Package 'mixIndependR'

October 13, 2022

Type Package

Title Genetics and Independence Testing of Mixed Genetic Panels

Version 1.0.0

Author Bing Song

Maintainer Bing Song <bing.song@utsouthwestern.edu>

Depends R (>= 3.6.0)

Imports stats (>= 3.3), utils(>= 3.6.1), data.table

Description Developed to deal with multi-locus genotype data, this package is especially designed for those panel which include different type of markers. Basic genetic parameters like allele frequency, genotype frequency, heterozygosity and Hardy-Weinberg test of mixed genetic data can be obtained. In addition, a new test for mutual independence which is compatible for mixed genetic data is developed in this package.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests testthat, knitr, rmarkdown, ggplot2

NeedsCompilation no

VignetteBuilder knitr

URL https://github.com/ice4prince/mixIndependR

Repository CRAN

Date/Publication 2021-03-17 09:30:05 UTC

R topics documented:

`AlleleFreq` *Calculate Allele Frequency*

Description

Calculate Allele Frequency

Usage

```
AlleleFreq(x,sep)
```

Arguments

`x`	a dataset of genotypes. Each row denotes each individual; each column contain each marker.
`sep`	the allele separator in the imported genotype data.Note: when using the special character like "|", remember to protect it as "|"(default).

Details

This function calculates the allele frequencies of one dataset.

Value

a matrix of allele frequencies. Each row denotes each allele; each column denotes each marker. The order of makers follows x.

Examples

```
require(mixIndependR)
x <- data.frame(STR1=c("12|12","13|14","13|13","14|15"),
                SNP1=c("A|A","T|T","A|T","A|T"))
AlleleFreq(x,"\\|")
```

`AlleleShare` *Calculate numbers of sharing alleles each pair at each locus*

Description

Calculate numbers of sharing alleles each pair at each locus

Usage

```
AlleleShare(df,sep,replacement=FALSE)
```

Arguments

`df`	a dataframe of genotype data with rownames of sample ID and column names of markers.
`sep`	allele separator in the imported genotype data. Note: when using the special character like "|", remember to protect it as "\\|"(default).
`replacement`	a logical variable. If it is TRUE, the pairs are sampled with replacement; if FALSE (default), the pairs are sampled without replacement.

Details

This function calculates the numbers of shared alleles between each pair of individuals for a dataset.

Value

a dataframe of numbers of shared alleles. Each row denotes each pair; Each column denotes each locus.

Examples

```
df <- data.frame(SNP1=c("A|A","T|T","A|T","A|T"),
                 STR1=c("12|12","13|14","13|13","14|15"))
AlleleShare(df,"\\|",replacement=FALSE)
```

ComposPare_K *Generate Comparison Observed and Expected No. of Heterozygous Loci.*

Description

Generate Comparison Observed and Expected No. of Heterozygous Loci.

Usage

```
ComposPare_K(h,Ex,trans)
```

Arguments

h	a double made up of "0" and "1" where 1 means heterozygous and 0 means homozygous; Outcome of function "Heterozygous"; Each column denotes each locus and each row denotes each individual.
Ex	a dataframe of expected density, outcome of function "DistHetero", on each possible total number of heterozygous loci.
trans	a logic variable, if True, the outcome is a dataframe of n x 2. n is the number of individuals of original imported database. First column is the observed No. of Heterozygous Loci and the second is the expected one. If False, the dataframe is 2n x 2, where n is the number of individuals of original imported database. The first column is a categorical variable denoting the frequency is observed or expected value; the second column is the frequency of No. of heterozygous loci.

Details

This function generates a dataframe in which the observed and expected heterozygous loci for each sample are included. The observed ones are calculated from the original dataset. However, the expected ones are simulated according to the expected probability with the same sample size as observed sample.

Value

a dataframe of observed and expected No. of heterozygous loci for each individual.

Examples

```
h<-matrix(rbinom(20,1,0.5),nrow=5)
Ex <- data.frame(K=c(0:5),Density=rnorm(6,mean = 0.5,sd=0.05))
ComposPare_K(h,Ex,trans = TRUE)
```

ComposPare_X *Generate Comparison Observed and Expected No. of Shared Alleles.*

Description

Generate Comparison Observed and Expected No. of Shared Alleles.

Usage

```
ComposPare_X(AS,Ex,trans=TRUE)
```

Arguments

AS	a double made up of "0","1" and "2" denoting number of shared alleles; Outcome of function "AlleleShare_Table"; Each column denotes each locus and each row denotes each pair of individuals.
Ex	a dataframe of expected density, outcome of function "DistAlleleShare", on each possible total number of shared Alleles.
trans	a logic variable, if True, the outcome is a dataframe of n x 2. n is the number of individuals of original imported database. First column is the observed No. of Heterozygous Loci and the second is the expected one. If False, the dataframe is 2n x 2, where n is the number of individuals of original imported database. The first column is a categorical variable denoting the frequency is observed or expected value; the second column is the frequency of No. of heterozygous loci.

Details

This function generates a dataframe in which the observed and expected shared alleles for each pair of individuals. The observed ones are calculated from the original dataset through "AlleleShare_Table". However, the expected ones are simulated according to the expected probability with the same sample size as the observed sample.

Value

a dataframe of observed and expected No. of shared alleles for each pair of individuals.

Examples

```
AS<-matrix(sample(c(0:2),20,replace=TRUE,prob=c(0.3,0.3,0.4)),nrow=5)
Ex <- data.frame(X=c(0:8),Density=rnorm(9,mean = 0.5,sd=0.05))
ComposPare_X(AS,Ex,trans = TRUE)
```

counta *Simple count including zero###*

Description

Simple count including zero###

Usage

```
counta(z, y)
```

Arguments

z	a vector you would like to check
y	an element you would like to count.(Even it is not included in z)

Details

This function counts how many the assigned elements there are in one vector.

Value

the times that y appears in z

Examples

```
z <-rbinom(20,1,0.5)
counta(z,0)
```

DistAlleleShare *Build Expected Distribution of Numbers of Shared Alleles*

Description

Build Expected Distribution of Numbers of Shared Alleles

Usage

```
DistAlleleShare(e)
```

Arguments

e	a matrix/dataframe of probability of shared alleles; outcome of "ExpProAlleleShare" or "RealProAlleleShare". Each row denotes each locus. The first column is the case of 0 shared alleles, the second column is the case of 1 shared alleles, the third column is the case of 2 shared alleles.

Details

This function build the expected distribution of numbers of shared alleles for known shared alleles of each pair of individuals.

Value

a dataframe of probabilities of each number of shared alleles(from 0 to 2*loci); the first column is No. of Shared Alleles; the Second Column is Expected Density

References

Chakraborty, R., Stivers, D. N., Su, B., Zhong, Y., & Budowle, B. (1999) <doi:10.1002/(SICI)1522-2683(19990101)20:8<1682::AID-ELPS1682>3.0.CO;2-Z>

Examples

```
e0<-data.frame("P0"=runif(5,min = 0,max = 0.5),"P1"=runif(5,0,0.5))
e<-data.frame(e0,"P2"=1-rowSums(e0))
DistAlleleShare(e)
```

`DistHetero` *Build Expected Distribution of Numbers of Heterozygous Loci*

Description

Build Expected Distribution of Numbers of Heterozygous Loci

Usage

```
DistHetero(H)
```

Arguments

`H`	a vector of average heterozygosity of each locus

Details

This function build the expected distribution of numbers of heterozygous loci for known heterozygosity of each loci.

Value

a dataframe of expected density on each possible total number of heterozygous loci.

References

Chakraborty, R. (1981, ISSN:0016-6731)

Examples

```
DistHetero(runif(10))
```

`Dist_SimuChisq` *Build a simulated distribution for Chi-Square*

Description

Build a simulated distribution for Chi-Square

Usage

```
Dist_SimuChisq(s,prob,b)
```

Arguments

`s`	a matrix of frequencies for each simulated sample. Each row for each sample.
`prob`	a vector of expected probability for each simulated sample.
`b`	the times of bootstrapping.

Details

This function build the distribution of Chi square statistics for simulated samples

Value

a vector of Chi-square statistics, length is the times of sampling.

Examples

```
require(mixIndependR)
h<-runif(10)
s<-Simulate_DistK(h,500,100)
Exp <- DistHetero(h)
Dist_SimuChisq(s,Exp$Density,10)
```

ExpProAlleleShare *Calculate the Expected Probability of 0,1 and 2 Shared Alleles###*

Description

Calculate the Expected Probability of 0,1 and 2 Shared Alleles###

Usage

```
ExpProAlleleShare(p)
```

Arguments

p	a matrix/double of frequency of alleles; Outcome of "AlleleFreq". Each column denotes each locus. Different alleles is ordered in different rows such as 11,11.3,12,12.2,13... and so on

Details

This function Calculates the Expected Probability of 0,1 and 2 Shared Alleles for a set of loci. Usually followed by write.csv(as.data.frame(y),file = "~/*.csv") to export the result of a n x3 matrix.

Value

a matrix/double of expected probabilities of 0,1 and 2 shared alleles for each locus. Each row denotes each locus. The first column denotes the probability of 0 shared alleles, the second denotes 1 shared allele, the third denotes 2 shared alleles.

References

Weir, B. S. (2004, ISSN:0022-1198)

Examples

```
a0<-matrix(runif(20),nrow=5)
a1<-colSums(a0)
a<-data.frame(STR1=a0[,1]/a1[1],STR2=a0[,2]/a1[2],STR3=a0[,3]/a1[3],STR4=a0[,4]/a1[4])
ExpProAlleleShare(a)
```

FreqAlleleShare *Build Observed Distribution of No. of Shared Alleles*

Description

Build Observed Distribution of No. of Shared Alleles

Usage

```
FreqAlleleShare(AS)
```

Arguments

AS	a matrix of number of shared alleles, made up with 0, 1 and 2, outcome of function "AlleleShare_Table". Rows for individuals, and columns for markers.

Details

This function build the observed distributions from observed Allele Share table, made up of 0,1 and 2.

Value

a dataframe of frequencies of each number of shared alleles(from 0 to 2*N0. of loci)

Examples

```
AS<-matrix(sample(c(0:2),20,replace=TRUE,prob=c(0.3,0.3,0.4)),nrow=5)
FreqAlleleShare(AS)
```

FreqHetero *Build Observed Distribution of No. of Heterozygous loci*

Description

Build Observed Distribution of No. of Heterozygous loci

Usage

```
FreqHetero(h)
```

Arguments

h	a dataframe of heterozygosity, made up with 0 and 1, outcome of function "Heterozygous" Rows for individuals, and columns for markers.

Details

This function build the observed distributions from observed heterozygosity table, made up of 0,1.

Value

a dataframe of frequencies of each number of heterozygous loci(from 0 to No. of loci)

Examples

```
h<-matrix(rbinom(20,1,0.5),nrow=5)
FreqHetero(h)
```

GenotypeFreq *Calculate Genotype Frequency###*

Description

Calculate Genotype Frequency###

Usage

```
GenotypeFreq(x,sep,expect=TRUE)
```

Arguments

x	a dataframe of genotype data with rownames of sample ID and column names of markers.
sep	allele separator in the imported genotype data. Note: when using the special character like "|", remember to protect it as "|"(default).
expect	a logic variable. If expect is true, the function will calculate the expected genotype probabilities. If false, calculate the observed genotype frequencies.

Details

This function calculates the observed or expected genotype frequency from dataset and allele frequency.#####

Value

a dataframe of genotype frequencies. Each row denotes each genotype; each column denotes each loci. The order of markers follows x; the genotypes are ordered from homozygous to heterozygous.

References

Chakraborty, R., Srinivasan, M. R., & Daiger, S. P. (1993, ISSN:0002-9297).

Examples

```
require(mixIndependR)
x <- data.frame(SNP1=c("A|A","T|T","A|T","A|T"),
                STR1=c("12|12","13|14","13|13","14|15"))
GenotypeFreq(x,"\\|",expect=TRUE)
```

Heterozygous *Test heterozygosity at each locus*

Description

Test heterozygosity at each locus

Usage

```
Heterozygous(x,sep)
```

Arguments

x	a dataset of genotypes with rownames of sample ID and column names of markers.
sep	allele separator in the imported genotype data. Note: when using the special character like "|", remember to protect it as "|"(default).

Details

This function test the heterozygosity of each individuals at each locus.Output a table and Usually followed by write.csv(as.data.frame(y),file = "~/*.csv") to export the results.

Value

a dataframe of heterozygosity.0 is homozygous;1 is heterozygous. Each row denotes each individual; Each column denotes each locus.

Examples

```
x <- data.frame(STR1=c("12|12","13|14","13|13","14|15"),
                SNP1=c("A|A","T|T","A|T","A|T"))
Heterozygous(x,"\\|")
```

HWE.Chisq *Test the Hardy Weinberg Equilibrium with Chi-square test####*

Description

Test the Hardy Weinberg Equilibrium with Chi-square test####

Usage

```
HWE.Chisq(G,G0,rescale.p=FALSE,simulate.p.value=TRUE,B=2000)
```

Arguments

G	a dataframe of observed genotype frequencies. Each row denotes each genotype; each column denotes each marker. The order of markers follows x; the genotypes are ordered by: from 1:l-th column, the genotypes are homozygous in order as : p1p1, p2p2,p3p3,...,plpl;from ll-th to u-th column, the genotypes are heterozygous in order as:choose(l,2) like: p1p2,p1p3,...,p1pl,p2p3,p2p4,...p2pl,...p(l-1)pl
G0	a dataframe of expected genotype probabilities;each row denotes each genotype; each column denotes each loci. The order of markers follows x; the genotypes are ordered by: from 1:l-th column, the genotypes are homozygous in order as : p1p1, p2p2,p3p3,...,plpl;from ll-th to u-th column, the genotypes are heterozygous in order as:choose(l,2) like: p1p2,p1p3,...,p1pl,p2p3,p2p4,...p2pl,...p(l-1)pl
rescale.p	a logical scalar; if TRUE then p is rescaled (if necessary) to sum to 1. If rescale.p is FALSE, and p does not sum to 1, an error is given.
simulate.p.value	a logical indicating whether to compute p-values by Monte Carlo simulation.
B	an integer specifying the number of replicates used in the Monte Carlo test.

Details

This function check the Hardy Weinberg Equilibrium from observed and expected distribution with Chi-square test#####

Value

a vector of result of p-values for chi-square test; the orders of markers follows x.

Examples

```
require(mixIndependR)
x <- data.frame(STR1=c("11|12","12|13","11|13","13|15"),
                STR2=c("12|12","13|14","13|13","14|15"),
                SNP1=c("A|T","A|A","T|A","A|T"),
                SNP2=c("A|A","T|T","A|T","T|A"))
G <- GenotypeFreq(x,expect = FALSE)
```

```
G0 <- GenotypeFreq(x,expect = TRUE)
HWE.Chisq(G,G0,rescale.p=FALSE,simulate.p.value=TRUE,B=2000)
```

`mixexample` *Genotype Data from A Selected Mix Panel*

Description

This dataset is the phased genotypes for a mix panel with 100 variants. These variants are selected from the reference haplotype data of Gymrek's lab (see Reference). This is a sample with 2504 individuals.

Usage

```
data(mixexample)
```

Format

A dataframe with 2504 observations on 100 variables. This dataframe is the phased genotype files for 100 variants (including SNPs and STRs) for 2504 individuals.

Source

1000 Genomes SNP-STR Haplotype Panel <https://gymreklab.com/2018/03/05/snpstr_imputation.html>
The genotypes of panel after selection <https://github.com/ice4prince/mixIndependR/tree/main/data>

References

Saini et al. (2018). A reference haplotype panel for genome-wide imputation of short tandem repeats. Nat Commun 9(1): 4397. <https://pubmed.ncbi.nlm.nih.gov/30353011/>

Examples

```
data(mixexample)
```

`mixIndependK` *Quick pvalue of total number of heterozygous loci*

Description

Quick pvalue of total number of heterozygous loci

Usage

```
mixIndependK(x,sep,t,B)
```

Arguments

x	a dataset of alleles. Each row denotes each individual.One allele in one cell.In the (2r-1)th column, there is the same locus with the 2r-th column; noted: no column for ID, make row.names=1 when importing.
sep	allele separator in the imported genotype data. Note: when using the special character like "|", remember to protect it as "|".
t	times of simulation in "Simulate_DistK" and "Simulate_DistX".
B	times of bootstrapping in Chi Squares Test.

Details

This function is a summary of pipeline for number of heterozygous loci (K), and generates the p-value of K for the target dataset.

Value

pvalue (1-cumulative probabilities) for the number of heterozygous loci(K)

Examples

```
x <- data.frame(SNP1=c("A|A","T|T","A|T","A|T"),
                STR1=c("12|12","13|14","13|13","14|15"))
mixIndependK(x,sep ="\\|",10,10)
```

---

mixIndependX *Quick pvalue of total number of shared alleles*

Description

Quick pvalue of total number of shared alleles

Usage

```
mixIndependX(x,sep,t,B)
```

Arguments

x	a dataset of alleles. Each row denotes each individual.One allele in one cell.In the (2r-1)th column, there is the same locus with the 2r-th column; noted: no column for ID, make row.names=1 when importing.
sep	allele separator in the imported genotype data. Note: when using the special character like "|", remember to protect it as "|".
t	times of simulation in "Simulate_DistK" and "Simulate_DistX".
B	times of bootstrapping in Chi Squares Test.

Details

This function is a summary of pipeline for number of shared alleles(X), and generates the p-value of K for the target dataset.

Value

pvalue (1-cumulative probabilities) for the number of shared alleles(K)

Examples

```
x <- data.frame(SNP1=c("A|A","T|T","A|T","A|T"),
                STR1=c("12|12","13|14","13|13","14|15"))
mixIndependX(x,sep="\\|",10,10)
```

`read_vcf_gt` *Import genotype data from vcf files/*

Description

Import genotype data from vcf files/

Usage

```
read_vcf_gt(x)
```

Arguments

`x`	The vcf file with its directory

Details

This function extract the genotypes and allele status from a vcf file.

Value

a list contains the genotype and allele status.

Examples

```
## Not run:
df<-read_vcf_gt("~/x.vcf")

## End(Not run)
```

RealProAlleleShare *Calculate the Real Probability of 0,1 and 2 Shared Alleles###*

Description

Calculate the Real Probability of 0,1 and 2 Shared Alleles###

Usage

```
RealProAlleleShare(AS)
```

Arguments

AS	a matrix/double of no. of Shared alleles, made up with 0,1 and 2; Outcome of "AlleleShare_Table". Each column denotes each locus. Each row denotes each individual.

Details

This function Calculates the density of 0,1 and 2 Shared Alleles for a set of loci. Usually followed by write.csv(as.data.frame(y),file = "~/*.csv") to export the result of a n x3 matrix.

Value

a matrix/double of real density of 0,1 and 2 shared alleles for each locus. Each row denotes each locus. The first column denotes the probability of 0 shared alleles, the second denotes 1 shared allele, the third denotes 2 shared alleles.

Examples

```
AS<-matrix(sample(c(0:2),20,replace=TRUE,prob=c(0.3,0.3,0.4)),nrow=5)
RealProAlleleShare(AS)
```

RxpHetero *Calculate Real or Expected Average Heterozygosity at each locus*

Description

Calculate Real or Expected Average Heterozygosity at each locus

Usage

```
RxpHetero(h,p,HWE)
```

Arguments

h	a dataset of heterozygosity, made up with 0 and 1. Output of function "Heterozygous". Each row denotes each individual. Each row denotes each locus.
p	a dataset of allele frequency, Output of function "AlleleFreq". Each row denotes each allele, and each column denotes each locus.
HWE	a logic variable. When TRUE, this function will calculate the expected heterozygosity under Hardy-Weinberg Equilibrium: H= 1-sum(q_i^2); q_i is the allele frequency; If FALSE, this function calculate the average heterozygosity from real heterozygosity table.

Details

This function calculate average heterozygosity at each locus.Output a vector of number of loci.

Value

a vector of average heterozygosity on each loci.

References

Chakraborty, R., & Jin, L. (1992, ISSN:1432-1203) <doi:10.1007/BF00197257>

Examples

```
x <- data.frame(STR1=c(12,13,13,14,15,13,14,12,14,15),
                STR1_1=c(12,14,13,15,13,14,13,12,14,15),
                SNP1=c("A","T","A","A","T","A","A","T","T","A"),
                SNP1_1=c("A","T","T","T","A","T","A","A","T","T"))
require(mixIndependR)
h <- Heterozygous(x)
p <- AlleleFreq(x)
RxpHetero(h,p,HWE=TRUE)
```

Simulate_DistK — *Generate a Bundle of Simulated distributions for No. of heterozygous loci with known heterozygosites*

Description

Generate a Bundle of Simulated distributions for No. of heterozygous loci with known heterozygosites

Usage

```
Simulate_DistK(H,m,t)
```

Arguments

H	a vector of average heterozygosity of each loci. Length of H is the number of loci.
m	the sample size you want, usually similar to the real sample size.
t	the number of samples you want to build

Details

This function generates multinomial distribution for loci known the heterozygosity and build the simulated distribution for no. of heterozygous loci.

Value

a matrix of frequencies of No. of Heterozygous Loci. Each row denotes each simulated sample; Each column denotes each No. of Heterozygous loci, from 0 to length of H.

Examples

```
Simulate_DistK(runif(10),500,100)
```

Simulate_DistX *Build a simulated distribution for No. of Shared Alleles*

Description

Build a simulated distribution for No. of Shared Alleles

Usage

```
Simulate_DistX(e,m,t)
```

Arguments

e	a matrix of Probability of Sharing 2,1 or 0 alleles at each loci. Each row denotes each locus. Three columns denote sharing 0,1 or 2 alleles.
m	the sample size you want, usually similar to the real sample size.
t	the number of samples you want to build/ the times to generate a sample

Details

This function generates multinomial distribution for loci known the Allele Frequency and Expected Probability of Shared 2,1 or 0 alleles

Value

a matrix of frequencies of No. of shared alleles. Each row denotes each simulated sample; Each column denotes each No. of shared alleles, from 0 to 2e length of e.

Examples

```
e0<-data.frame("P0"=runif(5,min = 0,max = 0.5),"P1"=runif(5,0,0.5))
e<-data.frame(e0,"P2"=1-rowSums(e0))
Simulate_DistX(e,500,10)
```

`splitGenotype` *Split Genotype Table to Duo-Allele Table*

Description

Split each column to two columns for a table of genotypes

Usage

```
splitGenotype(df,sep,dif,rowbind)
```

Arguments

`df`	a dataframe of genotype data with rownames of sample ID and column names of markers.
`sep`	allele separator in the imported genotype data. Note: when using the special character like "|", remember to protect it as "|"(default).
`dif`	a symbol differentiate the one marker on each allele.
`rowbind`	a logical variable. If rowbind is TRUE, the output is arranged with double rows but the same columns, and the table of the second allele is followed after the first allele table by rows with double individual IDs in the same order. If rowbind is false,the output is arranged by double columns and the same rows; the column names are in the order of alphabet by pairs.

Details

The function convert a genotype data to allele data with double columns or with double rows; the rownames are sample ID in the same order but twice if the rows are doubled, and the column names are in the same order or in the order of alphabet by pairs if columns are doubled.
The parameter "sep" is the symbol of allele separator in the imported genotype data.
The parameter "dif" is the difference between the second and the first appearance for the same marker. For example, if "dif = _1", the column names of output will be "marker1" "marker1 _1","marker2","marker2 _1", if the original list of column names is "marker1","marker2".

Value

a dataframe with doubled columns of import data and alleles in different columns

Examples

```
## Not run:
df <- data.frame(SNP1=c("A|A","T|T","A|T","A|T"),
                 STR1=c("12|12","13|14","13|13","14|15"))
splitGenotype(df)

## End(Not run)
```

Index