Package: rnmamod (via r-universe)

October 19, 2026

Type Package

Title Bayesian Network Meta-Analysis with Missing Participants

Version 0.5.1

Date 2026-07-12

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, and evaluation of the consistency assumption via the node-splitting approach and the unrelated mean effects model (original and revised model proposed by Spineli, (2021) <doi:10.1177/0272989X211068005>). Missing participant outcome data are addressed in all models of the package (see Spineli, (2019) <doi:10.1186/s12874-019-0731-y>, Spineli et al., (2019) <doi:10.1002/sim.8207>, Spineli, (2019) <doi:10.1016/j.jclinepi.2018.09.002>, and Spineli et al., (2021) <doi:10.1177/0962280220983544>). The robustness to primary analysis results can also be investigated using a novel intuitive index (see Spineli et al., (2021) <doi:10.1002/jrsm.1478> and Spineli et al., (2021) <doi:10.1186/s12916-021-02195-y>). Methods to evaluate the transitivity assumption using trial dissimilarities and hierarchical clustering are provided (see Spineli, (2024) <doi:10.1186/s12874-024-02436-7>, and Spineli et al., (2025) <doi:10.1002/sim.70068>). A novel index to facilitate interpretation of local inconsistency is also available (see Spineli, (2024) <doi:10.1186/s13643-024-02680-4> and Spineli, (2025) <doi:10.1186/s13643-025-02984-z>). 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, https://loukiaspin.github.io/rnmamod/

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

Depends R (>= 4.0.0)

Imports cluster, coda (>= 0.13), dendextend, gemtc, ggfittext, ggplot2, ggpubr, ggrepel, heatmaply, igraph, knitr, MASS, Matrix, R2jags, reshape2, scales, stringr, writexl

Suggests metafor, netmeta, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.3

NeedsCompilation no

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

Repository https://cran.r-universe.dev

Date/Publication 2026-07-13 07:10:02 UTC

RemoteUrl https://github.com/cran/rnmamod

RemoteRef HEAD

RemoteSha fe6c2a011d94f72ac96124f3aa899975223d264e

Contents

rnmamod-package 4
balloon_plot 8
baseline_model 10
bland_altman_plot 12
comp_clustering 13

covar_contribution_plot . . . 18
data_preparation . . . 20
dendro_heatmap . . . 22
describe_network . . . 24
distr_characteristics . . . 25
forestplot . . . 27
forestplot_juxtapose . . . 29
forestplot_metareg . . . 31
gower_distance . . . 33
heatmap_missing_dataset . . . 34
heatmap_missing_network . . . 36
heatmap_robustness . . . 37
heter_density_plot . . . 39
heterogeneity_param_prior . . . 42
improved_ume . . . 43
inconsistency_variance_prior . . . 45
internal_measures_plot . . . 46
intervalplot_panel_ume . . . 48
kld_barplot . . . 49
kld_inconsistency . . . 50
kld_inconsistency_user . . . 53
kld_measure . . . 57
league_heatmap . . . 58
league_heatmap_pred . . . 61
league_table_absolute . . . 64
league_table_absolute_user . . . 66
leverage_plot . . . 68
mcmc_diagnostics . . . 69
metareg_plot . . . 71
miss_characteristics . . . 73
missingness_param_prior . . . 76
netplot . . . 78
nma.baker2009 . . . 81
nma.bottomley2011 . . . 82
nma.dogliotti2014 . . . 82
nma.fluoride.donegan2018 . . . 83
nma.liu2013 . . . 84
nma.malaria.donegan2018 . . . 84
nma.schwingshackl2014 . . . 85
nma.stowe2011 . . . 86
nodesplit_plot . . . 86
plot_study_dissimilarities . . . 89
pma.hetrick2012 . . . 91
pma.taylor2004 . . . 91
prepare_model . . . 92
prepare_nodesplit . . . 94
prepare_ume . . . 95
rankosucra_plot . . . 96

robustness_index 98
robustness_index_user 100
run_metareg 103
run_model 106
run_nodesplit 113
run_sensitivity 116
run_series_meta 120
run_ume 122
scatterplot_sucra 125
scatterplots_dev 126
series_meta_plot 127
study_perc_contrib 129
table_tau2_prior 131
taylor_continuous 132
taylor_imor 134
ume_plot 136
unrelated_effects_plot 138

Index **141**

`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`);
- calculation of the mean and standard deviation for the prior distribution for the inconsistency variance, and the median inconsistency standard deviation (`inconsistency_variance_prior`) based on a selected empirical prior distribution for the between-study variance (proposed by Turner et al., 2015 and Rhodes et al., 2015);

- a panel of density plots for each target comparison (based on the back-calculation or the node-splitting approach) illustrating the posterior distribution of direct and indirect estimates, the inconsistency parameter estimate and 95 value and a percent stacked bar plot on the percentage contribution of approximating direct with indirect estimate (and vice-versa) to the total information loss for each split node (`kld_inconsistency` and `kld_inconsistency_user` when using the results of a published systematic review with network meta-analysis) (Spineli, 2024);
- 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, 1971) (`gower_distance` and `comp_clustering`) (Spineli et al., 2025; Spineli, 2024). Results on the clustered comparisons based on hierarchical agglomerative clustering are illustrated using a dendrogram with integrated heatmap (`dendro_heatmap`) (Spineli et al., 2025). 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

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

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, Papadimitropoulou K, Kalyvas C. Exploring the Transitivity Assumption in Network Meta-Analysis: A Novel Approach and Its Implications. *Stat Med* 2025;**44**(7):e70068. doi: 10.1002/sim.70068.

Spineli LM. An empirical study on 209 networks of treatments revealed intransitivity to be common and multiple statistical tests suboptimal to assess transitivity. *BMC Med Res Methodol* 2024;**24**(1):301. doi: 10.1186/s12874-024-02436-7.

Spineli LM. Local inconsistency detection using the Kullback-Leibler divergence measure. *Syst Rev* 2024;**13**(1):261. doi: 10.1186/s13643-024-02680-4.

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

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

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 as described in Spineli et al. (2025). It also calculates the non-statistical heterogeneity within-comparisons and between-comparisons using the dissimilarities among all trials of the network (Spineli et al., 2025).

Usage

```
comp_clustering(
  input,
  weight,
  drug_names,
  threshold,
  informative = TRUE,
  ranged_values = FALSE,
  optimal_clusters,
  get_plots = "none",
  override = FALSE,
  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.
weight	A vector of non-negative numbers to define the weight contribution of each characteristic. The default is a vector of 1s for all characteristics.
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.
override	Logical with values TRUE to run the function for a pairwise meta-analysis and FALSE to stop the function in case of two treatments. 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

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.

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

Spineli LM, Papadimitropoulou K, Kalyvas C. Exploring the Transitivity Assumption in Network Meta-Analysis: A Novel Approach and Its Implications. *Stat Med* 2025;**44**(7):e70068. doi: 10.1002/sim.70068.

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 = as.integer(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)
```

`covar_contribution_plot` *Visualising study percentage contributions against a covariate*

Description

A scatter plot of the study percentage contributions against the values of a continuous study-level covariate for the treatment effects of comparisons referring to the basic parameters, functional parameters or both. Contributions on the estimated regression coefficients are also presented. Study percentage contributions are based on the proposed methodology of Donegan and colleagues (2018).

Usage

```
covar_contribution_plot(
  contr_res,
  comparisons = "basic",
  drug_names,
  upper_limit = 100,
  name_x_axis = NULL,
  axis_title_size = 14,
  axis_text_size = 14,
  strip_text_size = 14,
  subtitle_size = 14,
  label_size = 4,
  seq_by = 0.1,
  percentage = FALSE
)
```

Arguments

`contr_res`	An object of S3 class `study_perc_contrib`. This object contains the study percentage contributions to the treatment effects (or regression coefficients, if relevant) of all possible comparisons in the network. See 'Value' in `study_perc_contrib`.
`comparisons`	Character string indicating the type of comparisons to plot, with possible values: `"basic"`, `"functional"`, or `"all"` to consider only the basic parameters, only the functional parameters, or both, respectively. The default argument is `"basic"`.
`drug_names`	A vector of labels with the name of the interventions in the order they appear in the argument `contr_res`. If `drug_names` is not defined, the order of the interventions as they appear in `contr_res` is used, instead.
`upper_limit`	A positive number to define the upper bound of range of percentage values for the y-axis. The default argument is 100.

`name_x_axis` Text for the x axis title through the `labs` function found in the R-package ggplot2.

`axis_title_size` A positive integer for the font size of x axis title. `axis_title_size` determines the axis.title (and legend.title) arguments found in the theme's properties in the R-package ggplot2.

`axis_text_size` A positive integer for the font size of axis text (both axes). `axis_text_size` determines the axis.text (and legend.text) arguments 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 facets. `strip_text_size` determines the strip.text argument found in the theme's properties in the R-package ggplot2.

`subtitle_size` A positive integer for the font size of subtitle. `subtitle_size` determines the plot.subtitle argument found in the theme's properties in the R-package ggplot2.

`label_size` A positive integer for the font size of labels appearing on each data point. `label_size` determines the size argument found in the geom's aesthetic properties in the R-package ggplot2.

`seq_by` A positive integer for the sequence of values in the x-axis. `seq_by` appears in the arguments breaks and labels found in the scale_x_continuous aesthetic properties in the R-package ggplot2.

`percentage` Logical with values `TRUE` if the covariate is measured in per cent and `FALSE` otherwise. The default argument is `FALSE`.

Details

A panel of scatter plots is returned on the study percentage contributions to the treatment effects (and also regression coefficients, if relevant) against a continuous covariate for each comparison defined by the argument `comparisons`; namely, only those referring to the basic or functional parameters or all possible pairwise comparisons. Blue and red points indicate the studies investigating the corresponding comparisons directly and indirectly, respectively. Each point displays the number of the corresponding study in the dataset.

If interest also lies on the study percentage contributions to the regression coefficients, the regression coefficients can be determined to be common across the comparisons, independent or exchangeable and this assumption is specified in the `study_perc_contrib` function.

Value

If interest lies only on the study percentage contributions to the summary treatment effects of all possible pairwise comparisons, the function returns one plot named 'plot_treat'. If interest lies also on the study percentage contributions to the regression coefficient(s), the function returns also the plot named 'plot_reg'.

Author(s)

Loukia M. Spineli

References

Donegan S, Dias S, Tudur-Smith C, Marinho V, Welton NJ. Graphs of study contributions and covariate distributions for network meta-regression. *Res Synth Methods* 2018;**9**(2):243–60. doi: 10.1002/jrsm.1292

See Also

study_perc_contrib

Examples

```
## Not run:
data("nma.fluoride.donegan2018")

# Get study contributions to random-effects network meta-regression
# results under the assumption of independent treatment-by-covariate
# interaction
res <- study_perc_contrib(study_name = nma.fluoride.donegan2018$study,
                          base_t = nma.fluoride.donegan2018$t1,
                          exp_t = nma.fluoride.donegan2018$t2,
                          ref_t = 1,
                          obs_se = nma.fluoride.donegan2018$SE,
                          obs_cov = nma.fluoride.donegan2018$Cov,
                          covar = nma.fluoride.donegan2018$year,
                          covar_assum = "independent",
                          model = "RE",
                          tau = sqrt(0.03))

# Covariate-contribution plot on the basic parameters only
covar_contribution_plot(contr_res = res,
                        comparisons = "basic",
                        drug_names = c("NT", "PL", "DE", "RI", "GE", "VA"),
                        upper_limit = 15,
                        name_x_axis = "Randomisation year",
                        seq_by = 10)

## End(Not run)
```

---

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 (Spineli et al., 2025).

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

References

Spineli LM, Papadimitropoulou K, Kalyvas C. Exploring the Transitivity Assumption in Network Meta-Analysis: A Novel Approach and Its Implications. *Stat Med* 2025;**44**(7):e70068. doi: 10.1002/sim.70068.

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 = as.integer(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 (Spineli et al., 2025).

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

References

Spineli LM, Papadimitropoulou K, Kalyvas C. Exploring the Transitivity Assumption in Network Meta-Analysis: A Novel Approach and Its Implications. *Stat Med* 2025;**44**(7):e70068. doi: 10.1002/sim.70068.

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_juxtapose *Forest plot of juxtaposing several network meta-analysis models*

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 several network meta-analyses models, as well as a forest plot with the corresponding SUCRA values.

Usage

```
forestplot_juxtapose(
  results,
  compar,
  name,
  drug_names,
  axis_title_size = 12,
  axis_text_size = 12,
  caption_text_size = 9,
  label_size = 3.5,
  position_width = 0.8
)
```

Arguments

results	A list of at least two objects of S3 class run_model or run_metareg. See 'Value' in run_model and run_metareg.

compar	A character to indicate the comparator intervention. It must be any name found in drug_names.
name	A vector of characters referring to the juxtaposed models. If the argument is left unspecified, the names of models appear as 'Model X' with 'X' being the order/position of each model in the argument results.
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.
axis_title_size	A positive integer for the font size of x axis title. 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 (both axes). axis_text_size determines the axis.text argument found in the theme's properties in the R-package ggplot2.
caption_text_size	A positive integer for the font size of caption text. caption_text_size determines the plot.caption argument found in the theme's properties in the R-package ggplot2.
label_size	A positive integer for the font size of labels appearing on each interval. label_size determines the size argument found in the geom's aesthetic properties in the R-package ggplot2.
position_width	A positive integer specifying the vertical position of the intervals. position_width is found in the geom's aesthetic properties in the R-package ggplot2.

Details

The y-axis of the forest plot on **forest_plots** displays the labels of the treatments in the network; the selected treatment that comprises the compar argument is annotated in the plot with the label 'Comparator intervention'. For each comparison with the selected treatment, the 95% credible and prediction intervals are displayed as overlapping lines. Black lines refer to estimation under both analyses. Coloured lines refer to prediction under each model, 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.

If one of the models refer to network meta-regression (run_metareg) the results on treatment effects (estimation and prediction) and SUCRA values refer to the covariate value selected when employing run_metareg.

The y-axis for the forest plot on **SUCRA** values displays the labels of the treatments in the network. The corresponding numerical results are displayed above each line.

In **forest_plots** and **tau_plot**, the treatments are sorted in the descending order of their SUCRA values based on the first model specified in results.

Important note: forestplot_juxtapose should be used to compare the results from several network meta-analysis models that contain the same treatments, have the same meta-analysis model (fixed-effect or random-effects) and the same effect measure; otherwise the execution of the function will be stopped and an error message will be printed on the R console.

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

Value

A list of the following two figures:

`forest_plots`	A panel of two forest plots: (1) a forest plot on the posterior median and 95% credible and prediction intervals for comparisons with the selected comparator treatment (specified with `compar`), and (2) a forest plot on the posterior mean and 95% credible interval of SUCRA values of the treatments (Salanti et al., 2011).
`tau_plot`	A forest plot on the posterior median and 95% credible interval of the between-study standard deviation.

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_metareg`, `run_model`

`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_name = "covariate 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_name	A character or text to indicate the name of the covariate.
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.

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

Description

`gower_distance` calculate the weighted Gower's dissimilarity coefficient for all pairs of trials included in a network of interventions, considering several characteristics measured at trial level (Spineli et al., 2025). It takes values from 0 to 1, with 0 implying complete similarity and 1 complete dissimilarity.

Usage

```
gower_distance(input, weight)
```

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.
`weight`	A vector of non-negative numbers to define the weight contribution of each characteristic. The default is a vector of 1s for all characteristics.

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

Spineli LM, Papadimitropoulou K, Kalyvas C. Exploring the Transitivity Assumption in Network Meta-Analysis: A Novel Approach and Its Implications. *Stat Med* 2025;**44**(7):e70068. doi: 10.1002/sim.70068.

See Also

`comp_clustering`

Examples

```
# Fictional dataset
data_set <- data.frame(Trial_name = as.character(1:7),
                       comp = c("1vs2", "1vs2", "1vs2", "1vs3", "1vs3", "2vs3", "2vs3"),
                       sample = c(140, 145, 150, 40, 45, 75, 80),
                       age = c(18, 18, 18, 48, 48, 35, 35),
                     blinding = as.integer(c("yes", "yes", "yes", "no", "no", "no", "no")))

# Calculate the weighted Gower dissimilarity of all study pairs in the network
gower_distance(input = data_set)
```

---

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)
```

`heter_density_plot` *Visualising the density of two prior distributions for the heterogeneity parameter*

Description

Creating the density plot of two prior distributions for the between-study variance (log-normal and location-scale t distributions) or between-study standard deviation (half-normal distribution).

Usage

```
heter_density_plot(
  distr,
  heter_prior1,
  heter_prior2,
  heter1 = "tau",
  heter2 = "tau",
  caption = FALSE,
  x_axis_name = TRUE,
  y_axis_name = TRUE,
  title_name = NULL,
  axis_title_size = 13,
  axis_text_size = 13,
  legend_title_size = 13,
  legend_text_size = 13
)
```

Arguments

`distr`	Character string indicating the prior distribution. Set `distr` equal to one of the following: `"lognormal"`, `"logt"`, or `"halfnormal"`, which refers to a log-normal, location-scale, or half-normal distribution, respectively.

heter_prior1 A numeric vector with two values for the first prior distribution: 1) the mean value and 2) the standard deviation. When distr = "halfnormal", the first value should zero and the second a non-negative value referring to the scale parameter of the distribution.

heter_prior2 A numeric vector with two values for the second prior distribution: 1) the mean value and 2) the standard deviation. When distr = "halfnormal", the first value should zero and the second a non-negative value referring to the scale parameter of the distribution.

heter1 Character string indicating the heterogeneity parameter for heter_prior1. Set heter1 equal to one of the following: "tau", or "tau_omega", which refers to a between-study heterogeneity or between-design heterogeneity (inconsistency), respectively. This argument is relevant only when distr = "lognormal" or distr = "logt". The default is "tau".

heter2 Character string indicating the heterogeneity parameter for heter_prior2. Set heter2 equal to one of the following: "tau", or "tau_omega", which refers to a between-study heterogeneity or between-design heterogeneity (inconsistency), respectively. This argument is relevant only when distr = "lognormal" or distr = "logt". The default is "tau".

caption Logical to indicate whether to report a caption at the bottom right of the plot. It is relevant only when distr = "lognormal" and distr = "logt". The default is FALSE (do not report).

x_axis_name Logical to indicate whether to present the title of x-axis ('Between-study standard deviation'). The default is TRUE (report).

y_axis_name Logical to indicate whether to present the title of y-axis ('Density'). The default is TRUE (report).

title_name Text for the title of the plot. title_name determines the labs argument of the R-package ggplot2.

axis_title_size A positive integer for the font size of axis title. axis_title_size determines the axis.title argument found in the theme's properties in the R-package ggplot2. The default option is 13.

axis_text_size A positive integer for the font size of axis text. axis_text_size determines the axis.text argument found in the theme's properties in the R-package ggplot2. The default option is 13.

legend_title_size A positive integer for the font size of legend title. legend_text_size determines the legend.text argument found in the theme's properties in the R-package ggplot2. The default option is 13.

legend_text_size A positive integer for the font size of legend text. legend_text_size determines the legend.text argument found in the theme's properties in the R-package ggplot2. The default option is 13.

Details

Use this function to inspect the shape of the distribution and the range of between-study variance or standard deviation values before you define the argument heter_prior in run_model) to run

random-effects network meta-analysis.

Turner et al. (2012), Turner et al. (2015), and Rhodes et al. (2016) provide predictive prior distributions for the between-study variance for a binary outcome, measured in the log-odds ratio scale, and a continuous outcome, measured in the standardised mean difference scale, respectively.

Value

A plot with the density of two selected prior distributions for the heterogeneity parameter. Two different colours are used to discern the distributions. A legend is also created with the name and hyper-parameters of the selected prior distributions. The filled area under each curved indicates the values up to the median of the corresponding distribution. The x-axis present the 0.1

`heter_density_plot` also returns a table with the percentiles of each distribution.

Author(s)

Loukia M. Spineli

References

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

Turner RM, Davey J, Clarke MJ, Thompson SG, Higgins JP. Predicting the extent of heterogeneity in meta-analysis, using empirical data from the Cochrane Database of Systematic Reviews. *Int J Epidemiol* 2012;**41**(3):818–27. doi: 10.1093/ije/dys041

See Also

`run_model`

Examples

```
## Not run:
## Two empirical priors for between-study variance of log odds ratio.
heter_density_plot(distr = "lognormal",
                   heter_prior1 = c(-2.56, 1.74),  # General healthcare setting
                   heter_prior2 = c(-1.83, 1.52))  # Pain and pharma vs. placebo/ctrl

## Two empirical priors for between-study variance of standardised mean
## difference.
heter_density_plot(distr = "logt",
                   heter_prior1 = c(-3.44, 2.59),  # General healthcare setting
              heter_prior2 = c(-0.60, 2.61)) # Pain and pharma vs. placebo/ctrl for cancer

## Two half-normal prior distributions for between-study standard deviation
heter_density_plot(distr = "halfnormal",
                   heter_prior1 = c(0, 1),
```

```
                    heter_prior2 = c(0, 0.5))

## End(Not run)
```

`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 4 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

`inconsistency_variance_prior` *Function for the hyper-parameters of the prior distribution of the inconsistency variance (network meta-analysis with random inconsistency effects)*

Description

Calculates the mean and standard deviation of the log-normal distribution and location-scale t-distribution of the inconsistency variance in the log-odds ratio and standardised mean difference scales, respectively, based on corresponding empirical distributions for the between-study variance proposed by Turner et al. (2015) and Rhodes et al. (2015). It also return the median value of the inconsistency standard deviation.

Usage

```
inconsistency_variance_prior(mean_tau2, sd_tau2, mean_scale, measure)
```

Arguments

`mean_tau2`	Mean value from the empirical prior distribution for the between-study variance.
`sd_tau2`	Standard deviation value from the empirical prior distribution for the between-study variance.
`mean_scale`	Positive (non-zero value) as a scaling factor of `mean_tau2`. See Law et al. (2016).
`measure`	Character string indicating the effect measure. For a binary outcome, use only `"OR"` for the odds ratio. For a continuous outcome, use only `"SMD"` for standardised mean difference.

Details

Law et al. (2016) suggested using the proposed empirical prior distributions for between-study variance to construct a prior distribution for the inconsistency variance. The authors provided the formulas for the hyper-parameters of the inconsistency variance for a binary outcome measured in the log odds ratio scale. We extended the idea for a continuous outcome measured in the standardised mean difference scale. Currently, the empirical prior distributions for the between-study variance have been proposed for these effect measures only (Turner et al. (2015), Rhodes et al. (2015)).

Value

A list of three elements: the mean and standard deviation for the prior distribution for the inconsistency variance, and the median inconsistency standard deviation according to the selected empirical prior distribution for the between-study variance.

Author(s)

Loukia M. Spineli

References

Law M, Jackson D, Turner R, Rhodes K, Viechtbauer W. Two new methods to fit models for network meta-analysis with random inconsistency effects. *BMC Med Res Methodol* 2016;**16**:87. doi: 10.1186/s12874-016-0184-5

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

`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. *Bioinformatics* 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 (missingness scenarios)*

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 re-analysis.

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)
```

---

kld_inconsistency *Density plots of local inconsistency results and Kullback-Leibler divergence when 'rnmamod', 'netmeta' or 'gemtc' R packages are used*

Description

A panel of density plots on the direct and indirect estimates of the selected comparisons based on approach for local inconsistency evaluation, such as back-calculation and node-splitting approaches (Dias et al., 2010; van Valkenhoef et al., 2016) and loop-specific approach (Bucher et al., 1997) accompanied by the average Kullback-Leibler divergence. Additionally, stacked bar plots on the percentage contribution of either Kullback-Leibler divergence (from direct to indirect, and vice-versa) to the total information loss for each selected comparison are presented (Spineli, 2024). The function handles results also from the R-packages gemtc and netmeta.

Usage

```
kld_inconsistency(
  node,
  threshold = 1e-05,
  drug_names = NULL,
  outcome = NULL,
  scales = "free",
  show_incons = TRUE,
  y_axis_name = TRUE,
  title_name = NULL,
  axis_title_size = 13,
  axis_text_size = 13,
  text_size = 3.5,
  strip_text_size = 13,
  legend_title_size = 13,
  legend_text_size = 13,
  str_wrap_width = 10
)
```

Arguments

node	An object of S3 class `run_nodesplit` or class `mtc.nodesplit` (see gemtc) or class `netsplit` (see netmeta).
threshold	A positive number indicating the threshold of not concerning inconsistency, that is, the minimally allowed deviation between the direct and indirect estimates for a split node that does raise concerns for material inconsistency. The argument is optional.
drug_names	A vector of labels with the name of the interventions in the order they appear in the argument `data`. It is not relevant for gemtc and netmeta.
outcome	Optional argument to describe the effect measure used (the x-axis of the plots).
scales	A character on whether both axes should be fixed (`"fixed"`) or free (`"free"`) or only one of them be free (`"free_x"` or `"free_y"`). `scales` determines the scales argument found in function (`facet_wrap`) in the R-package ggplot2. The default is (`"free"`).
show_incons	Logical to indicate whether to present the point estimate and 95 (report).
y_axis_name	Logical to indicate whether to present the title of y-axis ('Density'). The default is `TRUE` (report).
title_name	Text for the title of the plot. `title_name` determines the labs argument of the R-package ggplot2.
axis_title_size	A positive integer for the font size of axis title. `axis_title_size` determines the axis.title argument found in the theme's properties in the R-package ggplot2. The default option is 13.
axis_text_size	A positive integer for the font size of axis text. `axis_text_size` determines the axis.text argument found in the theme's properties in the R-package ggplot2. The default option is 13.

`text_size` A positive integer for the font size of labels. `text_size` determines the size argument found in the geom_text function in the R-package ggplot2. The default option is 3.5.

`strip_text_size`
A positive integer for the font size of facet labels. `legend_text_size` determines the legend.text argument found in the theme's properties in the R-package ggplot2. The default option is 13.

`legend_title_size`
A positive integer for the font size of legend title. `legend_text_size` determines the legend.text argument found in the theme's properties in the R-package ggplot2. The default option is 13.

`legend_text_size`
A positive integer for the font size of legend text. `legend_text_size` determines the legend.text argument found in the theme's properties in the R-package ggplot2. The default option is 13.

`str_wrap_width` A positive integer for wrapping the axis labels in the percent stacked bar-plot. `str_wrap_width` determines the `str_wrap` function of the R-package stringr.

Value

The first plot is a panel of density plots for each split node sorted in ascending order of the Kullback-Leibler divergence value. Blue and black lines refer to the direct and indirect estimates, respectively. The grey segment refers to the 95% credible (confidence) interval of the inconsistency parameter, when `run_nodesplit` (`netsplit`) has been applied, with a darker grey line referring to the point estimate. When `mtc.nodesplit` has been employed, the 95% confidence interval has been approximated using the Bucher's approach based on the corresponding direct and indirect results. This was necessary because `mtc.nodesplit` (version 1.0-2) returns only the inconsistency p-values rather than the posterior results on the inconsistency parameters. The mean estimate on the scale of the selected effect measure appears at the top of each density curve.

The Kullback-Leibler divergence value appears at the top left of each plot in three colours: black, if no threshold has been defined (the default), green, if the Kullback-Leibler divergence is below the specified `threshold` (not concerning inconsistency) and red, if the Kullback-Leibler divergence is at least the specified `threshold` (substantial inconsistency).

The second plot is a percent stacked bar plot on the percentage contribution of approximating direct with indirect estimate (and vice-versa) to the total information loss for each target comparison. Total information loss is defined as the sum of the KLD value when approximating the direct with indirect estimate (blue bars), and the KLD when approximating the indirect with direct estimate (black bars). Values parentheses refer to the corresponding KLD value.

The function also returns the data-frame `average_KLD` that includes the split comparisons and the corresponding average Kullback-Leibler divergence value.

Author(s)

Loukia M. Spineli

References

Bucher HC, Guyatt GH, Griffith LE, Walter SD. The results of direct and indirect treatment comparisons in meta-analysis of randomized controlled trials. *J Clin Epidemiol* 1997;**50**(6):683–91.

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

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

Spineli LM. Local inconsistency detection using the Kullback-Leibler divergence measure. *Syst Rev* 2024;**13**(1):261. doi: 10.1186/s13643-024-02680-4.

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

facet_wrap, mtc.nodesplit, kld_measure, netsplit, run_nodesplit, str_wrap

Examples

```
## Not run:
data("nma.baker2009")

# 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")

# Apply the function
kld_inconsistency(node = node,
                  threshold = 0.64,
                  drug_names = interv_names,
                  outcome = "Odds ratio (logarithmic scale)",
                  str_wrap_width = 15)

## End(Not run)
```

---

kld_inconsistency_user

Density plots of local inconsistency results and Kullback-Leibler divergence (When dataset is created by the user)

Description

When the user has extracted results obtained from a method of local inconsistency evaluation (e.g., loop-specific, back-calculation or node-splitting approaches) as reported in publication, this function provides the same output with the function `kld_inconsistency`. A panel of density plots on the direct and indirect estimates of the selected comparisons based on approach for local inconsistency evaluation, such as back-calculation and node-splitting approaches (Dias et al., 2010; van Valkenhoef et al., 2016) and loop-specific approach (Bucher et al., 1997) accompanied by the average Kullback-Leibler divergence. Additionally, stacked bar plots on the percentage contribution of either Kullback-Leibler divergence (from direct to indirect, and vice-versa) to the total information loss for each selected comparison are presented (Spineli, 2024).

Usage

```
kld_inconsistency_user(
  dataset,
  threshold = 1e-05,
  level = 0.05,
  outcome = NULL,
  scales = "free",
  show_incons = TRUE,
  y_axis_name = TRUE,
  title_name = NULL,
  axis_title_size = 13,
  axis_text_size = 13,
  text_size = 3.5,
  strip_text_size = 13,
  legend_title_size = 13,
  legend_text_size = 13,
  str_wrap_width = 10
)
```

Arguments

`dataset`	A data-frame of seven columns and as many rows as the split nodes. The first column contains the names of the split nodes, and the remaining columns have the point estimate and standard error of the direct, indirect and inconsistency parameter in that order.
`threshold`	A positive number indicating the threshold of not concerning inconsistency, that is, the minimally allowed deviation between the direct and indirect estimates for a split node that does raise concerns for material inconsistency. The argument is optional.
`level`	A number indicating the significance level. Suggested values are 0.05 and 0.10. The default value is 0.05.
`outcome`	Optional argument to describe the effect measure used (the x-axis of the plots).
`scales`	A character on whether both axes should be fixed (`"fixed"`) or free (`"free"`) or only one of them be free (`"free_x"` or `"free_y"`). `scales` determines the scales argument found in function (`facet_wrap`) in the R-package ggplot2. The default is (`"free"`).

show_incons Logical to indicate whether to present the point estimate and 95 (report).

y_axis_name Logical to indicate whether to present the title of y-axis ('Density'). The default is TRUE (report).

title_name Text for the title of the plot. title_name determines the labs argument of the R-package ggplot2.

axis_title_size A positive integer for the font size of axis title. axis_title_size determines the axis.title argument found in the theme's properties in the R-package ggplot2. The default option is 13.

axis_text_size A positive integer for the font size of axis text. axis_text_size determines the axis.text argument found in the theme's properties in the R-package ggplot2. The default option is 13.

text_size A positive integer for the font size of labels. text_size determines the size argument found in the geom_text function in the R-package ggplot2. The default option is 3.5.

strip_text_size A positive integer for the font size of facet labels. legend_text_size determines the legend.text argument found in the theme's properties in the R-package ggplot2. The default option is 13.

legend_title_size A positive integer for the font size of legend title. legend_text_size determines the legend.text argument found in the theme's properties in the R-package ggplot2. The default option is 13.

legend_text_size A positive integer for the font size of legend text. legend_text_size determines the legend.text argument found in the theme's properties in the R-package ggplot2. The default option is 13.

str_wrap_width A positive integer for wrapping the axis labels in the percent stacked bar-plot. str_wrap_width determines the str_wrap function of the R-package stringr.

Value

The first plot is a panel of density plots for each split node sorted in ascending order of the Kullback-Leibler divergence value. Blue and black lines refer to the direct and indirect estimates, respectively. The grey segment refers to the (1 - level)% 'pseudo' confidence interval of the inconsistency parameter based on the corresponding normal z-scores, with a darker grey line referring to the point estimate. The names of the selected comparisons appear at the top of each plot. The mean estimate on the scale of the selected effect measure appears at the top of each density curve.

The Kullback-Leibler divergence value appears at the top left of each plot in three colours: black, if no threshold has been defined (the default), green, if the Kullback-Leibler divergence is below the specified threshold (not concerning inconsistency) and red, if the Kullback-Leibler divergence is at least the specified threshold (substantial inconsistency).

The second plot is a percent stacked bar plot on the percentage contribution of approximating direct with indirect estimate (and vice-versa) to the total information loss for each target comparison. Total information loss is defined as the sum of the Kullback-Leibler divergence value when approximating the direct with indirect estimate (blue bars), and the Kullback-Leibler divergence value

when approximating the indirect with direct estimate (black bars). Values parentheses refer to the corresponding Kullback-Leibler divergence value. Bars are sorted in ascending order of the average Kullback-Leibler divergence value.

The function also returns the data-frame `average_KLD` that includes the split comparisons and the corresponding average Kullback-Leibler divergence value.

Author(s)

Loukia M. Spineli

References

Bucher HC, Guyatt GH, Griffith LE, Walter SD. The results of direct and indirect treatment comparisons in meta-analysis of randomized controlled trials. *J Clin Epidemiol* 1997;**50**(6):683–91.

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

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

Spineli LM. Local inconsistency detection using the Kullback-Leibler divergence measure. *Syst Rev* 2024;**13**(1):261. doi: 10.1186/s13643-024-02680-4.

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

`facet_wrap`, `kld_inconsistency`, `kld_measure`

Examples

```
## Not run:
## Data are taken from Table II in Dias et al. (2010)
# Treatments compared
treat <-
c("SK", "t-PA", "Acc t-PA", "SK+t-PA", "r-PA", "TNK", "PTCA", "UK", "ASPAC")

# Baseline arm (from each selected comparison)
base <- rep(1:3, c(6, 3, 5))

# Non-baseline arm (from each selected comparison)
nonbase <- c(2, 3, 5, 7, 8, 9, 7, 8, 9, 4, 5, 7, 8, 9)

# Compared treatments with their names
treat_comp <-
mapply(function(x, y) paste(treat[x], "vs", treat[y]), base, nonbase)

# Direct results
direct_mean <- c(0.000, -0.158, -0.060, -0.666, -0.369, 0.009, -0.545,
                 -0.295, 0.006, 0.126, 0.019, -0.216, 0.143, 1.409)
```

```
direct_sd <- c(0.030, 0.048, 0.089, 0.185, 0.518, 0.037, 0.417, 0.347, 0.037,
               0.054, 0.066, 0.118, 0.356, 0.415)

# Indirect results
indirect_mean <- c(0.189, -0.247, -0.175, -0.393, -0.168, 0.424, -0.475,
                   -0.144, 0.471, 0.630, 0.135, -0.477, -0.136, 0.165)
indirect_sd <- c(0.235, 0.092, 0.081, 0.120, 0.244, 0.252, 0.108, 0.290,
                 0.241, 0.697, 0.101, 0.174, 0.288, 0.057)

# Inconsistency
incons_mean <- c(-0.190, 0.088, 0.115, -0.272, -0.207, -0.413, -0.073,
                 -0.155, -0.468, -0.506, -0.116, 0.260, 0.277, 1.239)
incons_sd <- c(0.236, 0.104, 0.121, 0.222, 0.575, 0.253, 0.432, 0.452, 0.241,
               0.696, 0.120, 0.211, 0.461, 0.420)

# Collect results in a data-frame (exactly as required from the function)
dias_results <- data.frame(treat_comp, direct_mean, direct_sd, indirect_mean,
                           indirect_sd, incons_mean, incons_sd)

# Apply the function
kld_inconsistency_user(dataset = dias_results,
                       threshold = 0.13,
                       outcome = "Odds ratio (logarithmic scale)")

## End(Not run)
```

`kld_measure` *Function for the Kullback-Leibler Divergence of two normally distributed treatment effects for the same pairwise comparison*

Description

The user specify the (posterior) mean and standard error (or posterior standard deviation) of two estimated treatment effects, X and Y, that refer to the **same** pairwise comparison and are assumed to follow a normal distribution. The function returns the Kullback-Leibler Divergence (KLD) measure of 1) approximating X with Y, 2) approximating Y with X, and 3) their average.

Usage

```
kld_measure(mean_y, sd_y, mean_x, sd_x)
```

Arguments

`mean_y`	A real number that refers to the mean of the estimated treatment effect Y on the scale of the selected effect measure (in logarithmic scale for relative effect measures).
`sd_y`	A positive integer that refers to the posterior standard deviation or the standard error of the estimated treatment effect Y on the scale of the selected effect measure (in logarithmic scale for relative effect measures).

`mean_x` A real number that refers to the mean of the estimated treatment effect X on the scale of the selected effect measure (in logarithmic scale for relative effect measures).

`sd_x` A positive integer that refers to the posterior standard deviation or the standard error of the estimated treatment effect X on the scale of the selected effect measure (in logarithmic scale for relative effect measures).

Value

The function return the following numeric results:

kld_sym The symmetric KLD value as the average of two KLD values .

kld_x_true The KLD value when approximating X by Y (X is the 'truth').

kld_y_true The KLD value when approximating Y by X (Y is the 'truth').

References

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

See Also

`kld_inconsistency`, `kld_inconsistency_user`, `robustness_index`, `robustness_index_user`

`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 to create the barplot on the ratio of MCMC error to the posterior standard deviation and the barplot on the Gelman-Rubin R diagnostic:

`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 `traceplot`, `densplot`, and `autocorr.plot` functions of the R-package coda to return the trace plots, density plots, and autocorrelation plots for all or selected monitored parameters through the `par` argument.

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

`as.mcmc`, `traceplot`, `densplot`, `autocorr.plot`, `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_name = "covariate 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_name`	A character or text to indicate the name of the covariate.
`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.

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 covariate value specified in `run_metareg`.

`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 covariate value specified in `run_metareg` 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 covariate value specified in `run_metareg`. 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
# average publication year
metareg_plot(full = res,
             reg = reg,
             compar = "salmeterol",
             cov_name = "mean publication year",
             drug_names = interv_names)
```

---

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 (Spineli et al. 2025).

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

References

Spineli LM, Papadimitropoulou K, Kalyvas C. Exploring the Transitivity Assumption in Network Meta-Analysis: A Novel Approach and Its Implications. *Stat Med* 2025;**44**(7):e70068. doi: 10.1002/sim.70068.

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"))
```

`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

---

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.fluoride.donegan2018`

Topical fluoride interventions for preventing dental caries

Description

A dataset of 130 trials comparing different forms of topical fluoride interventions for preventing dental caries in children or adolescents with at least 1 year or school year. The analysed continuous outcome is the change from baseline in decayed, missing, and filled tooth surfaces. The dataset contains also multi-arm trials.

Usage

```
data(nma.fluoride.donegan2018)
```

Format

A data frame with 140 rows of arm-based data and 16 columns referring to the trial number, the treatment identifier in the compared arms, the standardised mean difference and its standard error for each trial and possible comparison (in the case of multi-arm trial), the randomisation year, the standard deviation and number randomised for each arm and trial, the pooled standard deviation and the within-study covariance in multi-arm trials.

Details

The interventions have been coded as follows: 1, no treatment; 2, placebo; 3, dentifrice; 4, rinse; 5, gel; and 6, varnish

Source

Donegan S, Dias S, Tudur-Smith C, Marinho V, Welton NJ. Graphs of study contributions and covariate distributions for network meta-regression. *Res Synth Methods* 2018;**9**(2):243–60. doi: 10.1002/jrsm.1292

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.malaria.donegan2018 *Artemether, artesunate and quinine for severe malaria*

Description

A dataset of 24 two-arm trials comparing artemether, artesunate and quinine with each other in patients with severe malaria. The analysed binary outcome is death.

Usage

```
data(nma.malaria.donegan2018)
```

Format

A data frame with 24 rows of arm-based data and 11 columns referring to the trial number, the treatment identifier in the compared arms, the odds ratio in the logarithmic scale and its standard error for each trial, the average age in years and its centered version, the number of events and number randomised for each arm and trial.

Details

The interventions have been coded as follows: 1, quinine; 2, artemether; and 3, artesunate

Source

Donegan S, Dias S, Tudur-Smith C, Marinho V, Welton NJ. Graphs of study contributions and covariate distributions for network meta-regression. *Res Synth Methods* 2018;**9**(2):243–60. doi: 10.1002/jrsm.1292

---

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)
```

`plot_study_dissimilarities` *Plot Gower's disimilarity values for each study (Transitivity evaluation)*

Description

Illustrating the range of Gower's dissimilarity values for each study in the network, as well as their between- and within-comparison dissimilarities

Usage

```
plot_study_dissimilarities(
  results,
  axis_title_size = 12,
  axis_text_size = 12,
  strip_text_size = 11,
  label_size = 3.5
)
```

Arguments

`results`	An object of S3 class `comp_clustering`. See 'Value' in `comp_clustering`.
`axis_title_size`	A positive integer for the font size of axis title (both axes). `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 (both axes). `axis_text_size` determines the axis.text argument found in the theme's properties in the R-package ggplot2.
`strip_text_size`	A positive integer for the font size of facet labels. `strip_text_size` determines the strip.text argument found in the theme's properties in the R-package ggplot2.
`label_size`	A positive integer for the font size of labels appearing on each study-specific segment. `label_size` determines the size argument found in the geom's aesthetic properties in the R-package ggplot2.

Details

The range of Gower's dissimilarity values for each study versus the remaining studies in the network for a set of clinical and methodological characteristics that may act as effect modifiers. Gower's dissimilarities take values from 0 to 1, with 0 and 1 implying perfect similarity and perfect dissimilarity, respectively.

The unique dissimilarity values appear as dotted, vertical, grey lines on each study

Value

A horizontal bar plot illustrating the range of Gower's dissimilarity values for each study with those found in other comparisons. The study names appear on the y-axis in the order they appear in `results` and the dissimilarity values appear on the x-axis. Red and blue points refer to the (average) within-comparison and between-comparison dissimilarity, respectively, for each study.

A data-frame on the (average) within-comparison and between-comparison dissimilarities for each study alongside the study name and comparison. The last two columns refer to the within-comparison and between-comparison dissimilarities, respectively, after replacing with the maximum value in the multi-arm trials. These two columns should be used as a covariate in the function `study_perc_contrib` to obtain the percentage contribution of each study based on the covariate values.

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`, `study_perc_contrib`

Examples

```
# Fictional dataset
data_set <- data.frame(Trial_name = paste("study", 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 = as.integer(c("yes", "yes", "yes", "no", "no", "no", "no")))

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

plot_study_dissimilarities(results = res,
```

```
axis_title_size = 12,
axis_text_size = 12,
strip_text_size = 11,
label_size = 3.5)
```

`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, prediction = FALSE, 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 or run_metareg indicating different re-analyses: the first object in the list should refer to the primary analysis.

prediction — Logical character on whether to consider the prediction (TRUE) or estimation of the summary treatment effects (FALSE). This is only relevant for a random-effects model and the default argument is FALSE (estimation).

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). Furhermore, robustness_index can be used in that context 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.

In the case of a general sensitivity analysis, the compared models should refer to the same effect measure and the same meta-analysis model (i.e., fixed-effect or random-effects).

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

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`, `kld_measure`, `run_metareg`, `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

kld_measure, 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,
  cov_value,
  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".
cov_value	Numeric for the covariate value of interest.
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.

`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` 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",
            cov_value = 2007,
            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.

`p_value` The two-sided Bayesian p-value (based on the posterior mean on step(diff)) for each split node defined.

`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_pred`	The predicted summary effect measure (according to the argument `measure` defined in `run_model`). This element does not appear in the case of a fixed-effect meta-analysis.
`EM_LOR`	The estimated summary odd ratio in the logarithmic scale when `measure = "RR"` or `measure = "RD"`.
`EM_LOR_pred`	The predicted summary odd ratio in the logarithmic scale when `measure = "RR"` or `measure = "RD"`. This element does not appear in the case of a fixed-effect meta-analysis.
`tau`	The between-trial standard deviation. This element does not appear in the case of a fixed-effect 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)
```

---

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_name = "covariate 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_name	A character or text to indicate the name of the covariate.
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.

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`

`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

---

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)
```

`study_perc_contrib` *Calculate study percentage contributions to summary treatment effects or regression coefficients*

Description

A data-frame on the percentage contributions of each study to every possible pairwise comparison in the investigated network. Study percentage contributions are based on the proposed methodology of Donegan and colleagues (2018).

Usage

```
study_perc_contrib(
  study_name,
  base_t,
  exp_t,
  ref_t,
  obs_se,
  obs_cov = NULL,
  covar,
  covar_assum,
  model,
  tau = NULL,
  tau_beta = NULL
)
```

Arguments

`study_name`	A vector of labels with the names of the studies included in the investigated network. For multi-arm studies, the study name should appear as many times as the number of possible comparisons among the compared treatments.
`base_t`	A vector of numbers referring to the treatment identifier for the baseline arm (comparator) of each study.
`exp_t`	A vector of numbers referring to the treatment identifier for the experimental arm of each study.
`ref_t`	A scalar for the selected reference treatment in the network.
`obs_se`	A vector of numbers referring to the estimated standard error of the treatment effect of each study. For multi-arm studies, the standard error of the treatment effect of each possible comparison among the compared treatments should be included.
`obs_cov`	A vector of numbers referring to the covariance in the block variance-covariance matrix of the estimated treatments effects for the multi-arm studies only. This argument should be left unspecified if there are no multi-arm studies in the network.

`covar` A vector of numbers referring to a continuous covariate that indicates a study characteristic or summary patient characteristic.

`covar_assum` Character string indicating the structure of the treatment-by-covariate interaction, as described in Cooper et al. (2009) if interest also lies on the study percentage contributions to the estimated regression coefficients. Set `covar_assumption` equal to `"no"`, `"exchangeable"`, `"independent"`, or `"common"`. When `covar_assum = "no"`, only the study percentage contributions to the summary treatment effects will be calculated. There is no default argument.

`model` Character string indicating the analysis model with values `"RE"`, or `"FE"` for the random-effects and fixed-effect model, respectively. There is no default argument.

`tau` A scalar referring to the estimated between-study standard deviation obtained from network meta-analysis, if `covar_assum = "no"`, or network meta-regression for a specific treatment-by-covariate interaction assumption. This argument should be left unspecified when `model = "FE"`.

`tau_beta` A scalar referring to the estimated standard deviation of the exchangeable regression coefficients obtained from network meta-regression with exchangeable treatment-by-covariate interaction. This argument should be left unspecified when `covar_assum` is not "exchangeable". There is no default argument.

Details

Note that the columns referring to the study percentage contributions to summary treatment effects are indicated by the letter 'd' with two numbers in decreasing order for the comparison: the first number refers to the comparator and the second number refers to the experimental treatment of the comparison. If interest lies also on the regression coefficients, the correspoding columns are indicated by 'beta'.

The function centers the covariate to the mean but presents the original version of the covariate.

Value

A list of the following two elements:

`perc_contribute`
A data-frame with four columns referring to the study name, baseline and experimental treatment arm, and the covariate and as many columns as the number of possible comparisons with the study percentage contributions to summary treatment effects referring to the basic parameters, and followed by the functional parameters. If interest lies also on the regression coefficients, extra columns appear referring to the study percentage contributions to the regression coefficients specified from the argument `covar_assum`.

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

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

Donegan S, Dias S, Tudur-Smith C, Marinho V, Welton NJ. Graphs of study contributions and covariate distributions for network meta-regression. *Res Synth Methods* 2018;**9**(2):243–60. doi: 10.1002/jrsm.1292

Examples

```
data("nma.malaria.donegan2018")

# Get study contributions to fixed-effect network meta-regression
# results under the assumption of independent treatment-by-covariate
# interaction
study_perc_contrib(study_name = nma.malaria.donegan2018$s,
                   base_t = nma.malaria.donegan2018$t1,
                   exp_t = nma.malaria.donegan2018$t2,
                   ref_t = 1,
                   obs_se = nma.malaria.donegan2018$se,
                   covar = nma.malaria.donegan2018$x,
                   covar_assum = "independent",
                   model = "FE")
```

---

table_tau2_prior *Predictive distributions for the between-study variance in a future meta-analysis on odds ratio or standardised mean difference*

Description

A table with the hyperparameters of the predictive distributions for the between-study variance developed by Turner et al. (2015) and Rhodes et al. (2015): log-normal distribution and t-distribution (with 5 degrees of freedom) when the outcome data are analysed in the odds ratio or standardised mean difference scale, respectively.

Usage

```
table_tau2_prior(measure, area)
```

Arguments

measure	Character string indicating the effect measure with possible values "OR" for odds ratio and "SMD" for standardised mean difference.

`area` Character string indicating the medical area relating to the predictive distributions for standardised mean difference with possible values `"cancer"` for medical areas of cancer, `"respiratory"` for medical areas of respiratory diseases, and `"other"` for medical areas other than cancer or respiratory diseases. The argument is not relevant for odds ratio.

Details

This table aids in selecting the hyperparameters for the function `heterogeneity_param_prior` when considering an informative prior distribution for the between-study variance parameter based on the two publications mentioned above (relevant for the function `run_model` to conduct random-effects network meta-analysis).

Value

A cross-sectional table as a heatmap showing the hyperparameters (mean and standard deviation) of the corresponding predictive distribution for all combinations between the outcome types and treatment-comparison types and according to the selected medical area (only relevant with standardised mean difference) as defined by Turner et al. (2015) and Rhodes et al. (2015). The tiles are coloured with different shades according to the corresponding median value: the larger the median, the darker the colour.

Author(s)

Loukia M. Spineli

References

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

`heterogeneity_param_prior`, `run_model`

`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`

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

∗ **datasets**
nma.baker2009, 81
nma.bottomley2011, 82
nma.dogliotti2014, 82
nma.fluoride.donegan2018, 83
nma.liu2013, 84
nma.malaria.donegan2018, 84
nma.schwingshackl2014, 85
nma.stowe2011, 86
pma.hetrick2012, 91
pma.taylor2004, 91
∗ **package**
rnmamod-package, 4

as.mcmc, *70*
autocorr.plot, *70*
autojags, *11, 12*, *104, 105*, *109*, *112*, *114, 115*, *118, 119*, *121*, *123, 124*

balloon_plot, *6*, 8
baseline_model, *5*, 10, *110*, *112*
bland_altman_plot, *6*, 12, *137*

comp_clustering, *6*, 13, *22, 23*, *26, 27*, *33, 34*, *47*, *89, 90*
cophenetic, *15*, *17*
covar_contribution_plot, 18

data_preparation, *12, 13*, 20, *24, 25*, *44*, *69*, *109*, *112*, *126, 127*
dendro_heatmap, *6*, 22
densplot, *70*
describe_network, *5*, 24
distr_characteristics, *6*, 25

facet_wrap, *51*, *53, 54*, *56*
forestplot, *5*, 27
forestplot_juxtapose, 29
forestplot_metareg, *5*, 31, *72, 73*

gower_distance, *6*, 33

hclust, *15*, *17*
heatmap_missing_dataset, *5*, *21, 22*, 34
heatmap_missing_network, *5*, *21, 22*, 36
heatmap_robustness, *6*, 37, *100*, *102*
heatmaply, *23*
heter_density_plot, 39
heterogeneity_param_prior, 42, *107*, *112*, *132*

improved_ume, 43, *48, 49*, *123*
inconsistency_variance_prior, *5*, 45
internal_measures_plot, *17*, 46
intervalplot_panel_ume, *6*, 48, *137*

jags, *10*, *12*, *93–96*, *103–105*, *108*, *111–115*, *117*, *119–122*, *124*

kld_barplot, *6*, 49, *100*
kld_inconsistency, *6*, 50, *54*, *56*, *58*
kld_inconsistency_user, *6*, 53, *58*
kld_measure, *53*, *56*, 57, *100*, *102*

league_heatmap, *5*, 58
league_heatmap_pred, *5*, 61
league_table_absolute, *5*, 64, *66*, *68*
league_table_absolute_user, *5*, 66
leverage_plot, *6*, 68, *137*

mcmc_diagnostics, *5*, 69, *105*, *111*, *124*
metareg_plot, *5*, *32, 33*, 71, *125, 126*
miss_characteristics, *6*, 73
missingness_param_prior, 76, *107*, *112*
mtc.nodesplit, *51–53*
mtc.nodesplit.comparisons, *87, 88*, *94, 95*, *114, 115*

netmeta, *101, 102*
netmetabin, *101, 102*
netplot, *5*, 78
netsplit, *51–53*
nma.baker2009, 81

nma.bottomley2011, 82
nma.dogliotti2014, 82
nma.fluoride.donegan2018, 83
nma.liu2013, 84
nma.malaria.donegan2018, 84
nma.schwingshackl2014, 85
nma.stowe2011, 86
nodesplit_plot, *5*, 86, *114, 115*

plot.igraph, *79–81*
plot_study_dissimilarities, 89
pma.hetrick2012, 91
pma.taylor2004, 91
prepare_model, *11, 12*, *21, 22*, 92, *109, 110*, *112*
prepare_nodesplit, *21, 22*, 94, *114, 115*
prepare_ume, *21, 22*, 95, *123, 124*

rankosucra_plot, *5*, 96
rma, *101, 102*
rma.glmm, *101, 102*
rma.mh, *101, 102*
rma.mv, *101, 102*
rma.peto, *101, 102*
rma.uni, *101, 102*
rnmamod *(*rnmamod-package*)*, 4
rnmamod-package, 4
robustness_index, *6*, *38*, *49, 50*, *58*, 98
robustness_index_user, *6*, *38*, *58*, 100
run_metareg, *4*, *10–13*, *21, 22*, *29–33*, *59–63*, *68–73*, *93*, *98*, *100*, 103, *125, 126*
run_model, *4*, *8–13*, *20–22*, *24, 25*, *28–33*, *35–38*, *40–44*, *48–50*, *59–65*, *68–73*, *76–78*, *81*, *86–88*, *93–100*, *103–105*, 106, *113–128*, *132*, *134–138*, *140*
run_nodesplit, *4*, *21, 22*, *51–53*, *69, 70*, *86–88*, *94, 95*, 113
run_sensitivity, *5*, *8, 9*, *21, 22*, *49, 50*, *69, 70*, *98–100*, 116
run_series_meta, *4*, *59*, *61*, *69, 70*, 120, *123, 124*, *127, 128*
run_ume, *5*, *12, 13*, *21, 22*, *44*, *48, 49*, *68–70*, *96*, 122, *126, 127*, *136, 137*

scatterplot_sucra, *5*, *72, 73*, 125
scatterplots_dev, *6*, 126, *137*
series_meta_plot, *6*, *121*, 127
silhouette, *17*, *47*
str_wrap, *15*, *17*, *52, 53*, *55*
study_perc_contrib, *18–20*, *90*, 129

table_tau2_prior, 131
taylor_continuous, 132, *139, 140*
taylor_imor, 134, *139, 140*
textConnection, *93–96*
traceplot, *70*

ume_plot, *6*, *12, 13*, *48, 49*, *68, 69*, *123, 124*, *126, 127*, 136
unrelated_effects_plot, *133–135*, 138

write_xlsx, *24, 25*, *67, 68*, *71–73*, *87, 88*, *127, 128*, *136, 137*, *139, 140*