Package 'forensim'

May 5, 2009

Type Package

Title Statistical tools for the interpretation of forensic DNA mixtures

Version 1.0-0

Date 2009-01-19

Author Hinda Haned <haned@biomserv.univ-lyon1.fr>

Maintainer Hinda Haned <haned@biomserv.univ-lyon1.fr>

Depends methods

Description Statistical methods and simulation tools for the interpretation of forensic DNA mixtures

License (>=2)

LazyLoad yes

Collate classes_definitions.R classes_constructors.R accessors.R simufreqD.R simupopD.R zzz.R
AuxFunc.R changepop.R PE.R likelihood.R likestim.R mincontri.R

R topics documented:

---

forensim-package   *The forensim package*

Description

forensim is dedicated to the interpretation of forensic DNA mixtures through statistical methods. It relies on three S4 classes that facilitate the manipulation and the storage of genetic data produced in forensic casework: tabfreq, simugeno and simumix.

tabfreq objects are used to store allele frequencies, simugeno objects are used to store genotypes and simumix objects are used to store DNA mixtures.

For more informations about these classes type 'class ?tabfreq', 'class ?simugeno' and 'class ?simumix'.

Author(s)

Hinda Haned <haned@biomserv.univ-lyon1.fr>

Accessors   *Accessors for forensim objects*

Description

Accessors for forensim objects: simugeno, simumix and tabfreq. "$" and "$<-" are used to access the slots of an object, they are equivalent to "@" and "@<-".

Value

A simugeno, a simumix or a tabfreq object.

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

Examples

```
data(strusa)
class(strusa)

strusa@pop.names
#equivalent
strusa$pop.names
```

---

changepop	*Function to change population related informations in forensim objects*

Description

The `changepop` function changes population related informations in tabfreq, simugeno and simumix objects

Usage

```
changepop(obj, oldpop, newpop)
```

Arguments

obj   a forensim object, either a tabfreq, a simugeno or a simumix object

oldpop   a character vector giving the population names to be changed

newpop   a character vector giving the new population names

Value

a forensim object where the slots containing population related informations have been modified

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

Examples

```
data(strveneto)
tab1 <- simugeno(strveneto,n=100)
tab2 <- changepop(tab1,"Veneto","VENE")
tab1$pop.names
tab2$pop.names
```

`Cmn`	*The number of all possible combinations of m elements among n with repetitions*

Description

The number of all possible combinations of m elements among n with repetitions.

Usage

```
Cmn(m, n)
```

Arguments

`m`	the `m` elements to combine among `n`
`n`	the `n` elements from which to combine `m` elements with repetitions

Details

There are (n+m-1)!/(m!(n-1)!) ways to combine m elements among n with repetitions.

Note

Cmn was implemented as an auxiliary function for the `dataL` function which computes the likelihood of a mixture alleles conditional on the number of contributors.

Author(s)

Hinda Haned <haned@biomserv.univ-lyon1.fr>

See Also

`comb` for all possible combinations of m elements among n with repetitions

Examples

```
Cmn(2,3)
comb(2,3)
```

| comb | *Generate all possible combinations of m elements among n with repetitions* |

Description

Generate all possible combinations of m elements among n with repetitions.

Usage

```
comb(m, n)
```

Arguments

| m | the number of elements to combine |
| n | the number of elements from which to combine the `m` elements |

Details

There are (n+m-1)!/(m!(n-1)!) ways to combine m elements among n with repetitions, `combn` generates all these possible combinations.

Value

A matrix of (n+m-1)!/(m!(n-1)!) rows, and n columns, each row is a possible combination of m elements among n .

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

See Also

[Cmn](Cmn) for the calculation of the number of all possible combinations of m elements among n with repetitions

Examples

```
#combine 2 objcets among 3 with repetitions
Cmn(2,3)
comb(2,3)
```

---

dataL	*Generic formula of the likelihood of the observed alleles in a mixture conditional on the number of contributors for a specific locus*

Description

The function `dataL` gives the likelihood of a set of alleles observed at a specific locus conditional on the number of contributors that gave these alleles. Calculation is based upon the frequencies of the observed alleles.

Usage

```
dataL(x = 1, p, theta = 0)
```

Arguments

x	an integer giving the number of contributors
p	a numeric vector giving the frequencies of the observed alleles in the mixture
theta	a float in [0,1[. `theta` is equivalent to Wright's Fst. In case of population subdivision, it allows a correction of the allele frequencies in the subpopulation of interest

Note

`dataL` function has several similarities with the `Pevid.gen` function of the *forensic* package which gives the match probability of the evidentiary DNA profile with a suspect's profile, `dataL` implements a particular case of this probability. Please see http://cran.r-project.org/web/packages/forensic/

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

References

Haned H, Pontier D, Lobry J R, Pene L, Dufour AB. Estimating the number of contributors to forensic DNA mixtures: does maximizing the likelihood performs better than the maximum allele count? In prep, 2009.
Curran JM, Triggs CM, Buckleton J, Weir BS. Interpreting DNA Mixtures in Structured Populations. *J Forensic Sci* 1999;44(5): 987-995

See Also

`lik.loc` and `lik` for calculating the likelihood of a given simumix object

Examples

```
#likelihood of observing two alleles at frequencies 0.1 and 0.01 when the number of
#contributors is 2, in two cases:  theta=0 and theta=0.03
dataL(x=2,p=c(0.1,0.01), theta=0)
dataL(x=2,p=c(0.1,0.01), theta=0.03)
```

| findfreq | *Finds the frequencies of the alleles of a mixture from a tabfreq object* |

Description

The `findfreq` function finds the frequencies of the alleles of a mixture stored in a simumix object, form a given tabfreq object. If the tabfreq object contains multiple populations, a reference population from which to extract the frequencies must be specified.

Usage

```
findfreq(mix, freq, refpop = NULL)
```

Arguments

mix	a `simumix` object
freq	a `tabfreq` object from which to extract the alleles frequencies of the mixture
refpop	a factor giving the reference population in `tabfreq` from which to extract the allele frequencies

Value

A list giving the allele frequencies for each locus.

Author(s)

Hinda Haned <haned@biomserv.univ-lyon1.fr>

See Also

simumix

Examples

```
data(strusa)
s2<-simumix(simugeno(strusa,n=c(0,2000,0)),ncontri=c(0,2,0))
findfreq(s2,strusa,refpop="Cauc")
```

| findmax | *Function to find the maximum of a vector and its position* |

Description

The `findmax` function finds the maximum of a vector and its position.

Usage

```
findmax(vec)
```

Arguments

vec   a numeric vector

Details

`findmax` finds the maximum value of a vector and its position.

Value

A matrix of two rows:
`max` the position of the maximum in `vec`
`maxval` the maximum

Note

`findmax` is an auxiliary function for the `dataL` function, used to compute the likelihood of a mixture alleles given the number of contributors.

Author(s)

Hinda Haned <haned@biomserv.univ-lyon1.fr>

Examples

```
findmax(1:10)
```

---

lik	*Likelihood of the alleles observed at different loci in a DNA mixture conditional on the number of contributors to the mixture*

Description

The `lik` function computes the likelihood of the observed alleles in a forensic DNA mixture, for a set of loci, conditional on the number of contributors to the mixture. The overall likelihood is computed as the product of loci likelihoods.

Usage

```
lik(x = 1, mix, freq, refpop = NULL, theta = NULL, loc=NULL)
```

Arguments

x   the number of contributors to the DNA mixture, default is 1

mix   a `simumix` object which contains the mixture to be analysed

freq   a `tabfreq` object from which to extract the allele frequencies

refpop   a factor giving the reference population in `tabfreq` from which to extract the mixture allele frequencies. This argument is used only if `freq` contains allele frequencies for multiple populations, otherwise it is by default set to NULL

theta   a float from [0,1[giving Wright's Fst coefficient. `theta` accounts for population subdivision while computing the likelihood of the data

loc   loci for which the overall likelihood shall be computed. Default (NULL) corresponds to all loci

Details

`lik` computes the likelihood of the alleles observed at all loci conditional on the number of contributors. This function implements a particular case of the general formula of the match probability in the case of subdivided populations (Curran et al, 1999), in the particular case where all contributors are unknown.

Wright's Fst coefficient given in the `theta` argument allows accounting for population subdivision when all contributors belong to the same subpopulation.

The likelihood for multiple loci is computed as the product of loci likelihoods.

Author(s)

Hinda Haned <haned@biomserv.univ-lyon1.fr>

References

Haned H, Pontier D, Lobry J R, Pene L, Dufour AB. Estimating the number of contributors to forensic DNA mixtures: does maximizing the likelihood performs better than the maximum allele count? In prep, 2009.

Curran JM, Triggs CM, Buckleton J, Weir BS. Interpreting DNA Mixtures in Structured Populations. *J Forensic Sci* 1999;44(5): 987-995

See Also

`lik.loc` for the likelihood per locus, `likestim` and `likestim.loc` for the estimation of the number of contributors to a DNA mixture through likelihood maximization

Examples

```
data(strusa)
#simulation of 1000 genotypes from the African American allele frequencies
gen<-simugeno(strusa,n=c(1000,0,0))
#3-person mixture
mix3<-simumix(gen,ncontri=c(3,0,0))
sapply(1:3, function(i) lik(x=i,mix3, strusa, refpop="Afri"))
```

| `lik.loc` | *Likelihood per locus of the alleles observed in a DNA mixture conditional on the number of contributors to the mixture* |

Description

The `lik.loc` function computes the likelihood of the observed data in a forensic DNA mixture, for each of the loci involved, conditional on the number of contributors to the mixture.

Usage

```
lik.loc(x = 1, mix, freq, refpop = NULL, theta = NULL, loc=NULL)
```

Arguments

x	the number of contributors to the DNA mixture
mix	a `simumix` object which contains the mixture to be analysed
freq	a `tabfreq` object from which to extract the mixture allele frequencies
refpop	a factor giving the reference population in `tabfreq` from which to extract the mixture allele frequencies
theta	a float from [0,1[giving Wright's Fst coefficien. `theta` acounts for population subdivision while computing the likelihood of the data.
loc	the loci for which the likelihood shall be computed. Default (set to NULL) corresponds to all loci.

Details

`lik.loc` computes the likelihood per locus of the observed alleles. This functions implements a particular case of the general formula of the match probability in the case of subdivised populations (Curran et al, 1999), in the particular case where all contributors are unknown.

The Fst coefficient given in the `theta` argument allows accounting for population subdivision when all contributors belong to the same subpopulation.

Value

The function `lik.loc` returns a vector, of length the number of loci in `loc`, giving the likelihood of the data for each locus.

Author(s)

Hinda Haned <haned@biomserv.univ-lyon1.fr>

References

Haned H, Pontier D, Lobry J R, Pene L, Dufour AB. Estimating the number of contributors to forensic DNA mixtures: does maximizing the likelihood performs better than the maximum allele count? In prep, 2009.

Curran JM, Triggs CM, Buckleton J, Weir BS. Interpreting DNA Mixtures in Structured Populations. *J Forensic Sci* 1999;44(5): 987-995

See Also

`lik` for the overall loci likelihood, `likestim` and `likestim.loc` for the estimation of the number of contributors to a DNA mixture through likelihood maximization

Examples

```
data(strusa)
#simulation of 1000 genotypes from the Caucasian allele frequencies
gen<-simugeno(strusa,n=c(0,100,0))

#4-person mixture
mix4 <- simumix(gen,ncontri=c(0,4,0))
lik.loc(x=2,mix4, strusa, refpop="Cauc")
```

```
lik.loc(x=2,mix4, strusa, refpop="Afri")
#You may also want to try:
#likestim(mix4,strusa,refpop="Cauc")
```

---

likestim	*Maximum of likelihood estimation of the number of contributors to a forensic DNA mixture for a set of loci*

Description

The `likestim` function gives multiloci estimation of the number of contributors to a forensic DNA mixture using likelihood maximization.

Usage

```
likestim(mix, freq, refpop = NULL, theta = NULL, loc=NULL)
```

Arguments

mix	a `simumix` object
freq	a `tabfreq` object containing the allele frequencies to use for the likelihood calculation
refpop	the reference population form which to extract the allele frequencies used in the likelihood calculation. If `tabfreq` contains more than one population, `refpop` must be specified, otherwise, `refpop` is set to default (NULL).
theta	a float from [0,1[giving Wright's Fst coefficient. `theta` accounts for population subdivision while computing the likelihood of the data.
loc	loci to be considered in th estimation. Default (set to NULL) corresponds to all loci.

Details

The number of contributors which maximizes the likelihood of the data observed in the mixture is searched in the discrete interval [1,6]. In most cases this interval is a plausible range for the number of contributors.

Value

A matrix, the first row, `max`, gives the maximum likelihood estimation of the number of contributors, the second row gives the corresponding likelihood value `maxvalue`.

Author(s)

Hinda Haned <haned@biomserv.univ-lyon1.fr>

References

Haned H, Pontier D, Lobry J R, Pene L, Dufour AB. Estimating the number of contributors to forensic DNA mixtures: does maximizing the likelihood performs better than the maximum allele count? In prep, 2009.

Egeland T, Dalen I, Mostad PF. Estimating the number of contributors to a DNA profile. *Int J Legal Med* 2003, 117: 271-275
Curran JM, Triggs CM, Buckleton J, Weir BS. Interpreting DNA Mixtures in Structured Populations. *J Forensic Sci* 1999, 44(5): 987-995

See Also

`likestim.loc` for maximum of likelihood estimations per locus

Examples

```
data(strusa)
#simulation of 1000 genotypes from the Hispanic allele frequencies
gen<-simugeno(strusa,n=c(0,0,100))
#4-person mixture
mix4 <- simumix(gen,ncontri=c(0,0,4))
likestim(mix4,strusa,refpop="Hisp")
```

---

likestim.loc	*Maximum of likelihood estimation per locus of the number of contributors to forensic DNA mixtures.*

Description

The `likestim.loc` function returns the estimation of the number of contributors, at each locus, obtained by maximizing the likelihood.

Usage

```
likestim.loc(mix, freq, refpop = NULL, theta = NULL, loc = NULL)
```

Arguments

`mix`	a `simumix` object
`freq`	a `tabfreq` object containing the allele frequencies to use for the likelihood calculation
`refpop`	the reference population form which to extract the allele frequencies used in the likelihood calculation. Default set to NULL, if `tabfreq` contains more than one population, `refpop` must be specified
`theta`	a float from [0,1[giving Wright's Fst coefficient. `theta` acounts for population subdivision while computing the likelihood of the data.
`loc`	loci to be considered in th estimation. Default (set to NULL) corresponds to all loci.

Details

The number of contributors which maximizes the likelihood of the data observed in the mixture is searched in the discrete interval [1,6]. In most cases this interval is a plausible range for the number of contributors.

Value

A matrix of dimension 2 x `loc`. The first row, `max`, gives the maximum likelihood estimation of the number of contributors for each locus in column. The second row, `maxvalue`, gives the corresponding likelihood value.

Author(s)

Hinda Haned <haned@biomserv.univ-lyon1.fr>

References

Haned H, Pontier D, Lobry J R, Pene L, Dufour AB. Estimating the number of contributors to forensic DNA mixtures: does maximizing the likelihood performs better than the maximum allele count? In prep, 2009.

Egeland T , Dalen I, Mostad PF. Estimating the number of contributors to a DNA profile. *Int J Legal Med* 2003, 117: 271-275

Curran, JM , Triggs CM, Buckleton J , Weir BS. Interpreting DNA Mixtures in Structured Populations. *J Forensic Sci* 1999, 44(5): 987-995

See Also

likestim for overall loci estimations

Examples

```
data(strusa)
#simulation of 1000 genotypes from the Hispanic allele frequencies
gen<-simugeno(strusa,n=c(0,0,100))
#4-person mixture
mix4 <- simumix(gen,ncontri=c(0,0,4))
likestim.loc(mix4,strusa,refpop="Hisp")
```

---

mincontri	*Minimum of contributors required to explain a forensic DNA mixture*

Description

`mincontri` gives the minimum number of contributors required to explain a forensic DNA mixture. This method is also known as the maximum allele count as it relies on the maximum number of alleles showed through all available loci

Usage

```
mincontri(mix, loc = NULL)
```

Arguments

`mix`	a `simumix` object
`loc`	the loci to consider for the calculation of the minimum of contributors, default (NULL) corresponds to all loci

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

See Also

`likestim` for the estimation of the number of contributors through likelihood maximization

Examples

```
data(strusa)
#simulation of 1000 genotypes from the African American allele frequencies
gen<-simugeno(strusa,n=c(1000,0,0))
#3-person mixture
mix3<-simumix(gen,ncontri=c(3,0,0))
#compare
likestim(mix3, strusa, refpop="Afri")
mincontri(mix3)
```

---

naomitab	*Handling of missing values in a data frame*

Description

`naomitab` handles missing values (NA) in a data frame: it returns a list of the columns where NA's have been removed.

Usage

```
naomitab(tab)
```

Arguments

`tab`	a data frame

Value

Returns a list of length the number of columns in `tab` where each component is a column of `tab`, and the values are the corresponding rows where NA's have been removed.

Note

This function was designed to handle missing values in data frames in the format of the Journal of Forensic Sciences for population genetic data: allele names are given in the first column, and frequencies for a given allele are read in rows for different loci. When a given allele is not observed, the value is coded NA (originally coded "-" in the journal).

Author(s)

Hinda Haned <haned@biomserv.univ-lyon1.fr>

See Also

tabfreq

Examples

```
data(Tu)
naomitab(Tu)
```

---

nball   *Number of alleles in a mixture*

Description

`nball` gives the number of alleles of a `simumix` object.

Usage

```
nball(mix, byloc = FALSE)
```

Arguments

mix	a `simumix` object
byloc	a logical indicating whether the number of alleles must be calculated by locus or for all loci (default)

Author(s)

Hinda Haned <haned@biomserv.univ-lyon1.fr>

See Also

simumix

Examples

```
data(strusa)
#simulating 100 genotypes with allele frequencies from the   African American population
gaa<-simugeno(strusa,n=c(100,0,0))
#simulating a 2-person mixture
maa2<-simumix(gaa,ncontri=c(2,0,0))
nball(maa2,byloc=TRUE)
```

PE   *Exclusion probability*

Description

Computes the exclusion probability of a mixture stored in a `simumix` object

Usage

```
PE(mix, freq, refpop = NULL, theta = 0, byloc = FALSE, digits = 2)
```

Arguments

`mix`	a `simumix` object
`freq`	a `tabfreq` object giving the allele frequencies from which to compute the exclusion probability
`refpop`	character giving the reference population, used only if `freq` contains allele frequencies for multiple populations
`theta`	a float from [0,1[giving Wright's Fst coefficient. `theta` accounts for population subdivision while computing the likelihood of the data.
`byloc`	logical, if TRUE, than the exclusion probability is computed per locus, if FALSE (default), the calculations are done for all loci simultaneously
`digits`	number of digits to display in the exclusion probability, default is 2

Details

`PE` gives the exclusion probability at a locus, or at several loci when conditions for Hardy Weinberg are met. If this condition is not met in the population, than a value for `theta` must be supplied to take into account dependencies between alleles. The formula of the exclusion probability that allows taking into account departure from Hardy Weinberg proportions due to population subdivision was provided by Bruce Weir, please see the references section.

Author(s)

Hinda Haned <haned@biomserv.univ-lyon1.fr>

References

Clayton T, Buckleton JS. Mixtures. In: Buckleton JS, Triggs CM, Walsh SJ, editors. Forensic DNA Interpretation. CRC Press 2005;217-74

Examples

```
data(strusa)
geno1<-simugeno(strusa,n=c(0,0,100))
mix2 <-simumix(geno1,ncontri=c(0,0,2))
PE(mix2,strusa,"Hisp",byloc=TRUE)
```

simufreqD	*Function to simulate allele frequencies for independent loci from a Dirichlet model*

Description

The `simufreqD` function simulate single population allele frequencies for independent loci. Allele frequencies are generated as random deviates from a Dirichlet distribution, which parameters control the mean and the variance of the simulated allele frequencies.

Usage

```
simufreqD(nloc = 1, nal = 2, alpha = 1)
```

Arguments

nloc   the number of loci to simulate

nal   the number of alleles per locus. Either an integer, if the loci have the same number of alleles, or an integer vector, if the number of alleles differ between loci

alpha   the parameter used to simulate allele frequencies from the Dirichlet distribution. If loci have the same allele number, `alpha` can either be the same for all alleles (default is one: uniform distribution), in this case `alpha` is an integer, or `alpha` can be different between alleles at a given locus, in this case, `alpha` is a matrix of dimension `nal` x `nloc`.

When the number of alleles differ between loci, `alpha` can either be the same or differ between alleles at a given locus. In the first case `alpha` is a vector of length `nloc`, in the second case, `alpha` is a matrix of dimensions `nal` x `loc` where NA's are introduced for alleles not seen at a given locus.

Details

Allele frequencies for independent loci are simulated using a Dirichlet distribution with parameter `alpha`. At a given locus L with n alleles, the allele frequencies are modelled as a vector of random variables p=(p1, ..., pn), following a Dirichlet distribution with parameters:
alpha = (alpha1, ..., alphan) where p1+...+pn=1 and alpha1,..., alphan > 0.

Value

A matrix containing the simulated allele frequencies. The data is presented in the format of the Journal of Forensic Sciences for genetic data: allele names are given in the first column, and frequencies for a given allele are read in rows for the different markers. When a given allele is not observed, the value is coded NA (instead of "-" in the original format).

Note

The code used here for the generation of random Dirichlet deviates was previously implemented in the gtools library.

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

References

Kotz and Balakrishnan. Continuous Univariate Distributions, vol. 2. New York, NY 1995: John Wiley

Wright S. The genetical structure of populations. Ann Eugen 1951;15:323-354.

See Also

`simupopD`

Examples

```
#simulate alleles frequencies for 5 markers with respectively 2, 3, 4, 5, and 6 alleles

simufreqD(nloc=5,na=c(2,3,4,5,6) , alpha=1)
```

---

  simugeno   *forensim class for simluated genotypes*

Description

The S4 `simugeno` class is used to store existing or simulated genotypes.

Slots

tab.freq: a list giving allele frequencies for each locus. If there are several populations, `tab.freq` gives allele frequencies in each population

nind: integer vector giving the number of individuals. If there are several populations, `nind` gives the number of individuals per population

pop.names: factor of populations names

popind: factor giving the population of each individual

which.loc: character vector giving the locus names

tab.geno: matrix giving the genotypes (in rows) for each locus (in columns). The genotype of a homozygous individual carrying the allele "12" is coded "12/12". A heterozygous individual carrying alleles "12" and "13" is coded "12/13" or "13/12".

indID: charcter vector giving the individuals ID's

Methods

names signature(x = "simugeno"): gives the names of the attributes of a simugeno object

show signature(object = "simugeno"): shows a simugeno object

print signature(object = "simugeno"): prints a simugeno object

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

See Also

`as.simugeno` for the simugeno class constructor, `is.simugeno`, `simumix` and `tabfreq`

Examples

```
showClass("simugeno")
```

```
simugeno constructor
```
simugeno constructor

Description

Constructor for simugeno objects.
The function simugeno creates a simugeno object from a tabfreq object.

The function as.simugeno is an alias for simugeno function.

is.simugeno tests if an object is a valid simugeno object.

Note: to get the manpage about simugeno, please type 'class ? simugeno'.

Usage

```
simugeno(tab,which.loc=NULL,n=1)
as.simugeno(tab,which.loc=NULL,n=1)
is.simugeno(x)
```

Arguments

tab	a tabfreq object created with constructor tabfreq
which.loc	a character vector giving the chosen loci for the genotypes simulation. The default is set to NULL, which corresponds to all the loci of the tabfreq object given in argument
n	integer vector giving the number of individuals. If there are several populations, n gives the number individuals to simulate per population. For a single population, default is 1.
x	an object

Value

For simugeno and as.simugeno, a simugeno object. For is.simugeno, a logical.

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

See Also

"simugeno", and tabfreq for creating a tabfreq object from a data file.

Examples

```
data(Tu)
tab<-tabfreq(Tu)
#simulation of 3 individual genotypes for the str  marker FGA
geno1 <- simugeno(tab,which.loc='FGA', n =1000)
geno1@tab.geno
```

---

simumix   *forensim class for DNA mixtures*

Description

The S4 `simumix` class is used to store DNA mixtures of individual genotypes along with informations about the individuals poulations and the loci used to simulate the genotypes.

Slots

ncontri: integer vector giving the number of contributors to the DNA mixture. If there are several populations, `ncontri` gives the number of contributors per population

mix.prof: matrix giving the contributors genotypes (in rows) for each locus (in columns). The genotype of a homozygous individual carrying the allele "12" is coded "12/12". A heterozygous individual carrying alleles "12" and "13" is coded "12/13" or "13/12".

mix.all: list giving the alleles present in the mixture for each locus

which.loc: character vector giving the locus names

popinfo: factor giving the population of each contributor

Methods

names signature(x = "simumix"): gives the names of the attributes of a simumix object

show signature(object = "simumix"): shows a simumix object

print signature(object = "simumix"): prints a simumix object

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

See Also

`simugeno`, `as.simumix`, `is.simumix`, `simugeno` and `tabfreq`

Examples

```
showClass("simumix")
data(strusa)
```

simumix constructor

simumix constructor

Description

Constructor for simumix objects.
The function `simumix` creates a simumix object from a tabfreq object.

The function `as.simumix` is an alias for `simumix` function.

`is.simumix` tests if an object is a valid simumix object.

Note: to get the manpage about simumix, please type 'class ? simumix'.

Usage

```
simumix(tab,which.loc=NULL,ncontri=1)
as.simumix(tab,which.loc=NULL,ncontri=1)
is.simumix(x)
```

Arguments

tab	a simugeno object created with constructor simugeno
which.loc	a character vector giving the chosen loci for the genotypes simulation. The default is set to NULL, which corresponds to all the loci of the `simugeno` object given in argument
ncontri	integer vector giving the number of individuals. If there are several populations, `ncontri` gives the number of individuals to simulate per population. Default is one.
x	an object

Value

For `simumix` and `as.simumix`, a simumix object. For `is.simumix`, a logical.

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

See Also

"simumix", simugeno for creating a simugeno object.

Examples

```
data(Tu)
tab<-simugeno(tabfreq(Tu),n=1200)
#simulation of 3 individual mixture with markers 'FGA', 'TH01' and 'TPOX'
simumix(tab,which.loc=c('FGA','TH01', 'TPOX') , n =3)
```

| simupopD | *Simulate multi-population allele frequencies for independent loci from a reference population, following a Dirichlet model* |

Description

Simulate multi-population allele frequencies for independent loci, from a given reference population, fallowing a Dirichlet model. Allele frequencies in the populations are generated as random deviates from a Dirichlet distribution, which parameters control the deviation of allele frequencies from the avreage values in the refrence population.

Usage

```
simupopD(npop = 1, nloc = 1, na = 2, globalfreq = NULL, which.loc = NULL, alpha1
```

Arguments

npop	the number of populations
nloc	the number of loci
na	an integer vector giving the number of alleles per locus
globalfreq	matrix of allele frequencies in the reference population. Data must be given in the format of the Journal of Forensic Sciences for genetic data. Default corresponds to allele frequencies generated form a Dirichlet distribution with parameter `alpha2` for all allele frequencies.
which.loc	which loci to simulate from the `globalfreq` matrix, default considers all loci
alpha1	a postive float vector of length `npop` giving the variance parameter of the Dirichlet distribution used to generate allele frequencies in the `npop` independent populations
alpha2	a positive float giving the parameter to be used to in the Dirichlet distribution to generate allele frequencies for the reference population

Details

In the reference population, allele frequencies for independent loci are simulated using a Dirichlet distribution with parameter `alpha2`.

At a given locus L with n alleles, the allele frequencies are modelled as a vector of random variables p=(p1, ..., pn) following a Dirichlet distribution with a parameter vector of length n, where each component is equal to alpha2, p1+...+pn=1 and alpha2 > 0.

Note that a more sophisticated generation of global allele frequencies is possible using the `simufreqD` function.

Similarly, allele frequencies in the independent populations are simulated using a Dirichlet Distribution. For example, for the first population to simualte, at a given locus L with n alleles, the allele frequencies are modelled as a vector of random variables p=(p1, ..., pn) following a Dirichlet distribution with a parameter vector of length n:

(p1(1-a1)/alpha1[1], ..., pn(1-alpha1[1])/alpha1[1]), where p1+...+pn=1 and alpha1[1] > 0 .

Value

The result is stored in a list with two elements :

globfreq   a `tabfreq` object giving the allele frequencies of the chosen reference population, with the chosen loci.

popfreq   a `tabfreq` object giving the allele frequencies of the simulated populations.

Note

The code used here for the generation of random Dirichlet deviates was previously implemented in the gtools library.

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

References

Nicholson G, Smith AV, Jonsson F, Gustafsson O, Stefansson K, Donnelly P. Assessing population differentiation and isolation from single-nucleotide polymorphism data. *J Roy Stat Soc B* 2002;64:695–715

Marchini J and Cardon LR. Discussion of Nicholson et al. *J Roy Stat Soc B* 2002;64: 740–741

Wright S. The genetical structure of populations. *Ann Eugen* 1951;15:323-354

See Also

simufreqD

Examples

```
# simulate allele frequencies for two populations
data(Tu)
simupopD(npop=2,globalfreq=Tu, which.loc=c("FGA","TH01","TPOX"),alpha1=c(0.2,0.3),alpha2=
```

---

strusa   *Allele frequencies for 15 autosomal short tandem repeats core loci on U.S. Caucasian, African American, and Hispanic populations.*

Description

Allele frequencies for 15 autosomal short tandem repeats loci on three American populations : Caucasians, African Americans and Hispanics. Among the 15 loci, 13 belong to the core Combined DNA Index System (CODIS) loci used by the Federal Bureau of Investigation (USA), in forensic DNA analysis, and two supplementary loci are more commonly used in Europe, see details.

Usage

```
data(strusa)
```

Format

`strusa` is a tabfreq object giving allele frequencies of 15 lociin three American populations.

Details

CSF1PO, FGA, TH01, TPOX, vWA, D3S1358, D5S818, D7S820, D8S1179, D13S317, D16S539, D18S51 and D21S11, belong to the core CODIS loci used in the US, whereas D2S1338 and D19S433 belong to the European core loci.

References

Butler JM, Reeder DJ. http://www.cstl.nist.gov/strbase/index.htm, last visited: March 4th 2009
Butler JM, Schoske R, Vallone MP, Redman JW, Kline MC. Allele frequencies for 15 autosomal STR loci on U.S. Caucasian, African American, and Hispanic populations. *J Forensic Sci* 2003;48(8):908-911.

Examples

```
data(strusa)
strusa
#genotypes simulations from each population
geno<- simugeno(strusa,n=c(100,100,100))
geno
#3-person mixture simulation with the contributors from the three different populations
mix3<- simumix(geno,ncontri=c(1,1,1))
mix3
```

---

strveneto   *Population study of three miniSTR loci in Veneto (Italy)*

Description

Allele frequencies for three short tandem repeats loci D10S1248, D2S441 and D22S1045 in a sample of 198 individuals born in Veneto, Italy. These loci are commonly used in forensic DNA characterization.

Usage

```
data(strveneto)
```

Format

`strveneto` is a tabfreq object

References

Turrina S, Atzei R, De Leo D. Population study of three miniSTR loci in Veneto (Italy). Forensic Sci Int Genetics 2008; 1(1);378-379

Examples

```
data(strveneto)
strveneto
```

`tabfreq`	*forensim class for population allele frequencies*

Description

The S4 `tabfreq` class is used to store allele frequencies, from either one or several populations.

Slots

`tab`: a list giving allele frequencies for each locus. If there are several populations, `tab` gives allele frequencies in each population

`which.loc`: character vector giving the names of the loci

`pop.names`: factor of populations names (optional)

Methods

names signature(x = "tabfreq"): gives the names of the attributes of a tabfreq object

show signature(object = "tabfreq"): shows a tabfreq object

print signature(object="tabfreq"): prints a tabfreq object

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

See Also

`as.tabfreq`, `is.tabfreq` and `simugeno` for genotypes simulation from allele frequencies stored in a `tabfreq` object

Examples

```
showClass("tabfreq")
```

tabfreq constructor	
	tabfreq constructor

Description

Constructor for tabfreq objects.
The function `tabfreq` creates a tabfreq object from a data frame or a matrix giving allele frequencies for a single population in the Journal of Forensic Sciences (JFS) format for population genetic data. Whene multiple populations are considered, data shall be given as a list, where each element is either a matrix or a data frame in the JFS format, and the populations names must be specified.

The function `as.tabfreq` is an alias for the `tabfreq` function.

`is.tabfreq` tests if an object is a valid tabfreq object.

Note: to get the manpage about tabfreq, please type 'class ? tabfreq'.

Usage

```
tabfreq(tab,pop.names=NULL)
as.tabfreq(tab,pop.names=NULL)
is.tabfreq(x)
```

Arguments

tab   either a matrix or a data.frame of markers allele frequencies given in the Journal
               of Forensic Sciences format for population genetic data

pop.names   (optional) a factor giving the populations names. For a single population in `tab`,
               default is set to NULL.

x   an object

Value

For `tabfreq` and `as.tabfreq`, a tabfreq object. For `is.tabfreq`, a logical.

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

See Also

"`tabfreq`", `simugeno` for creating a simugeno object from a tabfreq object.

Examples

```
data(Tu)
tabfreq(Tu,pop.names=factor("Tu"))
```

Tu   *Allele frequencies of 15 autosomal short tandem repeats loci on Chi-*
                                nese Tu ethnic minority group

---

Description

Population genetic analysis of 15 STR loci of Chinese Tu ethnic minority group.

Usage

```
data(Tu)
```

Format

a data frame presented in the format of the Journal of Forensic Sciences for genetic data: allele
names are given in the first column, and frequencies for a given allele are read in rows for the
different markers. When a given allele is not observed, value is coded NA (rather than "-" in the
original format).

Details

CSF1PO, FGA, TH01, TPOX, vWA, D3S1358, D5S818, D7S820, D8S1179, D13S317, D16S539, D18S51 and D21S11, belong to the core CODIS loci used in the US, whereas D2S1338 and D19S433 belong to the European core loci.

References

Zhu B, Yan J, Shen C, Li T, Li Y, Yu X, Xiong X, Muf H, Huang Y, Deng Y. (2008). Population genetic analysis of 15 STR loci of Chinese Tu ethnic minority group. *Forensic Sci Int*; 174: 255-258.

Examples

```
data(Tu)
tabfreq(Tu)
```

---

virtualClasses   *Virtual classes for forensim*

Description

Virtual classes that are only for internal use in forensim

Objects from the Class

A virtual Class: programming tool, not intended for objects creation.

Author(s)

Hinda Haned ⟨haned@biomserv.univ-lyon1.fr⟩

Index