Package 'multiplexGT'

February 20, 2022

Title Estimating the Prevalence of Multiple Infections

Version 1.0.0

Author Md S. Warasi

Maintainer Md S. Warasi <msarker@radford.edu>

Description Provides estimation techniques to estimate the prevalence of multiple infections from pooled testing data.

License GPL-3

NeedsCompilation yes

Roxygen list(markdown = TRUE)

Encoding UTF-8

RoxygenNote 7.1.1

R topics documented:

`multH2Cov` *Estimating Fisher information matrix for the MLE of multinomial parameter vector*

Description

This function implements the Louis's (1982) method to calculate the Fisher information matrix for the multinomial parameter p=(p00, p10, p01) based on two-stage hierarchical pooling data with multiple infections.

Usage

```
multH2Cov(p.true, gtData, Se, Sp, nburn = 1000, ngit = 3000)
```

Arguments

p.true	A value of the parameter (p00,p10,p01,p11) at which the covariance matrix is to be calculated.
gtData	A matrix or data.frame consisting of the pooled test outcomes and other information from a group testing application. Needs to be specified as shown in the example below.
Se	A vector of sensitivities, Se = c(Se1, Se2).
Sp	A vector of specificities, Sp = c(Sp1, Sp2).
nburn	The number of initial Gibbs iterates to be discarded.
ngit	The number of Gibbs iterates to be used in the E-step after discarding the initial iterates as a burn-in period.

Details

For information about the Fisher information matrix calculation, refer to Tebbs et al. (2013).

Value

Fisher information matrix.

Examples

```
library(multiplexGT)

set.seed(23)
p.t <- c(0.80, 0.10, 0.09, 0.01)
N <- 200
design <- c(4, 1)
Se.t <- c(0.95, 0.95)
Sp.t <- c(0.98, 0.98)
h2Data <- multH2Data(p=p.t, N=N, design=design, Se=Se.t, Sp=Sp.t)$gtData
FI <- multH2Cov(p.true=p.t,gtData=h2Data[ ,-(4:7)],Se=Se.t,Sp=Sp.t)
FI
```

---

multH2Data *Simulating two-stage hierarchical pooled testing data*

Description

This function generates pooled testing data from two-stage hierarchical protocol with multiple infections.

Usage

```
multH2Data(p, N, design, Se, Sp)
```

Arguments

p	The vector of coinfection probabilities, (p00,p10,p01,p11).
N	Sample size.
design	A vector of pool sizes, i.e., design = c(k, 1), where k is the initial pool size.
Se	The vector of sensitivities, i.e., Se = c(Se1, Se2).
Sp	The vector of specificities, i.e., Sp = c(Sp1, Sp2).

Details

This function is created to present the prevalence of multiple infections based from two-stage hierarchical testing protocol.

Value

A list with components:

gtData	Simulated data from two-stage hierarchical testing.
testExp	The number of tests expended.

Examples

```
set.seed(23)
p.t <- c(0.80, 0.10, 0.09, 0.01)
N <- 200
design <- c(4, 1)
Se.t <- c(0.95, 0.95)
Sp.t <- c(0.98, 0.98)
res <- multH2Data(p=p.t, N, design=design, Se=Se.t, Sp=Sp.t)

h2Data <- res$gtData
head( h2Data )
tail( h2Data )

T <- res$testExp
T
```

---

multH2MLE *EM Algorithm to Estimate the Prevalence of Multiple Infections from Pooled Testing Data*

Description

This function implements an expectation-maximization (EM) algorithm to find the maximum likelihood estimate (MLE) of the prevalence of multiple infections based on pooled responses. We take the estimation framework provided in Tebbs et al. (2013).

Usage

```
multH2MLE(
  p0 = rep(0.25, 4),
  gtData,
  Se,
  Sp,
  covariance = FALSE,
  nburn = 1000,
  ngit = 3000,
  maxit = 200,
  tol = 10^(-3),
  tracing = TRUE,
  conf.level = 0.95
)
```

Arguments

p0	An initial value of the vector of coinfection probabilities, (p00,p10,p01,p11).
gtData	A matrix or data.frame consisting of the pooled test outcomes and other information from a group testing application. Needs to be specified as shown in the example below.
Se	A vector of sensitivities, Se = c(Se1, Se2).
Sp	A vector of specificities, Sp = c(Sp1, Sp2).
covariance	When TRUE, the variance is calculated at the MLE.
nburn	The number of initial Gibbs iterates to be discarded.
ngit	The number of Gibbs iterates to be used in the E-step after discarding the initial iterates as a burn-in period.
maxit	The maximum number of EM steps (iterations) allowed in the EM algorithm.
tol	Convergence tolerance used in the EM algorithm.
tracing	When TRUE, progress in the EM algorithm is displayed.
conf.level	Confidence level to be used for the Wald confidence interval.

Details

More details about the EM algorithm and data structure will be provided later. Note that the Fisher information matrix is calculated by an appeal to the missing data principle and the method outlined in Louis (1982).

Value

A list with components:

param	The MLE of p=(p00,p10,p01).
covariance	Estimated variance of the MLE.
iterUsed	The number of EM iterations used for convergence.
convergence	0 if the EM algorithm converges successfully and 1 if the iteration limit maxit has been reached.
summary	Estimation summary with Wald confidence interval.
Fisher.info	Observed Fisher information matrix.

Examples

```
library(multiplexGT)

set.seed(23)
p.t <- c(0.80, 0.10, 0.09, 0.01)
N <- 200
design <- c(4, 1)
Se.t <- c(0.95, 0.95)
Sp.t <- c(0.98, 0.98)
h2Data <- multH2Data(p=p.t, N=N, design=design, Se=Se.t, Sp=Sp.t)$gtData

param0 <- rep(.25,4)
res <- multH2MLE(p0=param0,gtData=h2Data[ ,-(4:7)],Se=Se.t,Sp=Sp.t,covariance=TRUE)
res
```

`multIndData` *Simulating individual testing data with multiple infections*

Description

This function generates individual test responses with multiple infections, such as chlamydia and gonorrhea.

Usage

```
multIndData(p, N, Se, Sp)
```

Arguments

`p`	The vector of coinfection probabilities, (p00,p10,p01,p11).
`N`	Sample size.
`Se`	The vector of sensitivities, i.e., Se = c(Se1, Se2).
`Sp`	The vector of specificities, i.e., Sp = c(Sp1, Sp2).

Details

This function is created mainly to compare pooled testing estimates with individual testing estimates.

Value

A list component, which consists of a matrix object of individual responses.

Examples

```
set.seed(23)
p.t <- c(0.80, 0.10, 0.09, 0.01)
N <- 200
Se.t <- c(0.95, 0.95)
Sp.t <- c(0.98, 0.98)
indData <- multIndData(p=p.t, N, Se=Se.t, Sp=Sp.t)
head(indData)
tail(indData)
```

`multMptCov` *Estimating the Expected Covariance matrix for the MLE of Multinomial Parameter Vector*

Description

This function calculates the expected covariance matrix for the multinomial parameter p=(p00, p10, p01) from initial pooled responses with multiple infections.

Usage

```
multMptCov(p.true, m, k, Se, Sp)
```

Arguments

`p.true`	A value of the parameter (p00,p10,p01,p11) at which the covariance matrix is to be calculated.
`m`	Number of pools.
`k`	Pool size.
`Se`	A vector of sensitivities, Se = c(Se1, Se2).
`Sp`	A vector of specificities, Sp = c(Sp1, Sp2).

Details

Covariance matrix based on the expected Fisher information matrix using the method in Li et al. (2017).

Value

Covariance matrix.

Examples

```
p.t <- c(0.80, 0.10, 0.09, 0.01)
npools <- 50
psz <- 4
Se.t <- c(0.95, 0.95)
Sp.t <- c(0.98, 0.98)
Cov <- multMptCov(p.true=p.t, m=npools, k=psz, Se=Se.t, Sp=Sp.t)
Cov
```

`multMptData` *Simulating initial pooled testing data*

Description

This function generates pooled responses with multiple infections.

Usage

```
multMptData(p, N, k, Se, Sp)
```

Arguments

`p`	The vector of coinfection probabilities, (p00,p10,p01,p11).
`N`	Sample size.
`k`	Initial pool size.
`Se`	The vector of sensitivities, i.e., Se = c(Se1, Se2).
`Sp`	The vector of specificities, i.e., Sp = c(Sp1, Sp2).

Details

This function is created to present the prevalence of multiple infections based from two-stage hierarchical testing protocol.

Value

A list with components:

`gtData`	Simulated data from two-stage hierarchical testing.
`testExp`	The number of tests expended.

Examples

```
set.seed(23)
p.t <- c(0.80, 0.10, 0.09, 0.01)
N <- 200
pool.size <- 4
Se.t <- c(0.95, 0.95)
Sp.t <- c(0.98, 0.98)
res <- multMptData(p=p.t, N, k=pool.size, Se=Se.t, Sp=Sp.t)

mptData <- res$gtData
head( mptData )
tail( mptData )

T <- res$testExp
T
```

`multMptMLE` *Estimating the Prevalence of Multiple Infections from Pooled Responses*

Description

This function provides maximum likelihood estimates of the vector of coinfection probabilities (p00,p10,p01,p11) from initial pooled responses.

Usage

```
multMptMLE(gtData, Se, Sp, covariance = FALSE, conf.level = 0.95)
```

Arguments

`gtData`	A matrix or data.frame consisting of the pooled test outcomes and other information from a group testing application. Needs to be specified as shown in the example below.
`Se`	A vector of sensitivities, Se = c(Se1, Se2).
`Sp`	A vector of specificities, Sp = c(Sp1, Sp2).
`covariance`	When TRUE, the variance is calculated at the MLE.
`conf.level`	Confidence level to be used for the Wald confidence interval.

Details

The estimates are calculated based on the method presented in Li et al. (2017).

Value

A list with components:

`param`	The MLE of p=(p00,p10,p01).
`covariance`	Estimated variance of the MLE.
`summary`	Estimation summary with Wald confidence interval.
`Fisher.info`	Observed Fisher information matrix.

Examples

```
library(multiplexGT)

set.seed(23)
p.t <- c(0.80, 0.10, 0.09, 0.01)
N <- 200
pool.size <- 4
Se.t <- c(0.95, 0.95)
Sp.t <- c(0.98, 0.98)
mptData <- multMptData(p=p.t, N, k=pool.size, Se=Se.t, Sp=Sp.t)$gtData
res <- multMptMLE(gtData=mptData[ ,1:3],Se=Se.t,Sp=Sp.t,covariance=TRUE,conf.level=0.95)
res
```

Index