

Package ‘IFP’

February 14, 2012

Version 0.0.3

Date 2010-2-18

Title Identifying functional polymorphisms in genetic association studies

Author Leeyoung Park

Maintainer Leeyoung Park <lypark@yonsei.edu>

Depends R (>= 2.7.0), haplo.stats

Suggests

Description A suite for identifying causal polymorphisms in case-control genetic association data, especially with large controls re-sequenced data.

License GPL (>= 2)

URL <http://www.r-project.org>

Repository CRAN

Date/Publication 2010-03-17 15:49:28

R topics documented:

allele.freq	2
apoe	2
error.rates	3
hap.freq	4
lrt	5
Index	7

 allele.freq

Allele Frequency Computation from Genotype Data

Description

Computes allele frequencies from genotype data.

Usage

```
allele.freq(geno)
```

Arguments

geno matrix of alleles, such that each locus has a pair of adjacent columns of alleles, and the order of columns corresponds to the order of loci on a chromosome. If there are K loci, then $\text{ncol}(\text{geno}) = 2 * K$. Rows represent the alleles for each subject. Each allele should be represented as numbers (A=1,C=2,G=3,T=4).

Value

array of allele frequencies of each SNP. The computed allele is targeted as an order of alleles, "A", "C", "G", and "T".

Examples

```
data(apoe)
allele.freq(apoe7)
allele.freq(apoe)
```

 apoe

Genetic data of APOE gene region

Description

This data set came from a re-sequenced data of APOE gene region in the Molecular Diversity and Epidemiology of Common Disease (MDECODE) database. Sixteen polymorphic sites were included. "apoe7" data contains the genetic data of seven single nucleotide polymorphisms with allele frequencies higher than 0.1 from the apoe data.

Usage

```
apoe
```

Format

A matrix with 48 rows and 32 columns

Source

<http://droog.gs.washington.edu/mdecode/>

References

Nickerson, D. A., S. L. Taylor, S. M. Fullerton, K. M. Weiss, A. G. Clark et al. (2000) Sequence diversity and large-scale typing of SNPs in the human apolipoprotein E gene. *Genome Res* 10: 1532-1545.

error.rates	<i>Error Rates Estimation for Likelihood Ratio Tests Designed for Identifying Number of Functional Polymorphisms</i>
-------------	--

Description

Compute error rates for a given model.

Usage

```
error.rates(H0,Z, pMc, geno, no.ca, no.con=nrow(geno), sim.no = 1000)
```

Arguments

H0	the index number for a given model for functional SNPs
Z	number of functional SNPs for the given model
pMc	array of allele frequencies of case samples
geno	matrix of alleles, such that each locus has a pair of adjacent columns of alleles, and the order of columns corresponds to the order of loci on a chromosome. If there are K loci, then ncol(geno) = 2*K. Rows represent the alleles for each subject. Each allele should be represented as numbers (A=1,C=2,G=3,T=4).
no.ca	number of case chromosomes
no.con	number of control chromosomes
sim.no	number of simulations for error rates estimation

Value

array of results consisted of Type I error rate (alpha=0.05), Type I error rate (alpha=0.01), Type II error rate (beta=0.05), Type II error rate (beta=0.01), percent when the target model has the lowest corrected -2 log likelihood ratio.

See Also

allele.freq hap.freq lrtB

Examples

```
## LRT tests when SNP1 & SNP6 are the functional polymorphisms.

library(haplo.stats)
library(IFP)
data(apoe)

n<-c(2000, 2000, 2000, 2000, 2000, 2000, 2000) #case sample size = 1000
x<-c(1707, 281,1341, 435, 772, 416, 1797) #allele numbers in case samples

Z<-2 #number of functional SNPs for tests
n.poly<-ncol(apoe7)/2 #total number of SNPs

#index number for the model in this case is 5 for SNP1 and 6. apoe7 is considered to represent the true control allele

error.rates(5, 2, x/n, apoe7, 2000, 2000, sim.no=100)
```

hap.freq

Estimation of Haplotype Frequencies with Two SNPs

Description

EM computation of haplotype frequencies with two SNPs. The computation is relied on the package "haplo.stats".

Usage

```
hap.freq(geno)
```

Arguments

geno matrix of alleles, such that each locus has a pair of adjacent columns of alleles, and the order of columns corresponds to the order of loci on a chromosome. If there are K loci, then $\text{ncol}(\text{geno}) = 2 * K$. Rows represent the alleles for each subject. Each allele should be represented as numbers (A=1,C=2,G=3,T=4).

Value

matrix of haplotype frequencies consisted of two alleles from each SNP. These alleles are the same ones computed for frequency using the function "allele.freq".

See Also

```
allele.freq
```

Examples

```
data(apoe)
hap.freq(apoe7)
hap.freq(apoe)
```

lrt	<i>Likelihood Ratio Tests for Identifying Number of Functional Polymorphisms</i>
-----	--

Description

Compute p-values and likelihoods of all possible models for a given number of functional SNP(s).

Usage

```
lrt(n.fp, n, x, geno, no.con=nrow(geno))
```

Arguments

n.fp	number of functional SNPs for tests.
n	array of each total number of case sample chromosomes for SNPs
x	array of each total allele number in case samples
geno	matrix of alleles, such that each locus has a pair of adjacent columns of alleles, and the order of columns corresponds to the order of loci on a chromosome. If there are K loci, then $\text{ncol}(\text{geno}) = 2 * K$. Rows represent the alleles for each subject. Each allele should be represented as numbers (A=1,C=2,G=3,T=4).
no.con	number of control chromosomes.

Value

matrix of likelihood ratio test results. First n.fp rows indicate the model for each set of disease polymorphisms, and followed by p-values, $-2 \log(\text{likelihood ratio})$ with corrections for variances, maximum likelihood ratio estimates, and likelihood.

See Also

allele.freq hap.freq

Examples

```
## LRT tests when SNP1 & SNP6 are the functional polymorphisms.

library(haplo.stats)
library(IFP)
data(apoe)

n<-c(2000, 2000, 2000, 2000, 2000, 2000, 2000) #case sample size = 1000
```

```
x<-c(1707, 281,1341, 435, 772, 416, 1797) #allele numbers in case samples

Z<-2 #number of functional SNPs for tests
n.poly<-ncol(apoe7)/2 #total number of SNPs

#control sample generation( sample size = 1000 )
con.samp<-sample(nrow(apoe7),1000,replace=TRUE)
con.data<-array()
for (i in con.samp){
con.data<-rbind(con.data,apoe7[i,])
}
con.data<-con.data[2:1001,]

lrt(1,n,x,con.data)
lrt(2,n,x,con.data)
```

Index

*Topic **datasets**

apoe, [2](#)

allele.freq, [2](#)

apoe, [2](#)

apoe7 (apoe), [2](#)

error.rates, [3](#)

hap.freq, [4](#)

lrt, [5](#)