Package ‘nmarank’

September 19, 2021

Title Complex Hierarchy Questions in Network Meta-Analysis
Version 0.2-3
Date 2021-09-16
Depends R (>= 3.3.1), meta (>= 4.19-1), netmeta (>= 1.5-0), data.tree, mvtnorm, tidyr
Imports dplyr, tibble, rlang
Suggestions testthat

Author Adriani Nikolakopoulou [aut], Guido Schwarzer [aut] (<https://orcid.org/0000-0001-6214-9087>), Theodoros Papakonstantinou [aut, cre] (<https://orcid.org/0000-0002-6630-6817>)
Maintainer Theodoros Papakonstantinou <dev@tpapak.com>

URL https://github.com/esm-ispm-unibe-ch/nmarank

Description Derives the most frequent hierarchies along with their probability of occurrence. One can also define complex hierarchy criteria and calculate their probability. Methodology based on Papakonstantinou et al. (2021) <DOI:10.21203/rs.3.rs-858140/v1>.

License GPL-3
Encoding UTF-8
RoxygenNote 7.1.2
NeedsCompilation no
Repository CRAN
Date/Publication 2021-09-19 07:20:02 UTC

R topics documented:

  condition ................................................................. 2
  nmarank ................................................................. 3
  opposite ................................................................. 5
  %AND% ................................................................. 6
  %OR% ................................................................. 7
  %XOR% ................................................................. 8
Condition

Index

condition  Define which hierarchies to select

Description

Defines a condition that is of interest to be satisfied involving a set of treatments in the network.

Usage

condition(fn, ...)

Arguments

fn Character string specifying type of condition.

... Function arguments.

Details

The following types of conditions are available.

The condition fn = "sameHierarchy" checks whether a specific hierarchy occurs. One additional unnamed argument has to be provided in `...`: a vector with a permutation of all treatment names in the network.

The condition fn = "specificPosition" checks whether a treatment ranks in a specific position. Two additional unnamed arguments have to be provided in `...`: (1) name of the treatment of interest and (2) a single numeric specifying the rank position.

The condition fn = "betterEqual" checks whether a treatment has a position better or equal to a specific rank. Two additional unnamed arguments have to be provided in `...`: (1) name of the treatment of interest and (2) a single numeric specifying the rank position.

The condition fn = "retainOrder" checks whether a specific order of two or more treatments is retained anywhere in the hierarchy. One additional unnamed argument has to be provided in `...`: a vector with two or more treatment names providing the order of treatments.

The condition fn = "biggerCIV" checks whether the effect of a treatment is bigger than that of a second treatment by more than a given clinically important value (CIV) on an additive scale (e.g. log odds ratio, log risk ratio, mean difference). Three additional unnamed arguments have to be provided in `...`: (1) name of the first treatment, (2) name of the second treatment and (3) a numerical value for the CIV. Note that the actual value of the relative effect is considered independently of whether small.values is "good" or "bad".

Composition of conditions for more complex queries:

Conditions can be combined to express more complex decision trees. This can be done by using the special operators %AND%, %OR%, %XOR% and the opposite function. The combination should be defined as a binary tree with the use of parentheses. If A, B, C and D are conditions, we can for example combine them into a complex condition E:

E = A %AND% (B %OR% (opposite(C) %XOR% D))
Value

A list with the defined function and its arguments.

See Also

nmarank

Examples

data("Woods2010", package = "netmeta")
p1 <- pairwise(treatment, event = r, n = N, studlab = author,
               data = Woods2010, sm = "OR")
net1 <- netmeta(p1, small.values = "good")

# criterionA if all treatments are in the exact defined order
criterionA <-
  condition("sameHierarchy",
             c("SFC", "Salmeterol", "Fluticasone", "Placebo"))

# criterionB respects the relative order of two or more treatments
criterionB <-
  condition("retainOrder",
             c("SFC", "Fluticasone", "Placebo"))

# Below we define the condition that SFC and Salmeterol are in the
# first two positions.

# We first define conditions that each one of them is in position 1
# or 2

criterionC1 <- condition("betterEqual", "SFC", 2)
criterionC2 <- condition("betterEqual", "Salmeterol", 2)
# We then combine them with operator %AND%
criterionC <- criterionC1 %AND% criterionC2

# Next we can feed the condition into nmarank to get the
# probability of the selection
nmarank(net1, criterionC,
        text.condition =
        "SFC and Salmeterol are the two best options", nsim = 100)

# We can further combine criteria

criterionD <- criterionA %AND% (criterionB %OR% opposite(criterionC))

description

Specifies the frequencies of hierarchies along with their estimated probabilities and the probability
that a specified criterion holds.
Usage

nmrank(
  TE.nma,
  condition = NULL,
  text.condition = "",
  VCOV.nma = NULL,
  pooled,
  nsim = 10000,
  small.values
)

## S3 method for class 'nmrank'
print(
  x,
  text.condition = x$text.condition,
  nrows = 10,
  digits = gs("digits.prop"),
  ...
)

Arguments

 TE.nma Either a netmeta object or a matrix with network estimates.
 condition Defines the conditions that should be satisfied by the treatments in the network. Multiple conditions can be combined with special operators into any decision tree. See condition.
 text.condition Optional descriptive text for the condition.
 VCOV.nma Variance-covariance matrix for network estimates (only considered if argument TE.nma isn’t a netmeta object).
 pooled A character string indicating whether the hierarchy is calculated for the fixed effects ("fixed") or random effects model ("random"). Can be abbreviated.
 nsim Number of simulations.
 small.values A character string specifying whether small treatment effects indicate a "good" or "bad" effect.
 x A nmrank object.
 nrows Number of hierarchies to print.
 digits Minimal number of significant digits for proportions, see print.default.
 ... Additional arguments.

Details

A simulation method is used to derive the relative frequency of all possible hierarchies in a network of interventions. Users can also define the set of all possible hierarchies that satisfy a specified criterion, for example that a specific order among treatments is retained in the network and/or a treatment is in a specific position, and the sum of their frequencies constitute the certainty around the criterion.
Value
An object of class "nmarank" with corresponding print function. The object is a list containing the following components:

- **hierarchies**: A list of the most frequent hierarchies along with their estimated probability of occurrence.
- **probabilityOfSelection**: Combined probability of all hierarchies that satisfy the defined condition.
- **TE.nma, condition, VCOV.nma, pooled, nsim, small.values**: As defined above.

See Also
condition, netmeta

Examples
```r
data( "Woods2010", package = "netmeta" )
p1 <- pairwise( treatment, event = r, n = N, studlab = author, 
    data = Woods2010, sm = "OR" )
net1 <- netmeta( p1, small.values = "good" )
nmarank( net1, nsim = 100 )

criterionA <- condition( "sameHierarchy", 
    c("SFC", "Salmeterol", "Fluticasone", "Placebo") )
nmarank( net1, criterionA, nsim = 100 )
```

The NOT function for a selection statement
It simply reverses condition

Description
The NOT function for a selection statement It simply reverses condition

Usage
```
opposite( cond )
```

Arguments
```
cond condition.
```
Value

Object of class 'data.tree'.

See Also

condition, nmarank

Examples

data("Woods2010", package = "netmeta")
p1 <- pairwise(treatment, event = r, n = N, studlab = author,
  data = Woods2010, sm = "OR")
net1 <- netmeta(p1, small.values = "good")

A = condition("retainOrder", c("Placebo", "Salmeterol", "SFC"))
nmarank(net1, opposite(A), text.condition = "NOT order P-S-S", nsim=5000)


%AND%

Combine selections with AND

Description

Combine selections with AND

Usage

cond1 %AND% cond2

Arguments

cond1 First condition.
cond2 Second condition.

Value

Object of class 'data.tree'.

See Also

condition, nmarank
Examples

data("Woods2010", package = "netmeta")
p1 <- pairwise(treatment, event = r, n = N, studlab = author,
data = Woods2010, sm = "OR")
net1 <- netmeta(p1, small.values = "good")

A <- condition("retainOrder", c("Placebo", "Salmeterol", "SFC"))
B <- condition("betterEqual", "Fluticasone", 2)
nmarank(net1, A %AND% B, nsim=5000)

%OR%

Combine selections with OR

Description

Combine selections with OR

Usage

cond1 %OR% cond2

Arguments

cond1 First condition.
cond2 Second condition.

Value

Object of class 'data.tree'.

See Also

condition, nmarank

Examples

data("Woods2010", package = "netmeta")
p1 <- pairwise(treatment, event = r, n = N, studlab = author,
data = Woods2010, sm = "OR")
net1 <- netmeta(p1, small.values = "good")

A <- condition("retainOrder", c("Placebo", "Salmeterol", "SFC"))
B <- condition("betterEqual", "Fluticasone", 2)

nmarank(net1, A %OR% B, nsim=5000)
Combine selections with XOR

Description
Combine selections with XOR

Usage
cond1 %XOR% cond2

Arguments
cond1  First condition.
cond2  Second condition.

Value
Object of class ’data.tree’.

See Also
condition, nmarank

Examples
```r
data("Woods2010", package = "netmeta")
p1 <- pairwise(treatment, event = r, n = N, studlab = author,
               data = Woods2010, sm = "OR")
net1 <- netmeta(p1, small.values = "good")

A <- condition("retainOrder", c("Placebo", "Salmeterol", "SFC"))
B <- condition("betterEqual", "Fluticasone", 2)

nmarank(net1, A %XOR% B, nsim=3000)
```
Index

%AND%, 6
%OR%, 7
%XOR%, 8

c条件, 2, 4–8

netmeta, 4, 5
nmarank, 3, 3, 4, 6–8

opposite, 5

print.nmarank(nmarank), 3