Package 'pcnetmeta'

July 19, 2014

Type Package

Title Methods for patient-centered network meta-analysis

Version 1.2

Date 2014-07-18

Author Lifeng Lin, Jing Zhang <zhan0979@umn.edu>, and Haitao Chu <chux0051@umn.edu>

Maintainer Lifeng Lin <linl@umn.edu>

Depends R (>= 3.0.2), R2jags (>= 0.03-11), network (>= 1.8.1), runjags

SystemRequirements jags (>=3.0.0)

Description Provides functions to conduct network meta-analysis using arm-based method, which was proposed by Zhang et al (2014); contains functions to generate summary table for effect sizes (e.g., odds ratio, population-averaged event rate), plot network comparisons, and plot credible interval for population-averaged event rate.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2014-07-19 18:38:33

R topics documented:

pcnetmeta-package *Methods for patient-centered network meta-analysis*

Description

Provides functions to conduct network meta-analysis using arm-based method, which was proposed by Zhang et al (2014); contains functions to generate summary table for effect sizes (e.g., odds ratio, population-averaged event rate), plot network comparisons, and plot credible interval for population-averaged event rate.

Details

Package:	pcnetmeta
Type:	Package
Version:	1.2
Date:	2014-07-18
Depends:	R (>= 3.0.2), R2jags, network, runjags
License:	GPL (>=2)

Currently, most popular approaches in network meta-analysis are contrast-based, and they focus on modeling the relative treatment effects, such as odds ratio. However, the arm-based method, proposed by Zhang et al (2014), considers modeling from the perspective of missing data analysis, and it focuses on estimating the absolute risk for each treatment. This package conducts network meta-analysis by the arm-based method, and `nma.ab` is the most important function, which generates summary result file containing estimated absolute risk (AR), relative risk (RR), risk difference (RD), odds ratio (OR), etc.

Analysis in this package is conducted through JAGS. Note that this package does not include a copy of JAGS library, so you must install this software separately. Please refer to the JAGS home page at `http://mcmc-jags.sourceforge.net/` for instructions on downloading JAGS.

Author(s)

Lifeng Lin, Jing Zhang <zhan0979@umn.edu>, and Haitao Chu <chux0051@umn.edu>

Maintainer: Lifeng Lin <linl@umn.edu>

References

Zhang J, Carlin BP, Neaton JD, Soon GG, Nie L, Kane R, Virnig BA, Chu H (2014). "Network meta-analysis of randomized clinical trials: Reporting the proper summaries." *Clin Trials* **11**(2), 246–262.

Lu G, Ades AE (2004). "Combination of direct and indirect evidence in mixed treatment comparisons." *Stat Med* **23**(20), 3105–24.

Butts CT (2008). "network: A Package for Managing Relational Data in R." *J Stat Softw* **24**(2), 1–36.

Middleton LJ et al. (2010) "Hysterectomy, endometrial destruction, and levonorgestrel releasing intrauterine system (Mirena) for heavy menstrual bleeding: systematic review and meta-analysis of data from individual patients." *BMJ* **341**, c3929.

`ci.plot` *Plot credible interval*

Description

`ci.plot` generates credible interval plot for the population-averaged event rates given by the function `nma.ab`.

Usage

```
ci.plot(summary.stat, trtname, graphtitle = "")
```

Arguments

`summary.stat`	a data frame read from the summary result file generated by the function `nma.ab`. We can use the function `read.table` to import the summary file.
`trtname`	a vector of character string indicating treatment names. These will be shown on x-axis in the plot. The default is "trt1", "trt2", and so on.
`graphtitle`	a charater string indicating the graph title. The default is an empty character.

Value

A credible interval plot for population-averaged event rate is generated. Treatment names are placed on x-axis, while the y-axis shows the lower bound, upper bound and median of credible interval.

Author(s)

Lifeng Lin, Jing Zhang, and Haitao Chu.

References

Zhang J, Carlin BP, Neaton JD, Soon GG, Nie L, Kane R, Virnig BA, Chu H (2014). "Network meta-analysis of randomized clinical trials: Reporting the proper summaries." *Clin Trials* **11**(2), 246–262.

Examples

```
## CI plot for network Middleton 2010
data(Middleton10)
attach(Middleton10)
set.seed(12345)
nma.ab(s.id = sid, t.id = tid, event.n = r, total.n = n,
       f.name = "Middleton10_", n.iter = 500, dic = FALSE)
detach(Middleton10)
```

```
ci.plot(summary.stat = read.table("Middleton10_Summary.stat", header = TRUE),
        graphtitle = "CI plot of estimated event rate for network Middleton10",
        trtname = c("First generation", "Hysterectomy", "Second generation",
        "Mirena"))
```

`create.tab` *Generate a summary table file for arm-based method*

Description

`create.tab` generates a concise table file from the summary result given by the function `nma.ab`, and the table demonstrates mutual relationships between different treatments. In the table file, diagonal elements are population-averaged event rates; upper and lower triangular elements show the risk ratio (RR), risk difference (RD), or odds ratio (OR) with the corresponding credible interval or standard error.

Usage

```
create.tab(summary.stat, type = "CI", compare = 1, f.name = "", trtname,
           order = TRUE, o.path = getwd())
```

Arguments

`summary.stat`	a data frame read from the summary result file generated by the function `nma.ab`. We can use the function `read.table` to import the file.
`type`	a charater string. It should be set as either `"CI"` (the default) or `"SE"`. If set as `"CI"`, credible interval would follow the estimated value in the generated table file; otherwise, estimated value would be followed by the corresponding standard error.
`compare`	an integer chosen from `1` (the default), `2`, `3`, `4`, `5`, and `6`. `1` stands for upper triangular elements in the table being RR and lower being RD; `2` for upper RD and lower RR; `3` for upper RR and lower OR; `4` for upper OR and lower RR; `5` for upper RD and lower OR; `6` for upper OR and lower RD.
`f.name`	a character string indicating the name of generated table file. The default is an empty character.
`trtname`	a vector of character string indicating treatment names, and they would be used as the row and column names in the generated table. The default is "treat1", "treat2", and so on.
`order`	logical. If `TRUE` (the default), lower treatment ID is compared to higher treatment ID for effect sizes (e.g., RR, RD, and OR) in the off-diagonal elements in the generated table; otherwise, higher treatment ID is compared to lower one for corresponding effect size.
`o.path`	output path. If not specified, it would be set as the working directory before runing this function.

Value

A file containing a summary table for treatment comparisons, with diagonal elements being population-averaged event rates and off-diagonal elements being RRs, RDs, or ORs with their credible intervals or standard errors.

Author(s)

Lifeng Lin, Jing Zhang, and Haitao Chu.

References

Zhang J, Carlin BP, Neaton JD, Soon GG, Nie L, Kane R, Virnig BA, Chu H (2014). "Network meta-analysis of randomized clinical trials: Reporting the proper summaries." *Clin Trials* **11**(2), 246–262.

Examples

```
## summary RR-RD table for network Middleton 2010
data(Middleton10)
attach(Middleton10)
set.seed(12345)
nma.ab(s.id = sid, t.id = tid, event.n = r, total.n = n,
       f.name = "Middleton10_", n.iter = 500, dic = FALSE)
detach(Middleton10)
create.tab(summary.stat = read.table("Middleton10_Summary.stat", header = TRUE),
           type = "CI", f.name = "Middleton10-")
```

Middleton10 *Network meta-analysis: Middleton 2010*

Description

`Middleton10` is a network data set served as an example for the function `nma.networkplot` and the main function `nma.ab` in this **pcnetmeta** package.

Usage

```
data("Middleton10")
```

Format

A data frame containing 20 randomized controlled trials on 4 treatments.

`sid` a numeric vector indicating study IDs.

`tid` a numeric vector indicating treatment IDs.

`r` a numeric vector indicating event number for a certain treatment in the corresponding study.

`n` a numeric vector indicating total number of participants for a certain treatment in the corresponding study.

Details

Treatment IDs stand for 1: First generation; 2: Hysterectomy; 3: Second generation; 4: Mirena.

Source

Middleton LJ et al. (2010) "Hysterectomy, endometrial destruction, and levonorgestrel releasing intrauterine system (Mirena) for heavy menstrual bleeding: systematic review and meta-analysis of data from individual patients." *BMJ* **341**, c3929.

`nma.ab` *Arm-based method in network meta-analysis*

Description

`nma.ab` generates a summary file for effect sizes by conducting the arm-based approach, proposed by Zhang et al (2014), in network meta-analysis. The generated summary file contains odds ratio (OR) and patient-centered paramenters, such as risk ratio (RR), risk difference (RD), and absolute risk (AR). Also, it can provide DIC statistics for checking the goodness of fit; give trace plots to check the MCMC convergence; generate posterior density plot for the population-averaged event rates.

Usage

```
nma.ab(s.id, t.id, event.n, total.n, o.path = getwd(), f.name = "",
       model = "het1", sigma2.a = 0.001, sigma2.b = 0.001, mu.prec = 0.001,
       R, param = c("probt","RR","RD","OR","rk","best"), trtname,
       n.iter = 200000, n.burnin = floor(n.iter/2), n.chains = 2,
       n.thin = max(1, floor((n.iter - n.burnin)/100000)), dic = TRUE,
       trace = FALSE, postdens = FALSE)
```

Arguments

`s.id`	a numeric vector of natural numbers, indicating study ID.
`t.id`	a numeric vector of natural numbers, indicating treatment ID.
`event.n`	a numeric vector of non-negative numbers, indicating event number for a certain treatment in the corresponding study.
`total.n`	a numeric vector of non-negative numbers, indicating total number of participants for a certain treatment in the corresponding study.
`o.path`	a character string indicating output path. If not specified, it would be set as the working directory before runing this function.
`f.name`	a character string indicating the file name of output results, including network meta-analysis summary, DIC statistics (if `dic = TRUE`), trace plot (if `trace = TRUE`) and posterior density plot (if `postdens = TRUE`). The default is an empty character.

`model` a character string indicating the Bayesian hierarchical model applied in the network meta-analysis. This argument could be set as `"hom"` (homogeneous variance, denoted as model HOM), `"het1"` (model HET1), or `"het2"` (model HET2). Both model HET1 and HET2 account for heterogenous variances of random effects. The default is `"het1"`. See "Details" for the JAGS models.

`sigma2.a` a positive number, specifying the first parameter of the inverse gamma prior for variance(s) of random effects in model HOM and HET1. The default is 0.001.

`sigma2.b` a positive number, specifying the second parameter of the inverse gamma prior for variance(s) of random effects in model HOM and HET1. The default is 0.001. When both `sigma2.a` and `sigma2.b` are set as the default, the prior is weakly informative.

`mu.prec` a positive number, specifying the precision (the reciprocal of the variance) of the normal prior for fixed effects in model HOM, HET1, and HET2. The default is 0.001, in which case the prior is weakly informative.

`R` a `tN` by `tN` covariance matrix for Wishart prior in the model HET2 (`model = "het2"`), where `tN` is the number of treatments. The default is a matrix with diagonal elements being 1 and off-diagonal elements being 0.005, since this correlation matrix could cover the most general cases.

`param` a vector of character string indicating the node(s) to be included in the summary result file. The default is "probt" (population-averaged event rate), "RR", "RD", "OR", "rk" (rank of treatment), "best" (probability of being the best treatment). In addition to the default parameters, "mu" (fixed effect), "vi" (random effect), and "sigma" (variance for random effects, only if `model = "hom"` or `"het1"`) could be selected into `param`.

`trtname` a vector of character string indicating the treatment name for the corresponding treatment id specified by argument `t.id`. It is optional, and if specified, the names would be shown in the posterior density plot for event rate (when `postdens = TRUE`).

`n.iter` the total number of iterations in each chain. The default is 200,000.

`n.burnin` the number of iterations for burn-in. The default is the largest integer not greater than `n.iter/2`.

`n.chains` the number of parallel chains for the model. The default is 2.

`n.thin` a positive integer indicating thinning rate.

`dic` logical. If `TRUE` (the default), the function would generate a file containing the DIC statistics, and a node named "deviance" would be contained in the summary result file; otherwise, the DIC statistics would not be calculated.

`trace` logical. If `TRUE`, the function would save the trace plots for monitored nodes in "probt", "RR", "RD", and "OR". The default is `FALSE`.

`postdens` logical. If `TRUE`, the function would save the posterior density plot for population-averaged event rate (the node "probt"). The default is `FALSE`.

Details

The homogeneous model (`model = "hom"`) considers a common variance for the random effects, and it assumes that the random effects for different treatments are the same in each study. The JAGS model is given as follows:

```
model{
 for(i in 1:sN){
  p[i] <- phi(mu[t[i]] + sigma*vi[s[i]])
  r[i] ~ dbin(p[i], totaln[i])
 }
 for(j in 1:tS){
  vi[j] ~ dnorm(0, 1)
 }
 sigma <- 1/sqrt(tau)
 tau ~ dgamma(sigma2.a, sigma2.b)
 for(j in 1:tN){
  mu[j] ~ dnorm(0, mu.prec)
  probt[j] <- phi(mu[j]/sqrt(1 + 1/tau))
 }
 for(j in 1:tN){
  for(k in 1:tN){
   RR[j, k] <- probt[j]/probt[k]
   RD[j, k] <- probt[j] - probt[k]
   OR[j, k] <- probt[j]/(1 - probt[j])/probt[k]*(1 - probt[k])
  }
 }
 rk[1:tN] <- tN + 1 - rank(probt[])
 best[1:tN] <- equals(rk[], 1)
}
```

The first heterogeneous model (`model = "het1"`) accounts for the heterogeneity for the variances of random effects, but it still assumes that the random effects for different treatments are the same in each study. The following shows the corresponding JAGS model:

```
model{
 for(i in 1:sN){
  p[i] <- phi(mu[t[i]] + sigma[t[i]]*vi[s[i]])
  r[i] ~ dbin(p[i], totaln[i])
 }
 for(j in 1:tS){
  vi[j] ~ dnorm(0, 1)
 }
 for(j in 1:tN){
  mu[j] ~ dnorm(0, mu.prec)
  tau[j] ~ dgamma(sigma2.a, sigma2.b)
  sigma[j] <- 1/sqrt(tau[j])
  probt[j] <- phi(mu[j]/sqrt(1 + 1/tau[j]))
 }
 for(j in 1:tN){
  for(k in 1:tN){
   RR[j, k] <- probt[j]/probt[k]
   RD[j, k] <- probt[j] - probt[k]
   OR[j, k] <- probt[j]/(1 - probt[j])/probt[k]*(1 - probt[k])
  }
```

```
 }
 rk[1:tN] <- tN + 1 - rank(probt[])
 best[1:tN] <- equals(rk[], 1)
}
```

The second heterogeneous model (`model = "het2"`) covers the most general cases, and it employs a Wishart prior for the inverse of covariance matrix for random effects. The JAGS model is defined as follows:

```
model{
 for(i in 1:sN){
  p[i] <- phi(mu[t[i]] + vi[s[i], t[i]])
  r[i] ~ dbin(p[i], totaln[i])
 }
 for(j in 1:tS){
  vi[j, 1:tN] ~ dmnorm(mn[1:tN], T[1:tN, 1:tN])
 }
 invT[1:tN, 1:tN] <- inverse(T[,])
 for(j in 1:tN){
  mu[j] ~ dnorm(0, mu.prec)
  sigma[j] <- sqrt(invT[j, j])
  probt[j] <- phi(mu[j]/sqrt(1 + invT[j, j]))
 }
 T[1:tN, 1:tN] ~ dwish(R[1:tN, 1:tN], tN)
 for(j in 1:tN){
  for(k in 1:tN){
   RR[j, k] <- probt[j]/probt[k]
   RD[j, k] <- probt[j] - probt[k]
   OR[j, k] <- probt[j]/(1 - probt[j])/probt[k]*(1 - probt[k])
  }
 }
 rk[1:tN] <- tN + 1 - rank(probt[])
 best[1:tN] <- equals(rk[], 1)
}
```

Value

`nma.ab` generates a summary statistics file using the arm-based method. Furthermore, this function would give a DIC statistics file if `dic = TRUE`, a trace plot file if `trace = TRUE`, a posterior density file if `postdens = TRUE`.

In the summary file, each row contains statistics for corresponding OR, RD, RR, population-averaged event rate ("probt"), rank of treatment ("rk"), probability of being the best treatment ("best"), etc. Note that RR[i, j], RD[i, j] or OR[i, j] means that treatment i is compared with treatment j, e.g., RD[i,j] = probt[i] - probt[j]. The columns show the statistics of these nodes, including mean, standard deviance, 2.5% percentile, median, and 97.5% percentile. Also, potential scale reduction factor (PSRF) in Gelman and Rubin's MCMC convergence diagnostic is saved in the column "Rhat" for each node.

The DIC file contains the value of pD and DIC; the trace plot file shows the traces for each node in

"probt", "RR", "RD" and "OR"; posterior density plot file contains the combined posterior density for population-averaged event rate.

Note

If there exists a treatment with all of event numbers equal to 0 or corresponding total participant number, JAGS may come to an error or give incorrect results. To avoid this problem, you need to add (if event rate is 0%) or subtract (if event rate is 100%) a small value (e.g., 0.5) to/from those event numbers.

Author(s)

Lifeng Lin, Jing Zhang, and Haitao Chu.

References

Zhang J, Carlin BP, Neaton JD, Soon GG, Nie L, Kane R, Virnig BA, Chu H (2014). "Network meta-analysis of randomized clinical trials: Reporting the proper summaries." *Clin Trials* **11**(2), 246–262.

Lu G, Ades AE (2004). "Combination of direct and indirect evidence in mixed treatment comparisons." *Stat Med* **23**(20), 3105–24.

Examples

```
data(Middleton10)
attach(Middleton10)
set.seed(12345)
nma.ab(s.id = sid, t.id = tid, event.n = r, total.n = n,
       model = "hom", f.name = "Middleton10_hom_", n.iter = 500)
nma.ab(s.id = sid, t.id = tid, event.n = r, total.n = n,
       model = "het1", f.name = "Middleton10_het1_", n.iter = 500,
       trtname = c("FG", "H", "SG", "M"), trace = TRUE, postdens = TRUE)
detach(Middleton10)
```

---

nma.networkplot *Network plot of treatment comparisons*

Description

nma.networkplot plots a visual network presenting comparisons of treatments in various studies.

Usage

```
nma.networkplot(c1, c2, percomparison, trtname, weight = FALSE, VAR1,
                graphtitle, thickness, nodetextsize, nodesize)
```

Arguments

c1	a numeric vector of natural numbers. If the third argument percomparison is set as TRUE, c1 represents treatment ID; otherwise, it represents study ID.
c2	a numeric vector of natural numbers indicating treatment ID.
percomparison	logical. If TRUE, c1 would be treated as the treatment in the first arm of a comparison while c2 would be the second arm; otherwise, c1 is the study ID column while c2 is the treatment vector.
trtname	a vector of character string indicating the treatment names. It is optional, and the default is "treat.1", "treat.2", and so on.
weight	logical. If TRUE, the weights of edges (i.e., total number of corresponding comparison in the network study) would be shown in the network plot. The default is FALSE.
VAR1	a numeric vector used to plot the size of each node according to a treatment characteristic, e.g., by providing a vector with the sample size randomized in each treatment. It is an optional argument, and the default is to plot the node proportional to the number of trials that include the given treatment.
graphtitle	a character string indicating the gragh title. It is optional, and the default is to plot without a title.
thickness	a numeric value used to change the thickness of the edges. It is optional, and the default is 10.
nodetextsize	a numeric value used to change the text size of the node label. It is optional, and the default is 1.
nodesize	a numeric value used to change the size of nodes. It is optional, and the default is 5.

Value

A network plot is generated by this function. Each node in the plot represents a treatment, and the links between nodes indicate studies comparing pairs of corresponding treatments.

Warning

When large values given to the nodetextsize argument (e.g., larger than 4), the graph might be forced out of the margins.

Author(s)

Antonis Mairgiotis.

References

Butts CT (2008). "network: A Package for Managing Relational Data in R." *J Stat Softw*, **24**(2), 1–36.

Examples

```
## pairwise comparison between elements (IDs) in two vectors
t1 <- c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 7, 1, 1, 1, 1, 1, 1, 3, 2, 3)
t2 <- c(2, 3, 4, 4, 5, 5, 8, 8, 8, 8, 10, 7, 9, 9, 9, 10, 6, 6, 3, 4)
nma.networkplot(c1 = t1, c2 = t2, percomparison = TRUE)

## network plot for network Middleton10
data(Middleton10)
attach(Middleton10)
nma.networkplot(c1 = sid, c2 = tid, percomparison = FALSE,
                weight = TRUE, graphtitle = "Middleton 2010",
                trtname = c("FG", "H", "SG", "M"))
detach(Middleton10)
```

Index