Package ‘rnmamod’

April 6, 2022

Type Package
Title Bayesian Network Meta-Analysis with Missing Participants
Version 0.2.0
Maintainer Loukia Spineli <Spineli.Loukia@mh-hannover.de>
Description A comprehensive suite of functions to perform and visualise pairwise and network meta-analysis with aggregate binary or continuous missing participant outcome data. The package covers core Bayesian one-stage models implemented in a systematic review with multiple interventions, including fixed-effect and random-effects network meta-analysis, meta-regression, evaluation of the consistency assumption via the node-splitting approach and the unrelated mean effects model, and sensitivity analysis. Missing participant outcome data are addressed in all models of the package. The 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, and barplot of Kullback-Leibler divergence. The package also allows the user to export the results to an Excel file at the working directory.

License GPL (>= 3)
URL https://github.com/LoukiaSpin/rnmamod
Depends R (>= 4.0.0)
Imports coda, dplyr, fdrtool, gemtc, ggfittext, ggplot2, ggpubr, ggrepel, knitr, MASS, mcmcpplots, netmeta, pcnetmeta, R2jags, reshape2, scales, writexl
Suggests rmarkdown, testthat (>= 3.0.0)
VignetteBuilder knitr
Config/testthat/edition 3
Encoding UTF-8
Language en-US
LazyData true
RoxygenNote 7.1.2
NeedsCompilation no
Author Loukia Spineli [aut, cre], Chrysostomos Kalyvas [ctb], Katerina Papadimitropoulou [ctb]
Repository CRAN
Date/Publication 2022-04-06 17:40:02 UTC

R topics documented:

- rmamod-package ................................................. 3
- balloon_plot ................................................... 6
- bland_altman_plot ............................................... 8
- data_preparation ............................................... 9
- describe_network ............................................. 11
- forestplot ....................................................... 12
- forestplot_metareg ........................................... 14
- heatmap_missing_dataset .................................... 15
- heatmap_missing_network .................................... 16
- heatmap_robustness .......................................... 18
- heterogeneity_param_prior .................................. 19
- improved_ume .................................................. 21
- intervalplot_panel_ume ...................................... 22
- kld_barplot ..................................................... 23
- league_heatmap ............................................... 25
- league_heatmap_pred ......................................... 28
- league_table_absolute ....................................... 31
- leverage_plot ................................................. 32
- mcmc_diagnostics ............................................ 33
- metareg_plot ................................................... 35
- missingness_param_prior .................................... 37
- netplot ......................................................... 39
- nma.baker2009 .................................................. 41
- nma.bottomley2011 ............................................. 42
- nma.dogliotti2014 ............................................. 42
- nma.liu2013 ..................................................... 43
- nma.schwingshackl2014 ...................................... 44
- nma.stowe2011 .................................................. 44
- nodesplit_plot ............................................... 45
- pma.hetrick2012 ............................................... 47
- pma.taylor2004 .................................................. 48
- prepare_model ................................................ 49
- prepare_nodesplit .......................................... 50
- prepare_ume .................................................. 52
Description

An R package for performing Bayesian network meta-analysis while handling missing participant outcome data properly.

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).

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) following the PRISMA statement for systematic reviews with network meta-analysis (Hutton et al., 2015);
• tabulation of the R-hat (Gelman and Rubin, 1992) 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 or P-scores (league_heatmap and league_heatmap_pred, respectively) after performing network meta-analysis or network meta-regression (Salanti et al., 2011; Ruecker and Schwarzer, 2015);
• 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 (Salanti et al., 2011);
• tabulation of the estimated regression coefficient(s), the estimated and predicted summary effects, measures of model fit and estimated between-trial standard deviation before and after adjusting for a trial-specific covariate (metareg_plot), and visualisation of the summary effects and SUCRA values from both models (forestplot_metareg, and scatterplot_sucra, respectively–both found in metareg_plot);
• tabulation of the estimated direct and indirect effects of the split nodes and corresponding inconsistency factors, measures of model fit and estimated between-trial standard deviation after each split node, and visualisation of these results (nodesplit_plot);
• tabulation of the estimated summary effects of all comparisons observed in the network, measures of model fit and estimated between-trial standard deviation under the unrelated mean effects model and network meta-analysis, as well as visualisation of the summary effects from both models (intervalplot_panel_ume) and the goodness of fit of each model using a series of complementary plots (scatterplots_dev (Dias et al., 2013a), bland_altman_plot (Bland and Altman, 1999), and leverage_plot (Dias et al., 2010)–all found in ume_plot);
• tabulation of the estimated summary effects and corresponding between-trial standard deviation for comparisons with at least two trials under pairwise and network meta-analysis, as well as visualisation of these results (series_meta_plot);
• calculation and visualisation of the robustness index for all possible comparisons in the network (robustness_index and heatmap_robustness, respectively) (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).
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


---

**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**

```r
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**


**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 = 'rnsmamod'))

# 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)
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
**data_preparation**

**References**


**See Also**

data_preparation, run_metareg, run_model, run_ume, ume_plot

---

**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`
**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. See also 'Value' in netplot.

**Usage**

describe_network(data, drug_names, measure)

**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.

**Details**

describe_network calls data_preparation to facilitate the calculations.

**Value**

A list of scalar results and two data-frames to be passed to netplot. The scalar results include:

- **direct_comp**
  The number of observed comparisons in the network.

- **two_arm_ns**
  The number of two-arm trials in the network.

- **multi_arm_ns**
  The number of multi-arm trials in the network.

- **total_rand_network**
  The total number of randomised participants in the network.

- **prop_obs_network**
  The proportion of participants who completed the trial.

- **prop_event_network**
  The proportion of observed events in the network. When the outcome is continuous, this element is omitted.

- **trial_zero_event**
  The number of trials with at least one arm with zero events. When the outcome is continuous, this element is omitted.
The number of trials with zero events in all arms. When the outcome is continuous, this element is omitted.

The two data-frames include table_interventions and table_comparisons. See 'Value' in netplot for these data-frames.

Author(s)
Loukia M. Spineli

See Also
data_preparation, netplot, run_model

Description
Provides a forest plot with the posterior mean and 95% credible and prediction intervals for comparisons with the selected intervention (comparator) in the network.

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 the 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 the 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
forestplot  

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


See Also

run_model

Examples

data("nma.liu2013")

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

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

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

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

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

**Description**
This function illustrates a forest plot of the posterior mean and 95% credible and prediction interval of comparisons with the selected intervention of the network under the network meta-analysis and network meta-regression.

**Usage**

```r
generate_forestplot_metareg(full, reg, compar, cov_value, drug_names)
```

**Arguments**

- `full`: An object of S3 class `run_model`. See 'Value' in `run_model`
- `reg`: An object of S3 class `run_metareg`. See 'Value' in `run_metareg`
- `compar`: A character to indicate the comparator intervention. It must be any name found in `drug_names`
- `cov_value`: A list of two elements in the following order: a number for the covariate value of interest (see 'Arguments' in `run_metareg`), and a character to indicate the name of the covariate. See also 'Details'.
- `drug_names`: A vector of labels with the name of the interventions in the order they appear in the argument data of `run_model`. If `drug_names` is not defined, the order of the interventions as they appear in data is used, instead.

**Details**

In both plots, the y-axis displays all interventions in the network; the selected intervention that comprises the `compar` is indicated in the plot with a homonymous label. The numerical results are displayed above each line. Odds ratio, relative risks, and ratio of means are reported in the original scale after exponentiation of the logarithmic scale.

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` 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 estimated effect size of comparisons with the selected intervention of the network, and (2) a forest plot on the predicted effect size of comparisons with the selected intervention of the network. Both forest plots illustrate the results from network meta-analysis and network meta-regression using different colors for the corresponding lines.
**heatmap_missing_dataset**

**Author(s)**

Loukia M. Spineli

**References**


**See Also**

`metareg_plot`, `run_metareg`, `run_model`

---

**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**

```r
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


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.
heatmap_missing_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


See Also

run_model
Examples

```r
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)
```

**Description**

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

**Usage**

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

**Arguments**

- `robust` An object of S3 class `robustness_index`. See 'Value' in `robustness_index`.
- `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` 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. Then, it returns a lower triangular heatmap matrix with the robustness index value of all possible pairwise comparisons.

Author(s)

Loukia M. Spineli

References


See Also

robustness_index, run_model

Examples

data("nma.baker2009")

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

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

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

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

---

**heterogeneity_param_prior**

*Determine the prior distribution for the heterogeneity parameter*

Description

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

Usage

heterogeneity_param_prior(measure, model, heter_prior)

Arguments

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

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

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

Details

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

Value

A value to be passed to run_model.

Author(s)

Loukia M. Spineli

References


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').
intervalplot_panel_ume

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


See Also

data_preparation, run_model, run_ume

---

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


See Also

`improved_ume run_model, run_ume, ume_plot`

---

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

Description

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

```r
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


See Also

`robustness_index`, `run_model`, `run_sensitivity`
**Examples**

```r
data("pma.taylor2004")

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

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

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

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

---

**league_heatmap**  
*League heatmap for estimation*

**Description**

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

**Usage**

```r
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 mean 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 mean 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


See Also

`run_metareg`, `run_model`, `run_series_meta`
Examples

```r
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**

```r
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'.
league_heatmap_pred

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 mean 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 mean 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


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. It 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 homonymous label.

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


See Also

run_model

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)
mcmc_diagnostics

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


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 Carlo (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 three character strings that refer to three monitored parameters in jagsfit which is an object of S3 class run_metareg, run_model, and run_ume. These three 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 maximum R-hat and MCMC error and compares them with the thresholds 1.1 and 5%, respectively. Convergence is achieved for the monitored parameter, when the maximum R-hat and MCMC error are below the corresponding thresholds; otherwise, the MCMC algorithm has not converged for that parameter. If the monitored parameter is a vector with the posterior results, there is only one R-hat and one MCMC error. If the monitored parameter is a matrix of the posterior results, there are as many R-hats and MCMC errors as the number of rows for that parameter. In that case, the maximum R-hat and MCMC error are considered.

Value
mcmc_diagnostics returns a data-frame that contains the Gelman-Rubin convergence diagnostic, R-hat, the MCMC error, and the convergence status of the following monitored parameters: EM The estimated summary effect measure. EM_pred The predicted summary effect measure. delta The estimated trial-specific effect measure. tau The between-trial standard deviation. direct The direct estimate of the split node (see ‘Value’ in run_nodesplit). indirect The indirect estimate of the split node (see ‘Value’ in run_nodesplit). diff The inconsistency factor of the split node (see ‘Value’ in run_nodesplit). phi The informative missingness parameter. beta The regression coefficient.

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

Author(s)
Loukia M. Spineli

References
metareg_plot

See Also

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

Examples

data("nma.baker2009")

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

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

metareg_plot

End-user-ready results for network meta-regression

Description

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

Usage

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

Arguments

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

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

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

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

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

Value

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

`table_estimates`

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

`table_model_assessment`

The DIC, total residual deviance, number of effective parameters, and the posterior median and 95% credible interval of between-trial standard deviation ($\tau$) under each model (Spiegelhalter et al., 2002). When a fixed-effect model has been performed, `metareg_plot` does not return results on $\tau$. For a binary outcome, the results refer to the odds ratio scale.

`table_regression_coefficients`

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

`interval_plot`

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

`sucra_scatterplot`

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


See Also

run_model

netplot

Network plot and description of the evidence base

Description

Illustrates the network plot for one outcome and summarises the characteristics of the evidence base.

Usage

netplot(data, drug_names, 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.
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).

... Additional arguments of the nma.networkplot function of the R-package penmeta.
Details

netplot draws the network plot using the \texttt{nma.networkplot} function of the R-package \texttt{pcnetmeta}. The \texttt{mtc.data.studyrow} function of the R-package hrefhttps://CRAN.R-project.org/package=gemtc\texttt{gemtc} is additionally used to convert data from the required one-trial-per-row format into the one-arm-per-row format.

Furthermore, netplot exports the data-frames to separate 'xlsx' files (via the \texttt{write_xlsx} function of the R-package \texttt{writexl}) at the working directory of the user.

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. The edge thickness is proportional to the number of trials investigating the corresponding comparison, unless specified otherwise (see \texttt{nma.networkplot} function of the R-package \texttt{pcnetmeta}). The size of the node is weighted by the total number of trials investigating the corresponding intervention, unless specified otherwise (see \texttt{nma.networkplot} function of the R-package \texttt{pcnetmeta}).

netplot also returns the following data-frames that describe the evidence base:

\textbf{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.

\textbf{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.

\textbf{table\_comparisons}

Identical structure to \textbf{table\_interventions} but for each observed comparison in the network.

Author(s)

Loukia M. Spinieli

References


See Also

data\_preparation, mtc.data.studyrow, nma.networkplot, run\_model write\_xlsx
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,
        save_xls = FALSE)

---

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

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


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**


<table>
<thead>
<tr>
<th>nma.liu2013</th>
<th>Antidepressants in Parkinson’s disease</th>
</tr>
</thead>
</table>

**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**

**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**


**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


---

**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 mean 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 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 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 mean, 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

See Also
mtc.nodesplit.comparisons, run_model, run_nodesplit, write_xlsx

Examples
```r
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)
```

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**


**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**

**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; Spinelli, 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**

```r
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.
prepare_nodesplit

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


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

```r
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 function 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


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

\texttt{prepare_ume} \ (\texttt{measure}, \texttt{model}, \texttt{assumption}, \texttt{connected})

Arguments

\begin{itemize}
  \item \texttt{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.
  \item \texttt{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".
  \item \texttt{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.
  \item \texttt{connected} \ An integer equal to one or larger that indicates the number of subnetworks.
\end{itemize}

Details

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

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

Value

An R character vector object to be passed to \texttt{run_ume} through the \texttt{textConnection} function as the argument object.
**rankosucra_plot**

**Author(s)**

Loukia M. Spineli

**References**


**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**

```r
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 (red bars) with integrated blue 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


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", ...)
robustness_index

"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)

---

**Description**

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

**Usage**

`robustness_index(sens, threshold)`

**Arguments**

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

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

**Details**

Thresholds of robustness have been proposed only for the odds ratio and standardised mean difference effect measures (Spineli et al., 2021). 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 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. Spineli et al. (2021) offers a discussion on specifying the threshold of robustness.

In the case of binary outcome, `robustness_index` considers the results in the odds ratio scale to calculate the robustness index. This is because, the odds ratio is used as the 'best-case' effect measure in `run_model`. Then, relative risk, and risk difference are functions of the odds ratio and the selected baseline risk (See 'Details' in `run_model`).
In the case of missing participant outcome data, the primary analysis is considered to be the middle of the numbers in the argument mean_scenarios of run_sensitivity (see 'Arguments' and 'Details' in run_sensitivity).

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

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

Value

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

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

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

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

- threshold: The threshold used to be inherited by the heatmap_robustness function.

- scenarios: The scenarios considered to be inherited by the heatmap_robustness and kld_barplot functions.

Author(s)

Loukia M. Spineli

References


run_metareg

See Also

heatmap_robustness, kld_barplot, run_model, run_sensitivity

Examples

data("nma.baker2009")

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

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

run_metareg

Perform Bayesian pairwise or network meta-regression

Description

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

Usage

run_metareg(
full,
covariate,
covar_assumption,
n_chains,
n_iter,
n_burnin,
n_thin
)

Arguments

full An object of S3 class run_model. See 'Value' in run_model.
covariate A numeric vector or matrix for a trial-specific covariate that is a potential effect modifier. See 'Details'.
covar_assumption Character string indicating the structure of the intervention-by-covariate interaction, as described in Cooper et al. (2009). Set covar_assumption equal to "exchangeable", "independent", or "common".
n_chains Positive integer specifying the number of chains for the MCMC sampling; an argument of the jags function of the R-package R2jags. The default argument is 2.
run_metareg

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

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

Value

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

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

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

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

hat_par The fitted outcome at each trial-arm.

phi The informative missingness parameter.

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

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

effectiveness The ranking probability of each intervention for every rank.

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

EM_pred The predicted summary effect measure (according to the argument measure defined in run_model).
delta  The estimated trial-specific effect measure (according to the argument measure defined in \texttt{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, \texttt{EM} and \texttt{EM\_pred} refer to all possible pairwise comparisons of interventions in the network. Furthermore, \texttt{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 \texttt{delta} for comparisons with the baseline intervention of the trial (found in the first column of the element \texttt{t}), with $T$ being the number of interventions in the trial.

Furthermore, the output includes the following elements:

\texttt{leverage\_o}  The leverage for the observed outcome at each trial-arm.
\texttt{sign\_dev\_o}  The sign of the difference between observed and fitted outcome at each trial-arm.
\texttt{model\_assessment}  A data-frame on the measures of model assessment: deviance information criterion, number of effective parameters, and total residual deviance.
\texttt{jagsfit}  An object of S3 class \texttt{jags} with the posterior results on all monitored parameters to be used in the \texttt{mcmc\_diagnostics} function.

The \texttt{run\_metareg} function also returns the arguments \texttt{data}, \texttt{measure}, \texttt{model}, \texttt{assumption}, \texttt{covariate}, \texttt{covar\_assumption}, \texttt{n\_chains}, \texttt{n\_iter}, \texttt{n\_burnin}, and \texttt{n\_thin} to be inherited by other relevant functions of the package.

**Author(s)**

Loukia M. Spineli

**References**


**See Also**

\texttt{jags}, \texttt{run\_model}
### Examples

```r
data("nma.baker2009")

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

# Publication year

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

---

**run_model**  
*Perform Bayesian pairwise or network meta-analysis*

### Description

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

### Usage

```r
run_model(
  data, 
  measure, 
  model, 
  assumption, 
  heter_prior, 
  mean_misspar, 
  var_misspar, 
  D, 
  ref, 
  base_risk = NULL, 
  n_chains, 
  n_iter, 
  n_burnin, 
  n_thin
)
```
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 number in the interval (0, 1) that indicates the baseline risk for the selected reference intervention. If base_risk has not been defined, the function uses the median event risk for the reference intervention as calculated from the corresponding trials in data. This argument is only relevant for binary outcomes.

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 \texttt{jags} function of the R-package \texttt{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 \texttt{jags} function of the R-package \texttt{R2jags}. The default argument is 1000.

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

**Format**

The columns of the data-frame in the argument \texttt{data} refer to the following elements for a continuous outcome:

- \texttt{t}  An intervention identifier in each arm.
- \texttt{y} The observed mean value of the outcome in each arm.
- \texttt{sd} The observed standard deviation of the outcome in each arm.
- \texttt{m} The number of missing participant outcome data in each arm.
- \texttt{n} The number of randomised participants in each arm.

For a binary outcome, the columns of the data-frame in the argument \texttt{data} refer to the following elements:

- \texttt{t}  An intervention identifier in each arm.
- \texttt{r} The observed number of events of the outcome in each arm.
- \texttt{m} The number of missing participant outcome data in each arm.
- \texttt{n} The number of randomised participants in each arm.

The number of rows in \texttt{data} equals the number of collected trials. Each element appears in \texttt{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 \texttt{data} for a network with a maximum number of four arms. It is not a prerequisite of \texttt{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.

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. (2013) 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., 2013).

The code of Dias et al. (2013) 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.

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., 2013). 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".
- **effectiveness**
  The ranking probability of each intervention for every rank.

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

- **EM_pred**
  The predicted summary effect measure (according to the argument measure).
- **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:

- **abs_risk**: The 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.
- **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**


**See Also**

`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)
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)

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.

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.

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

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

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

Value

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

- **direct** The summary effect measure (according to the argument measure defined in run_model) of each split node based on the corresponding trials.
- **indirect** The indirect summary effect measure (according to the argument measure defined in run_model) of each split node based on the remaining network after removing (splitting) the corresponding node.
- **diff** The inconsistency parameter for each split node defined as the difference between the direct and indirect effect of the corresponding split node.
- **tau** The between-trial standard deviation after each split node, when the random-effects model has been specified.

Furthermore, the output includes the following element:

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

Author(s)

Loukia M. Spineli

References

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,
)
run_sensitivity

    var_misspar,
    n_chains,
    n_iter,
    n_burnin,
    n_thin
)

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.

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 difference), 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.

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

Value

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

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

In a random-effects network meta-analysis, EM refer to all possible pairwise comparisons of interventions in the network. Furthermore, tau is typically assumed to be common for all observed comparisons in the network.
Author(s)

Loukia M. Spineli

References


See Also

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**

```r
run_series_meta(full, n_chains, n_iter, n_burnin, n_thin)
```

**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.

**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 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


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**

```r
run_ume(full, n_iter, n_burnin, n_chains, n_thin)
```

**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.

**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 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


See Also

`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)
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


See Also
data_preparation, run_model, run_ume, ume_plot
Description

Creates a scatterplot of the SUCRA values from the network meta-analysis and the network meta-regression for a specified level or value of the investigated covariate.

Usage

scatterplot_sucra(full, reg, cov_value, drug_names)

Arguments

full
An object of S3 class run_model. See 'Value' in run_model.
reg
An object of S3 class run_metareg. See 'Value' in run_metareg.
cov_value
A list of two elements in the following order: 1) a number for the covariate value of interest (see 'Arguments' in run_metareg), and 2) a character to indicate the name of the covariate. See also 'Details'.
drug_names
A vector of labels with the name of the interventions in the order they appear in the argument data of run_model. If drug_names is not defined, the order of the interventions as they appear in data is used, instead.

Details

The names of the interventions appear above each point in the plot. Three coloured rectangles are drawn in the scatterplot: a red rectangle for SUCRA values up to 50%, a yellow rectangular for SUCRA values between 50% and 80%, and a green rectangle for SUCRA values over 80%. Interventions falling at the green area are considered as the highest ranked interventions, whilst interventions falling at the red area are considered as the lowest ranked interventions.

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

scatterplot_sucra is integrated in metareg_plot.

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

Value

A scatterplot of the SUCRA values under the network meta-analysis (y-axis) against the SUCRA values under the network meta-regression (x-axis) for a specified level or value of the investigated covariate.

Author(s)

Loukia M. Spineli
References


See Also

`metareg_plot, run_metareg, run_model`

---

**series_meta_plot**  
End-user-ready results for a series of pairwise meta-analyses

**Description**

Facilitates the comparison of the consistency model (via `run_model`) with a series of pairwise meta-analyses (via `run_series_meta`) regarding the estimated summary effect sizes and between-trial standard deviation for comparisons with at least two trials.

**Usage**

```
series_meta_plot(full, meta, drug_names, save_xls)
```

**Arguments**

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

**Details**

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

For a binary outcome, when `measure` is "RR" (relative risk) or "RD" (risk difference) in `run_model`, `series_meta_plot` currently presents the results in the odds ratio for being the base-case effect measure in `run_model` for a binary outcome (see also 'Details' in `run_model`).

The user can detect any inconsistencies in the estimated effects from the compared models and explore the gains in precision stemming from applying network meta-analysis. Furthermore, the user can investigate the plausibility of the common between-trial heterogeneity assumption which is typically considered in network meta-analysis.
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 mean 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


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,
Pattern-mixture model with Taylor series for continuous outcome

drug_names = interv_names)

taylor_continuous

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

```r
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.
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. The latter uses the pairwise function from the package netmeta to transform the dataset from the wide arm-based format into the long arm-based format (see, 'Arguments' for data in unrelated_effects_plot).

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


See Also

pairwise, run_model, unrelated_effects_plot

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. The latter uses the pairwise function from the package netmeta to transform the dataset from the wide arm-based format into the long arm-based format (see, 'Arguments' for data in unrelated_effects_plot).

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.
The standard error of the log OR adjusted for missing participants and obtained using the Taylor series.

Author(s)

Loukia M. Spineli

References


See Also

pairwise, run_model, unrelated_effects_plot

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>full</td>
<td>An object of S3 class run_model. See 'Value' in run_model.</td>
</tr>
<tr>
<td>ume</td>
<td>An object of S3 class run_ume. See 'Value' in run_ume.</td>
</tr>
<tr>
<td>drug_names</td>
<td>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.</td>
</tr>
<tr>
<td>save_xls</td>
<td>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).</td>
</tr>
</tbody>
</table>
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 mean, 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.

- `leverage_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
unrelated_effects_plot

References


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,
unrelated_effects_plot

```r
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**


**See Also**

`run_model`, `taylor_continuous`, `taylor_imor`, `write_xlsx`
Index

* datasets
  nma.baker2009, 41
  nma.bottomley2011, 42
  nma.dogliotti2014, 42
  nma.liu2013, 43
  nma.schwingshackl2014, 44
  nma.stowe2011, 44
  pma.hetrick2012, 47
  pma.taylor2004, 48

* package
  rnmamod-package, 3
  rnmamod-package, 3
  mcmcplot, 34, 35
  metareg_plot, 4, 14, 15, 35, 79, 80
  missingness_param_prior, 37, 61, 66
  mtc.data.studyrow, 40
  mtc.nodesplit.comparisons, 46, 47, 51, 52, 68, 69
  netplot, 4, 11, 12, 39
  nma.baker2009, 41
  nma.bottomley2011, 42
  nma.dogliotti2014, 42
  nma.liu2013, 43
  nma.networkplot, 39, 40
  nma.schwingshackl2014, 44
  nma.stowe2011, 44
  nodesplit_plot, 4, 45, 68, 69
  pairwise, 83–85
  pma.hetrick2012, 47
  pma.taylor2004, 48
  prepare_model, 9, 10, 49, 63, 66
  prepare_nodesplit, 9, 10, 50, 67, 69
  prepare_ume, 9, 10, 52, 76, 77
  rankosuca_plot, 4, 53
  rnmamod (rnmamod-package), 3
  rnmamod-package, 3
  robustness_index, 4, 18, 19, 24, 55
  run_metareg, 3, 8–10, 14, 15, 26–30, 33–37, 50, 57, 79, 80
  run_model, 3, 6–24, 26–40, 45–47, 50–55, 57–59, 60, 67–81, 83, 85–89
  run_nodesplit, 3, 9, 10, 34, 35, 45–47, 51, 52, 67
  run_sensitivity, 4, 6, 7, 9, 10, 24, 34, 35, 55–57, 69
  run_series_meta, 3, 26, 27, 34, 35, 73, 75, 77, 80, 81
  run_ume, 3, 8–10, 21–23, 33–35, 52, 53, 75, 78, 85, 87

balloon_plot, 4, 6
bland_altman_plot, 4, 8, 86, 87
data_preparation, 8, 9, 11, 12, 21, 22, 33, 40, 63, 66, 78
describe_network, 11
forestplot, 4, 12
forestplot_metareg, 4, 14, 36, 37
heatmap_missing_dataset, 4, 10, 15
heatmap_missing_network, 4, 10, 16
heatmap_robustness, 4, 18, 56, 57
heterogeneity_param_prior, 19, 61, 66
improved_ume, 21, 23, 75
intervalplot_panel_ume, 4, 22, 86, 87
league_heatmap, 4, 23, 56, 57
league_heatmap_pred, 4, 28
league_table_absolute, 31
leverage_plot, 4, 32, 86, 87
mcmc_diagnostics, 4, 33, 59, 65, 76
mcmcplot, 34, 35
metareg_plot, 4, 14, 15, 35, 79, 80
missingness_param_prior, 37, 61, 66
mtc.data.studyrow, 40
mtc.nodesplit.comparisons, 46, 47, 51, 52, 68, 69
netplot, 4, 11, 12, 39
nma.baker2009, 41
nma.bottomley2011, 42
nma.dogliotti2014, 42
nma.liu2013, 43
nma.networkplot, 39, 40
nma.schwingshackl2014, 44
nma.stowe2011, 44
nodesplit_plot, 4, 45, 68, 69
pairwise, 83–85
pma.hetrick2012, 47
pma.taylor2004, 48
prepare_model, 9, 10, 49, 63, 66
prepare_nodesplit, 9, 10, 50, 67, 69
prepare_ume, 9, 10, 52, 76, 77
rankosuca_plot, 4, 53
rnmamod (rnmamod-package), 3
rnmamod-package, 3
robustness_index, 4, 18, 19, 24, 55
run_metareg, 3, 8–10, 14, 15, 26–30, 33–37, 50, 57, 79, 80
run_model, 3, 6–24, 26–40, 45–47, 50–55, 57–59, 60, 67–81, 83, 85–89
run_nodesplit, 3, 9, 10, 34, 35, 45–47, 51, 52, 67
run_sensitivity, 4, 6, 7, 9, 10, 24, 34, 35, 55–57, 69
run_series_meta, 3, 26, 27, 34, 35, 73, 75, 77, 80, 81
run_ume, 3, 8–10, 21–23, 33–35, 52, 53, 75, 78, 85, 87
scatterplot_sucra, 4, 36, 37, 79
scatterplots_dev, 4, 77, 86
series_meta_plot, 4, 73, 74, 80
taylor_continuous, 82, 88, 89
taylor_imor, 83, 88, 89
textConnection, 49–53
ume_plot, 4, 8, 9, 23, 33, 76–78, 85
unrelated_effects_plot, 83–85, 87
write_xlsx, 35–37, 39, 40, 45–47, 80, 81, 85–89