Package ‘genepi’

August 31, 2023

Type Package

Title Genetic Epidemiology Design and Inference

Version 1.0.3

Depends R (>= 2.0.0), stats

Author Venkatraman E. Seshan

Maintainer Venkatraman E. Seshan <seshanv@mskcc.org>

Description

Package for Genetic Epidemiologic Methods Developed at MSKCC. It contains functions to calculate haplotype specific odds ratio and the power of two stage design for GWAS studies.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-08-31 16:30:02 UTC

R topics documented:

`genepi-package` *Functions for some genetic epidemiology methods.*

Description

These functions provide code for genetic epidemiology methods developed at MSKCC. They currently include estimating haplotype disease risk and two stage designs for GWAS.

Details

Package:	genepi
Type:	Package
Version:	1.0.3
Date:	2023-08-31
License:	GPL version 2 or later
LazyLoad:	yes

There are two functions `haplotypeOddsRatio` and `twoStagePower` in this package.

Package will be archived and functions added to clinfun package

Author(s)

Venkatraman E. Seshan

Maintainer: Venkatraman E. Seshan <seshanv@mskcc.org>

References

Satagopan JM, Venkatraman ES, Begg CB. (2004) Two-stage designs for gene-disease association studies with sample size constraints. *Biometrics*

Venkatraman ES, Mitra N, Begg CB. (2004) A method for evaluating the impact of individual haplotypes on disease incidence in molecular epidemiology studies. *Stat Appl Genet Mol Biol.* v3:Article27.

`haplotypeOddsRatio` *Calculate haplotype disease risk.*

Description

Haplotype disease risk is calculated resolving haplotype ambiguity and adjusting for covariates and population stratification.

Usage

```
haplotypeOddsRatio(formula, gtypevar, data, stratvar=NULL, nsim=100, tol=1e-8)
## S3 method for class 'haploOR'
print(x, ...)
```

Arguments

`formula`	The formula for logistic regression without the haplotype variable.
`gtypevar`	The variable names in the data frame corresponding to the loci of interest. Each variables counts the number of mutant genotypes a subject has at that locus. Legal values are 0, 1, 2 & NA.
`data`	The name of the dataframe being analyzed. It should have all the variables in the formula as well as those in genotype and stratvar.

`stratvar` Name of the stratification variable. This is used to account for population stratification. The haplotype frequencies are estimated within each stratum.

`nsim` Variance should be inflated to account for inferred ambiguous haplotypes. The estimates are recalculated by simulating the disease haplotype copy number and variance added to average.

`tol` Tolerance limit for the EM algorithm convergence.

`x` Object of class haploOR.

`...` Other print options.

Details

This implements the method in the reference below.

Value

It is a list of class haploOR

`call` The function call that produced this output.

`coef` Table with estimated coefficients, standard error, Z-statistic and p-value.

`var` Covariance matrix of the estimated log odds-ratiios.

`deviance` Average of the simulated deviances. Its theoretical properties are unknown.

`aic` Average of the simulated aic.

`null.deviance` Deviance of null model.

`df.null` Degrees of freedom of null model.

`df.residual` Degrees of freedom of full model.

The "print" method formats the results into a user-friendly table.

Author(s)

Venkatraman E. Seshan

References

Venkatraman ES, Mitra N, Begg CB. (2004) A method for evaluating the impact of individual haplotypes on disease incidence in molecular epidemiology studies. *Stat Appl Genet Mol Biol.* v3:Article27.

Examples

```
# simulated data with 2 loci haplotypes 1=00, 2=01, 3=10, 4=11
# control haplotype probabilities p[i]  i=1,2,3,4
# haplotype pairs (i<=j) i=j: probs = p[i]^2 ; i<j: p[i]*p[j]
p <- c(0.25, 0.2, 0.2, 0.35)
p0 <- rep(0, 10)
l <- 0
for(i in 1:4) {for(j in i:4) {l <- l+1; p0[l] <- 2*p[i]*p[j]/(1+1*(i==j))}}
controls <- as.numeric(cut(runif(1000), cumsum(c(0,p0)), labels=1:10))
```

```
# case probabilities disease haplotype is 11
or <- c(2, 5)
p1 <- p0*c(1,1,1,2,1,1,2,1,2,8); p1 <- p1/sum(p1)
cases <- as.numeric(cut(runif(1000), cumsum(c(0,p1)), labels=1:10))
# now pool them together and set up the data frame
dat <- data.frame(status=rep(0:1, c(1000, 1000)))
# number of copies of mutant variant for locus 1
dat$gtype1 <- c(0,0,1,1,0,1,1,2,2,2)[c(controls, cases)]
# number of copies of mutant variant for locus 2
dat$gtype2 <- c(0,1,0,1,2,1,2,0,1,2)[c(controls, cases)]
# true number of copies of disease haplotype
dat$hcn <- c(0,0,0,1,0,0,1,0,1,2)[c(controls, cases)]
# model with genotypes only
haplotypeOddsRatio(status ~ 1, c("gtype1","gtype2"), dat)
# model from the logistic fit using the number of copies of disease haplotype
glm(status ~ factor(hcn), dat, family=binomial)
```

twoStagePower *Calculate the power of a two stage design for GWAS*

Description

Calculate the power of a two stage design for GWAS under sample size or cost constraints. Implements methods in the refereces below.

Usage

```
twoStagePower(n=NULL, Cost=NULL, m=5000, mu=0.045, mu.loc=0.5, p=0.10,
              f=NULL, relcost=1, true.needed=1, rho=0, rho0=0, nsim=2000)
```

Arguments

`n`	The maximum sample size for the study.
`Cost`	Maximum available resource. One of `Cost` or `n` must be specified.
`m`	The number of markers in the study. Default is 5000. It will take a a long time to compute power for very large numbers e.g. 100000
`mu`	The mean vector for the markers that are associated with endpoint.
`mu.loc`	The locations of the true markers. Since the chromosome is mapped to the unit interval (0,1) the numbers should be between 0 and 1.
`p`	The proportion of markers taken to the second stage. The default is 0.1 which is found to be optimal.
`f`	The fraction of Cost or sample size allocated to the first stage. If not specified it uses 0.75 for the Cost constraint scenario and 0.5 for the sample size contraint scenario.
`relcost`	Specifies how expensive it is to genotype in the second stage compared to the first stage.

`true.needed` The number of markers selected in the end. Can be a maximum of length of mu.loc (or mu).

`rho, rho0` correlation between markers

`nsim` Number of Monte Carlo replications to compute power.

Details

This implements the method in the reference below.

Value

It returns the power as a single numeric value

Author(s)

Jaya M. Satagopan & Venkatraman E. Seshan

References

Satagopan JM, Venkatraman ES, Begg CB. (2004) Two-stage designs for gene-disease association studies with sample size constraints. *Biometrics*

Examples

```
twoStagePower(n=1000)
twoStagePower(Cost=1000)
```

Index