Package 'remaCor'

July 23, 2025

Type Package

Title Random Effects Meta-Analysis for Correlated Test Statistics

Version 0.0.18

Date 2024-02-08

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://diseaseneurogenomics.github.io/remaCor/

BugReports https://github.com/DiseaseNeurogenomics/remaCor/issues

Suggests knitr, RUnit, clusterGeneration, metafor

Depends R (>= 3.6.0), ggplot2, methods

Imports mvtnorm, grid, reshape2, compiler, Rcpp, EnvStats, Rdpack, stats

VignetteBuilder knitr

RdMacros Rdpack

Encoding UTF-8

RoxygenNote 7.2.3

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Author Gabriel Hoffman [aut, cre] (ORCID: <https://orcid.org/0000-0002-0957-0224>)

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

Repository CRAN

Date/Publication 2024-02-08 15:00:02 UTC

Contents

`hotelling` *Hottelling T^2 test for multivariate regression*

Description

Hottelling T^2 test compares estimated regression coefficients to specified values under the null. This tests a global hypothesis for all specified coefficients. It uses the F-distribution as the null for the test statistic which is exact under finite sample size.

Usage

```
hotelling(beta, Sigma, n, mu_null = rep(0, length(beta)))
```

Arguments

`beta`	regressioin coefficients
`Sigma`	covariance matrix of regression coefficients
`n`	sample size used for estimation
`mu_null`	the values of the regression coefficients under the null hypothesis. Defaults to all zeros

Details

The Hotelling T2 test is not defined when n - p < 1. Returns `data.frame` with `stat` = `pvalue` = `NA`.

Examples

```
library(clusterGeneration)
library(mvtnorm)

# sample size
n = 30

# number of response variables
m = 2

# Error covariance
```

```
Sigma = genPositiveDefMat(m)$Sigma

# regression parameters
beta = matrix(.6, 1, m)

# covariates
X = matrix(rnorm(n), ncol=1)

# Simulate response variables
Y = X %*% beta + rmvnorm(n, sigma = Sigma)

# Multivariate regression
fit = lm(Y ~ X)

# extract coefficients and covariance
# corresponding to the x variable
beta = coef(fit)['X',]
S = vcov(fit)[c(2,4), c(2,4)]

# perform Hotelling test
hotelling(beta, S, n)
```

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

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

```
library(clusterGeneration)
library(mvtnorm)

# sample size
n = 30

# number of response variables
m = 6

# Error covariance
Sigma = genPositiveDefMat(m)$Sigma

# regression parameters
beta = matrix(.6, 1, m)

# covariates
X = matrix(rnorm(n), ncol=1)

# Simulate response variables
Y = X %*% beta + rmvnorm(n, sigma = Sigma)

# Multivariate regression
fit = lm(Y ~ X)

# Correlation between residuals
C = cor(residuals(fit))

# Extract effect sizes and standard errors from model fit
df = lapply(coef(summary(fit)), function(a)
 data.frame(beta = a["X", 1], se = a["X", 2]))
df = do.call(rbind, df)

# Run fixed effects meta-analysis,
```

```
# assume identity correlation
LS( df$beta, df$se)

# Run fixed effects meta-analysis,
# account for correlation
LS( df$beta, df$se, C)
```

`LS.empirical` *Fixed effect meta-analysis for correlated test statistics*

Description

Fixed effect meta-analysis for correlated test statistics using the Lin-Sullivan method using Monte Carlo draws from the null distribution to compute the p-value.

Usage

```
LS.empirical(
  beta,
  stders,
  cor = diag(1, length(beta)),
  nu,
  n.mc.samples = 10000,
  seed = 1
)
```

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
`nu`	degrees of freedom
`n.mc.samples`	number of Monte Carlo samples
`seed`	random seed so results are reproducable

Details

The theoretical null for the Lin-Sullivan statistic for fixed effects meta-analysis is chisq when the regression coefficients are estimated from a large sample size. But for finite sample size, this null distribution is not well characterized. In this case, we are not aware of a closed from cumulative distribution function. Instead we draw covariance matrices from a Wishart distribution, sample coefficients from a multivariate normal with this covariance, and then compute the Lin-Sullivan statistic. A gamma distribution is then fit to these draws from the null distribution and a p-value is computed from the cumulative distribution function of this gamma.

See Also

LS()

Examples

```
library(clusterGeneration)
library(mvtnorm)

# sample size
n = 30

# number of response variables
m = 6

# Error covariance
Sigma = genPositiveDefMat(m)$Sigma

# regression parameters
beta = matrix(.6, 1, m)

# covariates
X = matrix(rnorm(n), ncol=1)

# Simulate response variables
Y = X %*% beta + rmvnorm(n, sigma = Sigma)

# Multivariate regression
fit = lm(Y ~ X)

# Correlation between residuals
C = cor(residuals(fit))

# Extract effect sizes and standard errors from model fit
df = lapply(coef(summary(fit)), function(a)
data.frame(beta = a["X", 1], se = a["X", 2]))
df = do.call(rbind, df)

# Meta-analysis assuming infinite sample size
# but the p-value is anti-conservative
LS(df$beta, df$se, C)

# Meta-analysis explicitly modeling the finite sample size
# Gives properly calibrated p-values
# nu is the residual degrees of freedom from the model fit
LS.empirical(df$beta, df$se, C, nu=n-2)
```

---

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

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

```
library(clusterGeneration)
library(mvtnorm)

# sample size
n = 30

# number of response variables
m = 6

# Error covariance
Sigma = genPositiveDefMat(m)$Sigma

# regression parameters
beta = matrix(.6, 1, m)

# covariates
X = matrix(rnorm(n), ncol=1)

# Simulate response variables
Y = X %*% beta + rmvnorm(n, sigma = Sigma)

# Multivariate regression
fit = lm(Y ~ X)

# Correlation between residuals
C = cor(residuals(fit))

# Extract effect sizes and standard errors from model fit
df = lapply(coef(summary(fit)), function(a)
 data.frame(beta = a["X", 1], se = a["X", 2]))
df = do.call(rbind, df)

# Run fixed effects meta-analysis,
# assume identity correlation
LS( df$beta, df$se)

# Run random effects meta-analysis,
# assume identity correlation
```

```
RE2C( df$beta, df$se)

# Run fixed effects meta-analysis,
# account for correlation
LS( df$beta, df$se, C)

# Run random effects meta-analysis,
# account for correlation
RE2C( df$beta, df$se, C)
```

Index