# Package ‘HAPim’

February 19, 2015

**Type** Package  
**Title** HapIM  
**Version** 1.3  
**Date** 2009-10-10  
**Author** S. Dejean, N. Oumouhou, D. Estivals, B. Mangin  
**Maintainer** Brigitte Mangin <brigitte.mangin@toulouse.inra.fr>  

**Description** The package provides a set of functions whose aim is to propose 4 methods of QTL detection. HAPimLD is an interval-mapping method designed for unrelated individuals with no family information that makes use of linkage disequilibrium. HAPimLDL is an interval-mapping method for design of half-sib families. It combines linkage analysis and linkage disequilibrium. HaploMax is based on an analysis of variance with a dose haplotype effect. HaploMaxHS is based on an analysis of variance with a sire effect and a dose haplotype effect in half-sib family design. Fundings for the package development were provided to the LDLmapQTL project by the ANR GENANIMAL program and APIS-GENE.

**License** GPL  
**Repository** CRAN  
**Date/Publication** 2012-10-29 08:57:07  
**NeedsCompilation** no

**R topics documented:**

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAPim-package</td>
<td>2</td>
</tr>
<tr>
<td>allele.marq</td>
<td>4</td>
</tr>
<tr>
<td>corresp</td>
<td>5</td>
</tr>
<tr>
<td>data.test</td>
<td>6</td>
</tr>
<tr>
<td>depart.LD</td>
<td>7</td>
</tr>
<tr>
<td>depart.LDL</td>
<td>8</td>
</tr>
<tr>
<td>descendant.pere</td>
<td>9</td>
</tr>
</tbody>
</table>
The package provides a set of functions whose aim is to propose 4 methods of QTL detection:

- **HAPimLD** is an interval-mapping method designed for unrelated individuals with no family information that makes use of linkage disequilibrium.
- **HAPimLDL** is an interval-mapping method for design of half-sib families. It combines linkage analysis and linkage disequilibrium.
- **HaploMax** is based on an analysis of variance with a dose haplotype effect.
- **HaploMaxHS** is based on an analysis of variance with a sire effect and a dose haplotype effect in half-sib family design.

Fundings for the package development were provided to the LDLmapQTL project by the ANR GENANIMAL program and APIS-GENE.
Details
**Description**

This function finds the alleles present at each marker. It can be viewed as an internal function. The user does not have to call it by himself.

**Usage**

```r
allele.marq(hap)
```

**Arguments**

- `hap` character matrix (n x p) of individual haplotypes.

**Value**

The returned value is a list of alleles for each marker.

**Author(s)**

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

**References**

The function can be viewed as an internal function. The user does not have to call it by himself.

Usage

```
corresp(hap.o, res.structure)
```

Arguments

- `hap.o`: numeric matrix (n x p).
- `res.structure`: results provided by `structure.hap()` function, list of numeric objects.

Value

A list containing the following components:

- `corresp`: numeric matrix (n x p).
- `assoc`: numeric vector, associated haplotype

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


See Also

`structure.hap`
**Description**

"data.test" is a list of 9 elements. We consider a 10 cM chromosomal region with 10 equally spaced biallelic markers and a design of 5 half-sib families, each sire having 10 sons. Data are prepared for a QTL detection in the middle of each marker intervalle.

**Usage**

```r
data(data.test)
```

**Format**

A list containing the following components:
- **map**: vector (9) with numerical values
- **hap.trans.mere**: matrix (50 x 10) with character values
- **hap.trans.pere**: matrix (50 x 10) with character values
- **hap.chrom1.pere**: matrix (5 x 10) with character values
- **hap.chrom2.pere**: matrix (5 x 10) with character values
- **perf**: vector (50) with numerical values
- **cd**: vector (50) with numerical values
- **pla**: matrix (50 x 9) with numerical values
- **genea**: matrix (50 x 2) with numerical values

**Examples**

```r
data(data.test)

#distance between two consecutive markers on the chromosome
map=data.test[[1]]
map

#haplotype transmitted by dams
#son information (lines) are ordered following genea[,1]
hap.trans.mere=data.test[[2]]
 hap.trans.mere

#haplotype transmitted by sires
#son information (lines) are ordered following genea[,1]
 hap.trans.pere=data.test[[3]]
 hap.trans.pere

#haplotype of the first chromosome for each sire
```
`depart.LD`

Starting values for the optimization of HAPimLD method

**Description**

The function calculates the starting values of the performance mean and the error variance for the optimization of HAPimLD method. It can be viewed as an internal function. The user does not have to call it by himself.

**Usage**

`depart.LD(perf, CD)`

**Arguments**

- `perf` numeric vector of length=number of individuals which contains the performances of individuals.
- `CD` numeric vector of length=number of individuals which contains the CD of individuals.
Value

The value returned is a numeric vector of length=2 which contains estimates of the performance mean and the error variance.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


depart.LDL

starting values for the optimization of HAPimLDL method

Description

The function calculates the starting value of the error variance and the starting value of the QTL effect for the optimization of HAPimLDL method. It can be viewed as an internal function. The user does not have to call it by himself.

Usage

depart.LDL(moyenne.pere, perf, CD, PLA, desc.pere)

Arguments

moyenne.pere results provided by moyenne.pere() function, mean of half-sib family performances.

perf numeric vector of length=number of individuals which contains the performances of individuals.

CD numeric vector of length=number of individuals which contains the CD of individuals. \( \text{var(perf}_i^2) = s / \text{CD}_i^2 \)

PLA numeric vector (number of individuals) which contains transmission probabilities at a single test position.

desc.pere results provided by descendant.pere() function, numeric matrix (number of sires x 2) which gives for each sire, the first and last indexes of its progeny.

Value

The returned value is a numeric vector of length=2 which contains estimates of the error variance and the Q allele effect.
The function defines, for each sire, the first and last indexes of its progeny. It can be viewed as an internal function. The user does not have to call it by himself.

**Usage**

descendant.pere(genea)

**Arguments**

genea numeric matrix (number of individuals x 2) which contains the progeny index and its father index.

**Details**

Progeny data are prealably ordered by family

**Value**

The returned value is a numeric matrix (number of sires x 2) which gives for each sire, the first and last indexes of its progeny.

**Author(s)**

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

**References**

distance.marqueurs  \hspace{1em} distance of markers

Description
The function gives the distance of each marker to the first marker. It can be viewed as an internal function. The user does not have to call it by himself.

Usage
distance.marqueurs(map)

Arguments
map \hspace{1em} numeric vector of length=(number of markers-1) giving the distance between two consecutive markers on all the chromosome.

Value
the returned value is a numeric vector of length=number of markers containing the positions of each marker from the beginning of the chromosome.

Author(s)
S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References

distance.test  \hspace{1em} localisation of test positions

Description
The function gives the distance of test positions from the first marker and sorts them out by marker interval. It can be viewed as an internal function. The user does not have to call it by himself.

Usage
distance.test(position, dist.marq)
esp.freq.hap

Arguments

  position       numeric vector of test positions.
  dist.marq      results provided by distance.marqueurs() function, numeric vector of length=number
                 of markers which gives the distance of each marker from the beginning of the
                 chromosome.

Value

  the returned value is a list of n numeric vectors (n=number of markers-1). The ȘjȘth vector contains
  the distance (from the first marker) of test positions belonging to the ȘjȘth marker interval.

Author(s)

  S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References

  publication to be submitted: C. Cierco-Ayrolles, S. Dejean, A. Legarra, H. Gilbert, T. Druet, F.
  Ytournel, D. Estivals, N. Oumouhou and B. Mangin. Combining linkage analysis and linkage
  disequilibrium for QTL fine mapping in animal pedigrees.

See Also

  distance.marqueurs

---

esp.freq.hap    expectation of extended haplotype frequencies

Description

  This function computes the expectation of (haplotype + Q allele) frequencies under a Wright-Fisher
  model. The function can be viewed as an internal function. The user does not have to call it by
  himself.

Usage

  esp.freq.hap(hap.assoc, piQ.t0, timeT, pi.hap, res.structure, poids.D)

Arguments

  hap.assoc     numeric value, associated haplotype.
  piQ.t0        frequency of Q allele at time t=0.
  timeT         time of population evolution.
  pi.hap        provided by pi.hap() function, list of numeric objects.
  res.structure provided by structure.hap() function, list of numeric objects.
  poids.D       provided by poids.D() function, list of numeric objects.
Value

The value returned is a numeric vector containing for each observed haplotype, the frequencies of its extension with the Q allele.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


See Also

pi.hap, structure.hap, poids.D

Description

The function calculates allelic frequencies by marker given a set of haplotypes. It can be viewed as an internal function. The user does not have to call it by himself.

Usage

freq.all(hap)

Arguments

hap numeric matrix (number of individuals x number of markers) which contains the haplotype of individuals.

Value

The returned value is a list of n elements (n=number of markers) which gives the allelic frequencies by marker, ranged in ascending order of allele index.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References

haldanem1

Description
The function calculates recombination rates using Haldane distance. It can be viewed as an internal function. The user does not have to call it by himself.

Usage
haldanem1(distance)

Arguments
distance vector of distances in Morgan.

Value
the returned value is a vector of recombination rates.

Author(s)
S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References
Haldane, J.B.S. J. Genet. (1919), 8:299-309.

HAPimLD add

Description
HAPimLD is a method of QTL(Quantitative Trait Loci) detection developed by Boitard et al. (2006). It is an interval-mapping method designed for unrelated individuals with no family information. It is based on a maximum-likelihood calculation and makes use of linkage disequilibrium through a Wright-Fisher modelisation of the population evolution.

Usage
hapim.LD.add(hap.trans.pere, hap.trans.mere, perf, CD, map, position,
temps.depart, perfectLD, marq.hap.left)
Arguments

**hap.trans.pere** character matrix (number of individuals x number of markers) which provides, for each individual, the haplotype transmitted by its father.

**hap.trans.mere** character matrix (number of individuals x number of markers) which provides, for each individual, the haplotype transmitted by its mother.

**perf** numeric vector of length=number of individuals which contains the performances of individuals.

**CD** numeric vector of length=number of individuals which contains the CD of individuals. \( \text{var(perf}_i\text{)}=\text{error variance/CD}_i^2 \)

**map** numeric vector of length=(number of markers-1) which contains the distance in Morgan between two consecutive markers on the chromosome.

**position** numeric vector which contains the distance in Morgan of test positions from the beginning of the chromosome (first marker).

**temps.depart** numeric value which provides a start value for the evolution time of the population.

**perfectLD** need to be equal to TRUE: linkage disequilibrium is complete between mutated haplotype and Q allele at time 0.

**marq.hap.left** (number of markers of the mutated haplotype)/2.

Details

Individual information have to be ranged in the same order in hap.trans.mere, hap.trans.pere, perf, CD.

All distances are assumed to be Haldane’s distance in Morgan.

Value

The returned value is a data frame which contains 8 columns:

- Test positions
- Value of Likelihood Ratio Test (LRT)
- Mutated (i.e. associated to Q allele) haplotype
- Estimate of the error variance
- Estimate of the Q allele effect
- Estimate of the time of population evolution
- Estimate of the Q allele frequency at time \( t=0 \)
- Estimate of the performance mean

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin
References


Examples

```r
data(data.test)
map=data.test[1]
hap.trans.mere=data.test[2]
hap.trans.pere=data.test[3]
perf=data.test[6]
CD=data.test[7]

# In this example, marker positions are: 0, 0.010, 0.020, 0.030, 0.040, 0.050, 0.060,
# 0.070, 0.080, 0.090.
# We want to test the presence/absence of a QTL between 2 consecutive markers, so
position=c(0.005,0.015,0.025,0.035,0.045,0.055,0.065,0.075,0.085)

# we use a 2 markers-associated haplotype.
marq.hap.left=1

# We assume an evolution of 50 generations.
temps.depart=50
perfectLD=TRUE

hapim.LD.add=hapim.LD.add(hap.trans.pere,hap.trans.mere,perf,CD,map,position,
temps.depart,perfectLD,marq.hap.left)

hapim.LD.add
```

HAPimLDL method

HAPimLDL is a method of QTL (Quantitative Trait Loci) detection for a design of half-sib families. It is an interval-mapping method which uses family information and combines linkage analysis and linkage disequilibrium. It is based on a maximum-likelihood calculation and makes use of linkage disequilibrium through a Wright-Fisher modelisation of the population evolution.
Usage

hapim.LDL.add(hap.chrom1.pere, hap.chrom2.pere, hap.trans.mere, perf, CD,

genea, PLA, map, position, temps.depart, perfectLD, marq.hap.left)

Arguments

hap.chrom1.pere
character matrix (number of sires x number of markers) which gives the haplotype of the first chromosome for each sire.

hap.chrom2.pere
character matrix (number of sires x number of markers) which gives the haplotype of the second chromosome for each sire.

hap.trans.mere
numeric matrix (number of individuals x number of markers) which provides, for each individual, the haplotype transmitted by its mother.

perf
numeric vector of length=number of individuals which contains the performances of individuals.

CD
numeric vector of length=number of individuals which contains the CD of individuals. \( \text{var}(\text{perf}_i) = \text{error variance}/\text{CD}_i^2 \)

genea
numeric matrix (number of individuals x 2) which contains the progeny index and its father index.

PLA
numeric matrix (number of individuals x number of test positions) which contains transmission probabilities i.e. probability that the progeny receives the first chromosome from its father at the test positions given marker information, see J.-M. Elsen, B. Mangin, B. Goffinet, D. Boichard, P. Le Roy. Alternative models for QTL detection in livestock. I. General introduction. Genet. Sel. Evol. 31 (1999) 213-224.

map
numeric vector of length=(number of markers-1) which contains the distance in Morgan between two consecutive markers on the chromosome.

position
numeric vector which contains the distance in Morgan of test positions from the beginning of the chromosome (first marker).

temps.depart
numeric value which provides a start value for the evolution time of the population.

perfectLD
need to be equal to TRUE: linkage disequilibrium is complete between mutated haplotype and Q allele at time 0.

marq.hap.left
(number of markers of the mutated haplotype)/2.

Details

Progeny information have to be ranged in the same order in genea, hap.trans.mere, perf, CD and PLA.

Columns of PLA have to correspond to test positions.

Sire information have to be ranged in the same order in unique(genea[,2]), hap.chrom1.pere and hap.chrom2.pere.

All distances are assumed to be Haldame’s distance in Morgan.
Value

The returned value is a data frame which contains 8 columns:
- Test positions
- Value of Likelihood Ratio Test (LRT)
- Mutated (i.e. associated to Q allele) haplotype
- Estimate of the error variance
- Estimate of the Q allele effect
- Estimate of the time of population evolution
- Estimate of the Q allele frequency at time t=0
- Estimate of the performance mean

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


Examples

data(data.test)
map=data.test[[1]]
hap.trans.mere=data.test[[2]]
hap.chrom1.pere=data.test[[3]]
hap.chrom2.pere=data.test[[4]]
perf=data.test[[5]]
CD=data.test[[6]]
PLA=data.test[[7]]
genea=data.test[[8]]

dataHdataNtestI
map\dataNtestI
dataNtestI
hap\transNmere\dataNtestI
hap\chromQNpere\dataNtestI
hap\chromRNpere\dataNtestI
perf\dataNtestI
CD\dataNtestI
PLA\dataNtestI
genea\dataNtestI

# In this example, marker positions are : (0, 0.010, 0.020, 0.030, 0.040, 0.050, 0.060, 0.070, 0.080, 0.090).
# We want to test the presence/absence of a QTL between 2 consecutive markers, so transmission probabilities are given for the middle of each interval and
position=c(0.005,0.015,0.025,0.035,0.045,0.055,0.065,0.075,0.085)

# we use a 2 markers-associated haplotype.
marq.hap.left=1

# We assume an evolution of 50 generations.
temps.depart=50
perfectLD=TRUE
hapim.LDL.add

hapim.LDL.add(hap.chrom1.pere, hap.chrom2.pere, hap.trans.mere, perf, CD, genea, PLA, map, position, temps.depart, perfectLD, marq.hap.left)

hapim.LDL.add

### haplomax.add

**HaploMax method in unrelated population**

#### Description

The function computes an analysis of variance with a dose haplotype effect.

#### Usage

haplomax.add(hap.trans.pere, hap.trans.mere, perf, CD, map, marq.hap)

#### Arguments

- **hap.trans.pere**: character matrix (number of individuals x number of markers) which provides, for each individual, the haplotype transmitted by its father.
- **hap.trans.mere**: character matrix (number of individuals x number of markers) which provides, for each individual, the haplotype transmitted by its mother.
- **perf**: numeric vector of length=number of individuals which contains the performances of individuals.
- **CD**: numeric vector of length=number of individuals which contains the CD of individuals. \(\text{var}(\text{perf}_i)=\text{error variance}/\text{CD}_i^2\)
- **map**: numeric vector of length=(number of markers-1) which contains the distance in Morgan between two consecutive markers on the chromosome.
- **marq.hap**: number of markers of the mutated haplotype

#### Details

Individual information have to be ranged in the same order in hap.trans.mere, hap.trans.pere, perf, CD.

All distances are assumed to be Haldane’s distance in Morgan.

Test positions are located on the middles of marq.hap marker sliding windows.
Value

The value returned is a data frame which contains 5 columns:
- Test positions
- Value of Fisher test
- Mutated (i.e. associated to Q allele) haplotype
- Estimate of the error variance
- Estimate of the Q allele effect

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


Examples

data(data.test)
map=data.test[[1]]
hap.trans.mere=data.test[[2]]
hap.trans.pere=data.test[[3]]
perf=data.test[[6]]
CD=data.test[[7]]

# In this example, marker positions are: (0, 0.010, 0.020, 0.030, 0.040, 0.050, 0.060,
# 0.070, 0.080, 0.090 ).  
# we use a 2 markers-associated haplotype.
marq.hap=2

haplomax=haplomax.add(hap.trans.pere,hap.trans.mere,perf,CD,map, marq.hap)

haplomax

HaploMax method in half-sib family design

Description

The function computes an analysis of variance with a sire effect and a dose haplotype effect.
Usage

haplomax.HS.add(hap.chrom1.pere, hap.chrom2.pere, hap.trans.pere,

hap.trans.mere, perf, CD, genea, map, marq.hap)

Arguments

hap.chrom1.pere
character matrix (number of sires x number of markers) which gives the haplotype of the first chromosome for each sire.

hap.chrom2.pere
character matrix (number of sires x number of markers) which gives the haplotype of the second chromosome for each sire.

hap.trans.pere
numeric matrix (number of individuals x number of markers) which provides, for each individual, the haplotype transmitted by its father.

hap.trans.mere
numeric matrix (number of individuals x number of markers) which provides, for each individual, the haplotype transmitted by its mother.

perf
numeric vector of length=number of individuals which contains the performances of individuals.

CD
numeric vector of length=number of individuals which contains the CD of individuals. \( \text{var}(\text{perf}_i) = \text{error variance/CD}_i^2 \)

genea
numeric matrix (number of individuals x 2) which contains the progeny index and its father index.

map
numeric vector of length=(number of markers-1) which contains the distance in Morgan between two consecutive markers on the chromosome.

marq.hap
number of markers of the mutated haplotype.

Details

Progeny information have to be ranged in the same order in genea, hap.trans.pere, hap.trans.mere, perf and CD.

Sire information have to be ranged in the same order in unique(genea[,2]), hap.chrom1.pere and hap.chrom2.pere.

All distances are assumed to be Haldane's distance in Morgan.

Test positions are located on the middles of marq.hap marker sliding windows.

Value

The returned value is a data frame which contains 5 columns:

- Test positions
- Value of Fisher test
- Mutated (i.e. associated to Q allele) haplotype
- Estimate of the error variance
- Estimate of the Q allele effect
Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


Examples

data(data.test)
map=data.test[[1]]
hap.trans.mere=data.test[[2]]
hap.trans.pere=data.test[[3]]
hap.chrom1.pere=data.test[[4]]
hap.chrom2.pere=data.test[[5]]
perf=data.test[[6]]
CD=data.test[[7]]
genea=data.test[[9]]

# In this example, marker positions are: 0, 0.010, 0.020, 0.030, 0.040, 0.050, 0.060, 0.070, 0.080, 0.090.
# we use a 2 markers-associated haplotype

marq.hap=2

haplomax.HS=haplomax.HS.add(hap.chrom1.pere,hap.chrom2.pere,hap.trans.pere,hap.trans.mere,
perf,CD,genea,map,marq.hap)

haplomax.HS

Description

The function computes the empirical performance mean per sire. It can be viewed as an internal function. The user does not have to call it by himself.

Usage

moyenne.pere(perf, CD, desc.pere)
Arguments

- **perf**: numeric vector of length=number of individuals which contains the performances of individuals.
- **CD**: numeric vector of length=number of individuals which contains the CD of individuals. \( \text{var(perf}_i\text{)=} \text{error variance}/\text{CD}_i^2 \)
- **desc.pere**: results provided by `descendant.pere()` function, numeric matrix (number of sires x 2) which gives for each sire, the first and last indexes of its progeny.

Value

The returned value is a vector of length=number of sires which contains the empirical performance mean per sire.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


See Also

`descendant.pere`

---

**obj.haploMax.add**  
Analyses of variance of the HaploMax method

Description

The function computes the regression analysis with a dose haplotype effect. It can be viewed as an internal function. The user does not have to call it by himself.

Usage

`obj.haploMax.add(perf, CD, assoc, res.structure, pi.hap, cor.pere, cor.mere)`

Arguments

- **perf**: numeric vector of length=number of individuals which contains the performances of individuals.
- **CD**: numeric vector of length=number of individuals which contains the CD of individuals. \( \text{var(perf}_i\text{)=} \text{error variance}/\text{CD}_i^2 \)
- **assoc**: numeric value, associated haplotype.
The function computes the regression analysis with a dose haplotype effect and a sire effect in a design of half-sib families. It can be viewed as an internal function. The user does not have to call it by himself.

Usage

```r
obj.haplo.max.HS.add(genea, perf, CD, assoc, res.structure, pi.hap, cor.pere,
                  cor.mere)
```

Arguments

- `genea`: numeric matrix (number of individuals x 2) which contains individual index and corresponding sire index of each individual.
- `perf`: numeric vector of length=number of individuals which contains the performances of individuals.
- `CD`: numeric vector of length=number of individuals which contains the CD of individuals. `var(perf[,i])=error variance/CD[i]^2`
assoc numeric value, associated haplotype
res.structure provided by structure.hap() function, list of objects.
pi.hap provided by pi.hap() function, list of numeric objects.
cor.pere provided by corresp() function, list of numeric objects.
cor.mere provided by corresp() function, list of numeric objects.

Value
The returned value is an object of aov class containing the dose haplotype + sire effect regression.

Author(s)
S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References

See Also
corresp, pi.hap, structure.hap

obj.LD.add log-likelihood value of HAPimLD method under H1

Description
The function calculates the log-likelihood value of HAPimLD method under alternative hypothesis H1. It can be viewed as an internal function. The user does not have to call it by himself.

Usage
obj.LD.add(param, don)

Arguments
param numeric vector of length=5 containing parameters to maximize:
-error variance
-Q allele effect
-evolution time
-frequency of Q allele at time t=0
-performances mean
don list of 8 objects. Some objects are results provided by corresp(), pi.hap(), structure.hap(), poids.D() functions.
Value

The returned value is the log-likelihood value of HAPimLD method under hypothesis H1.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


See Also

depart.LD, corresp, pi.hap, structure.hap, poids.D

| obj.LD.add.H0 | log-likelihood value of HAPimLD method under H0 |

Description

The function calculates the log-likelihood value of HAPimLD method under hypothesis H0. It can be viewed as an internal function. The user does not have to call it by himself.

Usage

obj.LD.add.H0(perf, CD)

Arguments

perf numeric vector of length = number of individuals which contains the performances of individuals.

CD numeric vector of length = number of individuals which contains the CD of individuals. var(perf$\_i$)=s/CD$^2_i$

Value

The returned value is the log-likelihood value of HAPimLD method under hypothesis H0.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References

**Description**

The function gives the log-likelihood value of the HAPimLDL method under alternative hypothesis H1. It can be viewed as an internal function. The user does not have to call it by himself.

**Usage**

```
obj.LDL.add(param, don)
```

**Arguments**

- `param`: numeric vector of length=5 containing parameters to maximize:
  - error variance
  - Q allele effect
  - evolution time
  - Q allele frequency at time t=0
  - performance mean

- `don`: list of 12 objects. Some objects are results provided by descendant.pere(), moyenne.pere(), corresp(), pi.hap(), structure.hap(), poids.D() functions.

**Value**

The returned value is the log-likelihood value of the HAPimLDL method under alternative hypothesis H1.

**Author(s)**

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

**References**


**See Also**

depart.LDL, descendant.pere, moyenne.pere, corresp, pi.hap, structure.hap, poids.D
**Description**

The function calculates the log-likelihood value of the HAPimLDL method under hypothesis H0. It can be viewed as an internal function. The user does not have to call it by himself.

**Usage**

```r
obj.LDL.add.H0(moyenne.pere, perf, CD, desc.pere)
```

**Arguments**

- `moyenne.pere`: results provided by `moyenne.pere()` function, mean of half-sib family performances.
- `perf`: numeric vector of length=number of individuals which contains the performances of individuals.
- `CD`: numeric vector of length=number of individuals which contains the CD of individuals. var(perf_i)=s/CD^2_i
- `desc.pere`: results provided by `descendant.pere()` function, numeric matrix (number of sires x 2) which gives for each sire, the first and last indexes of its progeny.

**Value**

The returned value is the log-likelihood value of HAPimLDL method under hypothesis H0.

**Author(s)**

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

**References**


**See Also**

`moyenne.pere`, `descendant.pere`
pi.hap  

haplotype frequencies

Description

This function computes the frequencies of each haplotype. The function can be viewed as an internal function. The user does not have to call it by himself.

Usage

pi.hap(freq.marq, res.structure)

Arguments

freq.marq  results provided by freq.all() function, list of numeric objects which contains the allele frequencies.
res.structure  results provided by structure.hap() function, list of objects.

Value

The returned value is a list of numeric vectors containing haplotypes frequencies estimated under marker independency.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


See Also

freq.all, structure.hap
Description

This function computes the observed frequencies of each haplotype. The function can be viewed as an internal function. The user does not have to call it by himself.

Usage

pi.hap.NI(res.structure, cor.hap)

Arguments

res.structure results provided by structure.hap() function, list of objects.
cor.hap results provided by corresp() function, list of numeric objects.

Value

The returned value is a list of numeric vectors containing haplotypes frequencies estimated on haplotypes with non missing information.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


See Also

structure.hap
Description

This function calculates the probability of no recombination between loci for each Bennett’s disequilibrium. The function can be viewed as an internal function. The user does not have to call it by himself.

Usage

```r
poids.D(dist.test, pos.QTL, res.structure)
```

Arguments

- `dist.test`: results provided by `distance.test()` function, list of numeric objects.
- `pos.QTL`: numeric value, interval of a test position.
- `res.structure`: results provided by `structure.hap()` function, list of objects.

Value

The returned value is a list of numeric objects.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


See Also

`structure.hap, distance.test`
Description

The function calculates conditional probabilities of having Q allele given marker information due to linkage disequilibrium.

Usage

proba.DL(piQ.t0, esp.freq.hap, res.structure, pi.hap, res.corresp)

Arguments

- `piQ.t0`: frequency of Q allele at time t=0.
- `esp.freq.hap`: results provided by `esp.freq.hap()` function, numeric vector.
- `res.structure`: results provided by `structure.hap()` function, list of numeric objects.
- `pi.hap`: provided by `pi.hap()` function, list of numeric objects.
- `res.corresp`: results provided by `corresp()` function, list of 2 numeric objects.

Value

The returned value is a numeric vector of length=number of individuals which contains conditional probabilities of having Q allele given marker information due to linkage disequilibrium.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


See Also

`esp.freq.hap, structure.hap, pi.hap, corresp`
proba.DL.diploType  

PROBABILITIES DUE TO LINKAGE DISEQUILIBRIUM

Description

The function calculates probabilities due to linkage disequilibrium for a diplotype. It can be viewed as an internal function. The user does not have to call it by himself.

Usage

proba.DL.diploType(DL.chrom1, DL.chrom2)

Arguments

DL.chrom1  
results provided by proba.DL() function, numeric vector of length=number of individuals which contains probabilities due to linkage disequilibrium on the first chromosome given a genotype.

DL.chrom2  
results provided by proba.DL() function, numeric vector of length=number of individuals which contains probabilities due to linkage disequilibrium on the second chromosome given a genotype.

Value

The returned value is a numeric matrix (number of individuals x 4) containing probabilities due to linkage disequilibrium for a diplotype.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


See Also

proba.DL


**recode.hap**

**recoding of haplotypes**

---

**Description**

The function recodes haplotypes of individuals in consecutive numeric values. It can be viewed as an internal function. The user does not have to call it by himself.

**Usage**

```
recode.hap(hap, all.marq)
```

**Arguments**

- `hap` character matrix (number of individuals x number of markers) which contains the haplotype of individuals.
- `all.marq` results provided by `allele.marq()` function, list of alleles (coded with character values) for each marker.

**Value**

the returned value is a numeric matrix (number of individuals x number of markers) which contains the recoded haplotype of individuals.

**Author(s)**

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

**References**


**See Also**

`allele.marq`
retrouve.all  

*haplotype reconstruction*

**Description**

The function finds alleles of the associated haplotype. It can be viewed as an internal function. The user does not have to call it by himself.

**Usage**

```r
retrouve.all(assoc, res.structure, all.marq)
```

**Arguments**

- `assoc` numeric value, associated haplotype.
- `res.structure` results provided by `structure.hap()` function, list of objects.
- `all.marq` results provided by `allele.marq()` function, list of alleles for each marker.

**Value**

The returned value is a character which corresponds to the concatenation of alleles of the associated haplotype.

**Author(s)**

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

**References**


**See Also**

`structure.hap`, `allele.marq`
structure.hap

Description
The function can be viewed as an internal function. The user does not have to call it by himself.

Usage

structure.hap(nbre.marq, nbre.all.marq)

Arguments

nbre.marq number of markers.
nbre.all.marq number of alleles per markers.

Value
The value returned is a list of objects.

Author(s)
S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References

vrais.LD.add

log likelihood of HAPimLD method under H1

Description
The function calculates the value of log likelihood of HAPimLD method under alternative hypothesis H1. It can be viewed as an internal function. The user does not have to call it by himself.

Usage

vrais.LD.add(mu, alpha.Q, s, CD, perf, DL.d)
Arguments

mu parameter of the mean performances.
alpha.Q parameter of QTL effect.
s parameter of the error variance.
CD numeric vector of length=number of individuals which contains the CD of individuals. var(perf$_i$)=s/CD$_i^2$
perf numeric vector of length=number of individuals which contains the performances of individuals.
DL.d results provided by proba.DL.diploplotype() function, numeric matrix (number of individuals x 4) containing probabilities due to linkage disequilibrium for a diploplotype.

Value

The returned value is the value of log likelihood of HAPimLD method under alternative hypothesis H1.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


See Also

proba.DL.diploplotype

Description

The function calculates the value of log likelihood of HAPimLDL method under alternative hypothesis H1. It can be viewed as an internal function. The user does not have to call it by himself.

Usage

vrais.LDL.add(moyenne.pere, alpha.Q, s, CD, perf, PLA, DL.m, DL.chrom1,
DL.chrom2, desc.pere, mean.gene)
Arguments

- `moyenne.pere`: results provided by `moyenne.pere()` function, mean of half-sib family performances.
- `alpha.Q`: parameter of QTL effect.
- `s`: parameter of the error variance.
- `CD`: numeric vector of length=number of individuals which contains the CD of individuals. \( \text{var}(\text{perf}_i) = s/\text{CD}_i^2 \)
- `perf`: numeric vector of length=number of individuals which contains the performances of individuals.
- `PLA`: numeric vector (number of individuals) which contains transmission probabilities at a single test position.
- `DL.m`: results provided by `proba.DL()` function, numeric vector of length=number of individuals which contains probabilities due to linkage disequilibrium on dam.
- `DL.chrom1`: results provided by `proba.DL()` function, numeric vector of length=number of individuals which contains probabilities due to linkage disequilibrium on the first chromosome of sire.
- `DL.chrom2`: results provided by `proba.DL()` function, numeric vector of length=number of individuals which contains probabilities due to linkage disequilibrium on the second chromosome of sire.
- `desc.pere`: results provided by `descendant.pere()` function, numeric matrix (number of sires x 2) which gives for each sire, the first and last indexes of its progeny.
- `mean.gene`: parameter of the performance mean.

Value

The returned value is the value of log likelihood of HAPimLDL method under alternative hypothesis H1.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin

References


See Also

`moyenne.pere, descendant.pere, proba.DL`
Description

The function calculates the value of intra-sire log likelihood of HAPimLDL method under alternative hypothesis H1. It can be viewed as an internal function. The user does not have to call it by himself.

Usage

vrais.LDL.add.pere(moyenne.pere, alpha.Q, s, CD, perf, PLA, LD.m, LD.chrom1, LD.chrom2, mean.gene)

Arguments

- **moyenne.pere**: results provided by moyenne.pere() function, mean of half-sib family performances.
- **alpha.Q**: parameter of QTL effect.
- **s**: parameter of the error variance.
- **CD**: numeric vector of length=number of individuals which contains the CD of individuals. \( \text{var}(\text{perf}_i) = s \cdot CD_i^2 \)
- **perf**: numeric vector of length=number of individuals which contains the performances of individuals.
- **PLA**: numeric vector of length=number of individuals which contains transmission probabilities at a single test position.
- **LD.m**: results provided by proba.DL() function, numeric vector of length=number of individuals which contains probabilities due to linkage disequilibrium on dam.
- **LD.chrom1**: results provided by proba.DL() function, numeric vector of length=number of individuals which contains probabilities due to linkage disequilibrium on the first chromosome of sire.
- **LD.chrom2**: results provided by proba.DL() function, numeric vector of length=number of individuals which contains probabilities due to linkage disequilibrium on the second chromosome of sire.
- **mean.gene**: parameter of performance mean.

Value

The returned value is the value of intra-sire log likelihood of HAPimLDL method under alternative hypothesis H1.

Author(s)

S. Dejean, N. Oumouhou, D. Estivals, B. Mangin
References


See Also

moyenne.pere, proba.DL
Index

*Topic datasets
  data.test, 6

*Topic models
  allele.marq, 4
  corresp, 5
  depart.LD, 7
  depart.LDL, 8
  descendant.pere, 9
  distance.marqueurs, 10
  distance.test, 10
  esp.freq.hap, 11
  freq.all, 12
  haldanem1, 13
  HAPim (HAPim-package), 2
  hapim.LD.add, 13
  hapim.LDL.add, 15
  haplomax.add, 18
  haplomax.HS.add, 19
  moyenne.pere, 21
  obj.haplomax.add, 22
  obj.haplomax.HS.add, 23
  obj.LD.add, 24
  obj.LD.add.H0, 25
  obj.LDL.add, 26
  obj.LDL.add.H0, 27
  pi.hap, 28
  pi.hap.NI, 29
  poids.D, 30
  proba.DL, 31
  proba.DL.diplotype, 32
  recouve.all, 34
  structure.hap, 35
  vrais.LD.add, 35
  vrais.LDL.add, 36
  vrais.LDL.add.pere, 38

  allele.marq, 4, 33, 34
  corresp, 5, 23–26, 31

  data.test, 6
  depart.LD, 7, 25
  depart.LDL, 8, 26
  descendant.pere, 9, 22, 26, 27, 37
  distance.marqueurs, 10, 11
  distance.test, 10, 30
  esp.freq.hap, 11, 31
  freq.all, 12, 28
  haldanem1, 13
  HAPim (HAPim-package), 2
  hapim.LD.add, 13
  hapim.LDL.add, 15
  haplomax.add, 18
  haplomax.HS.add, 19
  moyenne.pere, 9, 21, 26, 27, 37, 39
  obj.haplomax.add, 22
  obj.haplomax.HS.add, 23
  obj.LD.add, 24
  obj.LD.add.H0, 25
  obj.LDL.add, 26
  obj.LDL.add.H0, 27
  pi.hap, 12, 23–26, 28, 31
  pi.hap.NI, 29
  poids.D, 12, 25, 26, 30
  proba.DL, 23, 31, 32, 37, 39
  proba.DL.diplotype, 32, 36
  recouve.all, 34
  structure.hap, 5, 12, 23–26, 28–31, 34, 35
  vrais.LD.add, 35
  vrais.LDL.add, 36
  vrais.LDL.add.pere, 38