_table` Character string that indicates whether the systematic review reported a proper table of characteristics. The following values can be considered: `"No"`, `"No table"`, and `"Yes"`.

Details

A systematic review with `"Low"` quality of transitivity evaluation does not provide a protocol, nor describe the evaluation strategy in the methods section, does not report the evaluation results, nor discusses the transitivity evaluation and does not provide a table of characteristics. On the contrary, a systematic review with `"High"` quality of transitivity evaluation provides an evaluation plan in the protocol (including at least one direct method), describes the evaluation strategy in the methods section (including at least one direct method), reports the evaluation results in the results section, discusses the transitivity evaluation while considering at least one model parameter (if NMA has been conducted), and provides a proper table of characteristics. Otherwise, the systematic review is judged to have an `"Unclear"` quality of transitivity evaluation.

Value

A character with value `"Low"`, `"Unclear"`, or `"High"` to indicate low, unclear, or high-quality of transitivity evaluation.

Author(s)

Loukia M. Spineli

References

Spineli LM, Kalyvas C, Seide SE, Papadimitropoulou K. Low awareness of the transitivity assumption in complex networks of interventions: empirical evidence from 356 network meta-analyses. 2022 *submitted*

`ume_plot` *End-user-ready results for the unrelated mean effects model*

Description

ume_plot hosts a toolkit of functions that facilitates the comparison of the consistency model (via run_model) with the unrelated mean effects model (via run_ume) regarding the posterior summaries of the summary effect size for the pairwise comparisons observed in the network, the between-trial standard deviation (*tau*) and model assessment parameters.

Usage

```
ume_plot(full, ume, drug_names, save_xls)
```

Arguments

full	An object of S3 class run_model. See 'Value' in run_model.
ume	An object of S3 class run_ume. See 'Value' in run_ume.
drug_names	A vector of labels with the name of the interventions in the order they appear in the argument data of run_model. If drug_names is not defined, the order of the interventions as they appear in data is used, instead.
save_xls	Logical to indicate whether to export the tabulated results to an 'xlsx' file (via the write_xlsx function of the R-package writexl) to the working directory of the user. The default is FALSE (do not export).

Details

The deviance information criterion (DIC) of the consistency model is compared with the DIC of the unrelated mean effects model (Dias et al., 2013). If the difference in DIC exceeds 5, the unrelated mean effects model is preferred. If the difference in DIC is less than -5, the consistency is preferred; otherwise, there is little to choose between the compared models.

For a binary outcome, when measure is "RR" (relative risk) or "RD" (risk difference) in run_model, ume_plot currently presents the results from network meta-analysis and unrelated mean effects in the odds ratio for being the **base-case** effect measure in run_model for a binary outcome (see also 'Details' in run_model).

Furthermore, ume_plot exports table_effect_size and table_model_assessment to separate 'xlsx' files (via the write_xlsx function) to the working directory of the user.

ume_plot can be used only for a network of interventions. In the case of two interventions, the execution of the function will be stopped and an error message will be printed on the R console.

Value

ume_plot prints on the R console a message on the most parsimonious model (if any) based on the DIC (red text). Then, the function returns the following list of elements:

table_effect_size
: The posterior median, posterior standard deviation, and 95% credible interval of the summary effect size for each pairwise comparison observed in the network under the consistency model and the unrelated mean effects model.

table_model_assessment
: The DIC, number of effective parameters, and total residual deviance under the consistency model and the unrelated mean effects model (Spiegelhalter et al., 2002).

table_tau
: The posterior median and 95% credible interval of *tau* under the consistency model and the unrelated mean effects model. When a fixed-effect model has been performed, ume_plot does not return this element.

scatterplots
: The scatterplot and the Bland-Altman plot on the posterior mean deviance contribution of the individual data points under the consistency model and the unrelated mean effects model. See 'Details' and 'Value' in scatterplots_dev and bland_altman_plot, respectively.

levarage_plots
: The leverage plot under the consistency model and the unrelated mean effects model, separately. See 'Details' and 'Value' in leverage_plot.

intervalplots
: A panel of interval plots on the summary effect size under the consistency model and the unrelated mean effects model for each pairwise comparison observed in the network. See 'Details' and 'Value' in intervalplot_panel_ume.

Author(s)

Loukia M. Spineli

References

Dias S, Welton NJ, Sutton AJ, Caldwell DM, Lu G, Ades AE. Evidence synthesis for decision making 4: inconsistency in networks of evidence based on randomized controlled trials. *Med Decis Making* 2013;**33**(5):641–56. doi: 10.1177/0272989X12455847

Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A. Bayesian measures of model complexity and fit. *J R Stat Soc B* 2002;**64**(4):583–396. doi: 10.1111/1467-9868.00353

See Also

bland_altman_plot, intervalplot_panel_ume, leverage_plot, run_model, run_ume, write_xlsx

Examples

```
data("nma.liu2013")


# Read results from 'run_model' (using the default arguments)
res <- readRDS(system.file('extdata/res_liu.rds', package = 'rnmamod'))

# Read results from 'run_ume' (using the default arguments)
ume <- readRDS(system.file('extdata/ume_liu.rds', package = 'rnmamod'))

# The names of the interventions in the order they appear in the dataset
interv_names <- c("placebo", "pramipexole", "serotonin-norepinephrine
                  reuptake inhibitor", "serotonin reuptake inhibitor",
                  "tricyclic antidepressant", "pergolide")
```

```
# Plot the results from both models
ume_plot(full = res,
         ume = ume,
         drug_names = interv_names)
```

unrelated_effects_plot

End-user-ready results for unrelated trial effects model

Description

Performs the unrelated trial effects model (also known as fixed effects model) and illustrates the results of each trial and corresponding pairwise comparison.

Usage

```
unrelated_effects_plot(
  data,
  measure,
  char,
  drug_names,
  trial_names,
  mean_misspar,
  var_misspar,
  rho,
  save_xls
)
```

Arguments

`data`	A data-frame of a one-trial-per-row format containing arm-level data of each trial. See 'Format' in `run_model`.
`measure`	Character string indicating the effect measure with values `"OR"`, `"MD"`, `"SMD"`, or `"ROM"` for the odds ratio, mean difference, standardised mean difference and ratio of means, respectively.
`char`	A data-frame of three columns and number of rows equal to the number of trials in `data`. Each column refers to a trial-characteristic with **nominal** elements.
`drug_names`	A vector of labels with the name of the interventions in the order they appear in the argument `data`. If `drug_names` is not defined, the order of the interventions as they appear in `data` is used, instead.
`trial_names`	A vector of labels with the name of the trials in the order they appear in the argument `data`. If `trial_names` is not defined, the order of the trials as they appear in `data` is used, instead.

`mean_misspar` A numeric value for the mean of the normal distribution of the informative missingness parameter (see 'Details'). The default argument is 0 and corresponds to the missing-at-random assumption. The same value is considered across all trials of the dataset.

`var_misspar` A positive non-zero number for the variance of the normal distribution of the informative missingness parameter. When the `measure` is `"OR"`, `"MD"`, or `"SMD"` the default argument is 1. When the `measure` is `"ROM"` the default argument is 0.04. The same value is considered across all trials of the dataset.

`rho` A numeric value in the interval [-1, 1] that indicates the correlation coefficient between two informative missingness parameters in a trial. The same value is considered across all trials of the dataset. The default argument is 0 and corresponds to uncorrelated missingness parameters.

`save_xls` Logical to indicate whether to export the tabulated results to an 'xlsx' file (via the `write_xlsx` function of the R-package writexl) to the working directory of the user. The default is `FALSE` (do not export).

Details

The unrelated trial effects model may be an alternative to network meta-analysis, when the latter is not deemed appropriate (e.g., there is considerable statistical heterogeneity, or substantial intransitivity). In the presence of missing participant outcome data, the effect size and standard error are adjusted by applying the pattern-mixture model with Taylor series in trial-arms with reported missing participants (Mavridis et al., 2015; White et al., 2008). The `unrelated_effects_plot` function calls the `taylor_imor` and `taylor_continuous` functions (for a binary and continuous outcome, respectively) to employ pattern-mixture model with Taylor series. The `unrelated_effects_plot` function considers the informative missingness odds ratio in the logarithmic scale for binary outcome data (White et al., 2008), the informative missingness difference of means when `measure` is `"MD"` or `"SMD"`, and the informative missingness ratio of means in the logarithmic scale when `measure` is `"ROM"` (Mavridis et al., 2015).

The number of interval plots equals the number of observed comparisons in the network. In each interval plot, the y-axis refers to all trials of the network and x-axis refers to the selected effect measure. The odds ratio and ratio of means are calculated in the logarithmic scale but they are reported in their original scale after exponentiation.

`unrelated_effects_plot` depicts all three characteristics for each trial using different colours, line-types and point-shapes for the corresponding 95% confidence interval and point estimate. Ideally, each characteristic should have no more than three categories; otherwise, the plot becomes cluttered. For now, the `unrelated_effects_plot` function uses the default colour palette, line-types and point-shapes.

Value

A panel of interval plots for each observed comparison in the network, when there are up to 15 trials in the `data`. Otherwise, `unrelated_effects_plot` exports a data-frame to an 'xlsx' file at the working directory of the user. This data-frame includes the `data` in the long format, the within-trial effect measure and 95% confidence interval of the corresponding comparisons, the interventions compared, and the three characteristics (as defined in `char`). For datasets with more than 15 trials, the plot becomes cluttered and it is difficult to identify the trial-names. Hence, exporting the results in an Excel file is a viable alternative.

Author(s)

Loukia M. Spineli

References

Mavridis D, White IR, Higgins JP, Cipriani A, Salanti G. Allowing for uncertainty due to missing continuous outcome data in pairwise and network meta-analysis. *Stat Med* 2015;**34**(5):721–41. doi: 10.1002/sim.6365

White IR, Higgins JP, Wood AM. Allowing for uncertainty due to missing data in meta-analysis–part 1: two-stage methods. *Stat Med* 2008;**27**(5):711–27. doi: 10.1002/sim.3008

See Also

`run_model`, `taylor_continuous`, `taylor_imor`, `write_xlsx`

Index