

Package: gemtcPlus (via r-universe)

August 12, 2024

Type Package

Title Provides a suite of extension functions for NMA using the
`gemtc` package

Version 1.0.0

Description Functions for generating outputs: tables and plots for NMA
reports.

License Apache License (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R2jags

Imports gemtc, ggplot2, tidyr, reshape, dplyr, ggmcmc, coda, gplots,
rmarkdown, magrittr, methods, purrr, stringr, network

Suggests knitr, testthat, here, pander, reshape2

VignetteBuilder knitr

Repository <https://pharmaverse.r-universe.dev>

RemoteUrl <https://github.com/Roche/gemtcPlus>

RemoteRef HEAD

RemoteSha d6c4b4f1ad7afac7526567cc16d5e7967b2b0f51

Contents

bth_prior	3
create_jags_init	3
create_template	4
extract_BUGS_file	4
gemtcPlus	5
get_fp_1o	5
get_fp_1o_GoF	6
get_fp_1o_HR	6
get_fp_1o_S	7

get_fp_2o	8
get_fp_2o_GoF	9
get_fp_2o_HR	10
get_fp_2o_S	11
get_fp_comparison	12
get_fp_contrasts	12
get_fp_corrs	13
get_fp_elements	13
get_fp_GoF	14
get_fp_HR	15
get_fp_S	16
get_jags_info	17
get_mtc_allVsNew	18
get_mtc_newVsAll	18
get_mtc_probBetter	19
get_mtc_sum	19
get_nw_fromto	20
get_pwe_comparison	20
get_pwe_contrasts	21
get_pwe_conv_diag	22
get_pwe_elements	22
get_pwe_GoF	23
get_pwe_S	24
get_pw_segments	25
get_segments	25
groupedTTE_fp_pre_proc	26
groupedTTE_pwe_pre_proc	27
list_BUGS	28
match_args_to_func	28
mtc.prob.better.table	29
nma_fit	29
nma_pre_proc	30
plan_binary	30
plan_fp	32
plan_hr	33
plan_pwe	35
plot_fp_HR	36
plot_mtc_forest	37
process_binary	37
process_gsd	38
process_hr	38
pwe_Hu	39
pwe_S	39

bth_prior	<i>Creates a fractional polynomial model plan</i>
-----------	---

Description

Creates a fractional polynomial model plan

Usage

```
bth_prior(model = "PWE", bth.model = "RE", type, distr, param)
```

Arguments

model	One of "PWE" or "FP"
bth.model	If model is PWE then must be "RE". If model is FP then One of "RE" or "REINT"
type	If PWE RE or FP REINT then one of "sd" or "var". If FP RE then "vcov"
distr	If type is "sd" then one of "unif" or "hn". If type is "var" then "ln". If type is "vcov" then "invwish"
param	A list with ... if distribution is "unif" then named values min, max (numeric > 0). If distribution is hn or ln then list must contain mean and precision. Of distribution in invwish then names value "scale" (a matrix).

create_jags_init	<i>create_jags_init. Helper function to create jags init list dependant on length on chains provided</i>
------------------	--

Description

create_jags_init. Helper function to create jags init list dependant on length on chains provided

Usage

```
create_jags_init(n.chains = 3)
```

Arguments

n.chains	Number of chains'
----------	-------------------

create_template	<i>Creates a minimal project template for selected model type</i>
-----------------	---

Description

Creates a minimal project template for selected model type

Usage

```
create_template(type = c("binary", "gsd", "fp", "pwe"), file = NULL)
```

Arguments

type	model type
file	file name (optional); if not provided, default filename will be <type>_model.Rmd. The .Rmd extension will be added.

Examples

```
create_template("binary")
```

extract_BUGS_file	<i>Helper function to extract BUGS files for given input parameters</i>
-------------------	---

Description

Helper function to extract BUGS files for given input parameters

Usage

```
extract_BUGS_file(
  data.type,
  bth.model,
  bth.prior,
  model.pars = NULL,
  model.type = c("PWE", "FP")
)
```

Arguments

data.type	Only GSD currently supported
bth.model	FE or RE
bth.prior	list contains priors information
model.pars	Only needed for FP model, list containing exponents to determine order (1st of 2nd)
model.type	Either PWE or FP

`gemtcPlus`*gemtcPlus: A package for performing NMA in R*

Description

This package was designed to work alongside the `gemtc` package. Included are numerous convenience functions to aid in the production of reports.

Author(s)

Sandro Gsteiger <sandro.gsteiger@roche.com>

`get_fp_1o`*First order fractional polynomial*

Description

First order fractional polynomial

Usage

```
get_fp_1o(x, params, exponents, sums = NULL)
```

Arguments

<code>x</code>	A vector with the dependent variable.
<code>params</code>	A matrix with two columns giving the intercept and the "slope". If more than one rows given, the fractional polynomial is evaluated for each row.
<code>sums</code>	A function. If non-null, sums of the fractional polynomial at each <code>x</code> value (i.e. summaries over the different <code>params</code> values) are calculated.
<code>exponent</code>	A numerical value giving the exponent (p1) of the polynomial.

Value

A matrix with `length(x)` rows giving the FP values or FP summaries.

get_fp_1o_GoF	<i>Calculate the study and arm level survivor functions estimates from a 1st order fractional polynomial NMA. These estimates provide the basis for a goodness-of-fit graph when plotted along with the input data.</i>
---------------	---

Description

Calculate the study and arm level survivor functions estimates from a 1st order fractional polynomial NMA. These estimates provide the basis for a goodness-of-fit graph when plotted along with the input data.

Usage

```
get_fp_1o_GoF(fit, time = 1:24, bl.node = "mu", contrast.node = "d")
```

Arguments

fit	JAGS object with FP NMA fit.
time	Vector of time-points at which S(t) functions are calculated.
bl.node	Character to identify the node in the jags model that identifies the baseline estimates (default is "mu").
contrast.node	Character to identify the node in the jags model that identifies the baseline estimates (default is "d").

Details

The FP parameters to calculate the log-hazard curves for each survivor function are combinations bl.node + contrast.node. Here, every study-specific baseline estimate is combined with the arm-specific contrast to obtain arm level NMA estimates.

Value

a data.frame containing survivor function for each treatment

get_fp_1o_HR	<i>Calculate the time-dependent hazard ratios obtained from fitting a first order fractional polynomial model.</i>
--------------	--

Description

Calculate the time-dependent hazard ratios obtained from fitting a first order fractional polynomial model.

Usage

```
get_fp_1o_HR(
  x,
  fit,
  trt.nos,
  ref.no,
  trt.labs = NULL,
  node = "d",
  CI = TRUE,
  revert = FALSE
)
```

Arguments

x	A vector with the dependent variable.
fit	An rjags object with the output from the JAGS fit.
trt.nos	A vector with the numerical treatment IDs for which the HRs shall be calculated (including the ref).
ref.no	An integer with the numerical ID of the reference for the HR calculations.
trt.labs	A character vector of same length as trt.nos with the treatment labels.
node	A character string that identifies the node in the JAGS model giving the treatment effect estimates.
CI	Logical, shall CIs for the fractional polynomial be given? (Medians are always provided.)

Details

Requires the packages: dplyr, coda.

Value

A data frame with pointwise median (and CI) HRs for all comparisons of trt.nos vs ref.no.

get_fp_1o_S	<i>Calculate the survivor functions estimated in a 1st order fractional polynomial NMA model. The absolute $S(t)$ estimates combining the estimated baseline survival from a reference trial (in the NMA) with the fractional polynomial (log)hazard ratio estimates to construct the $S(t)$ functions for each treatment.</i>
-------------	--

Description

Calculate the survivor functions estimated in a 1st order fractional polynomial NMA model. The absolute $S(t)$ estimates combining the estimated baseline survival from a reference trial (in the NMA) with the fractional polynomial (log)hazard ratio estimates to construct the $S(t)$ functions for each treatment.

Usage

```

get_fp_1o_S(
  fit,
  ref.std,
  ref.arm,
  treatments,
  time = 1:24,
  bl.node = "mu",
  contrast.node = "d"
)

```

Arguments

<code>fit</code>	JAGS object with FP NMA fit.
<code>ref.std</code>	Numeric identifier of reference study to use for baseline survival estimate.
<code>ref.arm</code>	Numeric identifier of arm in reference study to use for baseline survival estimate.
<code>treatments</code>	Vector with character strings to label the treatments.
<code>time</code>	Vector of time-points at which S(t) functions are calculated.
<code>bl.node</code>	Character to identify the node in the jags model that identifies the baseline estimates (default is "mu").
<code>contrast.node</code>	Character to identify the node in the jags model that identifies the baseline estimates (default is "d").

Details

The FP parameters to calculate the log-hazard curves for each survivor function are combinations `bl.node` + `contrast.node`. This means the combination `ref.std`, `ref.arm` must identify a study arm in the NMA that used the (NMA) reference treatment. The basic parameters (contrasts vs reference) are then added to obtain the parameters for each treatment in the study.

Value

a `data.frame` containing survivor function for each treatment

get_fp_2o

Second order fractional polynomial

Description

Second order fractional polynomial

Usage

```
get_fp_2o(x, params, exponents, sums = NULL)
```


Arguments

x	A vector with the dependent variable.
params	A matrix with two columns giving the intercept and the "slope". If more than one rows given, the fractional polynomial is evaluated for each row.
exponents	A vector giving the exponents (p1, p2) of the polynomial.
sums	A function. If non-null, sums of the fractional polynomial at each x value (i.e. summaries over the different params values) are calculated.

Value

A matrix with length(x) rows giving the FP values or FP summaries.

get_fp_2o_GoF	<i>Calculate the study and arm level survivor functions estimates from a 2nd order fractional polynomial NMA. These estimates provide the basis for a goodness-of-fit graph when plotted along with the input data.</i>
---------------	---

Description

Calculate the study and arm level survivor functions estimates from a 2nd order fractional polynomial NMA. These estimates provide the basis for a goodness-of-fit graph when plotted along with the input data.

Usage

```
get_fp_2o_GoF(fit, time = 1:24, bl.node = "mu", contrast.node = "d")
```

Arguments

fit	JAGS object with FP NMA fit.
time	Vector of time-points at which S(t) functions are calculated.
bl.node	Charactor to identify the node in the jags model that identifies the baseline estimates (default is "mu").
contrast.node	Charactor to identify the node in the jags model that identifies the baseline estimates (default is "d").

Details

The FP parameters to calculate the log-hazard curves for each survivor function are combinations bl.node + contrast.node. Here, every study-specific baseline estimate is combined with the arm-specific contrast to obtain arm level NMA estimates.

Value

a data.frame containing survivor function for each treatment

get_fp_2o_HR	<i>Calculate the time-dependent hazard ratios obtained from fitting a second order fractional polynomial model.</i>
--------------	---

Description

Calculate the time-dependent hazard ratios obtained from fitting a second order fractional polynomial model.

Usage

```
get_fp_2o_HR(
  x,
  fit,
  trt.nos,
  ref.no,
  trt.labs = NULL,
  node = "d",
  CI = TRUE,
  revert = FALSE
)
```

Arguments

x	A vector with the dependent variable.
fit	An rjags object with the output from the JAGS fit.
trt.nos	A vector with the numerical treatment IDs for which the HRs shall be calculated (including the ref).
ref.no	An integer with the numerical ID of the reference for the HR calculations.
trt.labs	A character vector of same length as trt.nos with the treatment labels.
node	A character string that identifies the node in the JAGS model giving the treatment effect estimates.
CI	Logical, shall CIs for the fractional polynomial be given? (Medians are always provided.)

Details

Requires the packages: dplyr, coda.

Value

A data frame with pointwise median (and CI) HRs for all comparisons of trt.nos vs ref.no.

get_fp_2o_S	<i>Calculate the survivor functions estimated in a 2nd order fractional polynomial NMA model. The absolute S(t) estimates combining the estimated baseline survival from a reference trial (in the NMA) with the fractional polynomial (log)hazard ratio estimates to construct the S(t) functions for each treatment.</i>
-------------	--

Description

Calculate the survivor functions estimated in a 2nd order fractional polynomial NMA model. The absolute S(t) estimates combining the estimated baseline survival from a reference trial (in the NMA) with the fractional polynomial (log)hazard ratio estimates to construct the S(t) functions for each treatment.

Usage

```
get_fp_2o_S(
  fit,
  ref.std,
  ref.arm,
  treatments,
  time = 1:24,
  bl.node = "mu",
  contrast.node = "d"
)
```

Arguments

fit	JAGS object with FP NMA fit.
ref.std	Numeric identifier of reference study to use for baseline survival estimate.
ref.arm	Numeric identifier of arm in reference study to use for baseline survival estimate.
treatments	Vector with character strings to label the treatments.
time	Vector of time-points at which S(t) functions are calculated.
bl.node	Character to identify the node in the jags model that identifies the baseline estimates (default is "mu").
contrast.node	Character to identify the node in the jags model that identifies the baseline estimates (default is "d").

Details

The FP parameters to calculate the log-hazard curves for each survivor function are combinations bl.node + contrast.node. This means the combination ref.std, ref.arm must identify a study arm in the NMA that used the (NMA) reference treatment. The basic parameters (contrasts vs reference) are then added to obtain the parameters for each treatment in the study.

Value

a data.frame containing survivor function for each treatment

get_fp_comparison	<i>Extract model information and fit statistics from a list of fractional polynomial NMAs.</i>
-------------------	--

Description

Extract model information and fit statistics from a list of fractional polynomial NMAs.

Usage

```
get_fp_comparison(nmaout.list)
```

Arguments

nmaout.list	List of results lists (which must contain the names elements descr_s, model.pars, fit, DICsamp).
-------------	--

Value

matrix comparing the models

get_fp_contrasts	<i>Extract the treatment contrasts vs the reference in the network</i>
------------------	--

Description

Extract the treatment contrasts vs the reference in the network

Usage

```
get_fp_contrasts(fit.jg, treatments = NULL, node = "d", revert = FALSE)
```

Arguments

fit.jg	The jags output object.
treatments	A vector with the labels for the treatments in the network (in the correct order). If NULL treatments calculate from fit
node	The name of the node identifying the contrast.

get_fp_corr	<i>Calculate correlations between the contrast estimates for multi-dimensional effect estimates for all treatments in a FP NMA.</i>
-------------	---

Description

Calculate correlations between the contrast estimates for multi-dimensional effect estimates for all treatments in a FP NMA.

Usage

```
get_fp_corr(fit, node = "d")
```

Arguments

fit	Jags output
node	The name of the node with the contrasts (default is "d").

Value

a data.frame containing the posterior correlations

get_fp_elements	<i>Extract model information and fit statistics from NMA fit in jags of a fractional polynomial model.</i>
-----------------	--

Description

Extract model information and fit statistics from NMA fit in jags of a fractional polynomial model.

Usage

```
get_fp_elements(nmaout)
```

Arguments

nmaout	A list with the results from NMA fit (jags). The list must contain the names elements 'descr_s, model.pars, fit, DICsamp'.
--------	--

Value

list with description and fit metrics

get_fp_GoF	<i>Calculate the study and arm level survivor functions estimates from a fractional polynomial NMA. These estimates provide the basis for a goodness-of-fit graph when plotted along with the input data.</i>
------------	---

Description

Calculate the study and arm level survivor functions estimates from a fractional polynomial NMA. These estimates provide the basis for a goodness-of-fit graph when plotted along with the input data.

Usage

```
get_fp_GoF(fit, treatments, time = 1:24, bl.node = "mu", contrast.node = "d")
```

Arguments

fit	JAGS object with FP NMA fit.
treatments	Vector with character strings to label the treatments. Argument not currently used
time	Vector of time-points at which S(t) functions are calculated.
bl.node	Character to identify the node in the jags model that identifies the baseline estimates (default is "mu").
contrast.node	Character to identify the node in the jags model that identifies the baseline estimates (default is "d").

Details

The FP parameters to calculate the log-hazard curves for each survivor function are combinations bl.node + contrast.node. Here, every study-specific baseline estimate is combined with the arm-specific contrast to obtain arm level NMA estimates.

Value

a data.frame containing survivor function for each treatment

get_fp_HR	<i>Calculate the time-dependent hazard ratios obtained from fitting a fractional polynomial model (first or second order).</i>
-----------	--

Description

Calculate the time-dependent hazard ratios obtained from fitting a fractional polynomial model (first or second order).

Usage

```
get_fp_HR(
  x,
  fit,
  trt.nos,
  ref.no,
  trt.labs = NULL,
  node = "d",
  CI = TRUE,
  revert = FALSE
)
```

Arguments

x	A vector with the dependent variable.
fit	An rjags object with the output from the JAGS fit.
trt.nos	A vector with the numerical treatment IDs for which the HRs shall be calculated (including the ref).
ref.no	An integer with the numerical ID of the reference for the HR calculations.
trt.labs	A character vector of same length as trt.nos with the treatment labels.
node	A character string that identifies the node in the JAGS model giving the treatment effect estimates.
CI	Logical, shall CIs for the fractional polynomial be given? (Medians are always provided.)

Details

Requires the packages: dplyr, coda.

Value

A data frame with pointwise median (and CI) HRs for all comparisons of trt.nos vs ref.no.

get_fp_S	<i>Calculate the survivor functions estimated in a fractional polynomial NMA model. The absolute $S(t)$ estimates combining the estimated baseline survival from a reference trial (in the NMA) with the fractional polynomial (log)hazard ratio estimates to construct the $S(t)$ functions for each treatment.</i>
----------	--

Description

Calculate the survivor functions estimated in a fractional polynomial NMA model. The absolute $S(t)$ estimates combining the estimated baseline survival from a reference trial (in the NMA) with the fractional polynomial (log)hazard ratio estimates to construct the $S(t)$ functions for each treatment.

Usage

```
get_fp_S(
  fit,
  ref.std,
  ref.arm,
  treatments = NULL,
  time,
  bl.node = "mu",
  contrast.node = "d"
)
```

Arguments

fit	JAGS object with NMA fit.
ref.std	Numeric identifier of reference study to use for baseline survival estimate.
ref.arm	Numeric identifier of arm in reference study to use for baseline survival estimate.
treatments	Vector with character strings to label the treatments. If NULL treatments calculated from fit
time	Vector of time-points at which $S(t)$ functions are calculated.
bl.node	Character to identify the node in the jags model that identifies the baseline estimates (default is "mu").
contrast.node	Character to identify the node in the jags model that identifies the baseline estimates (default is "d").

Details

The FP parameters to calculate the log-hazard curves for each survivor function are combinations `bl.node + contrast.node`. This means the combination `ref.std, ref.arm` must identify a study arm in the NMA that used the (NMA) reference treatment. The basic parameters (contrasts vs reference) are then added to obtain the parameters for each treatment in the study.

Value

A data.frame containing survivor function for each treatment

get_jags_info	<i>Utility function to return jags data and model for reporting (e.g. in appendix)</i>
---------------	--

Description

Utility function to return jags data and model for reporting (e.g. in appendix)

Usage

```
get_jags_info(  
  sims,  
  include.comments = FALSE,  
  input.round.function = function(x) signif(x, digits = 4)  
)
```

Arguments

sims	rjags object
include.comments	Logical, should comments in the model be included or stripped out (default: FALSE)
input.round.function	Function to apply to data inputs - default: function(x) signif(x, digits = 4)

Value

jagsInfo object which contains jags simulation information accessed via a call to print

Author(s)

Iain Bennett (adaptations: Sandro Gsteiger)

get_mtc_allVsNew	<i>Utility function to extract effect estimates "other treatments vs new" from gemtc fit.</i>
------------------	---

Description

Utility function to extract effect estimates "other treatments vs new" from gemtc fit.

Usage

```
get_mtc_allVsNew(x, new.lab, transform = NULL, digits = NULL)
```

Arguments

x	Object of class <code>mtc.result</code> which is the output from a <code>mtc.run</code> .
new.lab	Character string with name of new intervention.
transform	Optional name of transformation to apply to output (e.g. "exp").
digits	Optional integer number of digits to round the output to.

Value

Returns a `data.frame` of effect estimates

get_mtc_newVsAll	<i>Utility function to extract effect estimates "new vs other treatments" from gemtc fit.</i>
------------------	---

Description

Utility function to extract effect estimates "new vs other treatments" from gemtc fit.

Usage

```
get_mtc_newVsAll(x, new.lab, transform = NULL, digits = NULL)
```

Arguments

x	Object of class <code>mtc.result</code> which is the output from a <code>mtc.run</code> .
new.lab	Character string with name of new intervention.
transform	Optional name of transformation to apply to output (e.g. "exp").
digits	Optional integer number of digits to round the output to.

Value

Returns a `data.frame` of effect estimates

get_mtc_probBetter	<i>Utility function to extract probabilities of new treatment being better from gemtc fit (e.g. $P(HR < 1)$ for HRs new vs other).</i>
--------------------	--

Description

Utility function to extract probabilities of new treatment being better from gemtc fit (e.g. $P(HR < 1)$ for HRs new vs other).

Usage

```
get_mtc_probBetter(
  x,
  new.lab,
  smaller.is.better = TRUE,
  threshold = 0,
  sort.by = c("name", "effect")[1],
  digits = 3
)
```

Arguments

x	Object of class mtc.result containing the NMA fit.
new.lab	Character string with name of new intervention.
smaller.is.better	Logical (default is TRUE).
threshold	Numerical value (default is 0) giving the threshold against which contrasts are being compared.
sort.by	Character string, must be either "name" or "effect" (default), indicating the sorting of the output table.
digits	Integer (default is 3); if provided, outputs will be rounded accordingly.

Value

A data.frame of probabilities of a treatment being better from a mtc model

get_mtc_sum	<i>Utility function to extract summary stats from mtc.result object.</i>
-------------	--

Description

Utility function to extract summary stats from mtc.result object.

Usage

```
get_mtc_sum(x, digits = 2)
```

Arguments

`x` Object of class `mtc.result` which is the output from a `mtc.run`.
`digits` Integer specifying the number of digits for rounding (default is 2).

Value

A `data.frame` of `mtc` model summary statistics

<code>get_nw_fromto</code>	<i>Extract edges information ("from-to matrix") from network data frame.</i>
----------------------------	--

Description

Extract edges information ("from-to matrix") from network data frame.

Usage

```
get_nw_fromto(dat)
```

Arguments

`dat` `data.frame` with network data in long format. Must contain columns `study` and `treatment`. Can be IPD, grouped data, or AD.

Value

A matrix with columns "from", "to".

<code>get_pwe_comparison</code>	<i>Extract model information and fit statistics from a list of piecewise-exponential NMA fits.</i>
---------------------------------	--

Description

Extract model information and fit statistics from a list of piecewise-exponential NMA fits.

Usage

```
get_pwe_comparison(nmaout.list)
```

Arguments

`nmaout.list` List of results lists (which must contain the names elements 'descr_s, model.pars, fit, DICsamp').

Value

matrix comparing the models

`get_pwe_contrasts` *Utility function to extract HR estimates from piece-wise exponential model fit in (format needed for ggplot)*

Description

Utility function to extract HR estimates from piece-wise exponential model fit in (format needed for ggplot)

Usage

```
get_pwe_contrasts(
  fit,
  treatments = NULL,
  ref,
  reverse = FALSE,
  exponentiate = TRUE,
  xmax = 24,
  digits = 3,
  alpha = 0.05
)
```

Arguments

`fit` `rjags` object which is the return of `R2jags::jags()`

`treatments` Vector of treatment names (character string) ordered according to array indexes in `jags` fit. If `NULL` treatments extracted from `fit`

`ref` Character string identifying reference treatment to use in calculating contrasts.

`reverse` Logical, if `TRUE` (default) "ref vs others" is calculated (instead of "others vs ref").

`exponentiate` Logical, if `TRUE` (default) contrast estimates are exponentiated in output.

`xmax` Numeric

`digits` Numeric

`alpha` Numeric

Value

A `data.frame` containing hazzard ratio estimates from an `rjags` object which is the return of `R2jags::jags()`

get_pwe_conv_diag	<i>Utility function: convergence diagnostics for piece-wise constant models</i>
-------------------	---

Description

Utility function: convergence diagnostics for piece-wise constant models

Usage

```
get_pwe_conv_diag(fit, file = NULL, patterns = c("d", "mu"), re.pattern = "sd")
```

Arguments

fit	rjags object which is the return of R2jags::jags()
file	Optional output filename (a pdf with this name will be created if provided).
patterns	Vector of character strings used to identify the patterns selected for conv assessment.
re.pattern	Character

Value

ggplot object

get_pwe_elements	<i>Extract model information and fit statistics from NMA fit in jags of a piecewise-exponential model.</i>
------------------	--

Description

Extract model information and fit statistics from NMA fit in jags of a piecewise-exponential model.

Usage

```
get_pwe_elements(nmaout)
```

Arguments

nmaout	A list with the results from NMA fit (jags). The list must contain the named elements 'descr_s, model.pars, fit, DICsamp'.
--------	--

Value

list with description and fit metrics

get_pwe_GoF	<i>Calculate the survivor function estimates for each study and arm. Calculate also the observed survival curves from the binned KM data to compare observed and estimated survivor functions.</i>
-------------	--

Description

Calculate the survivor function estimates for each study and arm. Calculate also the observed survival curves from the binned KM data to compare observed and estimated survivor functions.

Usage

```
get_pwe_GoF(
  fit,
  data.arms,
  data.jg,
  time = 0:60,
  bl.node = "mu",
  contrast.node = "d"
)
```

Arguments

fit	JAGS object with NMA fit.
data.arms	Data frame with one line per study arm and columns study, treatment, studyn, treatmentn.
data.jg	List with input data set that was used in jags fit.
time	Vector of time-points at which S(t) functions are calculated.
bl.node	Charactor to identify the node in the jags model that identifies the baseline estimates (default is "mu").
contrast.node	Charactor to identify the node in the jags model that identifies the baseline estimates (default is "d").

Value

data.frame

get_pwe_S	<i>Calculate the survivor functions estimated in piecewise-constant NMA model. The absolute $S(t)$ estimates combining the estimated baseline survival from a reference trial (in the NMA) with the piecewise-constant hazard ratio estimates to construct the $S(t)$ functions for each treatment.</i>
-----------	---

Description

Calculate the survivor functions estimated in piecewise-constant NMA model. The absolute $S(t)$ estimates combining the estimated baseline survival from a reference trial (in the NMA) with the piecewise-constant hazard ratio estimates to construct the $S(t)$ functions for each treatment.

Usage

```
get_pwe_S(
  fit,
  ref.std,
  ref.arm,
  treatments = NULL,
  time = 0:24,
  bl.node = "mu",
  contrast.node = "d"
)
```

Arguments

fit	JAGS object with NMA fit.
ref.std	Numeric identifier of reference study to use for baseline survival estimate.
ref.arm	Numeric identifier of arm in reference study to use for baseline survival estimate.
treatments	Vector with character strings to label the treatments. If NULL treatments extracted from fit
time	Vector of time-points at which $S(t)$ functions are calculated.
bl.node	Character to identify the node in the jags model that identifies the baseline estimates (default is "mu").
contrast.node	Character to identify the node in the jags model that identifies the baseline estimates (default is "d").

Value

a data.frame containing survivor function for each treatment

get_pw_segments	<i>Utility function to get segments (as character strings) from vector with cutpoints</i>
-----------------	---

Description

Utility function to get segments (as character strings) from vector with cutpoints

Usage

```
get_pw_segments(x = NULL, cuts, right = FALSE, ordered_results = TRUE)
```

Arguments

x	NULL value
cuts	Vector with cut points
right	Logical
ordered_results	Logical

Value

factor of segments

get_segments	<i>Utility function to get segments (as character strings) from vector with cutpoints</i>
--------------	---

Description

Utility function to get segments (as character strings) from vector with cutpoints

Usage

```
get_segments(cut.pts)
```

Arguments

cut.pts	Vector with cut points
---------	------------------------

Value

factor of segments

groupedTTE_fp_pre_proc

Utility function for pre-processing: prepare jags input data for FP model.

Description

Utility function for pre-processing: prepare jags input data for FP model.

Usage

```
groupedTTE_fp_pre_proc(
  dat,
  ref.std = NULL,
  nma.ref.trt = NULL,
  model.pars = NULL,
  feprior_mean = NULL,
  feprior_prec = NULL,
  both.prior = NULL
)
```

Arguments

dat	A data.frame with columns study, treatment, t.start, t.end, n.event, n.risk.
ref.std	A character of the reference study
nma.ref.trt	A character of the reference treatment
model.pars	list, containing exponents (numeric vector of exponents for FP model) and t.eval - one of "midpoint" (the default), "start", or "end", to identify the time-point within each interval used to calculate the hazard rate (over each interval, the hazard function is approximated with this value $h(t.eval)$); alternatively, a numeric value in the interval [0,1] used to define t.eval (where for example "midpoint" corresponds to the value 0.5).
feprior_mean	A numeric value representing feprior_mean (default 0)
feprior_prec	A numeric value representing feprior_prec default 0.0001

Value

A list with input data for jags fit of PWE model.

See Also

[nma_pre_proc](#), [groupedTTE_pwe_pre_proc](#)

groupedTTE_pwe_pre_proc

Utility function for pre-processing: prepare jags input data for PWE model.

Description

Utility function for pre-processing: prepare jags input data for PWE model.

Usage

```
groupedTTE_pwe_pre_proc(
  dat,
  ref.std = NULL,
  nma.ref.trt = NULL,
  model.pars = NULL,
  feprior_mean = NULL,
  feprior_prec = NULL,
  bth.prior = NULL
)
```

Arguments

dat	A data.frame with columns study, treatment, t.start, t.end, n.event, n.risk.
ref.std	A character of the reference study
nma.ref.trt	A character of the reference treatment
model.pars	list, containing exponents (numeric vector of exponents for FP model) and t.eval - one of "midpoint" (the default), "start", or "end", to identify the time-point within each interval used to calculate the hazard rate (over each interval, the hazard function is approximated with this value h(t.eval)); alternatively, a numeric value in the interval [0,1] used to define t.eval (where for example "midpoint" corresponds to the value 0.5).
feprior_mean	A numeric value representing feprior_mean (default 0)
feprior_prec	A numeric value representing feprior_prec default 0.0001
bth.prior	A numeric list containing between-trial heterogeneity priors

Value

A list with input data for jags fit of PWE model.

See Also

[nma_pre_proc](#), [groupedTTE_fp_pre_proc](#)

list_BUGS	<i>Lists all available model files inside the inst directory</i>
-----------	--

Description

Lists all available model files inside the inst directory

Usage

```
list_BUGS(full.path = FALSE)
```

Arguments

full.path	lgl Return full path to file?
-----------	-------------------------------

Value

character vector containing a file name/path

match_args_to_func	<i>Helper function to extract named elements from a list to match the arguments of supplied function</i>
--------------------	--

Description

Helper function to extract named elements from a list to match the arguments of supplied function

Usage

```
match_args_to_func(args, func)
```

Arguments

args	A named list containing arguments to match to 'func'
func	A function of whose argumetns to extract

`mtc.prob.better.table` *Utility function providing pairwise probability of being better (col vs row). (Adapted from `gemtc::relative.effect.table()`).*

Description

Utility function providing pairwise probability of being better (col vs row). (Adapted from `gemtc::relative.effect.table()`).

Usage

```
mtc.prob.better.table(x, smaller.is.better, threshold = 0, covariate = NA)
```

Arguments

`x` Object of class `mtc.result` which is the output from a `mtc.run`.

`smaller.is.better` Logical (default is TRUE).

`threshold` Numerical value (default is 0) giving the threshold against which contrasts are being compared.

`covariate` NA; for compatibility with `gemtc::relative.effect.table()`.

Value

An array of class ‘`mtc.prob.better.table`’

`nma_fit` *Takes input data and a model plan and passes to the model engine specified. Current supported engines are the ‘gemtc’ package (using `mtc.model` & `mtc.run`) or ‘rjags’ (using `jags` and `dic.samples.functions`)*

Description

Takes input data and a model plan and passes to the model engine specified. Current supported engines are the ‘gemtc’ package (using `mtc.model` & `mtc.run`) or ‘rjags’ (using `jags` and `dic.samples.functions`)

Usage

```
nma_fit(model_input)
```

Arguments

`model_input` a list containing named elements `fitting_data` (data which has been pre-processed) and `plan` (list object containing all input parameters)

Value

model object of class 'rjags' or 'mtc.result'

mma_pre_proc	<i>NMA data pre-processing</i>
--------------	--------------------------------

Description

NMA data pre-processing

Usage

```
mma_pre_proc(data, plan)
```

Arguments

data	input data.frame
plan	A list containing the model plan

Value

A network object or if gsd a list containing a network object and jags init parameters

See Also

[groupedTTE_fp_pre_proc](#), [groupedTTE_pwe_pre_proc](#)

plan_binary	<i>Creates a model plan for binary data</i>
-------------	---

Description

Creates a model plan for binary data

Usage

```
plan_binary(
  both.model = c("FE", "RE"),
  engine = "gemtc",
  data.type = "BINARY",
  binary_data_type = "arm_level",
  jags_init = NULL,
  n.chain = NULL,
  n.iter = NULL,
  n.adapt = NULL,
```

```

    thin = NULL,
    both.prior = NULL,
    rsd = 13579,
    model = "binom",
    link = "logit",
    om.scale = 5,
    ...
  )

```

Arguments

bth.model	One of "FE", or "RE"
engine	Only 'gemtc' currently supported
data.type	Only "BINARY" supported
binary_data_type	One of "relative_effect" or "arm_level"
jags_init	A list containing the jags seeds to set.
n.chain	A numeric value representing n.chains
n.iter	A numeric value representing n.iter
n.adapt	A numeric value representing n.adpat/ n.burnin
thin	A numeric value representing thin
bth.prior	Output of using mtc.hy.prior. Only required if bth.model == "RE"
rsd	numeric seed to be set
model	default "binom"
link	default "logit"
om.scale	default is 5
...	any other names arguments must match arguments names from mtc.model

Value

A list containing model, engine, binary_data_type and model params

Examples

```

plan_binary(bth.model = "RE",
            n.chain = 3,
            n.iter = 6000,
            thin = 1,
            n.adapt = 1000,
            link = "logit",
            bth.prior = gemtc::mtc.hy.prior(type = "var", distr = "dlnorm", -4.18, 1 / 1.41 ^ 2))

```

plan_fp	<i>Creates a fractional polynomial model plan</i>
---------	---

Description

Creates a fractional polynomial model plan

Usage

```
plan_fp(
  model.pars,
  both.model = c("FE", "REINT", "RE"),
  ref.std,
  nma.ref.trt,
  engine = "rjags",
  data.type = "GSD",
  feprior_mean = 0,
  feprior_prec = 1e-04,
  descr = "Fractional polynomial model",
  descr_s = "FP",
  n.chains = NULL,
  n.iter = NULL,
  n.burnin = NULL,
  n.thin = NULL,
  both.prior = NULL,
  rsd = 13579,
  model.file = NULL,
  ...
)
```

Arguments

model.pars	A named list containing ‘exponents’ & ‘t.val’ when using GSD
both.model	One of "FE", "REINT", or "RE"
ref.std	A string containing the reference study
nma.ref.trt	A string containing the reference treatment
engine	Only ‘rjags’ currently supported
data.type	Only "GSD" supported
feprior_mean	A numeric value representing feprior_mean (default 0)
feprior_prec	A numeric value representing feprior_prec default 0.0001
descr	A character string to describe the model
descr_s	Short description
n.chains	A numeric value representing n.chains
n.iter	A numeric value representing n.iter

n.burnin	A numeric value representing n.burnin
n.thin	A numeric value representing n.thin
bth.prior	A list containing type & distr and the parameters of distr. Only required if bth.model != "FE"
rsd	numeric seed to be set
model.file	Path to BUGS file. If NULL file name will be built from input parameters and extracted from BUGScode folder if present
...	any other named arguments must match arguments names from dic.samples or jags functions

Details

See the vignettes for the specification of bth.prior. Currently parameter names not fully aligned between PWE and FP.

Value

A list containing model, engine, analysis and model params

Examples

```
plan_fp(model.pars = list(exponents = 0, t.eval = "midpoint"),
        bth.model = "FE", ref.std = "STUDY2", nma.ref.trt = "B",
        model.file = system.file("BUGScode", "gsd_fracpoly-1o_fe.txt", package = "gemtcPlus"))
```

plan_hr	<i>Creates a model plan for hazard ratio</i>
---------	--

Description

Creates a model plan for hazard ratio

Usage

```
plan_hr(
  bth.model = c("FE", "RE"),
  engine = "gemtc",
  data.type = "HR",
  binary_data_type = "relative_effect",
  jags_init = NULL,
  n.chain = NULL,
  n.iter = NULL,
  n.adapt = NULL,
  thin = NULL,
  bth.prior = NULL,
  rsd = 13579,
```

```

    model = "binom",
    link = "identity",
    om.scale = 5,
    ...
)

```

Arguments

bth.model	One of "FE", or "RE"
engine	Only 'gemtc' currently supported
data.type	Only "HR" supported
binary_data_type	One of "relative_effect" or "arm_level"
jags_init	A list containing the jags seeds to set.
n.chain	A numeric value representing n.chains
n.iter	A numeric value representing n.iter
n.adapt	A numeric value representing n.adpat/ n.burnin
thin	A numeric value representing thin
bth.prior	Output of using mtc.hy.prior. Only required if bth.model == "RE"
rsd	numeric seed to be set
model	default "binom"
link	default "logit"
om.scale	default is 5
...	any other names arguments must match arguments names from mtc.model

Value

A list containing model, engine, binary_data_type and model params

Examples

```

plan_hr(bth.model = "RE",
        n.chain = 3,
        n.iter = 6000, thin = 1, n.adapt = 1000,
        link = "identity",
        linearModel = "random",
        bth.prior = gemtc::mtc.hy.prior(type = "var", distr = "dlnorm", -4.18, 1 / 1.41 ^ 2))

```

plan_pwe	<i>Creates a fractional polynomial model plan</i>
----------	---

Description

Creates a fractional polynomial model plan

Usage

```
plan_pwe(
  model.pars,
  both.model = c("FE", "RE"),
  ref.std,
  nma.ref.trt,
  engine = "rjags",
  data.type = "GSD",
  feprior_mean = 0,
  feprior_prec = 1e-04,
  descr = "Piecewise Exponential model",
  descr_s = "PWE",
  n.chains = NULL,
  n.iter = NULL,
  n.burnin = NULL,
  n.thin = NULL,
  both.prior = NULL,
  rsd = 13579,
  model.file = NULL,
  ...
)
```

Arguments

model.pars	A named list containing 'cut.pts' when using GSD
both.model	One of "FE" or "RE"
ref.std	A string containing the reference study
nma.ref.trt	A string containing the reference treatment
engine	Only 'rjags' currently supported
data.type	Only "GSD" supported
feprior_mean	A numeric value representing feprior_mean (default 0)
feprior_prec	A numeric value representing feprior_prec default 0.0001
descr	A character string to describe the model
descr_s	Short description
n.chains	A numeric value representing n.chains
n.iter	A numeric value representing n.iter

n.burnin	A numeric value representing n.burnin
n.thin	A numeric value representing n.thin
bth.prior	A list containing type & distr. Only required if bth.model == "FE"
rsd	A numeric representing seed to be set
model.file	Path to BUGS file. If NULL file name will be built from input parameters and extracted from BUGScode folder if present
...	any other named arguments must match arguments names from dic.samples or jags functions

Value

A list containing model, engine, analysis and model params

Examples

```
model_plan <- plan_pwe(model.pars = list(cut.pts = c(3, 10)),
  bth.model = "FE", ref.std = "STUDY2", nma.ref.trt = "B",
  n.chains = 2,
  n.iter = 6000,
  n.burnin = 1000,
  n.thin = 1)
```

plot_fp_HR	<i>Produce ggplot from HR values in dataframe (medians vs time for several trts, all in one plot)</i>
------------	---

Description

Produce ggplot from HR values in dataframe (medians vs time for several trts, all in one plot)

Usage

```
plot_fp_HR(
  HRs,
  xlab = "Month",
  legend.pos = "right",
  breaks = c(0.125, 0.25, 0.5, 1, 2, 4, 8),
  facet = FALSE,
  ncol = NULL
)
```

Arguments

HRs

Value

fig

plot_mtc_forest	<i>Utility function to do forest plot from data.frame with effect estimates.</i>
-----------------	--

Description

Utility function to do forest plot from data.frame with effect estimates.

Usage

```
plot_mtc_forest(
  x,
  sort.by = c("name", "effect")[1],
  lab = NULL,
  do.log = TRUE,
  breaks = c(0.25, 0.5, 0.8, 1, 1.25, 2, 4)
)
```

Arguments

x	A data.frame with columns: Comparator, Med, CIlo, CIup.
sort.by	Character string defining how estimates are sorted, either "name" (the default) or "effect".
lab	Label to plot underneath x-axis.
do.log	Logical, whether log transformed x-axis or not (default = TRUE).
breaks	Where to put x-axis breaks and labels.

Value

A ggplot2 plot object which

process_binary	<i>Transforms binary data</i>
----------------	-------------------------------

Description

Transforms binary data

Usage

```
process_binary(data)
```

Arguments

data	input data.frame
------	------------------

Value

data.frame

process_gsd *Transforms grouped survival data*

Description

Transforms grouped survival data

Usage

```
process_gsd(data)
```

Arguments

data input data . frame

Value

data.frame

process_hr *Transforms hazard ratio data*

Description

Transforms hazard ratio data

Usage

```
process_hr(data)
```

Arguments

data input data . frame

Value

data.frame

pwe_Hu	<i>Calculate the cumulative hazard over [0, tmax] from piecewise constant model.</i>
--------	--

Description

Calculate the cumulative hazard over [0, tmax] from piecewise constant model.

Usage

```
pwe_Hu(time, cut.pts, haz.rates)
```

Arguments

time	Time-point at which cumulative hazard evaluated, $H(\text{time}) = \int_0^{\text{time}} h(u) du$.
cut.pts	Vector of cut points.
haz.rates	Vector of hazard rates (must have one element more than the cut.pts vector).

Value

numeric estimate of the hazard over [0, tmax]

pwe_S	<i>Calculate the survivor function S(t) from a piecewise exponential model.</i>
-------	---

Description

Calculate the survivor function S(t) from a piecewise exponential model.

Usage

```
pwe_S(time, cut.pts, haz.rates)
```

Arguments

time	Time-point at which survivor fct is evaluated, $S(\text{time}) = \exp(-H(\text{time}))$.
cut.pts	Vector of cut points.
haz.rates	Vector of hazard rates (must have one element more than the cut.pts vector).

Value

numeric estimate of survivor function at time t, S(t)

Index

bth_prior, 3

create_jags_init, 3
create_template, 4

extract_BUGS_file, 4

gemtcPlus, 5
get_fp_1o, 5
get_fp_1o_GoF, 6
get_fp_1o_HR, 6
get_fp_1o_S, 7
get_fp_2o, 8
get_fp_2o_GoF, 9
get_fp_2o_HR, 10
get_fp_2o_S, 11
get_fp_comparison, 12
get_fp_contrasts, 12
get_fp_corrs, 13
get_fp_elements, 13
get_fp_GoF, 14
get_fp_HR, 15
get_fp_S, 16
get_jags_info, 17
get_mtc_allVsNew, 18
get_mtc_newVsAll, 18
get_mtc_probBetter, 19
get_mtc_sum, 19
get_nw_fromto, 20
get_pw_segments, 25
get_pwe_comparison, 20
get_pwe_contrasts, 21
get_pwe_conv_diag, 22
get_pwe_elements, 22
get_pwe_GoF, 23
get_pwe_S, 24
get_segments, 25
groupedTTE_fp_pre_proc, 26, 27, 30
groupedTTE_pwe_pre_proc, 26, 27, 30

list_BUGS, 28

match_args_to_func, 28
mtc.prob.better.table, 29

nma_fit, 29
nma_pre_proc, 26, 27, 30

plan_binary, 30
plan_fp, 32
plan_hr, 33
plan_pwe, 35
plot_fp_HR, 36
plot_mtc_forest, 37
process_binary, 37
process_gsd, 38
process_hr, 38
pwe_Hu, 39
pwe_S, 39