Package ‘remaCor’

September 7, 2022

Type Package

Title Random Effects Meta-Analysis for Correlated Test Statistics

Version 0.0.9

Date 2022-09-06

Description Meta-analysis is widely used to summarize estimated effects sizes across multiple statistical tests. Standard fixed and random effect meta-analysis methods assume that the estimated of the effect sizes are statistically independent. Here we relax this assumption and enable meta-analysis when the correlation matrix between effect size estimates is known. Fixed effect meta-analysis uses the method of Lin and Sullivan (2009) <doi:10.1016/j.ajhg.2009.11.001>, and random effects meta-analysis uses the method of Han, et al. <doi:10.1093/hmg/ddw049>.

License Artistic-2.0

URL `https://github.com/GabrielHoffman/remaCor`

BugReports `https://github.com/GabrielHoffman/remaCor/issues`

Suggests knitr, metafor

Depends R (>= 3.6.0), RUnit, clusterGeneration, ggplot2, grid, reshape2, methods

Imports mvtnorm, compiler, Rdpack, stats

VignetteBuilder knitr

RdMacros Rdpack

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation no

Author Gabriel Hoffman [aut, cre]

Maintainer Gabriel Hoffman <`gabriel.hoffman@mssm.edu`>

Repository CRAN

Date/Publication 2022-09-07 08:30:02 UTC

R topics documented:

LS *Fixed effect meta-analysis for correlated test statistics*

Description

Fixed effect meta-analysis for correlated test statistics using the Lin-Sullivan method.

Usage

```
LS(beta, stders, cor = diag(1, length(beta)))
```

Arguments

`beta`	regression coefficients from each analysis
`stders`	standard errors corresponding to betas
`cor`	correlation matrix between of test statistics. Default considers uncorrelated test statistics

Details

Perform fixed effect meta-analysis for correlated test statistics using method of Lin and Sullivan (2009). By default, correlation is set to identity matrix to for independent test statistics.

This method requires the correlation matrix to be symmatric positive definite (SPD). If this condition is not satisfied, results will be NA. If the matrix is not SPD, there is likely an issue with how it was generated.

However, evaluating the correlation between observations that are not pairwise complete can give correlation matricies that are not SPD. In this case, consider running `Matrix::nearPD( x, corr=TRUE)` to produce the nearest SPD matrix to the input.

Value

Return values:

- `beta`: effect size
- `se`: effect size standard error
- `p`: p-value

References

Lin D, Sullivan PF (2009). "Meta-analysis of genome-wide association studies with overlapping subjects." *The American Journal of Human Genetics*, **85**(6), 862–872. https://doi.org/10.1016/j.ajhg.2009.11.001.

Examples

```
# Generate effects
library(mvtnorm)
library(clusterGeneration )

n = 4
Sigma = cov2cor(genPositiveDefMat(n)$Sigma)
beta = t(rmvnorm(1, rep(0, n), Sigma))
stders = rep(.1, n)

# Run fixed effects meta-analysis,
# assume identity correlation
LS( beta, stders)

# Run random effects meta-analysis,
# assume identity correlation
RE2C( beta, stders)

# Run fixed effects meta-analysis,
# account for correlation
LS( beta, stders, Sigma)

# Run random effects meta-analysis,
# account for correlation
RE2C( beta, stders, Sigma)
```

`pkg.env` *Local environment*

Description

Local environment

Usage

```
pkg.env
```

Format

An object of class `environment` of length 0.

plotCor *Correlation plot*

Description

Correlation plot

Usage

```
plotCor(cor)
```

Arguments

cor	correlation matrix between of test statistics. Default considers uncorrelated test statistics

Value

Plot of correlation matrix

Examples

```
# Generate effects
library(mvtnorm)
library(clusterGeneration )

n = 4
Sigma = cov2cor(genPositiveDefMat(n)$Sigma)
beta = t(rmvnorm(1, rep(0, n), Sigma))
stders = rep(.1, n)

# set names
rownames(Sigma) = colnames(Sigma) = LETTERS[1:n]
rownames(beta) = names(stders) = LETTERS[1:n]

# Run random effects meta-analysis,
# account for correlation
RE2C( beta, stders, Sigma)

# Make plot
plotCor( Sigma )
```

plotForest *Forest plot of coefficients*

Description

Forest plot of coefficients

Usage

```
plotForest(beta, stders)
```

Arguments

beta	regression coefficients from each analysis
stders	standard errors corresponding to betas

Value

Forest plot of effect sizes and standard errors

Examples

```
# Generate effects
library(mvtnorm)
library(clusterGeneration )

n = 4
Sigma = cov2cor(genPositiveDefMat(n)$Sigma)
beta = t(rmvnorm(1, rep(0, n), Sigma))
stders = rep(.1, n)

# set names
rownames(Sigma) = colnames(Sigma) = LETTERS[1:n]
rownames(beta) = names(stders) = LETTERS[1:n]

# Run random effects meta-analysis,
# account for correlation
RE2C( beta, stders, Sigma)

# Make plot
plotForest( beta, stders )
```

RE2C *Random effect meta-analysis for correlated test statistics*

Description

Random effect meta-analysis for correlated test statistics using RE2C

Usage

```
RE2C(beta, stders, cor = diag(1, length(beta)), twoStep = FALSE)
```

Arguments

`beta` regression coefficients from each analysis

`stders` standard errors corresponding to betas

`cor` correlation matrix between of test statistics. Default considers uncorrelated test statistics

`twoStep` Apply two step version of RE2C that is designed to be applied only after the fixed effect model.

Details

Perform random effect meta-analysis for correlated test statistics using RE2 method of Han and Eskin (2011), or RE2 for correlated test statistics from Han, et al., (2016). Also uses RE2C method of Lee, Eskin and Han (2017) to further test for heterogenity in effect size. By default, correlation is set to identity matrix to for independent test statistics.

This method requires the correlation matrix to be symmatric positive definite (SPD). If this condition is not satisfied, results will be NA. If the matrix is not SPD, there is likely an issue with how it was generated.

However, evaluating the correlation between observations that are not pairwise complete can give correlation matricies that are not SPD. In this case, consider running `Matrix::nearPD( x, corr=TRUE)` to produce the nearest SPD matrix to the input.

Value

Return values:

- `stat1`: statistic testing effect mean
- `stat2`: statistic testing effect heterogeneity
- `RE2Cp`: RE2 p-value accounting for correlelation between tests
- `RE2Cp.twoStep`: two step RE2C test after fixed effect test. Only evaluated if twoStep==TRUE
- `QE`: test statistic for the test of (residual) heterogeneity
- `QEp`: p-value for the test of (residual) heterogeneity
- `Isq`: I^2 statistic

`QE`, `QEp` and `ISq` are only evaluted if correlation is diagonal

References

Lee CH, Eskin E, Han B (2017). “Increasing the power of meta-analysis of genome-wide association studies to detect heterogeneous effects.” *Bioinformatics*, **33**(14), i379–i388. https://doi.org/10.1093/bioinformatics/btx242.

Han B, Duong D, Sul JH, de Bakker PI, Eskin E, Raychaudhuri S (2016). “A general framework for meta-analyzing dependent studies with overlapping subjects in association mapping.” *Human Molecular Genetics*, **25**(9), 1857–1866. https://doi.org/10.1093/hmg/ddw049.

Han B, Eskin E (2011). “Random-effects model aimed at discovering associations in meta-analysis of genome-wide association studies.” *The American Journal of Human Genetics*, **88**(5), 586–598. https://doi.org/10.1016/j.ajhg.2011.04.014.

Examples

```
# Generate effects
library(mvtnorm)
library(clusterGeneration )

set.seed(1)
n = 4
Sigma = cov2cor(genPositiveDefMat(n)$Sigma)
beta = t(rmvnorm(1, rep(0, n), Sigma))
stders = rep(.1, n)

# Run fixed effects meta-analysis,
# assume identity correlation
LS( beta, stders)

# Run random effects meta-analysis,
# assume identity correlation
RE2C( beta, stders)

# Run fixed effects meta-analysis,
# account for correlation
LS( beta, stders, Sigma)

# Run random effects meta-analysis,
# account for correlation
RE2C( beta, stders, Sigma)
```

Index