

# Package: forensim (via r-universe)

June 4, 2026

**Type** Package

**Title** Interpretation of Forensic DNA Mixtures

**Version** 4.3.3

**Date** 2025-05-02

**Maintainer** Maarten Kruijver <maarten.kruijver@esr.cri.nz>

**Depends** methods, tcltk,tcltk2,tkrplot

**Description** Statistical methods and simulation tools for the interpretation of forensic DNA mixtures. The methods implemented are described in Haned et al. (2011) <doi:10.1111/j.1556-4029.2010.01550.x>, Haned et al. (2012) <doi:10.1016/j.fsigen.2012.11.002> and Gill & Haned (2013) <doi:10.1016/j.fsigen.2012.08.008>.

**License** GPL (>= 2)

**LazyLoad** yes

**SystemRequirements** Tcl/Tk package TkTable

**Collate** classes\_definitions.R classes\_constructors.R accessors.R simufreqD.R simupopD.R AuxFunc.R changepop.R PE.R likelihood.R likestim.R mincontri.R A2.simu.R A3.simu.R A4.simu.R mastermix.R N2Exact.R N2error.R simMixSNP.R wrapdataL.R simPCR2.R simPCR2TK.R PV.R Hbsimu.R LR.R LRmixTK.R

**Config/pak/sysreqs** tcl tk tk-dev tk-table libx11-dev

**Repository** https://mkruijver.r-universe.dev

**Date/Publication** 2025-05-07 15:33:20 UTC

**RemoteUrl** https://github.com/mkruijver/forensim

**RemoteRef** HEAD

**RemoteSha** 63c8d4eb858b73af67c6381196b32a402adc4a0b

## Contents

forensim-package . . . . .	3
----------------------------	---

A2.simu . . . . .	3
A3.simu . . . . .	5
A4.simu . . . . .	6
Accessors . . . . .	8
changepop . . . . .	8
Cmn . . . . .	9
comb . . . . .	10
dataL . . . . .	11
findfreq . . . . .	12
findmax . . . . .	13
Hbsimu . . . . .	14
lik . . . . .	14
lik.loc . . . . .	16
likestim . . . . .	17
likestim.loc . . . . .	19
likEvid . . . . .	20
LR . . . . .	22
LRmixTK . . . . .	24
mastermix . . . . .	24
mincontri . . . . .	26
N2error . . . . .	27
N2Exact . . . . .	27
naomitab . . . . .	28
nball . . . . .	29
ngm . . . . .	30
PE . . . . .	30
PV . . . . .	31
sgmNorway . . . . .	32
simMixSNP . . . . .	33
simPCR2 . . . . .	34
simPCR2TK . . . . .	35
simufreqD . . . . .	36
simugeno . . . . .	37
simugeno constructor . . . . .	38
simumix . . . . .	39
simumix constructor . . . . .	40
simupopD . . . . .	42
strusa . . . . .	44
strveneto . . . . .	45
tabfreq . . . . .	45
tabfreq constructor . . . . .	46
Tu . . . . .	47
virtualClasses . . . . .	48
wrapdataL . . . . .	48

---

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 <hinda@owlsandarrows.nl>

---

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.

### Value

No return value, called to show GUI.

### 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 <[hinda@owlsandarrows.nl](mailto:hinda@owlsandarrows.nl)>

### 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	<i>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</i>
---------	---

---

## 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.

## Value

No return value, called to show GUI.

**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 <hinda@owlsandarrows.nl>

**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.

### Value

No return value, called to show GUI.

### 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`](http://forensim.r-forge.r-project.org/misc/forensim-tutorial.pdf).

### Author(s)

Hinda Haned <[hinda@owlsandarrows.nl](mailto:hinda@owlsandarrows.nl)>

### 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 <[hinda@owlsandarrows.nl](mailto:hinda@owlsandarrows.nl)>

### 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 <a href="#">tabfreq</a> , a <a href="#">simugeno</a> or a <a href="#">simumix</a> 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 <hinda@owlsandarrows.nl>

**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.

**Value**

Numeric with the number of combinations.

**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 <hinda@owlsandarrows.nl>

**See Also**

[comb](#) for all possible combinations of m elements among n with repetitions

**Examples**

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

---

comb	<i>Generate all possible combinations of m elements among n with repetitions</i>
------	--

---

**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 <hinda@owlsandarrows.nl>

**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	<i>Generic formula of the likelihood of the observed alleles in a mixture conditional on the number of contributors for a specific locus</i>
-------	--

---

### 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

<code>x</code>	an integer giving the number of contributors
<code>p</code>	a numeric vector giving the frequencies of the observed alleles in the mixture
<code>theta</code>	a float in $[0,1[$ . <code>theta</code> is equivalent to Wright's $F_{st}$ . In case of population subdivision, it allows a correction of the allele frequencies in the subpopulation of interest

### Value

Numeric likelihood value.

### 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 <https://cran.r-project.org/package=forensic>

### Author(s)

Hinda Haned <[hinda@owlsandarrows.nl](mailto:hinda@owlsandarrows.nl)>

### 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)
```

---

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, from 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 <hinda@owlsandarrows.nl>

### 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	<i>Function to find the maximum of a vector and its position</i>
---------	--

---

**Description**

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

**Usage**

```
findmax(vec)
```

**Arguments**

<code>vec</code>	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 <hinda@owlsandarrows.nl>

**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()

### Value

No return value, called to show GUI.

### Author(s)

Hinda Haned <hinda@owlsandarrows.nl>

### 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): 632-643.

### 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 <code>simumix</code> object which contains the mixture to be analyzed
freq	a <code>tabfreq</code> object from which to extract the allele frequencies
refpop	a factor giving the reference population in <code>tabfreq</code> from which to extract the allele frequencies. This argument is used only if <code>freq</code> contains allele frequencies for multiple populations, otherwise it is by default set to <code>NULL</code>
theta	a float from <code>[0,1[</code> giving Wright's $F_{st}$ coefficient. <code>theta</code> accounts for population subdivision while computing the likelihood of the data
loc	loci for which the overall likelihood shall be computed. Default ( <code>NULL</code> ) 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.

## Value

Numeric likelihood value.

## Author(s)

Hinda Haned <hinda@owlsandarrows.nl>

## 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	<i>Likelihood per locus of the observed alleles in a DNA mixture conditional on the number of contributors to the mixture</i>
---------	---

---

### 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

<code>x</code>	the number of contributors to the DNA mixture
<code>mix</code>	a <code>simumix</code> object which contains the mixture to be analyzed
<code>freq</code>	a <code>tabfreq</code> object from which to extract the allele frequencies
<code>refpop</code>	a factor giving the reference population in <code>tabfreq</code> from which to extract the allele frequencies
<code>theta</code>	a float from $[0,1[$ giving Wright's $F_{st}$ coefficient. <code>theta</code> accounts for population subdivision while computing the likelihood of the data.
<code>loc</code>	the loci for which the likelihood shall be computed. Default (set to <code>NULL</code> ) 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 subpopulation.

The  $F_{st}$  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 <hinda@owlsandarrows.nl>

## 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	<i>Maximum likelihood estimation of the number of contributors to a forensic DNA mixture for a set of loci</i>
----------	--

---

## 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 <code>simumix</code> object
freq	a <code>tabfreq</code> 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 <code>tabfreq</code> contains more than one population, <code>refpop</code> must be specified, otherwise, <code>refpop</code> 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 <hinda@owlsandarrows.nl>

### 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	<i>Maximum likelihood estimation per locus of the number of contributors to forensic DNA mixtures.</i>
--------------	--

---

## 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

<code>mix</code>	a <code>simumix</code> object
<code>freq</code>	a <code>tabfreq</code> object containing the allele frequencies to use for the calculation
<code>refpop</code>	the reference population from which to extract the allele frequencies used in the likelihood calculation. Default set to <code>NULL</code> , if <code>tabfreq</code> contains more than one population, <code>refpop</code> must be specified
<code>theta</code>	a float from $[0,1[$ giving Wright's $F_{st}$ coefficient. <code>theta</code> accounts for population subdivision while computing the likelihood of the data.
<code>loc</code>	loci to be considered in the estimation. Default (set to <code>NULL</code> ) 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 column, `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 <hinda@owlsandarrows.nl>

## 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")
```

---

 likEvid

*Likelihood of DNA evidence conditioned on a given hypothesis*

---

## Description

likEvid allows the calculation of likelihood for a piece of DNA evidence, for any number of replicates, any number of contributors, and when drop-in and drop-out are possible.

## Usage

```
likEvid(Repliste, Tg, Vg, x, theta, prDHet, prDHom, prC, freq)
```

## Arguments

Repliste	vector of alleles present at a given locus for any number of replicates. If there are two replicates, showing alleles 12,13, and 14 respectively, then Repliste should be given as c(12,13,0,14), where the 0 is used as a separator. An empty replicate is simply 0. For example, replicates (12,13) and one empty replicate must be given as: c(12,14,0,0).
Tg	vector of genotypes for the known contributors under Hp. Genotype 12/17 should be given as a vector c(12,17) and genotypes 12/17,14/16, should be given as a unique vector: c(12,17,14,16). If T is empty, set to 0.

Vg	vector of genotypes for the known non-contributors (see References section) under Hp. See T for format.
x	Number of unknown individuals under H. Set to 0 if there are no unknown contributors.
theta	thete correction, value must be taken in [0,1)
prDHet	probability of dropout for heterozygotes. It is possible to assign different values per contributor. In this case, prDHet must be a vector, of length the number of contributors in T + x, and the probabilities must be given in this order. if the probability of dropout for T is d1, and for the unknown is d2, then prDHet=(d1,d2). In case T is not a heterozygote, the given vector must still be of length length(T) +x, but the given value for T does not matter, because it won't be used, the value in prDHom is used instead. This is a bit ad hoc and an improvement is currently under development.
prDHom	probability of dropout for homozygotes. See description ofr argument PrDHom.
prC	probability of drop-in applied per locus
freq	vector of the corresponding allele frequencies of the analysed locus in the target population

**Value**

Numeric likelihood value.

**Author(s)**

Hinda Haned <hinda@owlsandarrows.nl>

**References**

- Gill, P.; Kirkham, A. & Curran, J. LoComatioN: A software tool for the analysis of low copy number DNA profiles *Forensic Science International*, 2007, 166(2-3), 128-138
- Curran, J. M.; Gill, P. & Bill, M. R. Interpretation of repeat measurement DNA evidence allowing for multiple contributors and population substructure *Forensic Science International*, 2005, 148, 47-53

**See Also**

[LRmixTK](#)

**Examples**

```
#load allele frequencies
library(forensim)
data(ngm)
#create vector of allele frequencies
d10<-ngm$tab$D10
# evaluate the evidence under Hp; contributors are the suspect and one unknown,
# dropout probabilities for the suspect and the unknown are the same: 0.2 for heterozygotes,
# and 0.1 for homozygotes.
```

```
likEvid(Repliste=c(12,13,14),Tg=c(12,13),Vg=0,x=1,theta=0,prDHet=c(0.2,0.2),
prDHom=c(0.04,0.04),prC=0,
freq=d10)
# evaluate the evidence under Hd; contributors are two unknown people, the dropout
# probabilities for the unknowns is kept the same under Hd
likEvid(Repliste=c(12,13,14),Tg=0,Vg=0,x=2,theta=0,prDHet=c(0.2,0.2),
prDHom=c(0.04,0.04),prC=0,freq=d10)
```

---

LR *Likelihood ratio for DNA evidence interpretation (2): a sophisticated version of function LR()*

---

### Description

LR Allows the calculation of likelihood ratios for a piece of DNA evidence, for any number of replicates, any number of contributors, and when drop-in and drop-out are possible.

### Usage

```
LR(Repliste, Tp, Td, Vp, Vd, xp, xd, theta, prDHet, prDHom, prC, freq)
```

### Arguments

Repliste	vector of alleles present at a given locus for any number of replicates. If there are two replicates, showing alleles 12,13, and 14 respectively, then Repliste should be given as c(12,13,0,14), where the 0 is used as a separator. An empty replicate is simply 0. For example, replicates (12,13) and and one empty replicate must be given as: c(12,14,0,0).
Tp	vector of genotypes for the known contributors under Hp. Genotype 12/17 should be given as a vector c(12,17) and genotypes 12/17,14/16, should be given as a unique vector: c(12,17,14,16).
Td	vector of genotypes for the known contributors under Hd. Should be in the same format as Tp. If there are no known contributors under Hd, then set Td to 0.
Vp	vector of genotypes for the known non-contributors (see References section) under Hp. See Tp for format.
Vd	vector of genotypes for the known non-contributors (see References section) under Hd. Should be in the same format than Vp, if empty, set to 0.
xp	Number of unknown individuals under Hd. Set to 0 if there are no unknown contributors.
xd	Number of unknown individuals under Hd. Set to 0 if there are no unknown contributors.
theta	theta correction, value must be taken in [0,1)

prDHet	probability of dropout for heterozygotes. It is possible to assign different values per contributor. In this case, prDHet must be a vector, of length the number of contributors in T + x, and the probabilities must be given in this order. if the probability of dropout for T is d1, and for the unknown is d2, then prDHet=(d1,d2). In case T is not a heterozygote, the given vector must still be of length length(T) +x, but the given value for T does not matter, because it won't be used, the value in prDHom is used instead. This is a bit ad hoc and an improvement is currently under development.
prDHom	probability of dropout for homozygotes. See description ofr argument PrDHom.
prC	probability of drop-in applied per locus
freq	vector of the corresponding allele frequencies of the analysed locus in the target population

### Value

List with named elements for numerator likelihood (num), denominator likelihood (deno) and likelihood ratio (LR)

### Author(s)

Hinda Haned <hinda@owlsandarrows.nl>

### References

Gill, P.; Kirkham, A. & Curran, J. LoComatioN: A software tool for the analysis of low copy number DNA profiles *Forensic Science International*, 2007, 166(2-3), 128-138

Curran, J. M.; Gill, P. & Bill, M. R. Interpretation of repeat measurement DNA evidence allowing for multiple contributors and population substructure *Forensic Science International*, 2005, 148, 47-53

### See Also

[LRmixTK](#)

### Examples

```
#load allele frequencies
library(forensim)
data(ngm)
#create vector of allele frequencies
d10<-ngm$tab$D10
# heterozygote dropout probability (resp. homozygote) is set to 0.2 for all
# contributors (0.04 for homozygotes)
LR(Repliste=c(12,13,14),Tp=c(12,13),Td=0,Vp=0,Vd=0,xd=2,xp=1,theta=0,prDHet=c(0.2,0.2),
prDHom=c(0.04,0.04),prC=0,freq=d10)
```

LRmixTK

*GUI for the LR function*

---

**Description**

User-friendly graphical user interface for the LR calculator LR.

**Usage**

LRmixTK(verbose)

**Arguments**

verbose            if TRUE, progress is written to the console

**Value**

No return value, called to show GUI.

**Author(s)**

Hinda Haned <hinda@owlsandarrows.nl>

---

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.

## Value

No return value, called to show GUI.

## 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 <[hinda@owlsandarrows.nl](mailto:hinda@owlsandarrows.nl)>

## 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	<i>Minimum number of contributors required to explain a forensic DNA mixture</i>
-----------	--

---

### 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 <code>simumix</code> object
loc	the loci to consider for the calculation of the minimum of contributors, default (NULL) corresponds to all loci

### Value

Integer with minimum number of contributors.

### Author(s)

Hinda Haned <hinda@owlsandarrows.nl>

### 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	<i>Calculates exact error for maximum allele count for two markers</i>
---------	--

---

**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	<i>Calculates exact allele distribution for 2 contributors</i>
---------	--

---

**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 distributed 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 alleles 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 <hinda@owlsandarrows.nl>

**See Also**[tabfreq](#)**Examples**

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

---

nball	<i>Number of alleles in a mixture</i>
-------	---------------------------------------

---

**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)

**Value**

If byloc=TRUE, the number of alleles by locus; otherwise the sum.

**Author(s)**

Hinda Haned <hinda@owlsandarrows.nl>

**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)
```

---

ngm	<i>Allele frequencies for the new generation markers NGM, for the Caucasian US population</i>
-----	---

---

**Description**

Allele frequencies for 15 autosomal short tandem repeats loci in the American Caucasian population.

**Usage**

```
data(ngm)
```

**References**

Budowle, B.; Ge, J.; Chakraborty, R.; Eisenberg, A.; Green, R.; Mulero, J.; Lagace, R. & Hennessy, L. Population genetic analyses of the NGM STR loci International Journal of Legal Medicine, 2011, 1-9

**Examples**

```
library(forensim)
data(ngm)
boxplot(ngm$tab)
```

---

PE	<i>The random man exclusion probability</i>
----	---

---

**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 <code>simumix</code> object
freq	a <code>tabfreq</code> object giving the allele frequencies from which to compute the exclusion probability
refpop	character giving the reference population, used only if <code>freq</code> contains allele frequencies for multiple populations
theta	a float from $[0,1[$ giving Wright's $F_{st}$ coefficient. <code>theta</code> 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 the  $\theta$  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.

**Value**

Numeric vector with exclusion probability (by locus if `byloc = TRUE`).

**Author(s)**

Hinda Haned <hinda@owlsandarrows.nl>

**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)
```

---

PV	<i>Predictive value of the maximum likelihood estimator of the number of contributors to a DNA mixture</i>
----	--

---

**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 <hinda@owlsandarrows.nl>

**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)
```

---

sgmNorway

*Allele frequencies for the new generation markers sgm, for the Norwegian population*

---

**Description**

Allele frequencies for 10 autosomal short tandem repeats loci in the Norwegian population.

**Usage**

data(sgmNorway)

## References

Andreassen, R., S. Jakobsen, and Mevaag, B., Norwegian population data for the 10 autosomal STR loci in the AMPFISTR(R) SGM Plus(TM) system. Forensic Science International, 2007. 170(1): p. 59-61.

---

simMixSNP	<i>Simulates SNP mixtures</i>
-----------	-------------------------------

---

## 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 $2 \cdot 10^7$ 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 $1 - \text{probSperm}$
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 Tdrop 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 <hinda@owlsandarrows.nl>

**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\_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): 632-643.

**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()
```

**Value**

No return value, called to show GUI.

**Author(s)**

Hinda Haned <hinda@owlsandarrows.nl>

**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): 632-643.

**See Also**[simPCR2](#)**Examples**

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

---

simufreqD	<i>Function to simulate allele frequencies for independent loci from a Dirichlet model</i>
-----------	--

---

**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  $p=(p_1, \dots, p_n)$ , following a Dirichlet distribution with parameters:  $\alpha = (\alpha_1, \dots, \alpha_n)$  where  $p_1 + \dots + p_n = 1$  and  $\alpha_1, \dots, \alpha_n > 0$ .

**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 <hinda@owlsandarrows.nl>

**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 simulated 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 <hinda@owlsandarrows.nl>

## 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 <hinda@owlsandarrows.nl>

**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 populations 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 <hinda@owlsandarrows.nl>

### 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 <hinda@owlsandarrows.nl>

**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	<i>Simulate multi-population allele frequencies for independent loci from a reference population, following a Dirichlet model</i>
----------	---

---

## 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 from 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  $p=(p_1, \dots, p_n)$  following a Dirichlet distribution with a parameter vector of length n, where each component is equal to alpha2,  $p_1+\dots+p_n=1$  and  $\alpha_2 > 0$ .

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  $p=(p_1, \dots, p_n)$  following a Dirichlet distribution with a parameter vector of length n:

$(p_1(1-\alpha_1)/\alpha_1[1], \dots, p_n(1-\alpha_1[1])/alpha_1[1]),$  where  $p_1+\dots+p_n=1$  and  $\alpha_1[1] > 0$ .

$\alpha_1[1]$  is the variance parameter for population 1 and is equivalent to Wright's  $F_{st}$ . The closer 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 <hinda@owlsandarrows.nl>

### 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
```

---

`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 <hinda@owlsandarrows.nl>

## 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. When 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 <hinda@owlsandarrows.nl>

**See Also**

"[tabfreq](#)", [simugeno](#) for creating a `simugeno` object from a `tabfreq` object.

**Examples**

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

---

Tu	<i>Allele frequencies of 15 autosomal short tandem repeats loci on Chinese Tu ethnic minority group</i>
----	---

---

**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	<i>Virtual classes for forensim</i>
----------------	-------------------------------------

---

**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 <hinda@owlsandarrows.nl>

---

wrapdataL	<i>ML estimate of number of contributors for SNPs</i>
-----------	---

---

**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**

<code>fil</code>	Input file
<code>plotte</code>	If T, plot
<code>nInMixture</code>	Alternatives for number of contributors, say 1:5
<code>tit</code>	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=FALSE,outfile="sim.txt",ncont=3) #Simulates data
# run with writeFile = TRUE for plot
# aa<-simMixSNP(nSNP=5,writeFile=TRUE,outfile="sim.txt",ncont=3)
# res<-wrapdataL(fil="sim.txt") # Calculates and plots
```

# Index

## \* classes

simugeno, 37  
simumix, 39  
tabfreq, 45  
virtualClasses, 48

## \* datagen

forensim-package, 3  
Hbsimu, 14  
simPCR2, 34  
simPCR2TK, 35  
simufreqD, 36  
simugeno, 37  
simugeno constructor, 38  
simumix, 39  
simumix constructor, 40  
simupopD, 42  
tabfreq, 45  
tabfreq constructor, 46

## \* datasets

ngm, 30  
sgmNorway, 32  
strusa, 44  
strveneto, 45  
Tu, 47

## \* htest

A2.simu, 3  
A3.simu, 5  
A4.simu, 6  
dataL, 11  
lik, 14  
lik.loc, 16  
likestim, 17  
likestim.loc, 19  
likEvid, 20  
LR, 22  
LRmixTK, 24  
mastermix, 24  
mincontri, 26  
PE, 30

PV, 31

## \* manip

Accessors, 8  
changepop, 8  
forensim-package, 3  
naomitab, 28  
simugeno, 37  
simugeno constructor, 38  
simumix, 39  
simumix constructor, 40  
tabfreq, 45  
tabfreq constructor, 46

## \* misc

findfreq, 12  
findmax, 13  
nball, 29

## \* models

Cmn, 9  
comb, 10

\$, simugeno-method (Accessors), 8  
\$, simumix-method (Accessors), 8  
\$, tabfreq-method (Accessors), 8  
\$<-, simugeno-method (Accessors), 8  
\$<-, simumix-method (Accessors), 8  
\$<-, tabfreq-method (Accessors), 8

A2.simu, 3, 6, 7, 25

A3.simu, 4, 5, 7, 25

A4.simu, 4, 6, 6, 25

Accessors, 8

as.simugeno, 38

as.simugeno (simugeno constructor), 38

as.simumix, 40

as.simumix (simumix constructor), 40

as.tabfreq, 46

as.tabfreq (tabfreq constructor), 46

changepop, 8

characterOrNULL-class (virtualClasses),  
48

- Cmn, [9](#), [10](#)
- comb, [9](#), [10](#)
- dataL, [9](#), [11](#), [13](#)
- factorOrNULL-class (virtualClasses), [48](#)
- findfreq, [12](#)
- findmax, [13](#)
- forensim, [8](#)
- forensim (forensim-package), [3](#)
- forensim-package, [3](#)
- Hbsimu, [14](#)
- is.simugeno, [38](#)
- is.simugeno (simugeno constructor), [38](#)
- is.simumix, [40](#)
- is.simumix (simumix constructor), [40](#)
- is.tabfreq, [46](#)
- is.tabfreq (tabfreq constructor), [46](#)
- lik, [11](#), [14](#), [17](#)
- lik.loc, [11](#), [15](#), [16](#)
- likestim, [15](#), [17](#), [17](#), [20](#), [26](#), [32](#)
- likestim.loc, [15](#), [17](#), [18](#), [19](#)
- likEvid, [20](#)
- listOrdataframe-class (virtualClasses), [48](#)
- LR, [22](#)
- LRmixTK, [21](#), [23](#), [24](#)
- mastermix, [24](#)
- matrixOrdataframe-class (virtualClasses), [48](#)
- mincontri, [26](#)
- N2error, [27](#)
- N2Exact, [27](#)
- names, simugeno-method (simugeno), [37](#)
- names, simumix-method (simumix), [39](#)
- names, tabfreq-method (tabfreq), [45](#)
- naomitab, [28](#)
- nball, [29](#)
- ngm, [30](#)
- PE, [30](#)
- print, simugeno-method (simugeno), [37](#)
- print, simumix-method (simumix), [39](#)
- print, tabfreq-method (tabfreq), [45](#)
- PV, [31](#)
- sgmNorway, [32](#)
- show, simugeno-method (simugeno), [37](#)
- show, simumix-method (simumix), [39](#)
- show, tabfreq-method (tabfreq), [45](#)
- simMixSNP, [33](#)
- simPCR2, [34](#), [36](#)
- simPCR2TK, [35](#), [35](#)
- simufreqD, [36](#), [42](#), [43](#)
- simugeno, [3](#), [8](#), [37](#), [38–41](#), [46](#), [47](#)
- simugeno (simugeno constructor), [38](#)
- simugeno constructor, [38](#)
- simugeno-class (simugeno), [37](#)
- simugeno-methods (simugeno constructor), [38](#)
- simumix, [3](#), [8](#), [12](#), [26](#), [29](#), [30](#), [38](#), [39](#), [40](#), [41](#)
- simumix (simumix constructor), [40](#)
- simumix constructor, [40](#)
- simumix-class (simumix), [39](#)
- simumix-methods (simumix constructor), [40](#)
- simupopD, [37](#), [42](#)
- strusa, [44](#)
- strveneto, [45](#)
- tabfreq, [3](#), [8](#), [29](#), [38–40](#), [45](#), [46](#), [47](#)
- tabfreq (tabfreq constructor), [46](#)
- tabfreq constructor, [46](#)
- tabfreq-class (tabfreq), [45](#)
- tabfreq-methods (tabfreq constructor), [46](#)
- Tu, [47](#)
- vectorOrdataframe-class (virtualClasses), [48](#)
- vectorOrNULL-class (virtualClasses), [48](#)
- virtualClasses, [48](#)
- wrapdataL, [48](#)