Package 'HaploSim'

February 19, 2015

Type Package

Title Functions to simulate haplotypes

Version 1.8.4

Date 2012-12-15

Depends methods,utils, R (>= 2.6.0)

Suggests pedigree (>= 1.3.1)

Description Simulate haplotypes through meioses. Allows specification of population parameters.

License GPL (>= 2)

Author Albart Coster [aut, cre], John Bastiaansen [aut]

Maintainer Albart Coster <albart@dairyconsult.nl>

NeedsCompilation no

Repository CRAN

Date/Publication 2013-11-13 11:57:09

R topics documented:

AssignQTL *Assigns QTL to a list of Haplotypes*

Description

AssignQTL assigns QTL to the qtl of objects of class haplotype. ListQTL returns a list of qtl which for the object of class haploList.

Usage

```
AssignQTL(hList,nqtl = NA,frqtl = NA,sigma2qtl = NULL,shQTL = 1,scQTL = 1,
                      nTraits = 1,overlap = 0,MAF = 0.1,rmCausSNP =
TRUE)
ListQTL(hList,nqtl = NA,frqtl = NA,sigma2qtl = NULL,shQTL = 1,scQTL = 1,
                      nTraits = 1,overlap = 0,MAF = 0.1)
```

Arguments

hList	List of haplotype objects.
nqtl	If specified, the number of qtl which are placed on the genome.
frqtl	If specified, the fraction of heterozygous SNP loci which become QTL.
sigma2qtl	If specified, the qtl variance. If length(sigma2qtl)<nqtl, sigma2qtl is replicated until obtaining a vector of appropriate length. Can be specified as a list of length equal to nTraits.
shQTL	If alpha is not specified, shQTL specifies the shape parameter of the gamma distribution from which allele substitution effects are sampled.
scQTL	If alpha is not specified, scQTL specifies the scale parameter of the gamma distribution from which allele substitution effects are sampled.
nTraits	The number of traits.
overlap	Numeric between 0 and 1. Specifies the percantage of pleiotropic QTL.
MAF	Minor Allele Frequency. Loci with maf below MAF are not considered to become QTL. Do not count for frqtl.
rmCausSNP	Remove causative SNPs.

Value

A list of length nHaplotpes.

See Also

SampleHaplotype

Examples

```
hList <- SampleHaplotypes(nHaplotypes = 20,nLoc = 100,genDist =
1,nDec = 3) ## create objects
 hListd <- SampleHaplotypes(orig = hList,genDist = 1,nDec = 3)
hListQTL <- AssignQTL(hList,frqtl = 0.1,MAF = 0.0)
hListd <- SampleHaplotypes(orig = hListQTL,genDist = 1,nDec
= 3,QTL = TRUE)
qtlList <- ListQTL(hList,frqtl = 0.1,MAF = 0.0)
```

---

buildhPedigree *Build a haplotype pedrigree from a haplotype list*

Description

Builds a haplotype pedigree from a list of objects of class haplotype. Objects have attributes hID and phID0, phID1, the last two refer to the two haplotypes in the parent. Function can construct a new pedigree or continue a given pedigree. Simulate a list of haplotypes, either sampling using population parameters or from a previous list of haplotypes through a series of meioses.

Usage

```
buildhPedigree(hPedigree = NULL,hList)
```

Arguments

hPedigree	If unspecified, the previous haplotype pedigree upon which buildhPed builds the additional haplotypes.
hList	A list with objects of class haplotype.

Details

The function buildhPedigree uses the attributes hID, phID0, phID1 of objects of class haplotype to build a pedigree of haplotypes. Each haplotype originates from a pair of parental haplotypes betwee occured or has no known parental haplotypes.

Value

A data.frame.

See Also

SampleHaplotypes

Examples

```
example(SampleHaplotypes)
hPedigree <- buildhPedigree(hList = hList)
for(g in 1:10)
  {
    hList <- SampleHaplotypes(orig = hList,genDist
        = 1,roundDec = 3)
    hPedigree <- buildhPedigree(hPedigree=hPedigree,hList = hList)
  }
```

`getAll` *Get alleles*

Description

Extract the sequence of snp alleles from a list of objects of class `haplotype`.

Usage

```
getAll(hList,what = c("snp","qtl"),removeHomozygotes =
FALSE,translatePos = TRUE)
```

Arguments

`hList`	List of objects of class `haplotype`.
`what`	Specifies if snp or qtl alleles are extracted from haplotype objects.
`removeHomozygotes`	If TRUE, homozygote genotypes are removed from the marker data.
`translatePos`	Translates marker positions to positions in Morgan, else keeps the positions as integers.

Details

Function removes homozygous snp loci before extracting these.

Value

Matrix.

Examples

```
example(AssignQTL)
hh <- getAll(hList = hList)
qq <- getAll(hList = hList,what = 'qtl')
```

haploList *Function to create objects of class 'haploList'*

Description

Function to create objects of class 'haploList'. Can eiter copy the attributes of an earlier object of class 'haploList' or create a new 'haploList' object. Function mostly used within other functions.

Usage

```
haploList(list = NULL,hList = NULL,nDec,genDist,nChrom = 1)
```

Arguments

list	List of objects of class 'haplotype'.
hList	Object of class 'haploList'.
nDec	Number of decimal positions of new object.
genDist	Genome size of new object measured in Morgan.
nChrom	Number of chromosomes.

Value

An object of class 'haploList'. If list is not NULL, object of filled with objects of class 'haplotype'.

See Also

haplotype, haploList

Examples

```
hList <- haploList(nDec=  1,genDist = 1)
validhaploListObject(hList)
```

haploList-class *Class "haploList"*

Description

Class definition of haploList. Extends lists to contian objects of class haplotype. Attributes are nDec, the number of decimal positions and genDist, the genome size measured in Morgan.

Objects from the Class

Objects can be created by calls of the form new("haploList", ...).

Slots

.Data: Object of class "list" containing objects of class 'haplotype'

genDist: Object of class "numeric" expressing the genome size in Morgan.

nDec: Object of class "integer" expressing the number of decimal positions of the haplotypes.

nChrom: Object of class "integer" expressing the number of chromosomes.

Extends

Class "list", from data part. Class "vector", by class "list", distance 2.

Methods

[signature(x = "haploList", i = "ANY", j = "missing"): subset and extract from object of class 'haploList'

c signature(x = "haploList"): concatenate object of class 'haploList'.

See Also

SampleHaplotypes, haplotype

Examples

```
showClass("haploList")
```

haplotype-class *Class "haplotype" ~~~*

Description

Definition of 'haplotype' class.

Objects from the Class

Objects can be created by calls of the form new("haplotype", ...).

Slots

snp: Object of class "integer" containing the positions on the genome where the haplotype has a 1 allele. The number of snp positions is the genome size in Morgan times the number of decimal positions, both stored in 'haploList' objects.

qtl: Object of class "list". Names of the list are the QTL positions, on the same scale as snp positions. Objects in the list are the QTL alleles.

hID: Object of class "numeric" identifying the current object of class 'haplotype'.

phID0: Object of class "numeric" pointing to the first parental haplotype.

phID1: Object of class "numeric" pointing to the second parental haplotype.

Methods

No methods defined with class "haplotype" in the signature.

See Also

SampleHaplotype, SampleHaplotypes, haploList

Examples

```
showClass("haplotype")
```

hPed2Ped *Transform a haplotype pedigree into a pedigree.*

Description

Transforms a haplotype pedigree into a pedigree. Individuals in a haplotype pedigree are identified through meiosis. The number of rows of the pedigree equals the number of unique combinations of haplotypes plus the number of haplotypes which did not 'participate' in a meiosis event. The latter haplotypes form individuals with only one haplotype and thus only one parent. Row number of the pedigree identifies individuals. The pedigree has four columns. Columns 3 and 4 identify the haplotypes of the individual. Columns 1 and 2 identify parental individuals of the individuals by their row number in the pedigree.

Usage

```
hPed2Ped(hPed)
```

Arguments

hPed	The haplotype pedigree from which the pedigree is build.

Value

A data.frame.

See Also

SampleHaplotype, buildhPedigree

Examples

```
example(buildhPedigree)
ped <- hPed2Ped(hPedigree)
```

RemoveHomozygotes *Remove the homozygotic snp loci from a list of haplotypes*

Description

Finds homozygotic marker loci in a list of haplotypes and removes these from all haplotypes.

Usage

```
RemoveHomozygotes(hList)
```

Arguments

hList	List of objects of class haplotype

Value

A list of objects of class haplotype.

See Also

SampleHaplotypes, SampleHaplotype.

Examples

```
hList <- SampleHaplotypes(nHaplotypes = 20,nLoc = 100,genDist =
1,nDec = 3) ## create objects
hList <- RemoveHomozygotes(hList)
```

SampleBaseHaplotype *Simulate a base population haplotype*

Description

Simulates a base population haplotype.

Usage

```
SampleBaseHaplotype(genDist,nDec,nLoc,pSnp = seq(0,1,length.out = nLoc))
```

Arguments

genDist	Map size of the simulated genome in Morgan.
nDec	Number of decimals until which marker positions are rounded. Partially specifies marker density on the chromosome.
nLoc	Maximal number of snp loci on genome, default is number of available positions on genome.
pSnp	Optional,if specified the snp positions. Default: uniform distribution of loci over the whole genome. Consequence will be that loci have allele frequency equal to one. If NULL, positions will be randomly sampled for each haplotype and allele frequencies will be close to 0, depending on $nLoc/(genDist * 10^{nDec})$.

Details

Generally called by function SampleHaplotypes.

Value

An object of class haplotype.

See Also

SampleHaplotypes, SampleBaseHaplotype

Examples

```
hList <- sapply(1:10,function(x)SampleBaseHaplotype(genDist = 1,nDec
= 2,nLoc = 50))
```

SampleHaplotype *Simulate a haplotype*

Description

Simulates a haplotype, either sampling using population parameters or through a meiosis event with two parental haplotypes.

Usage

```
SampleHaplotype(H0 = NULL,H1 = NULL,genDist,nDec,nChrom = 1,prMut =
1E-5,QTL = F,checkValidity = TRUE)
```

Arguments

H0	If specified, the first parental haplotype.
H1	If specified, the second parental haplotype. If neither H0 nor H1 are specified, a new haplotype is sampled from a base population. Errors message is displayed when only one haplotype is provided as meiosis occurs between two haplotypes.
genDist	Map size of the simulated genome in Morgan.
nDec	Number of decimals until which marker positions are rounded. Partially specifies marker density on the chromosome.
nChrom	Number of chromosomes, default at 1
prMut	Probability of marker bp mutation.
QTL	If TRUE, qtl alleles are inherited to the next generation. See function AssignQTL for assigning qtl to a list of haplotypes.
checkValidity	If TRUE, tests if a pair of haplotypes is compatible; e.g. if the number of traits in both is equal (or 0) and if the sizes are equal.

Details

Markers are continually spaced over the whole genome. Marker density is specified in SampleBaseHaplotype. Position of 1 alleles is recorded and stored in @snp attribute of the object of class haplotype. If QTL is TRUE, haplotypes with QTL's need to be provided. If not, nothing happens apart from mutations (same prob. as for single basepairs). Function SampleHaplotype is generally called by function SampleHaplotypes.

Value

An object of class haplotype.

See Also

SampleHaplotypes, SampleBaseHaplotype

Examples

```
hList <- SampleHaplotypes(nHaplotypes = 20,genDist =
1,nDec = 3,nLoc = 20) ## create objects
h <- SampleHaplotype(H0 = hList[[1]],H1 = hList[[2]],genDist =
1,nDec = 3)
```

SampleHaplotypes *Simulate a list of haplotypes*

Description

Simulate a list of haplotypes, either sampling using population parameters or from a previous list of haplotypes through a series of meioses.

Usage

```
SampleHaplotypes(orig = NULL,nHaplotypes = 10,nMeioses = 2,gg = NULL,...)
```

Arguments

orig	If unspecified, the function samples base population haplotypes. If specified, the function requests a list containing objects of class haplotype.
nHaplotypes	The requested number of haplotypes.
nMeioses	The number of offspring from each individual. For details see below.
gg	If specified, the combinations of haplotypes in individuals. Meiosis only occur within individuals. If not specified, individuals are sampled as random combinations of haplotypes.
...	Additional arguments to be passed to function SampleHaplotype, haploList and SampleBaseHaplotype.

Details

The function SampleHaplotypes creates individuals by randomly combining haplotypes from the list. Meiosis events in individuals create new haplotypes. Argument nOff sets the number of meiosis events within each individual. Argument nHaplotypes has only effect when sampling a base population. See SampleBaseHaplotype for sampling base haplotypes and for arguments of this function.

Value

A list of length nHaplotpes or nOff * length(orig) of objects of class haplotype.

See Also

SampleHaplotype, SampleBaseHaplotype

Examples

```
hList <- SampleHaplotypes(nHaplotypes = 20,nLoc = 100,genDist =
1,nDec = 3) ## create objects
for(g in 1:10)hList <- SampleHaplotypes(orig = hList,genDist
= 1,nDec = 3)
```

SamplePedigree *Simulate a haplotypes in a pedigree*

Description

Simulates haplotypes within a given pedigree. Haplotypes for a base individual are sampled from a list of base haplotypes. Parameters for sampling haplotypes are passed to function SampleHaplotype.

Usage

```
SamplePedigree(orig,ped,...)
```

Arguments

orig	List of objects of class haplotype. Haplotypes for base individuals are the result of a meiosis event between two haplotypes in this list.
ped	data.frame of three columns. Column 1 contains id's, column two and three id's of parental individuals. Pedigree is first ordered with function orderPed from the package pedigree.
...	Arguments to be passed to function SampleHaplotype.

Details

Samples haplotypes for individuals in a pedigree and returns a pedigree with two additional columns which identify the two haplotypes of an individual together with a list of haplotypes. Uses function SampleHaplotype to sample a meiosis event between two parental haplotypes.

Value

A list with a pedigree and a list of objects of class haplotype.

See Also

SampleHaplotype, orderPed

Examples

```
example(SampleHaplotypes)
ID <- 1:10
pID0 <- c(rep(0,5),1,1,3,3,5)
pID1 <- c(rep(0,4),2,2,2,4,4,6)
ped <- data.frame(ID,pID0,pID1)
phList <- SamplePedigree(orig = hList,ped = ped)
```

validhaploListObject *Function to validate an object of class 'haploList'*

Description

Checks of object is of class `'haploList'` and checks if all entries in `object@.Data` are of class `'haplotype'`.

Usage

```
validhaploListObject(object)
```

Arguments

`object`	Any object.

Value

Logical

See Also

`haploList`, `haplotype`

Index