Package 'clipp'

July 6, 2020

Type Package

Title Calculating Likelihoods by Pedigree Paring

Version 0.1.0

Description A fast and general implementation of the Elston-Stewart algorithm that can calculate the likelihoods of large and complex pedigrees without loops. References for the Elston-Stewart algorithm are Elston & Stewart (1971) <doi:10.1159/000152448>, Lange & Elston (1975) <doi:10.1159/000152714> and Cannings et al. (1978) <doi:10.2307/1426718>.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author James Dowty [aut, cre], Kevin Wong [aut]

Maintainer James Dowty <jgdowty@gmail.com>

Repository CRAN

Date/Publication 2020-07-06 14:00:02 UTC

R topics documented:

dat_large — *Simulated data on one family with approximately 10,000 members.*

Description

A dataset giving the relationship structure of one large family and phenotypic data on the family members

Usage

```
dat_large
```

Format

A data frame with 10,002 rows (corresponding to persons) and 6 variables:

indiv an identifier (ID) for the person

mother the individual ID of the person's mother

father the individual ID of the person's father

sex the person's sex (1 = male, 2 = female)

aff the person's disease status (1 = case, 0 = control)

age the person's age, in years

Source

Simulated

dat_small — *Simulated data on 10 families with approximately 100 members each.*

Description

A dataset giving the relationship structure of 10 families and phenotypic data on the family members

Usage

```
dat_small
```

Format

A data frame with 1018 rows (corresponding to persons) and 7 variables:

family an identifier (ID) for the person's family

indiv an ID for the person

mother the individual ID of the person's mother

father the individual ID of the person's father

sex the person's sex (1 = male, 2 = female)

aff the person's disease status (1 = case, 0 = control)

age the person's age, in years

Source

Simulated

`genotype_probabilities` *Calculate the genotype probabilities of a target person*

Description

For a given individual within a given family, calculate the person's conditional genotype probabilities, given the family's phenotypes and relationship structure

Usage

```
genotype_probabilities(target_ID, fam, geno_freq, trans, penet)
```

Arguments

`target_ID`	The individual identifier (`indiv`) of the person in the pedigree `fam` whose genotype probabilities are sought.
`fam`	A data frame with rows corresponding to people and columns corresponding to the following variables (other variables can be included but will be ignored), which will be coerced to `character` type: • `indiv`, an identifier for each individual person, with no duplicates in `fam`. • `mother`, the individual identifier of each person's mother, or missing (`NA`) for founders. • `father`, the individual identifier of each person's father, or missing (`NA`) for founders.
`geno_freq`	A vector of strictly positive numbers that sum to `1`. The possible genotypes of the underlying genetic model are `1:length(geno_freq)`, and `geno_freq[j]` is interpreted as the population frequency of genotype `j`. For certain genetic models that often occur in applications, these genotype frequencies can be calculated by `geno_freq_HWE`.

trans — An ngeno^2 by ngeno matrix of non-negative numbers whose rows all sum to 1, where ngeno = length(geno_freq) is the number of possible genotypes. The rows of trans correspond to joint parental genotypes and the columns to offspring genotypes. The element trans[ngeno * gm + gf -ngeno,go] is interpreted as the conditional probability that a person has genotype go, given that his or her biological mother and father have genotypes gm and gf, respectively. For certain genetic models that often occur in applications, this transmission matrix can be calculated by trans_monogenic.

penet — An nrow(dat) by length(geno_freq) matrix of non-negative numbers. The element penet[i,j] is interpreted as the conditional probability of the phenotypes of the person corresponding to row i of fam, given that his or her genotype is j. Note that known genotype data can be incorporated into penet by regarding known genotypes as phenotypes, i.e. by regarding known genotypes as (possibly noisy) measurements of the underlying true genotypes. If any row of penet consists entirely of zeroes then the family's phenotypes are impossible, and -Inf is returned.

Details

The genotype probabilities are calculated by essentially the same algorithm as the one that underlies pedigree_loglikelihood; see there for details.

Value

A vector of length length(geno_freq) whose jth element is the conditional probability that the target person has genotype j, given the family's relationship structure and phenotypes.

Examples

```
# Read in some sample data
data("dat_small", "penet_small")
str(dat_small)
str(penet_small)

# Calculate the genotype probabilities for individual "ora008" in the family "ora"
w <- which(dat_small$family == "ora")
fam <- dat_small[w, -1]
penet <- penet_small[w, ]
trans <- trans_monogenic(2)
geno_freq <- geno_freq_HWE(p_alleles = c(0.9, 0.1))
genotype_probabilities(target_ID = "ora008", fam, geno_freq, trans, penet)
```

---

geno_freq_HWE — *Calculate genotype frequencies from allele frequencies, assuming Hardy-Weinberg equilibrium*

Description

A function to calculate the unphased genotype frequencies for a single autosomal genetic locus that has given allele frequencies and is at Hardy-Weinberg equilibrium (HWE).

Usage

```
geno_freq_HWE(p_alleles, annotate = FALSE)
```

Arguments

p_alleles	A vector of strictly positive numbers that sum to 1, with p_alleles[i] interpreted as the allele frequency of the ith allele of the genetic locus.
annotate	A logical flag. When FALSE (the default), the function returns a vector suitable to be used as the geno_freq argument of pedigree_loglikelihood. When TRUE, the function adds a names attribute to this vector to indicate which genotype corresponds to which element.

Details

For a genetic locus at HWE, the population allele frequencies at the locus determine the population genotype frequencies; see Section 1.3 of (Lange, 2002). Given a vector p_alleles containing the allele frequencies, this function returns the frequencies of the possible unphased genotypes, in a particular order. If the alleles are named 1:length(p_alleles), with p_alleles[i] being the frequency of allele i, then the unphased genotypes are of the form 1/1, 1/2, ..., and setting annotate to TRUE names each element of the output vector with the corresponding genotype. Note that if the output of this function is to be used as the geno_freq argument of pedigree_loglikelihood then the annotate option must be set to FALSE.

Value

A vector of strictly positive numbers (the genotype frequencies) that sum to 1, with genotype names added when annotate is TRUE

References

Lange K. Mathematical and Statistical Methods for Genetic Analysis (second edition). Springer, New York. 2002.

Examples

```
# Genotype frequencies for a biallelic locus at HWE and with a minor allele frequency of 10%
p_alleles <- c(0.9, 0.1)
geno_freq_HWE(p_alleles, annotate = TRUE)

# Genotype frequencies for a triallelic locus at HWE
p_alleles <- c(0.85, 0.1, 0.05)
geno_freq_HWE(p_alleles, annotate = TRUE)
sum(geno_freq_HWE(p_alleles))
```

pedigree_loglikelihood

Calculate the log-likelihoods of pedigrees

Description

For one or more pedigrees, this function calculates the natural logarithm of the pedigree likelihood on page 117 of (Lange, 2002), given inputs that correspond to the terms in this formula.

Usage

```
pedigree_loglikelihood(
  dat,
  geno_freq,
  trans,
  penet,
  sum_loglik = TRUE,
  ncores = 1
)
```

Arguments

dat	A data frame with rows corresponding to people and columns corresponding to the following variables (other variables can be included but will be ignored), which will be coerced to `character` type: • `family` (optional), an identifier for each person's family, constant within families. If this variable is not supplied then `dat` will be treated as a single pedigree. • `indiv`, an identifier for each individual person, with no duplicates across the dataset. • `mother`, the individual identifier of each person's mother, or missing (`NA`) for founders. • `father`, the individual identifier of each person's father, or missing (`NA`) for founders.
geno_freq	A vector of strictly positive numbers that sum to `1`. The possible genotypes of the underlying genetic model are `1:length(geno_freq)`, and `geno_freq[j]` is interpreted as the population frequency of genotype `j`, so `geno_freq` is essentially the function `Prior` of (Lange, 2002). For certain genetic models that often occur in applications, these genotype frequencies can be calculated by `geno_freq_HWE`.
trans	An `ngeno^2` by `ngeno` matrix of non-negative numbers whose rows all sum to `1`, where `ngeno = length(geno_freq)` is the number of possible genotypes. The rows of `trans` correspond to joint parental genotypes and the columns to offspring genotypes. The element `trans[ngeno * gm + gf - ngeno, go]` is interpreted as the conditional probability that a person has genotype `go`, given that

his or her biological mother and father have genotypes `gm` and `gf`, respectively. So `trans` is essentially the transmission function `Tran` of (Lange, 2002). For certain genetic models that often occur in applications, this transmission matrix can be calculated by `trans_monogenic`.

`penet` An `nrow(dat)` by `length(geno_freq)` matrix of non-negative numbers. The element `penet[i,j]` is interpreted as the conditional probability of the phenotypes of the person corresponding to row `i` of `dat`, given that his or her genotype is `j`. So `penet` is essentially the penetrance function `Pen` of (Lange, 2002). Note that genotype data can be incorporated into `penet` by regarding known genotypes as phenotypes, i.e. by regarding known genotypes as (possibly noisy) measurements of the underlying true genotypes. If any row of `penet` consists entirely of zeroes then the likelihood is `0`, so the returned log-likelihood is `-Inf`.

`sum_loglik` A logical flag. Return a named vector giving the log-likelihood of each family if `sum_loglik` is `FALSE`, or return the sum of these log-likelihoods if `sum_loglik` is `TRUE` (the default).

`ncores` The number of cores to be used, with `ncores = 1` (the default) corresponding to non-parallel computing. When `ncores > 1`, the `parallel` package is used to parallelize the calculation by dividing the pedigrees among the different cores.

Details

This function provides a fast and general implementation of the Elston-Stewart algorithm to calculate the log-likelihoods of large and complex pedigrees without loops. General references for the Elston-Stewart algorithm are (Elston & Stewart, 1971), (Lange & Elston, 1975) and (Cannings et al., 1978).

Each family within `dat` should be a complete pedigree, meaning that each non-missing mother or father ID should correspond to a row, and each person should either have both parental IDs missing (if a founder) or non-missing (if a non-founder). No family should contain pedigree loops, such as those caused by inbreeding or by two sisters having children with two brothers from a different family; see (Totir et al., 2009) for a precise definition. The code is not currently suitable for families containing identical twins, since these are not distinguished from full siblings.

In `geno_freq`, `trans` and `penet`, the order of the possible genotypes must match (so that the same genotype corresponds to element `j` of `geno_freq` and column `j` of `trans` and `penet`, for each `j` in `1:length(geno_freq)`).

Value

Either a named vector giving the log-likelihood of each family or the sum of these log-likelihoods, depending on `sum_loglik` (see above).

References

Cannings C, Thompson E, Skolnick M. Probability functions on complex pedigrees. Advances in Applied Probability, 1978;10(1):26-61.

Elston RC, Stewart J. A general model for the genetic analysis of pedigree data. Hum Hered. 1971;21(6):523-542.

Lange K. Mathematical and Statistical Methods for Genetic Analysis (second edition). Springer, New York. 2002.

Lange K, Elston RC. Extensions to pedigree analysis I. Likehood calculations for simple and complex pedigrees. Hum Hered. 1975;25(2):95-105.

Totir LR, Fernando RL, Abraham J. An efficient algorithm to compute marginal posterior genotype probabilities for every member of a pedigree with loops. Genet Sel Evol. 2009;41(1):52.

Examples

```
# Load pedigree files and penetrance matrices
data("dat_small", "penet_small", "dat_large", "penet_large")

# Settings for a biallelic locus in Hardy-Weinberg equilibrium and with a
# minor allele frequency of 10%
geno_freq <- geno_freq_HWE(c(0.9, 0.1))
trans <- trans_monogenic(2)

# Calculate the log-likelihoods for 10 families, each with approximately 100 family members
pedigree_loglikelihood(
  dat_small, geno_freq, trans, penet_small, sum_loglik = FALSE, ncores = 2
)

# Calculate the log-likelihood for one family with approximately 10,000 family members
# Note:  this calculation takes approximately one minute on a standard desktop computer
# Note:  parallelization achieves nothing here because there is only one family
str(dat_large)

system.time(
  ll <- pedigree_loglikelihood(dat_large, geno_freq, trans, penet_large)
)
ll
```

`penet_large` *A penetrance matrix relating the phenotypes in* `dat_large` *to three genotypes*

Description

A matrix relating the phenotypes of `dat_large` to the three unphased genotypes of a single biallelic, autosomal genetic locus. The element `penet_small[i,j]` is the conditional probability of the the phenotypes (i.e. `sex`, `aff` and `age`) of the person in row `i` of `dat_large`, given that his or her genotype is `j` (here labelling the genotypes as 1, 2, 3, where genotype 2 is the heterozygous genotype).

Usage

```
penet_large
```

Format

A matrix with 10,000 rows (corresponding to persons) and 3 columns (corresponding to genotypes).

Source

Simulated

---

penet_small — *A penetrance matrix relating the phenotypes of* dat_small *to three genotypes*

Description

A matrix relating the phenotypes of `dat_small` to the three unphased genotypes of a single biallelic, autosomal genetic locus. The element `penet_small[i,j]` is the conditional probability of the the phenotypes (i.e. `sex`, `aff` and `age`) of the person in row `i` of `dat_small`, given that his or her genotype is `j` (here labelling the genotypes as 1, 2, 3, where genotype 2 is the heterozygous genotype)

Usage

```
penet_small
```

Format

A matrix with 1018 rows (corresponding to persons) and 3 columns (corresponding to genotypes).

Source

Simulated

---

trans_monogenic — *The transmission matrix for a common genetic model*

Description

A function to calculate the transmission matrix for a single autosomal genetic locus with an arbitrary number of alleles and unphased genotypes, based on Mendel's laws of inheritance.

Usage

```
trans_monogenic(n_alleles, annotate = FALSE)
```

Arguments

n_alleles	A positive integer, interpreted as the number of possible alleles at the genetic locus.
annotate	A logical flag. When FALSE (the default), the function returns a matrix suitable to be used as the trans argument of pedigree_loglikelihood. When TRUE, the function annotates this matrix (and converts it to a data frame) to make the output more easily understood by humans.

Details

When annotate is FALSE, a matrix of genetic transmission probabilities is returned, with rows corresponding to the possible joint parental genotypes and columns corresponding to the possible offspring genotypes. There are ngeno = n_alleles*(n_alleles + 1)/2 possible unphased genotypes, and by choosing an order on these genotypes (see below) we can take the set of possible genotypes to be 1:ngeno. Then the (ngeno * gm + gf - ngeno, go)th element of the outputted matrix is the conditional probability that a person has genotype go, given that his or her biological mother and father have genotypes gm and gf, respectively. When annotate is TRUE, the function converts this matrix to a data frame, adds column names giving the offspring genotype corresponding to each column, and adds columns gm and gf describing the parental genotypes corresponding to each row. In this data frame, all genotypes are written in the usual form 1/1, 1/2, ... for alleles 1:n_alleles, so these annotations show the genotype order referred to above. Note that if the output of this function is to be used as the trans argument of pedigree_loglikelihood then the annotate option must be set to FALSE.

Value

Either a matrix of genetic transmission probabilities suitable to be used as the trans argument of pedigree_loglikelihood (if annotate is FALSE), or a data frame that is an annotated version of this matrix (if annotate is TRUE).

Examples

```
# The transition matrix for a biallelic, autosomal locus with unphased genotypes
trans_monogenic(2)
trans_monogenic(2, annotate = TRUE)
```

Index