Package ‘pedprobr’

November 1, 2019

Type Package
Title Probability Computations on Pedigrees
Version 0.1.0
Description An implementation of the Elston-Stewart algorithm for calculating pedigree likelihoods given genetic marker data (Elston and Stewart (1971) <doi:10.1159/000152448>). The standard algorithm is extended to allow inbred founders. Mutation modelling is supported by the 'pedmut' package. 'pedprobr' is part of the ped suite, a collection of packages for pedigree analysis in R, based on 'pedtools' for handling pedigrees and markers.
License GPL-3
URL https://github.com/magnusdv/pedprobr
Encoding UTF-8
Language en-GB
LazyData true
Depends R (>= 3.1.0), pedtools
Imports pedmut
Suggests testthat
RoxygenNote 6.1.1
NeedsCompilation no
Author Magnus Dehli Vigeland [aut, cre]
  (<https://orcid.org/0000-0002-9134-4962>)
Maintainer Magnus Dehli Vigeland <m.d.vigeland@medisin.uio.no>
Repository CRAN
Date/Publication 2019-11-01 12:30:02 UTC

R topics documented:

  allGenotypes ......................................................... 2
  genoCombinations .................................................. 2
  HWprob ............................................................... 3
**allGenotypes**

*Description*

An autosomal marker with \( n \) alleles has \( \text{choose}(n+1, 2) \) possible unordered genotypes. This function returns these as rows in a matrix.

*Usage*

```r
allGenotypes(n)
```

*Arguments*

- `n`: A positive integer.

*Value*

An integer matrix with two columns and \( \text{choose}(n+1, 2) \) rows.

*Examples*

```r
allGenotypes(3)
```

---

**genoCombinations**

*Description*

Returns the possible genotype combinations in a pedigree, given partial marker data. This function is mainly for internal use.

*Usage*

```r
genoCombinations(x, partialmarker, ids, make.grid = TRUE)
```
Arguments

- `x` a `ped()` object.
- `partialmarker` a `marker()` object compatible with `x`.
- `ids` a vector with ID labels of one or more pedigree members.
- `make.grid` a logical indicating if the result should be simplified to a matrix.

Value

If `make.grid = FALSE` (the default) the function returns a list of integer vectors, one vector for each element of `ids`. Each integer represents a genotype, in the form of a row number of the matrix `allGenotypes(n)`, where `n` is the number of alleles of the marker.

If `make.grid = TRUE`, the cartesian product of the vectors is taken, resulting in a matrix with one column for each element of `ids`.

---

**HWprob**

*Hardy-Weinberg probabilities*

**Description**

Hardy-Weinberg probabilities

**Usage**

```r
HWprob(allele1, allele2, afreq, f = 0)
```

**Arguments**

- `allele1, allele2` Vectors of equal length, containing alleles in the form of indices of `afreq`
- `afreq` A numeric vector with allele frequencies
- `f` A single number in `[0,1]`; the inbreeding coefficient

**Value**

A numeric vector of the same length as `allele1` and `allele2`

**Examples**

```r
p = 0.1; q = 1-p
hw = HWprob(c(1,1,2), c(1,2,2), c(p, q))
stopifnot(all.equal(hw, c(p^2, 2*p*q, q^2)))
```
likelihood  

**Pedigree likelihood**

**Description**

This function is the heart of pedprobr. It computes the likelihood of a pedigree (or a list of pedigrees) given genotypes for a marker or a pair of linked markers.

**Usage**

```r
likelihood(x, ...)  
## S3 method for class 'ped'
likelihood(x, marker1, marker2 = NULL, theta = NULL,
setup = NULL, eliminate = 0, logbase = NULL,
loop_breakers = NULL, verbose = FALSE, ...)

## S3 method for class 'singleton'
likelihood(x, marker1, marker2 = NULL,
logbase = NULL, ...)

## S3 method for class 'list'
likelihood(x, marker1, marker2 = NULL, logbase = NULL,
total = TRUE, ...)
```

**Arguments**

- `x`: a ped object, a singleton object, or a list of such objects.
- `...`: further arguments.
- `marker1`: a marker() object compatible with `x`. If `x` is a list, then `marker1` must be a list of corresponding marker objects.
- `marker2`: either NULL, or a marker() object compatible with `x`. See Details.
- `theta`: the recombination rate between `marker1` and `marker2`. To make biological sense theta should be between 0 and 0.5.
- `setup`: for internal use.
- `eliminate`: mostly for internal use: a non-negative integer indicating the number of iterations in the internal genotype-compatibility algorithm. Positive values can save time if the number of alleles is large.
- `logbase`: a numeric, or NULL. If numeric the log-likelihood is returned, with `logbase` as basis for the logarithm.
- `loop_breakers`: a vector of ID labels indicating loop breakers. If NULL (default), automatic selection of loop breakers will be performed. See `breakLoops()`.
- `verbose`: a logical
- `total`: a logical; if TRUE, the product of the likelihoods is returned, otherwise a vector with the likelihoods for each pedigree in the list.
**Details**

The implementation is based on the peeling algorithm of Elston and Stewart (1971). A variety of situations are covered; see the Examples section for some demonstrations.

- complex inbred pedigrees
- pedigrees with inbred founders
- autosomal and X-linked markers
- a single marker or two linked markers
- markers with mutation models

**Value**

The likelihood of the data. If the parameter logbase is a positive number, the output is log(likelihood, logbase).

**Author(s)**

Magnus Dehli Vigeland

**References**

Elston and Stewart (1971), doi: 10.1159/000152448

**Examples**

```r
### Example 1: Likelihood of trio with inbred father

x = cousinPed(0, child = TRUE)
x = addSon(x, 5)
x = relabel(x, old = 5:7, new = c("father", "mother", "child"))

# Equifrequent SNP marker: father homozygous, child heterozygous
m = marker(x, father = 1, child = 1:2)
x = addMarkers(x, m)

# Plot with genotypes
plot(x, marker = 1)

# Compute the likelihood
lik1 = likelihood(x, marker1 = 1)

### Example 2: Same as above, but using founder inbreeding

# Extract the trio
y = subset(x, c("father", "mother", "child"))

# Indicate that the father has inbreeding coefficient 1/4
founderInbreeding(y, "father") = 1/4
```
likelihoodMerlin

Pedigree likelihood computed by MERLIN

Description

This function is a wrapper of the "--likelihood" functionality of the MERLIN software. It computes the total likelihood of the pedigree given the indicated marker data. For this function to work, MERLIN must be installed and correctly pointed to in the PATH variable.

Usage

likelihoodMerlin(x, markers = seq_len(nMarkers(x)), logbase = NULL, verbose = FALSE, generateFiles = TRUE, cleanup = generateFiles, logfile = "")

Arguments

x

a ped object

markers

a vector of names or indices of markers attached to x. (Default: all markers).

logbase

a positive number, or NULL. If numeric, the log-likelihood is returned, with logbase as basis for the logarithm.

verbose

a logical.

generateFiles

a logical. If TRUE, input files to MERLIN named '_merlin.ped', '_merlin.dat', '_merlin.map', and '_merlin.freq' are created in the current directory. If FALSE, no files are created.

cleanup

a logical. If TRUE, the MERLIN input files are deleted after the call to MERLIN.

logfile

a character. If this is given, the MERLIN screen output will be written to a file with this name.

Details

By default the following MERLIN command is run via system(), after creating appropriate files in the current working directory:

_merlin.freq --likelihood --bits:100 --megabytes:4000 --quiet
oneMarkerDistribution

Value
A number.

Author(s)
Magnus Dehli Vigeland

References
http://csg.sph.umich.edu/abecasis/Merlin/

Examples

### Requires MERLIN to be installed ###

```r
x = nuclearPed(1)
m = marker(x, "3" = 1:2)
x = setMarkers(x, m)

# Likelihood computation by MERLIN:
likelihoodMerlin(x)
```

Description
Computes the genotype probability distribution of one or several pedigree members, possibly conditional on known genotypes for the marker.

Usage

```r
oneMarkerDistribution(x, ids, partialmarker, loop_breakers = NULL,
                      eliminate = 0, grid.subset = NULL, verbose = TRUE)
```

Arguments

- `x` A ped object.
- `ids` A numeric with ID labels of one or more pedigree members.
- `partialmarker` Either a marker object or the name (or index) of a marker attached to `x`.
- `loop_breakers` (Only relevant if the pedigree has loops). A vector with ID labels of individuals to be used as loop breakers. If NULL (default) loop breakers are selected automatically. See `breakLoops()`.
eliminate  A non-negative integer, indicating the number of iterations in the internal genotype-compatibility algorithm. Positive values can save time if `partialmarker` has many alleles.

grid.subset  (Optional; not relevant for most users.) A numeric matrix describing a subset of all marker genotype combinations for the `ids` individuals. The matrix should have one column for each of the `ids` individuals, and one row for each combination: The genotypes are described in terms of the matrix \( M = \text{allGenotypes}(n) \), where \( n \) is the number of alleles for the marker. If the entry in column \( j \) is the integer \( k \), this means that the genotype of individual \( ids[j] \) is row \( k \) of \( M \).

verbose  A logical.

Value

A named \( k \)-dimensional array, where \( k = \text{length}(ids) \), with the joint genotype distribution for the `ids` individuals. The probabilities are conditional on the known genotypes and the allele frequencies of `partialmarker`.

Author(s)

Magnus Dehli Vigeland

See Also

twoMarkerDistribution()

Examples

```r
# Trivial example giving Hardy-Weinberg probabilities
s = singleton(id = 1)
m = marker(s, alleles = 1:2)  # equifrequent SNP
oneMarkerDistribution(s, ids = 1, partialmarker = m)

# Conditioning on a partial genotype
genotype(m, id = 1) = c(1, NA)
oneMarkerDistribution(s, ids = 1, partialmarker = m)

# Genotype distribution for a child of heterozygous parents
trio = nuclearPed(father = "fa", mother = "mo", child = "ch")
m1 = marker(trio, fa = 1:2, mo = 1:2)
oneMarkerDistribution(trio, ids = "ch", partialmarker = m1)

# Joint distribution of the parents, given that the child is heterozygous
m2 = marker(trio, ch = 1:2, alleles = 1:2, afreq = c(0.5, 0.5))
oneMarkerDistribution(trio, ids = c("fa", "mo"), partialmarker = m2)

# A different example: The genotype distribution of an individual (id = 5)
# whose half cousin (id = 9) is homozygous for a rare allele.
y = halfCousinPed(degree = 1)
snp = marker(y, \(^9\) = "a", alleles = c("a", "b"), afreq = c(0.01, 0.99))
plot(y, snp)
```
Description

An implementation of the Elston-Stewart algorithm for calculating pedigree likelihoods given genetic marker data (Elston and Stewart (1971), doi: 10.1159/000152448). The standard algorithm is extended to allow inbred founders. Mutation modelling is included via the ‘pedmut’ package. ‘pedprobr’ is part of the ped suite, a collection of packages for pedigree analysis in R, based on ‘pedtools’ for handling pedigrees and markers.

twoMarkerDistribution  Genotype distribution for two linked markers

Description

Computes the joint genotype distribution of two markers for a specified pedigree member, conditional on known genotypes and the recombination rate between the markers.

Usage

twoMarkerDistribution(x, id, partialmarker1, partialmarker2, theta, 
  loop_breakers = NULL, eliminate = 99, verbose = TRUE)

Arguments

x  A ped object.

id  A single ID label.

partialmarker1, partialmarker2
  Either a marker object, or the name (or index) of a marker attached to x.

theta  A single numeric in the interval [0, 0.5]: the recombination fraction between the two markers.

loop_breakers  (Only relevant if the pedigree has loops). A vector with ID labels of individuals to be used as loop breakers. If NULL (default) loop breakers are selected automatically. See breakLoops().

eliminate  A non-negative integer, indicating the number of iterations in the internal algorithm for reducing the genotype space. Positive values can save time if partialmarker1 and/or partialmarker2 have many alleles.

verbose  A logical.
Value

A named matrix giving the joint genotype distribution.

Author(s)

Magnus Dehli Vigeland

See Also

oneMarkerDistribution()

Examples

# A sib-pair pedigree
x = nuclearPed(children = c("bro1", "bro2"))

# Two SNP markers; first brother homozygous for the '1' allele
SNP1 = SNP2 = marker(x, bro1 = c(1,1), alleles = 1:2)

plot(x, marker = list(SNP1, SNP2))

# Genotype distribution for the brother: Depends on theta
twoMarkerDistribution(x, id = "bro2", SNP1, SNP2, theta = 0)
twoMarkerDistribution(x, id = "bro2", SNP1, SNP2, theta = 0.5)

# X-linked
chrom(SNP1) = chrom(SNP2) = "X"

plot(x, marker = list(SNP1, SNP2))

twoMarkerDistribution(x, id = "bro2", SNP1, SNP2, theta = 0)
twoMarkerDistribution(x, id = "bro2", SNP1, SNP2, theta = 0.5)
Index

allGenotypes, 2
breakLoops(), 4, 7, 9
genoCombinations, 2
HWprob, 3
likelihood, 4
likelihoodMerlin, 6
marker(), 3, 4
oneMarkerDistribution, 7
oneMarkerDistribution(), 10
ped, 6
ped(), 3
pedprobr, 9
pedprobr-package (pedprobr), 9
system(), 6
twoMarkerDistribution, 9
twoMarkerDistribution(), 8