Package 'forensim'

June 1, 2010

Type Package

Title Statistical tools for the interpretation of forensic DNA mixtures

Version 1.1-5

Date 2010-05-25

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

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

Suggests gdata,gtools,MASS,mvtnorm,genetics,tcltk,tkrplot

Depends methods

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

License GPL (>= 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 Pevid2.R LR.R RMP.R A2.simu.R A3.simu.R A4.simu.R mastermix.R N2Exact.R N2error.R simMixSNP.R wrapdataL.R recordDrop.R DNAproxy.R tabDNAproxy.R recordHeights.R tabSPH.R simPCR2.R simPCR2TK.R PV.R Hbsimu.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 information about these classes type 'class ?tabfreq', 'class ?simugeno' and 'class ?simumix'.

Author(s)

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

`A2.simu`	*A Tcl/Tk graphical user interface for simple DNA mixtures resolution using allele peak heights or areas information when two alleles are observed at a given locus*

Description

The `A2.simu` function launches a Tcl/Tk graphical interface with functionalities devoted to two-person DNA mixtures resolution, when two alleles are observed at a given locus.

Usage

```
A2.simu()
```

Details

When two alleles are observed at a given locus in the DNA stain, seven genotype combinations are possible for the two contributors: (AA,AB), (AB,AB), (AA,BB), (AB,AA), (BB,AA), (AB,BB) and (BB,AB), where A and B are the two observed alleles (in ascending order of molecular weight). Having previously obtained an estimation for the mixture proportion, it is possible to reduce the number of possible genotype combinations by keeping those only supported by the observed data. This is achieved by computing the sum of square differences between the expected allelic ratio and the observed allelic ratio, for all possible mixture combinations. The likelihood of peak heights (or areas), given the combination of genotypes, is high if the residuals are low. Genotype combinations are thus selected according to the peak heights with the highest likelihoods.

The `A2.simu()` function launches a dialog window with three buttons:
-`Plot simulations`: plot of the residuals of each possible genotype combination for varying values of the mixture proportion across the interval [0.1, 0.9]. The observed mixture proportion is also reported on the plot.
-`Simulation details`: a matrix containing the simulation results. Simulation details and genotype combinations with the lowest residuals can be saved as a text file by clicking the "Save" button. It is also possible to choose specific paths and names for the save files.
-`Genotypes filter`: a matrix giving the mixture proportion conditional on the genotype combination. This conditional mixture proportion helps filter the most plausible genotypes among the seven possible combinations. The matrix can be saved as a text file by clicking the "Save" button. It is also possible to choose a specific path and a name for the save file.

Note

-Linux users may have to download the `libtktable` package to their system before using the `A2.simu` function. This is due to the `Tktable` widget, used in `forensim`, which is not (always) downloaded with the Tcl/Tk package.
-For the computational details, please see forensim tutorial at http://forensim.r-forge.r-project.org/misc/forensim-tutorial.pdf.

Author(s)

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

References

Gill P, Sparkes P, Pinchin R, Clayton, Whitaker J, Buckleton J. Interpreting simple STR mixtures using allele peak areas. *Forensic Sci Int* 1998;91:41-53.

See Also

`A3.simu`: the three-allele model, and `A4.simu`: the four-allele model

Examples

```
A2.simu()
```

`A3.simu`	*A Tcl/Tk graphical user interface for simple DNA mixtures resolution using allele peak heights or areas when three alleles are observed at a given locus*

Description

The `A3.simu` function launches a Tcl/Tk graphical interface with functionalities devoted to two-person DNA mixtures resolution, when three alleles are observed at a given locus.

Usage

```
A3.simu()
```

Details

When three alleles are observed at a given locus in the DNA stain, twelve genotype combinations are possible for the two contributors: (AA,BC), (BB,AC), (CC,AB), (AB,AC), (BC,AC), (AB,BC), (BC,AA), (AC,BB), (AB,CC), (AC,AB), (AC,BC) and (BC,AB) where A, B and C are the three observed alleles (in ascending order of molecular weights). Having previously obtained an estimation for the mixture proportion, it is possible to reduce the number of possible genotype combinations by keeping those only supported by the observed data. This is achieved by computing the sum of square differences between the expected allelic ratio and the observed allelic ratio, for all possible mixture combinations. The likelihood of peak heights (or areas), given the combination of genotypes, is high if the residuals are low. Genotype combinations are thus selected according to the peak heights with the highest likelihoods.

The `A3.simu()` function launches a dialog window with three buttons:
-`Plot simulations`: plot of the residuals of each possible genotype combination for varying values of the mixture proportion across the interval [0.1, 0.9]. The observed mixture proportion is also reported on the plot.
-`Simulation details`: a matrix containing the simulation results. Simulation details and genotype combinations with the lowest residuals can be saved as a text file by clicking the "Save" button. It is also possible to choose specific paths and names for the save files.

-`Genotypes filter`: a matrix giving the mixture proportion conditional on the genotype combination. This conditional mixture proportion helps filter the most plausible genotypes among the twelve possible combinations. The matrix can be saved as a text file by clicking the "Save" button. It is also possible to choose a specific path and a name for the save file.

Note

-Linux users may have to download the `libtktable` package to their system before using the `A3.simu` function. This is due to the `Tktable` widget, used in `forensim`, which is not (always) downloaded with the Tcl/Tk package.
-For the computational details, please see forensim tutorial at http://forensim.r-forge.r-project.org/misc/forensim-tutorial.pdf.

Author(s)

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

References

Gill P, Sparkes P, Pinchin R, Clayton, Whitaker J, Buckleton J. Interpreting simple STR mixtures using allele peak areas. *Forensic Sci Int* 1998;91:41-53.

See Also

[A2.simu](): the two-allele model, and [A4.simu](): the four-allele model

Examples

```
A3.simu()
```

| A4.simu | *A Tcl/Tk graphical user interface for simple DNA mixtures resolution using allele peak heights or areas when four alleles are observed at a given locus* |

Description

The `A4.simu` function launches a Tcl/Tk graphical interface with functionalities devoted to two-person DNA mixtures resolution, when four alleles are observed at a given locus.

Usage

```
A4.simu()
```

Details

When four alleles are observed at a given locus in the DNA stain, six genotype combinations are possible for the two contributors: (AB,CD),(AC,BD),(AD,BC),(BC,AD),(BD,AC) and (CD,AB) where A, B, C and D are the four observed alleles (in ascending order of molecular weights). Having previously obtained an estimation for the mixture proportion, it is possible to reduce the number of possible genotype combinations by keeping those only supported by the observed data. This is achieved by computing the sum of square differences between the expected allelic ratio and the observed allelic ratio, for all possible mixture combinations. The likelihood of peak heights (or areas), given the combination of genotypes, is high if the residuals are low. Genotype combinations are thus selected according to the peak heights with the highest likelihoods.

The `A4.simu()` function launches a dialog window with three buttons:
-`Plot simulations`: plot of the residuals of each possible genotype combination for varying values of the mixture proportion across the interval [0.1, 0.9]. The observed mixture proportion is also reported on the plot.
-`Simulation details`: a matrix containing the simulation results. Simulation details and genotype combinations with the lowest residuals can be saved as a text file by clicking the "Save" button. It is also possible to choose specific paths and names for the save files.
-`Genotypes filter`: a matrix giving the mixture proportion conditional on the genotype combination. This conditional mixture proportion helps filter the most plausible genotypes among the six possible combinations. The matrix can be saved as a text file by clicking the "Save" button. It is also possible to choose a specific path and a name for the save file.

Note

-Linux users may have to download the `libtktable` package to their system before using the `A4.simu` function. This is due to the `Tktable` widget, used in `forensim`, which is not (always) downloaded with the Tcl/Tk package.
-For the computational details, please see forensim tutorial at http://forensim.r-forge.r-project.org/misc/forensim-tutorial.pdf.

Author(s)

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

References

Gill P, Sparkes P, Pinchin R, Clayton, Whitaker J, Buckleton J. Interpreting simple STR mixtures using allele peak areas. *Forensic Sci Int* 1998;91:41-53.

See Also

`A2.simu`: the two-allele model, and `A3.simu`: the three-allele model

Examples

```
A4.simu()
```

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 information in forensim objects*

Description

The `changepop` function changes population-related information 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 information 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 the observed alleles in a mixed DNA stain 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](#) for the calculation of the number of all possible combinations of m elements among n with repetitions

Examples

```
#combine 2 objects 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 computes the probability of the DNA evidence, `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, Pene L, Lobry JR, Dufour AB, Pontier D. Estimating the number of contributors to forensic DNA mixtures: Does maximum likelihood perform better than maximum allele count? *J Forensic Sci*, accepted 2010.

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)
```

`DNAproxy`	*Approximation of the amount of DNA contributed by a person based on the observed peak heights of the alleles present in the analyzed sample*

Description

`DNAproxy` gives an estimation of the amount of DNA contributed by a person to a DNA stain based on the observed peak heights of the present alleles. The estimation is performed using data across all available loci, data can either consist of single-contributor or mixed DNA stains. The computation of the DNA proxies from experimental data are described by Tvedebrink et al. (cf. the references sections).

Usage

```
DNAproxy(tab, x)
```

Arguments

`tab`	a table produced by the `recordDrop` function, giving the allelic dropouts observations and the corresponding allelic peak heights
`x`	a character giving the label of the individual for whom the DNA proxy must be specified, this argument is to be specified only when data in `tab` is made of mixtures. In case data is consist of single-contributor stains, the argument must be left empty. `x` must match the name given in the `tab` table.

Note

`DNAproxy` is an auxiliary function of the `tabDNAproxy` function that implements the methodology proposed by Tvedebrink et al. to estimate the probability of allelic dropout using experimental DNA mixtures.

Author(s)

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

References

Tvedebrink T, Eriksen PS, Mogensen HS, Morling N. Estimating the probability of allelic drop-out of STR alleles in forensic genetics. *Forensic Science International: Genetics*, 2009, 3(4), 222-226.

See Also

`recordDrop`, `tabDNAproxy`

Examples

```
#load the exemple data
data(dropdata)
tabcsv<-dropdata$tabcsv
genot<-dropdata$genot
#individuals' labels are 1 and 2
```

```
#DNA proxy for individual one, when data is composed of a 2-person mixture
DNAproxy(recordDrop(1,2,geno=genot,tabcsv=tabcsv),"c1")
```

---

dropdata	*Dropout example data*

Description

`dropdata` gives is an extract of a series of experiments used to determine the probability of dropout

Usage

```
data(dropdata)
```

Format

A list of two components: 'tabcsv' and 'genot'. `tabcsv` is an extract of the validation table of a two-person mixture (Genemapper format) and `genot` is the matrix of genotypes of the individuals contributing to the mixture.

Details

The mixture is characterized using the Applied Biosystems AmpFlSTR Identifiler^TM kit.

Source

Data communicated by Elodie Suzanne and Laurent P\'ene, Laboratoire de Police Scientifique, Ecully, France.

Examples

```
data(dropdata)
names(dropdata)
dropdata$tabcsv
dropdata$genot
```

---

findfreq	*Finds the allele frequencies of a mixture from a tabfreq object*

Description

The `findfreq` function finds the allele frequencies 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 allele 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 columns:
`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 the observed alleles in a mixed DNA stain given the number of contributors.

Author(s)

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

Examples

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

Hbsimu   *A Tcl/Tk simulator of the heterozygous balance*

Description

`Hbsimu` is a user-friendly graphical interface simulating the heterozygous balance of heterozygous profiles generated according to the simulation model described in Gill et al. (2005)

Usage

```
Hbsimu()
```

Author(s)

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

References

Gill P, Curran J and Elliot K. A graphical simulation model of the entire DNA process associated with the analysis of short tandem repeat loci. *Nucleic Acids Research* 2005, 33(2): 632<96>643.

Examples

```
Hbsimu()
```

---

lik | *Likelihood of the observed alleles 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 analyzed

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 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 the general formula for the interpretation of DNA mixtures in case of population subdivision (Curran et al, 1999), in the particular case where all contributors are unknown and 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, Pene L, Lobry JR, Dufour AB, Pontier D. Estimating the number of contributors to forensic DNA mixtures: Does maximum likelihood perform better than maximum allele count? *J Forensic Sci*, accepted 2010.

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 observed alleles 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 analyzed
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 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 function implements the general formula for the interpretation of DNA mixtures in case of subdivided populations (Curran et al, 1999), in the particular case where all contributors are unknown and belong to the same sub-population.
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, Pene L, Lobry JR, Dufour AB, Pontier D. Estimating the number of contributors to forensic DNA mixtures: Does maximum likelihood perform better than maximum allele count? *J Forensic Sci*, accepted 2010.

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 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 calculation
refpop	the reference population from 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 the 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 1 x 2, the first column, `max`, gives the maximum likelihood estimation of the number of contributors, the second column gives the corresponding likelihood value `maxvalue`.

Author(s)

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

References

Haned H, Pene L, Lobry JR, Dufour AB, Pontier D. Estimating the number of contributors to forensic DNA mixtures: Does maximum likelihood perform better than maximum allele count? *J Forensic Sci*, accepted 2010.

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 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 calculation
`refpop`	the reference population from 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 the 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 `loc` x 2. The first colum, `max`, gives the maximum likelihood estimation of the number of contributors for each locus in row. The second column, `maxvalue`, gives the corresponding likelihood value.

Author(s)

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

References

Haned H, Pene L, Lobry JR, Dufour AB, Pontier D. Estimating the number of contributors to forensic DNA mixtures: Does maximum likelihood perform better than maximum allele count? *J Forensic Sci*, accepted 2010.

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 multiloci 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")
```

LR   *Likelihood ratio for DNA evidence interpretation*

Description

The `LR` function calculates the likelihood ratio for a DNA evidence, when two competing hypotheses Hd and Hp, respectively the defence and the prosecution hypotheses, are weighted about the origin of the DNA evidence. The evidence can either be a simple or a mixed stain.

Usage

```
LR(stain, freq, xp=0, xd=0, Tp=NULL, Vp=NULL, Td=NULL, Vd=NULL, theta=0)
```

Arguments

stain	a vector giving the set of (distinct) alleles present in the DNA stain
freq	vector of the corresponding allele frequencies in the global population
xp	the number of unknown contributors to the stain under the prosecution hypothesis Hp. Default is 0.
xd	the number of unknown contributors to the stain under the defence hypothesis Hd. Default is 0.
Tp	a vector of strings where each string contains two alleles separated by '/', corresponding to one known contributor under the prosecution hypothesis Hp. The length of the vector equals the number of known contributors. Default is NULL.
Vp	a vector of strings where each string contains two alleles separated by '/', corresponding to one known non-contributor under the prosecution hypothesis Hp. The length of the vector equals the number of known non-contributors. Default is NULL.
Td	a vector of strings where each string contains two alleles separated by '/', corresponding to one known contributor under the defence hypothesis Hd. The length of the vector equals the number of known contributors. Default is NULL.

Vd	a vector of strings where each string contains two alleles separated by '/', corresponding to one known non-contributor under the defence hypothesis Hd. The length of the vector equals the number of known non-contributors. Default is NULL.
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

Details

`LR` is the implementation of the general formula of Curran et al (1999) for the evaluation of forensic DNA mixtures through likelihood ratios. The likelihood ratio is computed as a ratio of two probabilities of the DNA evidence, E, conditional on the evaluated hypotheses:

$$LR = \frac{P(E|H_p)}{P(E|H_d)},$$

where H_p denotes the prosecution hypothesis and H_d the defence hypothesis.

In case of population subdivision, contributors to the DNA stain are considered to come from the same subpopulation. Allele dependencies within subpopulations are accounted for through Wright's Fst coefficient, denoted here θ.

Note

Please note that the `LR` function is based on functions initially implemented in the forensic package by Miriam Marusiakova http://cran.r-project.org/web/packages/forensic/

Author(s)

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

References

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

See Also

the exclusion probability PE.

Examples

```
# A rape case in Hong Kong (Hu and Fung, Int J Legal Med 2003)
# The stain shows alleles 14, 15, 17 and 18 at locus D3S1358.
stain =c(14,15,17,18)
# suspect's profile: "14/17"
suspect<-"14/17"
# victim's profile: "15/18"
victim<-"15/18"
# corresponding allele frequencies
freq<-c(0.033,0.331,0.239,0.056)

# Prosecution hypothesis: Contributors were the victim and the suspect
# defence hypothesis: Contributors were the victim and 1 unknown contributor
# Likelihood ratios for DNA evidence for different alternatives:
```

```
LR(stain,freq,xp=0,Tp=c(victim,suspect),Vp=NULL,Td=victim,Vd=suspect,xd=1)
```

| mastermix | *A Tcl/Tk graphical user interface for simple DNA mixtures resolution using allele peak heights/ or areas information* |

Description

The `mastermix` function launches a Tcl/Tk graphical user interface dedicated to the resolution of two-person DNA mixtures using allele peak heights/ or areas information. `mastermix` is the implementation of a method developed by Gill et al (see the references section), and previously programmed into an Excel macro by Dr. Peter Gill.

Usage

```
mastermix()
```

Details

`mastermix` is a Tcl/Tk graphical user interface implementing a method developed by Gill et al (1998) for simple mixtures resolution, using allele peak heights or areas information.

This method searches through simulation the most likely combination(s) of the contributors' genotypes. Having previously obtained an estimation for the mixture proportion, it is possible to reduce the number of possible genotype combinations by keeping only those supported by the observed data. This is achieved by computing the sum of square differences between the expected allelic ratio and the observed allelic ratio, for all possible mixture combinations. The likelihood of peak heights (or areas), conditional on the combination of genotypes, is high if the residuals are low. Genotype combinations are thus selected according to the peak heights with the highest (conditioned) likelihoods.

`mastermix` offers a graphical representation of the simulation for three models:
-The two allele model: at a given locus, two alleles are observed in the DNA stain.
-The three allele model: at a given locus, three alleles are observed in the DNA stain.
-The four allele model: at a given locus, four alleles are observed in the DNA stain.

A left-click on each button launches a simulation dialog window for the corresponding model, while a right-click opens the corresponding help page.

Note

-Each implemented model can either be launched using the `mastermix` interface, or the `A2.simu`, `A3.simu` and `A4.simu` functions, depending on the considered model.
-For the computational details, please see forensim tutorial at `http://forensim.r-forge.r-project.org/misc/forensim-tutorial.pdf`.

Author(s)

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

References

Gill P, Sparkes P, Pinchin R, Clayton, Whitaker J, Buckleton J. Interpreting simple STR mixtures using allele peak areas. *Forensic Sci Int* 1998;91:41-5.

See Also

`A2.simu`, `A3.simu` and `A4.simu`

Examples

```
mastermix()
```

---

mincontri	*Minimum number 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))
#5-person mixture
mix5<-simumix(gen,ncontri=c(5,0,0))
#compare
likestim(mix5, strusa, refpop="Afri")
mincontri(mix5)
```

N2error   *Calculates exact error for maximum allele count for two markers*

Description

The maximum allele count principle leads to wrong conclusion for two contributors if only a maximum of one or two alleles is seen. This probability of error is calculated.

Usage

```
N2error(dat)
```

Arguments

dat   a data frame, first column gives the alleles size, remaining columns give their
                    frequencies

Value

The probability of error is returned.

Author(s)

Thore Egeland <Thore.Egeland@medisin.uio.no>

Examples

```
#Example based on 15 markers of Tu data
library(forensim)
data(Tu)
N2error(Tu)
```

N2Exact   *Calculates exact allele distribution for 2 contributors*

Description

The distribution of N, the number of alleles showing is calculated exactly assuming 2 contributors. Theta-correction is not implemented. The function may be used to check accuracy of simulations and indicate required number of simulations for one example.

Usage

```
N2Exact(p)
```

Arguments

p   vector of allele frequencies. Must sum to 1. Default: for uniformly distrubted
                    alleles.

Value

Returns(P(N=i) for i=1,2,3,4

Author(s)

Thore Egeland `<Thore.Egeland@medisin.uio.no>`

Examples

```
#Distribution for a marker with 20 alles of equal frequency
N2Exact(p=rep(0.05,20))
```

---

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 NAs 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 NAs 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 4-person mixture
maa4<-simumix(gaa,ncontri=c(4,0,0))
nball(maa4,byloc=TRUE)
```

PE   *The random man exclusion probability*

Description

Computes the random man exclusion probability of a mixture stored in a [simumix](#) object

Usage

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

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

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)
```

Pevid2	*Conditional profile probabilities*

Description

Calculates the probability of observing a set of DNA profiles conditional on a given hypothesis specifying who were the contributors to the observed profiles. All the individuals involved in the analyzed case are assumed to come from the same subpopulation with a given coancestry coefficient.

Usage

```
Pevid2(stain, freq, x, T = NULL, V = NULL, theta = 0)
```

Arguments

`stain`	vector of distinct alleles (from one specific locus) found in the crime sample.
`freq`	vector of the corresponding allele frequencies in the global population
`x`	the number of unknown contributors to the mixture
`T`	object of class genotype (package **genetics**), or a vector of strings where each string contains two alleles separated by '/', corresponding to one known contributor. The length of the vector equals the number of known contributors. Default is NULL.
`V`	object of class genotype (package **genetics**), or a vector of strings where each string contains two alleles separated by '/', corresponding to one known non-contributor. The length of the vector equals the number of known non-contributors. Default is NULL.
`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

Please note that the `Pevid2` function is an improved version of the `Pevid.gen` function from the forensic package by Miriam Marusiakova (which explains the 2 in the function name). `Pevid2` calls external functions in C code.

Here we define the conditional profile probability as the probability of the profiles under a certain hypothesis stating who gave the observed alleles, hence, Pr(stain="A"|U=0,V=0,T="A/A",H="suspect A/A gave the profile) would equal one rather than 2*p(A)*p(A) in the original formula in Curran et al.

Author(s)

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

References

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

See Also

LR, RMP

Examples

```
# A rape case in Hong Kong (Hu and Fung, Int J Legal Med 2003)
# The stain shows alleles 14, 15, 17 and 18 at locus D3S1358.
stain=c(14,15,17,18)
# suspect's profile: "14/17"
suspect<-"14/17"
# victim's profile: "15/18"
victim<-"15/18"
# corresponding allele frequencies
freq<-c(0.033,0.331,0.239,0.056)
```

```
# Prosecution proposition: Contributors were the victim and the suspect
# defence proposition: Contributors were the victim and 1 unknown contributor
# from the same subpopulation as the victim
# Evaluation of the defence proposition, in case of independence between alleles
Pevid2(stain, freq, x=1, T = victim)

# note that if theta=0, the suspect's profile plays no role in the calculation
#and the same result is obtained
Pevid2(stain, freq, x=1, T = victim, V = suspect)
# In case of allele dependencies, measured by theta=0.03
Pevid2(stain, freq, x=1, T = victim, V = suspect, theta = 0.03)
```

PV   *Predictive value of the maximum likelihood estimator of the number of contributors to a DNA mixture*

Description

The `PV` function implements the predictive value of the maximum likelihood estimator of the number of contributors to a DNA mixture

Usage

```
PV(mat, prior)
```

Arguments

mat   matrix giving the estimates of the conditional probabilities that the maximum likelihood estimator classifies a given stain as a mixture of i contributors given that there are k contributor(s) to the stain. Estimates i must be given in columns for each possible value of the number of contributors given in rows.

prior   numeric vector giving the prior probabilities of encountering a mixture of i contributors. `prior` must be of length the number of rows in `mat`.

Value

Vector of the predictive values

Author(s)

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

References

Haned H., Pene L., Sauvage F., Pontier D., The predictive value of the maximum likelihood estimator of the number of contributors to a DNA mixture, submitted, 2010.

See Also

maximum likelihood estimator `likestim`

Examples

```
# the following examples reproduce some of the calculations appearing
# in the article cited above, for illustrative purpose, the maximum
#number of contributors is set here to 5
#matcondi: Table 2 in Haned et al. (2010)
matcondi<-matrix(c(1,rep(0,4),0,0.998,0.005,0,0,0,0.002,0.937,0.067,0,0,0,0.058,
0.805,0.131,rep(0,3),0.127,0.662,rep(0,3),0.001,0.207),ncol=6)
#prior defined by a forensic expert (Table 3 in Haned et al., 2010)
prior1<-c(0.45,0.04,0.30,0.15,0.06)
#uniform prior, for each mixture type, the probability of occurrence is 1/5,
#5 being the threshold for the number of contributors
prior2<-c(rep(1/5,5))
#predictive values for prior1
PV(matcondi,prior1)
#for prior2
PV(matcondi, prior2)
```

---

recordDrop	*Records the allelic dropout events matched with individual DNA proxies*

Description

The `recordDrop` function records the dropout events from experimental data. The function aims to facilitate the manipulation of experimental data used for the estimation of the probability of allelic dropout (cf. the references sections).

Usage

```
recordDrop(x, y, geno, tabcsv,s=40)
```

Arguments

x	numeric label of the contributing individual, if the stain is a mixture, `x` should give the label of the first individual contributing to the mixture
y	numeric label of the second contributing individual, default is NULL. If the stain is a mixture, `y` should give the label of the second individual contribution to the mixture. This argument is skipped if the stain is not a mixture (default case: `y` set to NULL).
geno	a matrix giving the genotypes of the individuals contributing to the analyzed data for each locus. An individual genotype is given in rows for each locus in column. A homozygous carrying allele 9 is coded '9/9', a heterozygous carrying alleles 8 and 9 is coded '8/9'. Individual labels are coded using integers that are simply the order of introduction in the data frame.
tabcsv	a matrix giving the validation table of the analysed DNA stain. `tabcsv` must have a "genemapper" validation table structure, namely, information about the present alleles and the corresponding peak heights must be given.
s	numeric giving the detection threshold for alleles in Relative fluorescence units (RFU) , default is set to 40 RFUS. An observed allele with a peak height smaller (<) than 40 RFUS is considered as dropped-out.

Value

A list of length the number of analyzed loci, each component of the list is a matrix with the following information: - The names of expected alleles

- The expected allele counts for the first contributor (when date is a mixture)
- The expected allele counts for the second contributor (when date is a mixture)
- The observed alleles
- The observed peak heights
- The dropout variable D, takes 1 if the allele has dropped out, 0 otherwise

Note

`recordDrop` is an auxiliary function of the `tabDNAproxy` function that implements the methodology proposed by Tvedebrink et al. to estimate the probability of allelic dropout using experimental DNA mixtures.

Author(s)

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

References

Tvedebrink T, Eriksen PS, Mogensen HS, Morling N. Estimating the probability of allelic drop-out of STR alleles in forensic genetics. *Forensic Science International: Genetics*, 2009, 3(4), 222-226.

See Also

DNAproxy, tabDNAproxy

Examples

```
#load the exemple data
data(dropdata)
tabcsv<-dropdata$tabcsv
genot<-dropdata$genot
#individuals' labels are 1 and 2
#record the dropout the surviving peak heights for heterozygotes whith non shared alleles
recordDrop(1,2,geno=genot,tabcsv=tabcsv,s=40)
```

| recordHeights | *Records the peak heights of the alleles present in the analyzed stains* |

Description

The `recordHeights` function records the peak heights of the alleles present in the analyzed stains. The function aims to facilitate the manipulation of experimental data used for the estimation of the probability of allelic dropout (cf. the references sections).

Usage

```
recordHeights(x,y=NULL,geno,tabcsv,byloc=FALSE)
```

Arguments

x	numeric label of the contributing individual, if the stain is a mixture, x should give the label of the first individual contributing to the mixture
y	numeric label of the second contributing individual, default is NULL. If the stain is a mixture, y should give the label of the second individual contribution to the mixture. This argument is skipped if the stain is not a mixture (default case: y set to NULL).
geno	a matrix giving the genotypes of the individuals contributing to the analyzed data for each locus. An individual genotype is given in rows for each locus in column. A homozygous carrying allele 9 is coded '9/9', a heterozygous carrying alleles 8 and 9 is coded '8/9'. Individual labels are coded using integers that are simply the order of introduction in the data frame.
tabcsv	a matrix giving the validation table of the analysed DNA stain. tabcsv must have a "genemapper" validation table structure, namely, information about the present alleles and the corresponding peak heights must be given.
byloc	logical indicating whether data should be displayed per locus (TRUE) or overall loci (FALSE, default)

Value

A list of length the number of analyzed loci, each component of the list is a matrix with the following information: - The names of expected alleles - The expected allele counts for the first contributor (when date is a mixture) - The expected allele counts for the second contributor (when date is a mixture) - The observed alleles - The observed peak heights - The dropout variable D, takes 1 if the allele has dropped out, 0 otherwise

Note

recordHeights is an auxiliary function of the tabSPH function that implements the methodology proposed by Gill et al. to estimate the probability of allelic dropout using experimental DNA mixtures.

Author(s)

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

References

Gill P, Puch-Solis R, Curran J. The low-template-DNA (stochastic) threshold-Its determination relative to risk analysis for national DNA databases. *Forensic Science International: Genetics*, 2009, 3, 104-111

See Also

recordDrop for an alternative method, tabSPH

Examples

```
#load the exemple data
data(dropdata)
tabcsv<-dropdata$tabcsv
genot<-dropdata$genot
```

```
#individuals' labels are 1 and 2
#peak heights of heterozygote genotypes with non shared alleles
recordHeights(1,2,geno=genot,tabcsv=tabcsv)
```

RMP   *The Random Match Probability of DNA evidence (RMP)*

Description

RMP computes the random match probability of DNA evidence given in a matrix (or data frame) or in a text file. Several situations are handled: the suspect and an unknown offender are unrelated, or are members of the same subpopulation with a given coancestry coefficient theta, or are close relatives. For the latter case, the relationship is described by the kinship coefficients.

Usage

```
RMP(suspect=NULL, filename=NULL, freq, k=c(1,0,0), theta=0,refpop=NULL)
```

Arguments

suspect   a matrix or a data frame of dimension L x 2, L being the number of loci involved in the DNA evidence. The first column gives the loci names, and the second column gives the suspect's genotype at each locus. A genotype is coded as a character where each string contains two alleles separated by '/'. The DNA evidence can also be given in a text file, see argument filename.

filename   the file name from which the input data should be read. Data mut be a matrix of dimension L x 2, L being the number of loci involved in the DNA evidence.The first column gives the loci names, and the second column gives the suspect's genotype at each locus. A genotype is coded as a character where each string contains two alleles separated by '/'.

freq   a tabfreq object giving the allele frequencies

k   vector of kinship coefficients (k0, k1, k2), where ki is the probability that two people (the suspect and an unknown offender) will share i alleles identical by descent, i = 0, 1, 2.

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

refpop   the reference population in freq from which to extract the allele frequencies fro the RMP calculation. This argument is obligatory only if freq contains allele frequencies from several populations

Details

The match probability is derived from Balding and Nichols (1994) and is computed as:

$$k_2 + k_1 Z_1 + k_0 Z_2$$

where k_0, k_1, k_2 are the kinship coefficients,
Z_1 is the match probability when the suspect an the unknown offender share one allele identical-by-descent.

Z_2 is the match probability in the unrelated case, when the suspect an the unknown offender share 0 allele identical-by-descent.

In the homozygous case, with the allele frequency p_i:

$$Z_1 = \frac{2\theta + (1 - \theta)p_i}{1 + \theta}$$

$$Z_2 = \frac{[2\theta + (1 - \theta)p_i]\,[3\theta + (1 - \theta)p_i]}{(1 + \theta)(1 + 2\theta)}$$

In the heterozygous case, with allele frequencies p_i and p_j:

$$Z_1 = \frac{2\theta + (1 - \theta)(p_i + p_j)}{2(1 + \theta)}$$

$$Z_2 = \frac{2\,[\theta + (1 - \theta)p_i]\,[\theta + (1 - \theta)p_j]}{(1 + \theta)(1 + 2\theta)}$$

θ is Wright's Fst coefficient, usually called the coancestry coefficient in forensic studies. Main effects of allele dependencies between loci in the suspect's subpopulation are taken into account though the coancestry coefficient, hence, the match probability at all loci is, to a close approximation, the product of single-locus probabilities.

Value

`RMP` returns a list with the following components:

`RMP.loc`	single-locus match probabilities
`RMP`	multiloci match probability (product of single-locus match probabilities)

Author(s)

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

References

Balding DJ, Nichols RA. DNA profile match probability calculation: How to allow for population stratification, relatedness, database selection and single bands. *Forensic Sci I* 1994;64:125-140.

See Also

LR for the evaluation of DNA evidence through likelihood ratio

Examples

```
# random match probability
# data input

data <- matrix(c("CSF1PO","FGA","TH01","TPOX","VWA","D3S1358","D5S818",
"D7S820","D8S1179","D13S317","D16S539","D18S51","D21S11","D2S1338","D19S433",
"12/11","22/19","6/7","10/8","17/18","18/17","12/12","8/8","13/13","11/11",
"12/10","14/15","33.2/32.2","23/22","14/14"),nc=2)
colnames(data)<- c('locus','genotype')
#15-locus genotype
data
```

```
#allele frequencies are taken from the strusa data set

data(strusa)

RMP(suspect=data,freq=strusa,refpop="Cauc")

# using a preexisting file from the forensim package
RMP(filename=system.file("files/exprofile.txt", package = "forensim"),
freq=strusa,refpop="Cauc")
```

---

simMixSNP	*Simulates SNP mixtures*

Description

Simulates SNP mixtures and outputs optionally file suitable for wrapdataL function for estimation of number of contributors

Usage

```
simMixSNP(nSNP , p , ncont, writeFile, outfile , id )
```

Arguments

nSNP	Integer number of SNPs>1
p	Minor allele frequency
ncont	Number of contributors >= 1
writeFile	If TRUE, output written to file
outfile	Name of output file
id	Column one of output file identifying run

Value

Returns a data frame with columns Id, marker, allele, frequency and height (=1 for now)

Author(s)

Thore Egeland <Thore.Egeland@medisin.uio.no>

Examples

```
simMixSNP()
```

---

simPCR2   *Polymorphism chain reaction simulation model*

Description

simPCR2 implements a simulation model for the polymorphism chain reaction (Gill et al., 2005). Giving several input parameters, simPCR2 outputs the number of amplified DNA molecules and their corresponding peak heights (in RFUs).

Usage

```
simPCR2(ncells,probEx,probAlq, probPCR, cyc = 28, Tdrop = 2 * 10^7,
probSperm = 0.5, dip = TRUE,KH=55)
```

Arguments

ncells	initial number of cells
probEx	probability that a DNA molecule is extracted (probability of surviving the extraction process)
probAlq	probability that a DNA molecule is selected for PCR amplification
probPCR	probability that a DNA molecule is amplified during a given round of PCR
cyc	number of PCR cycles, default is 28 cycles
Tdrop	threshold of detection: number of molecules (in the total PCR reaction mixture) that is needed to generate a signal, default is set to $2*10^7$ molecules
probSperm	probability of observing alleles of type A in the initial sample of haploid cells (e.g. sperm cells). Probability of observing allele B is given by 1-probSperm
dip	logical indicating the cell ploidy, default is diploid cells (TRUE), FALSE is for haploid cells
KH	positive constant used to scale the peak heights obtained from the number of amplified molecules (see reference section)

Details

A threshold of Tdrop (must be a multiple of 10^7) is needed to generate a signal, then, a log-linear relationship is used to determine the intensity of the signal with respect to the number of successfully amplified DNA molecules. Dropout events occur whenever less than Tdorp molecules are generated.

Value

A matrix with the following components:

HeightA	Peak height of allele A
DropA	Dropout variable for allele A
HeightB	Peak height of allele B
DropB	Dropout variable for allele B

Author(s)

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

References

Jeffreys AJ, Wilson V, Neumann R and Keyte J. Amplification of human minisatellites by the polymerase chain reaction: towards DNA fingerprinting of single cells. *Nucleic Acids Res* 1988;16: 10953<96>10971.

Gill P, Curran J and Elliot K. A graphical simulation model of the entire DNA process associated with the analysis of short tandem repeat loci. *Nucleic Acids Research* 2005, 33(2): 632<96>643.

See Also

simPCR2TK

Examples

```
#simulation of a 28 cycles PCR, with the initial stain containing 5 cells
simPCR2(ncells=5,probEx=0.6,probAlq=0.30,probPCR=0.8,cyc=28, Tdrop=2*10^7,dip=TRUE,KH=55)
```

---

simPCR2TK	*A Tcl/Tk graphical interface for the polymorphism chain reaction simulation model*

Description

simPCR2TK is a user-friendly graphical interface for the simPCR2 function that implements a simulation model for the polymorphism chain reaction.

Usage

```
simPCR2TK()
```

Author(s)

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

References

Gill P, Curran J and Elliot K. A graphical simulation model of the entire DNA process associated with the analysis of short tandem repeat loci. *Nucleic Acids Research* 2005, 33(2): 632<96>643.

See Also

simPCR2

Examples

```
#launch the graphical interface
simPCR2TK()
```

---

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, whose 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 numbers 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 the `nloc` 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 `nloc`
            where NAs 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 modeled 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 in columns. When an allele is
not observed for a given locus, 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

Johnson NL, Kotz S, Balakrishnan N. Continuous Univariate Distributions, vol 2. John Wiley & Sons, 1995.

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 numbers 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`: character vector giving the individuals ID

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 numbers of individuals to simulate per population. For a single population, default is 1.
x	an object

Details

At a given locus, an individual's genotype is simulated by randomly drawing two alleles (with replacement) at their respective allele frequencies in the target population.

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 =100)
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 numbers of individuals to simulate per population. Default is one.
x	an object

Details

DNA mixtures are created by randomly drawing individual genotypes with a uniform probability. If there are N individuals in the sample (the simugeno object), then each individual has a probability of 1/N to be selected.

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 a 3-person mixture characterized 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, following a Dirichlet model. Allele frequencies in the populations are generated as random deviates from a Dirichlet distribution, whose parameters control the deviation of allele frequencies from the values in the reference population.

Usage

```
simupopD(npop = 1, nloc = 1, na = 2, globalfreq = NULL, which.loc = NULL,
alpha1, alpha2 = 1)
```

Arguments

npop	the number of populations
nloc	the number of loci
na	an integer vector giving the numbers 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 positive 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 modeled 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 simulate, at a given locus L with n alleles, the allele frequencies are modeled 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.

alpha1[1] is the variance parameter for population 1 and is equivalent to Wright's Fst. The closest this parameter is to one, the more the population allele frequencies are different from the values of the reference population.

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, Cardon LR. Discussion on the meeting on "Statistical modelling and analysis of genetic data" *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=1)
```

---

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 loci in 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: May 11th 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 3 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)
#allele frequencies
strveneto@tab
```

---

tabDNAproxy   *Builds a list of tables that record the dropout events matched with the appropriate DNA proxies*

Description

The `tabDNAproxy` function builds a list of tables that record the dropout events matched with the appropriate "DNAproxies", these are the approximations of the amount of DNA contributed by the individuals in the analyzed DNA stains. Each table is specific to a locus. This function builds the data frames on which the logistic model, proposed by Tvedebrink et al (cf. references section), can be performed.

Usage

```
tabDNAproxy(x, y = NULL, geno, tabcsv)
```

Arguments

x	numeric label of the contributing individual, if the stain is a mixture, x should give the label of the first individual contributing to the mixture
y	numeric label of the second contributing individual, default is NULL. If the stain is a mixture, y should give the label of the second individual contribution to the mixture. This argument is skipped if the stain is not a mixture (default case: y set to NULL).
geno	a matrix giving the genotypes of the individuals contributing to the analyzed data for each locus. An individual genotype is given in rows for each locus in column. A homozygous carrying allele 9 is coded '9/9', a heterozygous carrying alleles 8 and 9 is coded '8/9'. Individual labels are coded using integers that are simply the order of introduction in the data frame.
tabcsv	a matrix giving the validation table of the analysed DNA stain. tabcsv must have a "genemapper" validation table structure, namely, information about the present alleles and the corresponding peak heights must be given.

Value

A list of length the number of analyzed loci, each component of the list is a matrix with the following information:

Dloc	the (per locus) dropout variable D, takes 1 if the allele has dropped out, 0 otherwise
Hestim	mean peak heights derived from the DNA proxies, see the references section for further details

Author(s)

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

References

Tvedebrink T, Eriksen PS, Mogensen HS, Morling N. Estimating the probability of allelic drop-out of STR alleles in forensic genetics. *Forensic Science International: Genetics*, 2009, 3(4), 222-226.

See Also

recordDrop, DNAproxy

Examples

```
#load the exemple data
data(dropdata)

tabcsv<-dropdata$tabcsv
genot<-dropdata$genot
#individuals' labels are 1 and 2
#lets record the dropout events and the corresponding DNA proxies
tabDNAproxy(1,2,geno=genot,tabcsv=tabcsv)
```

`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"))
```

---

tabSPH   *Builds a matrix of the dropout variable and the corresponding surviving peak heights*

Description

The tabSPH function builds a matrix of the dropout variable and the corresponding surviving peak heights, for each available locus or across all loci (default). The constructed matrices have two columns: the dropout variable and the surviving peak heights. The logistic model proposed to model the dropout probability from experimental data (see the references section) can be performed directly on the data yielded by tabSPH.

Usage

```
tabSPH(x, y = NULL, geno, tabcsv, byloc = FALSE,s=40)
```

Arguments

x   numeric label of the contributing individual, if the stain is a mixture, x should
                     give the label of the first individual contributing to the mixture

y   numeric label of the second contributing individual, default is NULL. If the stain
                     is a mixture, y should give the label of the second individual contribution to the
                     mixture. This argument is skipped if the stain is not a mixture (default case: y
                     set to NULL).

geno   a matrix giving the genotypes of the individuals contributing to the analyzed
                     data for each locus. An individual genotype is given in rows for each locus in
                     column. A homozygous carrying allele 9 is coded '9/9', a heterozygous carrying
                     alleles 8 and 9 is coded '8/9'. Individual labels are coded using integers that are
                     simply the order of introduction in the data frame.

tabcsv   a matrix giving the validation table of the analysed DNA stain. tabcsv must
                     have a "genemapper" validation table structure, namely, information about the
                     present alleles and the corresponding peak heights must be given.

byloc   logical indicating whether data should be displayed per locus (TRUE) or overall
                     loci (FALSE, default)

s   numeric giving the detection threshold for alleles in Relative fluorescence units
                     (RFU) , default is set to 40 RFUS. An observed allele with a peak height smaller
                     (<) than 40 RFUS is considered as dropped-out.

Details

Both mixed and unmixed samples can be used in tabSPH, setting the y argument to NULL (de-
fault) will produce results considering data for x only. In case of mixtures, note that only heterozy-
gote genotypes with no shared alleles are considered.

Value

If argument byloc is TRUE, tabSPH yields a list of length the number of available loci, each
elements of the list contain a matrix with two columns:

D   the dropout variable

H   the surviving peak height

If argument byloc is FALSE, tabSPH yields a single matrix with columns D and H described
above.

Author(s)

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

References

Gill P, Puch-Solis R, Curran J. The low-template-DNA (stochastic) threshold-Its determination rel-
ative to risk analysis for national DNA databases. *Forensic Science International: Genetics*, 2009,
3, 104-111

See Also

tabDNAproxy

Examples

```
#load the example data
data(dropdata)
tabcsv<-dropdata$tabcsv
genot<-dropdata$genot
#individuals' labels are 1 and 2
#recording dropout variable matched with the surviving peak heights
#for heterozygotes whith non shared alleles
tabSPH(1,2,geno=genot,tabcsv=tabcsv,s=0)
```

Tu	*Allele frequencies of 15 autosomal short tandem repeats loci on Chinese 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>

---

wrapdataL   *ML estimate of number of contributors for SNPs*

Description

Wrap up of dataL in forensim. Given file with columns: "No, Marker, Allele, Frequency and Height" the log likelihood for requested number of contributors is calculated. For now only "Frequency" column is used.

Usage

```
wrapdataL(fil , plotte , nInMixture , tit )
```

Arguments

fil	Input file
plotte	If T, plot
nInMixture	Alternatives for number of contributors, say 1:5
tit	Title to be used in plot

Value

Plot (optional) and log likelihoods

Author(s)

Thore Egeland <Thore.Egeland@medisin.uio.no>

Examples

```
aa<-simMixSNP(nSNP=5,writeFile=TRUE,outfile="sim.txt",ncont=3) #Simulates data
res<-wrapdataL(fil="sim.txt") # Calculates and plots
```

Index