Package ‘clipp’

July 12, 2022

Type Package

Title Calculating Likelihoods by Pedigree Paring

Version 1.1.1

Description A fast and general implementation of the Elston-Stewart algorithm
that can calculate the likelihoods of large and complex pedigrees.
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

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

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

Maintainer James Dowty <jgdowty@gmail.com>

Repository CRAN

Date/Publication 2022-07-12 05:30:02 UTC

R topics documented:

clipp-package *clipp: Calculate Likelihoods by Pedigree Paring*

Description

clipp provides a fast and general implementation of the Elston-Stewart algorithm, and can calculate the log-likelihoods of large and complex pedigrees, including those with loops. General references for the Elston-Stewart algorithm are (Elston & Stewart, 1971), (Lange & Elston, 1975) and (Cannings et al., 1978).

Details

The main function is pedigree_loglikelihood, which calculates the pedigree likelihood on page 117 of (Lange, 2002) for almost any choice of genotype frequencies, transmission matrix and penetrance matrix. Helper functions are provided to calculate the genotype frequencies and transmission matrices for genetic models that often arise in applications. The function genotype_probabilities calculates genotype probabilities for a target person within a family, given the family's phenotypes.

The clipp package can handle pedigree loops, such as those caused by inbreeding or by two sisters having children with two brothers from an unrelated family (see (Totir et al., 2009) for a precise definition). However, pedigrees with more than a few loops could greatly reduce the speed of the calculation.

It is feasible to apply clipp to very large families. For instance, in the examples for pedigree_loglikelihood, the log-likelihood of one family with approximately 10,000 members is calculated in less than one minute on a standard desktop computer. Numerical issues will eventually limit the family size, though clipp takes care to avoid arithmetic underflow and other issues.

Author(s)

Maintainer: James Dowty <jgdowty@gmail.com>

Authors:

- Kevin Wong <wongck.kevin@gmail.com>

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.

`combine_loci` *Combine the genotype frequencies and transition matrices of two genetic loci*

Description

A function to calculate the genotype frequencies and the transition matrices for the joint genotypes of two unlinked genetic loci in linkage equilibrium, given the corresponding objects for the separate loci. The results from this function can be used as inputs to `pedigree_loglikelihood` or `genotype_probabilities` to model the combined effect of the two loci on phenotypes.

Usage

```
combine_loci(geno_freq1, geno_freq2, trans1, trans2, annotate = FALSE)
```

Arguments

`geno_freq1` A vector of strictly positive numbers that sum to 1, with `geno_freq1[i]` interpreted as the population genotype frequency of the `i`th possible genotype at a genetic locus (locus 1). When `annotate` is `TRUE`, the names of the genotypes at locus 1 will be taken to be `names(geno_freq1)` or, if `names(geno_freq1)` is `NULL`, to be `1:length(geno_freq1)`.

`geno_freq2` Similar to `geno_freq1` (above) but interpreted as the population genotype frequencies for a different genetic locus (locus 2).

`trans1` An `ngeno1^2` by `ngeno1` matrix of non-negative numbers whose rows sum to `1`, where `ngeno1 = length(geno_freq1)`. This matrix is usually generated by `trans_monogenic` or a similar helper function, and its elements are interpreted as genetic transmission probabilities for locus 1 (see `trans_monogenic` for more details). If `trans1` has `ngeno1 + 2` instead of `ngeno1` columns, as could occur if it was generated by `trans_monogenic` with `annotate = TRUE`, then the first two columns will be deleted and `trans1` will be converted to a matrix.

`trans2` Similar to `trans1` (above) but interpreted as the genetic transmission probabilities for locus 2.

`annotate` A logical flag. When `FALSE` (the default), the function returns objects that can be used as the `geno_freq` and `trans` arguments of `pedigree_loglikelihood`. When `TRUE`, the function annotates these objects (and converts `trans` to a data frame) to make the output more easily understood by humans.

Details

This function combines the genotype frequencies and transition probabilities of two unlinked genetic loci that are in linkage equilibrium in a given population. Because the loci are unlinked, any person's genotypes at the two loci are conditionally independent given his or her parental genotypes, and because the loci are in linkage equilibrium, the genotypes at the two loci for a random person from the population are independent. This function uses these assumptions to calculate the population frequencies and transition probabilities for the joint genotypes of the two loci, where a joint genotype is just a pair consisting of a genotype at locus 1 and a genotype at locus 2. If the `annotate` option is set to `FALSE` then these frequencies and probabilities can be used in `pedigree_loglikelihood` to model the combined effect of the two loci on the phenotypes. By a repeated application of this function, more than two genetic loci can be included in the genetic model.

Value

A list with the following components:

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

`trans` 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`).

`genotype_decoder`
A data frame giving the locus 1 and locus 2 genotypes that correspond to each joint genotype. In some cases, this could aid the user's calculation of the `penet` argument of `pedigree_loglikelihood`.

Examples

```
pa1 <- c(0.9, 0.1); names(pa1) <- c("-","+")
pa2 <- c(0.5, 0.5); names(pa2) <- c("A","a")
(geno_freq1 <- geno_freq_monogenic(pa1, TRUE))
(geno_freq2 <- geno_freq_monogenic(pa2, TRUE))
(trans1 <- trans_monogenic(2, TRUE))
(trans2 <- trans_monogenic(2))
(cl <- combine_loci(geno_freq1, geno_freq2, trans1, trans2, TRUE))
sum(cl$geno_freq)
apply(cl$trans[,-(1:2)], 1, sum)
```

`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 the following 6 variables:

`indiv` an individual identifier for each person

`mother` the individual identifier of the person's mother

`father` the individual identifier 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 last known age in years (for controls) or age at diagnosis in years (for cases)

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 the following 7 variables:

`family` an identifier for each person's family

`indiv` an individual identifier for each person

`mother` the individual identifier of the person's mother

`father` the individual identifier 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 last known age in years (for controls) or age at diagnosis in years (for cases)

`geno` the person's genotype, or blank (`""`) if not known

Source

Simulated

`genotype_probabilities` *Calculate genotype probabilities for a target person*

Description

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

Usage

```
genotype_probabilities(target, fam, geno_freq, trans, penet, monozyg = NULL)
```

Arguments

`target` The individual identifier (an element of `fam$indiv`) of the person in the pedigree `fam` whose genotype probabilities are being sought.

`fam` A data frame specifying the family's relationship structure, 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`. If the possible genotypes of the underlying genetic model are `1:length(geno_freq)` then `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_monogenic`, `geno_freq_phased`, etc.

`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 correspond to offspring genotypes. If the possible genotypes are `1:length(geno_freq)` then 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`, `trans_phased`, etc.

`penet` An `nrow(fam)` by `length(geno_freq)` matrix of non-negative numbers. The element `penet[i,j]` is interpreted as the conditional probability (or probability density) of the phenotype of the person corresponding to row `i` of `fam`, given that his or her genotype is `j` (where the possible genotypes are `1:length(geno_freq)`). Note that genotype data can be incorporated into `penet` by regarding observed genotypes as part of the phenotype, i.e. by regarding observed genotypes as (possibly noisy) measurements of the underlying true genotypes. For example, if the observed genotype of person `i` is `1` (and if genotype measurement error is negligible) then `penet[i,j]` should be `0` for `j != 1` and `penet[i,1]` should be the same as if person `i` were ungenotyped.

`monozyg` An optional list that can be used to specify genetically identical persons, such as monozygotic twins, monozygotic triplets, a monozygotic pair within a set of dizygotic triplets, etc. Each element of the list should be a vector containing the individual identifiers of a group of genetically identical persons, e.g. if `fam` contains a set of monozygotic twins (and no other genetically identical persons) then `monozyg` will be a list with one element, and that element will be a vector of length two containing the individual identifiers of the twins. The order of the list and the orders of its elements do not affect the output of the function. Each group of genetically identical persons should contain two or more persons, the groups should not overlap, and all persons in each group must have the same (non-missing) parents.

Details

The genotype probabilities are calculated by essentially the same algorithm as `pedigree_loglikelihood`; see there for details. The genotype probabilities only depend on the connected component of the pedigree that contains `target`, so the function first restricts `fam` and `penet` to the rows corresponding to this connected component. For example, if `fam` is the union of two unrelated families then this function will restrict to the subfamily containing `target` before performing the calculation.

Value

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

NAs will be returned if a row of penet consists entirely of zeroes or if the pedigree is impossible for any other reason (after restricting fam and penet to the connected component of the pedigree containing target).

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, ]
monozyg <- list(c("ora024", "ora027"))  # ora024 and ora027 are identical twins
trans <- trans_monogenic(2)
geno_freq <- geno_freq_monogenic(p_alleles = c(0.9, 0.1))
genotype_probabilities(target = "ora008", fam, geno_freq, trans, penet, monozyg)
```

geno_freq_monogenic *Calculate genotype frequencies from allele frequencies using 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_monogenic(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. When annotate is TRUE, the names of the alleles will be taken to be names(p_alleles) or, if names(p_alleles) is NULL, to be 1:length(p_alleles).

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 Hardy-Weinberg equilibrium, the population allele frequencies at the locus determine the population genotype frequencies; see Sections 1.2 and 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 that can be viewed by setting `annotate` to `TRUE`. If the alleles are named `1:length(p_alleles)`, so that `p_alleles[i]` is the frequency of allele `i`, then the unphased genotypes are named `1/1, 1/2, ....` 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`, named with the genotype names if `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_monogenic(p_alleles, annotate = TRUE)

# Genotype frequencies for a triallelic locus at Hardy-Weinberg equilibrium
p_alleles <- c(0.85, 0.1, 0.05)
geno_freq_monogenic(p_alleles, annotate = TRUE)
sum(geno_freq_monogenic(p_alleles))
```

`geno_freq_phased` *Calculate phased genotype frequencies from allele frequencies, assuming Hardy-Weinberg equilibrium*

Description

A function to calculate the population frequencies of the phased genotypes at a single autosomal genetic locus that has given allele frequencies and is at Hardy-Weinberg equilibrium. Phased genotypes can be used to investigate parent-of-origin effects, e.g. see (van Vliet et al., 2011).

Usage

```
geno_freq_phased(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 `i`th allele of the genetic locus. When `annotate` is `TRUE`, the names of the alleles will be taken to be `names(p_alleles)` or, if `names(p_alleles)` is `NULL`, to be `1:length(p_alleles)`.

`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 element corresponds to which phased genotype.

Details

For a genetic locus that is at Hardy-Weinberg equilibrium in a particular population, the population allele frequencies at the locus determine the population genotype frequencies; see Sections 1.2 and 1.3 of (Lange, 2002) for the unphased version of this law. When a genetic locus is at Hardy-Weinberg equilibrium, the maternal and paternal alleles of a random person from the population are independent. A phased genotype at a genetic locus is an ordered pair consisting of a maternal and paternal allele at the locus. So to any heterozygous unphased genotype, there are two corresponding phased genotypes, and these two phased genotypes have equal frequencies under Hardy-Weinberg equilibrium.

Given a vector `p_alleles` containing the allele frequencies, this function returns the frequencies of the possible phased genotypes, in a particular order that can be viewed by setting `annotate` to `TRUE`. If the alleles are named `1:length(p_alleles)`, so that `p_alleles[i]` is the frequency of allele `i`, then the phased genotypes are of the form `1|1, 1|2, ...`, where `a|b` means the maternal allele is `a` and the paternal allele is `b`. 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`, named with the genotype names if `annotate` is `TRUE`.

References

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

van Vliet CM, Dowty JG, van Vliet JL, et al. Dependence of colorectal cancer risk on the parent-of-origin of mutations in DNA mismatch repair genes. Hum Mutat. 2011;32(2):207-212.

Examples

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

# Genotype frequencies for a triallelic locus at Hardy-Weinberg equilibrium
p_alleles <- c(0.85, 0.1, 0.05)
```

```
geno_freq_phased(p_alleles, annotate = TRUE)
sum(geno_freq_phased(p_alleles))
```

geno_freq_polygenic *Genotype frequencies for the hypergeometric polygenic model*

Description

A function to calculate the genotype frequencies for the hypergeometric polygenic model of (Cannings et al., 1978), see Section 8.9 of (Lange, 2002) for a nice description of this model.

Usage

```
geno_freq_polygenic(n_loci, annotate = FALSE)
```

Arguments

`n_loci` A positive integer, interpreted as the number of biallelic genetic loci that contribute to the polygene. The polygene will have `2*n_loci + 1` genotypes, so `n_loci` is typically fairly small, e.g. `4`.

`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 element corresponds to which genotype.

Details

The hypergeometric polygenic model (Cannings et al., 1978; Lange, 2002) is a computationally feasible genetic model that approximates the combined effect of a given number (`n_loci`) of unlinked biallelic genetic loci. This model is often used to model the effect of such loci on a trait when the alleles at these loci either increase the trait by a certain, locus-independent amount (if a 'positive' allele) or decrease the trait by the same amount (if a 'negative' allele), with 'positive' and 'negative' alleles equally likely at each locus. In this case, the only relevant aspect of the `3 ^ n_loci` possible joint genotypes is the total number of 'positive' alleles, so the possible genotypes of the hypergeometric polygenic model are taken to be `0:(2*n_loci)`. The transmission probabilities and genotype frequencies of the hypergeometric polygenic model approximate these quantities for the combination of the `n_loci` biallelic genetic loci described above. Under this model, the polygenic genotype for each person is approximately normally distributed, and these genotypes are correlated within families with correlation coefficients (in non-inbred families) equal to the kinship coefficients (Lange, 2002).

Setting `annotate` to `TRUE` names each element of the output vector with the corresponding genotype. The `annotate` option must be set to `FALSE` if the output of this function is to be used as the `geno_freq` argument of `pedigree_loglikelihood`.

Value

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

References

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

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

Examples

```
geno_freq_polygenic(4, annotate = TRUE)
sum(geno_freq_polygenic(4))
```

---

pedigree_loglikelihood

Calculate the log-likelihoods of pedigrees

Description

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

Usage

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

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 individual identifier for each person. If there are any duplicated identifiers in the dataset then the family and an underscore (_) will be prepended to all identifiers, and if any duplicates remain after this then the function will stop executing, with an error message.
- 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. If the possible genotypes of the underlying genetic model are 1:length(geno_freq) then geno_freq[j] is interpreted as the population frequency of genotype j, so geno_freq is essentially the function Prior in the pedigree likelihood on page 117 of (Lange, 2002). For certain genetic models that often occur in applications, these genotype frequencies can be calculated by geno_freq_monogenic, geno_freq_phased, etc.

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 correspond to offspring genotypes. If the possible genotypes are 1:length(geno_freq) then 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 on page 117 of (Lange, 2002). For certain genetic models that often occur in applications, this transmission matrix can be calculated by trans_monogenic, trans_phased, etc.

penet

An nrow(dat) by length(geno_freq) matrix of non-negative numbers. The element penet[i,j] is interpreted as the conditional probability (or probability density) of the phenotype of the person corresponding to row i of dat, given that his or her genotype is j (where the possible genotypes are 1:length(geno_freq)). Therefore, penet is essentially the penetrance function Pen on page 117 of (Lange, 2002). If any row of penet consists entirely of zeroes then the likelihood is 0, so the returned log-likelihood will be -Inf. Note that genotype data can be incorporated into penet by regarding observed genotypes as part of the phenotype, i.e. by regarding observed genotypes as (possibly noisy) measurements of the underlying true genotypes. For example, if the observed genotype of person i is 1 (and if genotype measurement error is negligible) then penet[i,j] should be 0 for j != 1 and penet[i,1] should be the same as if person i were ungenotyped.

monozyg

An optional list that can be used to specify genetically identical persons, such as monozygotic twins, monozygotic triplets, a monozygotic pair within a set of dizygotic triplets, etc. Each element of the list should be a vector containing the individual identifiers of a group of genetically identical persons, e.g. if dat contains six sets of monozygotic twins and one set of monozygotic triplets then monozyg will be a list with seven elements, one element a vector of length three and the other six elements all vectors of length two. The order of the list and the orders within its elements do not affect the output of the function. Each group of genetically identical persons should contain two or more persons, the groups should not overlap, and all persons in each group must have the same (non-missing) parents.

`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.

`load_balancing` A logical flag. When `ncores > 1`, parallelization is achieved either with the function `parallel::parLapply` (if `load_balancing` is `FALSE`) or with the load-balancing function `parallel::parLapplyLB` (if `load_balancing` is `TRUE`, the default). The load-balancing version will usually, but not always, be faster.

Details

This function provides a fast and general implementation of the Elston-Stewart algorithm to calculate the log-likelihoods of potentially large and complex pedigrees. 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 person should either have both parental identifiers missing (if a founder) or both non-missing (if a non-founder), and each (non-missing) mother or father should have a corresponding row of `dat`.

Observed genotypes should be incorporated into `penet`, as described above.

The function can handle pedigree loops, such as those caused by inbreeding or by two sisters having children with two brothers from an unrelated family (see (Totir et al., 2009) for a precise definition), though pedigrees with more than a few loops could greatly reduce the speed of the calculation.

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

Sex-specific genetics, such as X-linked genes or genetic loci with sex-specific recombination fractions, can be modelled by letting genotypes `1:nm` be the possible male genotypes and letting `(nm+1):(nm+nf)` be the possible female genotypes, where `nm` and `nf` are the number of possible genotypes for males and females, respectively. Then, for example, `penet[i,j]` will be `0` if `j %in% 1:nm` and row `i` of `dat` corresponds to a female, and `penet[i,j]` will be `0` if `j %in% (nm+1):(nm+nf)` and row `i` of `dat` corresponds to a male.

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

# In dat_small, ora024 and ora027 are identical twins, and so are aey063 and aey064
monozyg_small <- list(c("ora024", "ora027"), c("aey063", "aey064"))

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

# Calculate the log-likelihood for one family with approximately 10,000 family members
# Note:  this calculation should take less than a minute on a standard desktop computer
# Note:  parallelization would achieve 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_large[i,j] is the conditional probability of 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 10,002 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 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 single genetic locus*

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`, this function returns a matrix of genetic transmission probabilities, whose rows corresponding to the possible joint parental genotypes and whose 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 (which can be viewed by setting `annotate` to `TRUE`, see below) we can label the set of possible genotypes as `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, genotypes are written in the usual form `1/1, 1/2, ...` for the alleles `1:n_alleles`.

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)
```

trans_phased *The transmission matrix for phased genotypes*

Description

A function to calculate the transmission matrix for a single autosomal genetic locus with an arbitrary number of alleles and phased genotypes, based on Mendel's laws of inheritance. Phased genotypes can be used to investigate parent-of-origin effects, e.g. see (van Vliet et al., 2011).

Usage

```
trans_phased(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^2 possible phased genotypes, and by choosing an order on these genotypes (which can be viewed by setting annotate to TRUE, see below) we can label the set of possible phased genotypes as 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, phased genotypes are written in the usual form 1|1, 1|2, ... for the alleles 1:n_alleles, where a|b means the maternal allele is a and the paternal allele is b.

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`).

References

van Vliet CM, Dowty JG, van Vliet JL, et al. Dependence of colorectal cancer risk on the parent-of-origin of mutations in DNA mismatch repair genes. Hum Mutat. 2011;32(2):207-212.

Examples

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

`trans_polygenic` *The transmission matrix for the hypergeometric polygenic model*

Description

A function to calculate the transmission matrix for the hypergeometric polygenic model of (Cannings et al., 1978), see also Section 8.9 of (Lange, 2002) for a nice description of this model.

Usage

```
trans_polygenic(n_loci, annotate = FALSE)
```

Arguments

`n_loci`	A positive integer, interpreted as the number of biallelic genetic loci that contribute to the polygene. The polygene will have `2*n_loci + 1` genotypes, so `n_loci` is typically fairly small, e.g. `4`.
`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

This function calculates the genetic transmission probabilities (i.e. the conditional probability of a person's genotype, given his or her biological parents' genotypes) for the hypergeometric polygenic model, which is described in `geno_freq_polygenic`.

When `annotate` is `FALSE`, a matrix of transmission probabilities is returned, with rows corresponding to the possible joint parental genotypes and columns corresponding to the possible offspring genotypes. Setting annotate to TRUE shows which rows and columns correspond to which genotypes, by adding offspring genotypes as column names and adding columns gm and gf containing (respectively) the mother's and father's genotypes. 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).

References

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

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

Examples

```
trans_polygenic(4, annotate = TRUE)
apply(trans_polygenic(4), 1, sum)
```

Index