

Package: mispitoools (via r-universe)

September 9, 2024

Title Missing Person Identification Tools

Version 1.2.0

Description An open source software package written in R statistical language. It consist in a set of decision making tools to conduct missing person searches. Particularly, it allows computing optimal LR threshold for declaring potential matches in DNA-based database search. More recently 'mispitoools' incorporates preliminary investigation data based LRs. Statistical weight of different traces of evidence such as biological sex, age and hair color are presented. For citing mispitoools please use the following references: Marsico and Caridi, 2023 <[doi:10.1016/j.fsigen.2023.102891](https://doi.org/10.1016/j.fsigen.2023.102891)> and Marsico, Vigeland et al. 2021 <[doi:10.1016/j.fsigen.2021.102519](https://doi.org/10.1016/j.fsigen.2021.102519)>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Imports forrel, pedtools, dplyr, tidyr, tidyverse, DirichletReg, stats, purrr, patchwork, reshape2, graphics, ggplot2, shiny

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

URL <https://github.com/MarsicoFL/mispitoools>

BugReports <https://github.com/MarsicoFL/mispitoools/issues>

Depends R (>= 2.10)

Repository <https://marsicofl.r-universe.dev>

RemoteUrl <https://github.com/marsicofl/mispitoools>

RemoteRef HEAD

RemoteSha d531f7639bc64976babe9f56946acd22364f7a72

Contents

Argentina	3
Asia	3
Austria	3
bidirectionalKL	4
BosniaHerz	4
China	5
Cmodel	5
combLR	6
compute_LR_colors	7
conditionedProp	8
CondPlot	8
CPT_MP	9
CPT_POP	10
deplot	10
DeT	11
Europe	12
getfreqs	12
Japan	13
klPIE	13
LRage	14
LRcol	15
LRcolors	16
LRdate	16
LRdist	17
LRsex	18
makeMPprelim	19
makePOIgen	20
makePOIprelim	21
mispiApp	22
multi_kl_divergence	22
postSim	23
refProp	24
simLR2dataframe	25
simLRgen	25
simLRprelim	26
simRef	27
Trates	28
USA	28

Index

29

Argentina	<i>STRs allelic frequencies from specified country.</i>
-----------	---

Description

STRs allelic frequencies from specified country.

Usage

Argentina

Format

A data frame allele frequencies

Asia	<i>STRs allelic frequencies from specified country.</i>
------	---

Description

A dataset of allele frequencies.

Usage

Asia

Format

A data frame allele frequencies

Austria	<i>STRs allelic frequencies from specified country.</i>
---------	---

Description

STRs allelic frequencies from specified country.

Usage

Austria

Format

A data frame allele frequencies

bidirectionalKL	<i>Kullback-Leibler Divergence Calculation for Genetic Markers</i>
-----------------	--

Description

This function calculates the Kullback-Leibler divergence for shared genetic markers between two populations, considering allele frequencies. It normalizes data, adjusts zero frequencies, and calculates divergence in both directions.

Usage

```
bidirectionalKL(data1, data2, minFreq = 1e-10)
```

Arguments

data1	DataFrame with allele frequencies for the first population.
data2	DataFrame with allele frequencies for the second population.
minFreq	Minimum frequency to be considered for unobserved or poorly observed alleles.

Value

A list containing the Kullback-Leibler divergence from data1 to data2 and vice versa.

Examples

```
bidirectionalKL(Argentina, BosniaHerz)
```

BosniaHerz	<i>STRs allelic frequencies from specified country.</i>
------------	---

Description

STRs allelic frequencies from specified country.

Usage

```
BosniaHerz
```

Format

A data frame allele frequencies

China	<i>STRs allelic frequencies from specified country.</i>
-------	---

Description

STRs allelic frequencies from specified country.

Usage

China

Format

A data frame allele frequencies

Cmodel	<i>Epsilon hair color matrix</i>
--------	----------------------------------

Description

Epsilon hair color matrix

Usage

```
Cmodel(
  errorModel = c("custom", "uniform")[1],
  ep = 0.01,
  ep12 = 0.01,
  ep13 = 0.005,
  ep14 = 0.01,
  ep15 = 0.003,
  ep23 = 0.01,
  ep24 = 0.003,
  ep25 = 0.01,
  ep34 = 0.003,
  ep35 = 0.003,
  ep45 = 0.01
)
```

Arguments

errorModel	custom allows selecting a specific epsilon for each MP-UHR pair, uniform use ep for all.
ep	epsilon
ep12	epsilon

ep13	epsilon
ep14	epsilon
ep15	epsilon
ep23	epsilon
ep24	epsilon
ep25	epsilon
ep34	epsilon
ep35	epsilon
ep45	epsilon

Value

A value of Likelihood ratio based on preliminary investigation data. In this case, sex.

Examples

```
Cmodel()
```

combLR	<i>Combine LRs: a function for combining LRs obtained from simulations.</i>
--------	---

Description

Combine LRs: a function for combining LRs obtained from simulations.

Usage

```
combLR(LRdatasim1, LRdatasim2)
```

Arguments

LRdatasim1	A data frame object with the results of simulations. Outputs from simLRgen or simLRprelim functions.
LRdatasim2	A second data frame object with the results of simulations. Outputs from simLRgen or simLRprelim functions.

Value

An object of class data.frame combining the LRs obtained from simulations (the function multiplies the LRs).

Examples

```
library(mispitools)
library(forrel)
x = linearPed(2)
x = setMarkers(x, locusAttributes = NorwegianFrequencies[1:5])
x = profileSim(x, N = 1, ids = 2)
LRdatasim1 = simLRgen(x, missing = 5, 10, 123)
LRdatasim2 = simLRprelim("sex")
combLR(LRdatasim1,LRdatasim2)
```

compute_LR_colors	<i>Compute Likelihood Ratios based con color characteristics</i>
-------------------	--

Description

This function calculates the Likelihood Ratios (LRs) for each combination of hair colour, skin colour, and eye colour between two datasets. It assumes one dataset (conditioned) contains numerators and the other (unconditioned) contains denominators.

Usage

```
compute_LR_colors(conditioned, unconditioned)
```

Arguments

conditioned	A dataframe with at least the columns 'hair_colour', 'skin_colour', 'eye_colour', and 'numerators'.
unconditioned	A dataframe with at least the columns 'hair_colour', 'skin_colour', 'eye_colour', and 'f_h_s_y'.

Value

A dataframe with the merged data and computed LRs.

Examples

```
data <- simRef()
conditioned <- conditionedProp(data, 1, 1, 1, 0.01, 0.01, 0.01)
unconditioned <- refProp(data)
compute_LR_colors(conditioned, unconditioned)
```

conditionedProp	<i>Compute Conditioned Proportions for UPs</i>
-----------------	--

Description

This function calculates the conditioned proportions for pigmentation traits for UP, when UP is MP. It considers error rates for observations of hair color, skin color, and eye color.

Usage

```
conditionedProp(data, h, s, y, eh, es, ey)
```

Arguments

data	A data.frame containing the characteristics of UPs.
h	An integer representing the MP's hair color.
s	An integer representing the MP's skin color.
y	An integer representing the MP's eye color.
eh	A numeric value representing the error rate for observing hair color.
es	A numeric value representing the error rate for observing skin color.
ey	A numeric value representing the error rate for observing eye color.

Value

A numeric vector containing the conditioned proportion (numerator) for each individual in the dataset. These values are calculated based on the probability of observing the given combination of characteristics in the MP, compared to each UP.

CondPlot	<i>General plot for conditioned probabilities and LR combining variables</i>
----------	--

Description

General plot for conditioned probabilities and LR combining variables

Usage

```
CondPlot(CPT_POP, CPT_MP)
```

Arguments

CPT_POP	Population conditioned probability table
CPT_MP	Missing person conditioned probability table

Value

A value of Likelihood ratio based on preliminary investigation data. In this case, sex.

Examples

```
Cmodel()
```

CPT_MP	<i>Missing person based conditioned probability</i>
--------	---

Description

Missing person based conditioned probability

Usage

```
CPT_MP(MPs = "F", MPc = 1, eps = 0.05, epa = 0.05, epc = Cmodel())
```

Arguments

MPs	Missing person sex
MPc	Missing person hair color
eps	sex epsilon
epa	age epsilon - Age is not specified in this first version, because it assumes uniformity.
epc	color model

Value

A value of Likelihood ratio based on preliminary investigation data. In this case, sex.

Examples

```
CPT_MP()
```

CPT_POP

Population based conditioned probability

Description

Population based conditioned probability

Usage

```
CPT_POP(
  propS = c(0.5, 0.5),
  MPa = 40,
  MPr = 6,
  propC = c(0.3, 0.2, 0.25, 0.15, 0.1)
)
```

Arguments

propS	age epsilon - Age is not specified in this first version, because it assumes uniformity.
MPa	Missing person sex
MPr	Missing person hair color
propC	sex epsilon

Value

A value of Likelihood ratio based on preliminary investigation data. In this case, sex.

Examples

```
CPT_POP()
```

deplot

Decision making plot: a function for plotting false positive and false negative rates for each LR threshold.

Description

Decision making plot: a function for plotting false positive and false negative rates for each LR threshold.

Usage

```
deplot(datasim, LRmax = 1000)
```

Arguments

datasim	Input dataframe containing expected LR _s for related and unrelated POIs. It should be the output from makeLRsims function.
LRmax	Maximum LR value used as a threshold. 1000 setted by default.

Value

A plot showing false positive and false negative rates for each likelihood ratio threshold.

Examples

```
library(forrel)
x = linearPed(2)
x = setMarkers(x, locusAttributes = NorwegianFrequencies[1:5])
x = profileSim(x, N = 1, ids = 2)
datasim = simLRgen(x, missing = 5, 10, 123)
deplot(datasim)
```

DeT	<i>Decision Threshold: a function for computing likelihood ratio decision threshold.</i>
-----	--

Description

Decision Threshold: a function for computing likelihood ratio decision threshold.

Usage

```
DeT(datasim, weight)
```

Arguments

datasim	Input dataframe containing expected LR _s for related and unrelated POIs. It should be the output from makeLRsims function.
weight	The differential weight between false positives and false negatives. A value of 10 is suggested.

Value

A value of Likelihood ratio suggested as threshold based on false positive-false negative trade-off.

Examples

```
library(forrel)
x = linearPed(2)
x = setMarkers(x, locusAttributes = NorwegianFrequencies[1:5])
x = profileSim(x, N = 1, ids = 2)
datasim = simLRgen(x, missing = 5, 10, 123)
DeT(datasim, 10)
```

Europe	<i>STRs allelic frequencies from specified country.</i>
--------	---

Description

STRs allelic frequencies from specified country.

Usage

Europe

Format

A data frame allele frequencies

getfreqs	<i>Function for getting STR allele frequencies from different world populations.</i>
----------	--

Description

Function for getting STR allele frequencies from different world populations.

Usage

```
getfreqs(region)
```

Arguments

region	select the place of the allele frequency database. Possible values are listed: "Argentina", "Asia", "Europe", "USA", "Austria", "BosniaHerz", "China" and "Japan".
--------	--

Value

An allele frequency database adapted compatible with pedtools format.

Source

<https://doi.org/10.1016/j.fsigss.2009.08.178>; <https://doi.org/10.1016/j.fsigen.2016.06.008>; <https://doi.org/10.1016/j.fsigen.2016.06.008>

Japan	<i>STRs allelic frequencies from specified country.</i>
-------	---

Description

STRs allelic frequencies from specified country.

Usage

Japan

Format

A data frame allele frequencies

k1PIE	<i>Calculate Kullback-Leibler Divergence with Base 10 Logarithm</i>
-------	---

Description

This function computes the Kullback-Leibler (KL) divergence between two probability distributions represented by matrices, using a base 10 logarithm. The function calculates KL divergence in both directions ($P \parallel Q$ and $Q \parallel P$) and handles zero probabilities by replacing them with a minimum value to avoid undefined logarithms.

Usage

```
k1PIE(P, Q, min_value = 1e-12)
```

Arguments

P	A numeric matrix representing the first probability distribution. The entire matrix should sum to 1.
Q	A numeric matrix representing the second probability distribution. The entire matrix should sum to 1.
min_value	A numeric value representing the minimum value to replace any zero probabilities. Defaults to 1e-12.

Value

A named numeric vector with two elements:

"P \parallel Q" The KL divergence from P to Q ($P \parallel Q$).

"Q \parallel P" The KL divergence from Q to P ($Q \parallel P$).

LRage

Likelihood ratio for age variable

Description

Likelihood ratio for age variable

Usage

```
LRage(
  MPa = 40,
  MPr = 6,
  UHRr = 1,
  gam = 0.07,
  nsims = 1000,
  epa = 0.05,
  erRa = epa,
  H = 1,
  modelA = c("uniform", "custom")[1],
  LR = FALSE,
  seed = 1234
)
```

Arguments

MPa	Missing person age
MPr	Missing person age range.
UHRr	Unidentified person range
gam	Simulation parameter for UHR ages.
nsims	number of simulations.
epa	epsilon age
erRa	error rate in the database.
H	hypothesis tested, H1: UHR is MP, H2: UHR is not MP.
modelA	reference database probabilities, uniform assumes equally probable ages. Custom needs a vector with ages frequencies.
LR	compute LR values
seed	For reproducible simulations

Value

A value of Likelihood ratio based on preliminary investigation data. In this case, Age.

`LRcol`*Likelihood ratio for color variable*

Description

Likelihood ratio for color variable

Usage

```
LRcol(  
  MPc = 1,  
  epc = Cmodel(),  
  erRc = epc,  
  nsims = 1000,  
  Pc = c(0.3, 0.2, 0.25, 0.15, 0.1),  
  H = 1,  
  Qprop = MPc,  
  LR = FALSE,  
  seed = 1234  
)
```

Arguments

<code>MPc</code>	MP hair color
<code>epc</code>	epsilon parameter.
<code>erRc</code>	error rate in the database.
<code>nsims</code>	number of simulations performed.
<code>Pc</code>	hair color probabilities.
<code>H</code>	hypothesis tested, H1: UHR is MP, H2: UHR is no MP
<code>Qprop</code>	Query color tested.
<code>LR</code>	compute LR values
<code>seed</code>	For reproducible simulations

Value

A value of Likelihood ratio based on preliminary investigation data. In this case, hair color.

Examples

```
LRcol()
```

LRcolors

Simulate LR values considering H1 and H2

Description

Simulate LR values considering H1 and H2

Usage

```
LRcolors(df, seed = 1234, nsim = 500)
```

Arguments

df	A data.frame containing the characteristics of individuals, numerator, f_h_s_y and LRs. Output from compute_LRs function.
seed	For replication purposes.
nsim	Number of LRs simulated.

Value

LR distribution considering H1 (Related) and H2 (Unrelated).

LRdate

Likelihood ratio for birth date in missing person searches

Description

Likelihood ratio for birth date in missing person searches

Usage

```
LRdate(
  ABD = "1976-05-31",
  DBD = "1976-07-15",
  PrelimData,
  alpha = c(1, 4, 60, 11, 6, 4, 4),
  cuts = c(-120, -30, 30, 120, 240, 360),
  draw = 500,
  type = 1,
  seed = 123
)
```


Arguments

ABD	Actual birth date of the missing person.
DBD	Declared birth date of the person of interest.
PrelimData	Used when type = 2, is the dataframe with the DBD of the persons of interest in the database.
alpha	A vector containing the alpha values for the dirichlet. It should contain the number of categories of differences between DBD and ABD.
cuts	Value of differences between DBD and ABD used for category definition.
draw	Number of simulations for Dirichlet distribution computation.
type	Type of scenario, type 1 is an "open search", where it is unknown if the missing person is in the database. Type 2 refers to a scenario where the missing person is in the database.
seed	Seed for simulations.

Value

A value of Likelihood ratio based on preliminary investigation data. In this case, birth date.

Examples

```
library(DirichletReg)
LRdate(ABD = "1976-05-31", DBD = "1976-07-15",
PrelimData, alpha = c(1, 4, 60, 11, 6, 4, 4),
cuts = c(-120, -30, 30, 120, 240, 360),
type = 1, seed = 123)
```

LRdist	<i>Likelihood ratio distribution: a function for plotting expected log₁₀(LR) distributions under relatedness and unrelatedness.</i>
--------	--

Description

Likelihood ratio distribution: a function for plotting expected log₁₀(LR) distributions under relatedness and unrelatedness.

Usage

```
LRdist(datasim)
```

Arguments

datasim	Input dataframe containing expected LRs for related and unrelated POIs. It should be the output from makeLRsims function.
---------	---

Value

A plot showing likelihood ratio distributions under relatedness and unrelatedness hypothesis.

Examples

```
library(forrel)
x = linearPed(2)
x = setMarkers(x, locusAttributes = NorwegianFrequencies[1:5])
x = profileSim(x, N = 1, ids = 2)
datasim = simLRgen(x, missing = 5, 10, 123)
LRdist(datasim)
```

LRsex

Likelihood ratio for sex variable

Description

Likelihood ratio for sex variable

Usage

```
LRsex(
  MPs = "F",
  eps = 0.05,
  erRs = eps,
  nsims = 1000,
  Ps = c(0.5, 0.5),
  H = 1,
  LR = FALSE,
  seed = 1234
)
```

Arguments

MPs	MP sex
eps	epsilon parameter.
erRs	error rate in the database.
nsims	number of simulations performed.
Ps	Sex probabilities in the population.
H	hypothesis tested, H1: UHR is MP, H2: UHR is no MP
LR	compute LR values
seed	For reproducible simulations

Value

A value of Likelihood ratio based on preliminary investigation data. In this case, sex.

Examples

```
LRsex()
```

makeMPprelim	<i>Make preliminary investigation MP data simulations: a function for obtaining a database of preliminary investigation data for a missing person search.</i>
--------------	---

Description

Make preliminary investigation MP data simulations: a function for obtaining a database of preliminary investigation data for a missing person search.

Usage

```
makeMPprelim(
  casetype = "children",
  dateinit = "1975/01/01",
  scenario = 1,
  femaleprop = 0.5,
  ext = 100,
  numsims = 10000,
  seed = 123,
  region = c("North America", "South America", "Africa", "Asia", "Europe", "Oceania"),
  regionprob = c(0.2, 0.2, 0.2, 0.1, 0.2, 0.1)
)
```

Arguments

casetype	Type of missing person search case. Two options are available: "migrants" or "children".
dateinit	Minimum birth date of simulated missing person. Casetype: Children.
scenario	Birth date distribution scenarios: (1) non-uniform, (2) uniform. Casetype: Children.
femaleprop	Proportion of females. Casetype: All.
ext	Time extension for minimum birth date, range in scenario 1 and days in scenario 2. Casetype: Children.
numsims	Number of simulated MPs. Casetype: All.
seed	Select a seed for simulations. If it is defined, results will be reproducible. Casetype: All.
region	Birth region or place in missing children case or place of place of the last seen in missing migrant case. Casetype: All.
regionprob	Region proportions. Casetype: All.

Value

An object of class `data.frame` with preliminary investigation data.

Examples

```
makeMPprelim()
```

makePOIgen	<i>Make POIs gen: a function for obtaining a database with genetic information from simulated POIs or UHRs.</i>
------------	---

Description

Make POIs gen: a function for obtaining a database with genetic information from simulated POIs or UHRs.

Usage

```
makePOIgen(numsims = 100, reference, seed = 123)
```

Arguments

numsims	Number of simulations performed (number of POIs or UHRs).
reference	Indicate the reference STRs/SNPs frequency database used for simulations.
seed	Select a seed for simulations. If it is defined, results will be reproducible. Suggested, seed = 123

Value

An object of class `data.frame` with genetic information from POIs (randomly sampled from the frequency database).

Examples

```
library(forrel)
freqdata <- getfreqs(Argentina)
makePOIgen(numsims = 100, reference = freqdata, seed = 123)
```

makePOIprelim	<i>Make preliminary investigation POI/UHR data simulations: a function for obtaining a database of preliminary investigation data for a missing person search.</i>
---------------	--

Description

Make preliminary investigation POI/UHR data simulations: a function for obtaining a database of preliminary investigation data for a missing person search.

Usage

```
makePOIprelim(
  casetype = "children",
  dateinit = "1975/01/01",
  scenario = 1,
  femaleprop = 0.5,
  ext = 100,
  numsims = 10000,
  seed = 123,
  birthprob = c(0.09, 0.9, 0.01),
  region = c("North America", "South America", "Africa", "Asia", "Europe", "Oceania"),
  regionprob = c(0.2, 0.2, 0.2, 0.1, 0.2, 0.1)
)
```

Arguments

casetype	Type of missing person search case. Two options are available: "migrants" or "children".
dateinit	Minimum birth date of simulated persons of interest. Casetype: Children.
scenario	Birth date distribution scenarios: (1) non-uniform, (2) uniform. Casetype: Children.
femaleprop	Proportion of females. Casetype: All.
ext	Time extension for minimum birth date, range in scenario 1 and days in scenario 2. Casetype: Children.
numsims	Number of simulated POIs/UHRs. Casetype: All.
seed	Select a seed for simulations. If it is defined, results will be reproducible. Casetype: All.
birthprob	Birth type probabilities: home birth, hospital birth and unknown-adoption. Casetype: Children.
region	Birth region or place in missing children case or place of discovery of the human remain in missing migrant case. Casetype: All.
regionprob	Region proportions. Casetype: All.

Value

An object of class `data.frame` with preliminary investigation data.

Examples

```
makePOIprelim(
  dateinit = "1975/01/01",
  scenario = 1,
  femaleprop = 0.5,
  ext = 100,
  numsims = 10000,
  seed = 123,
  birthprob = c(0.09, 0.9, 0.01),
  region = c("North America", "South America", "Africa", "Asia", "Europe", "Oceania"),
  regionprob = c(0.2, 0.2, 0.2, 0.1, 0.2, 0.1))
```

`mispiApp`

Missing person shiny app

Description

Missing person shiny app

Usage

```
mispiApp()
```

Value

An user interface for computing non-genetic LRs and conditioned probability tables.

Examples

```
CPT_MP()
```

`multi_kl_divergence`

Multi-dataset Kullback-Leibler Divergence Calculation

Description

This function calculates the Kullback-Leibler divergence for all pairs of provided datasets, considering allele frequencies. It normalizes data, adjusts zero frequencies, and computes KL divergence in both directions for each pair.

Usage

```
multi_kl_divergence(datasets, minFreq = 1e-10)
```

Arguments

`datasets` List of dataframes, each containing allele frequencies for different populations.
`minFreq` Minimum frequency to be considered for unobserved or poorly observed alleles.

Value

A matrix containing the Kullback-Leibler divergence for each dataset pair.

Examples

```
kl_matrix <- multi_kl_divergence(list(Argentina, BosniaHerz, Europe))
```

postSim

postSim: A function for simulating posterior odds

Description

postSim: A function for simulating posterior odds

Usage

```
postSim(
  datasim,
  Prior = 0.01,
  PriorModel = c("prelim", "uniform")[1],
  eps = 0.05,
  erRs = 0.01,
  epc = Cmodel(),
  erRc = Cmodel(),
  MPc = 1,
  epa = 0.05,
  erRa = 0.01,
  MPa = 10,
  MPr = 2
)
```

Arguments

`datasim` Output from simLRgen function.
`Prior` Prior probability for H1
`PriorModel` Prior odds model: "prelim" is based on preliminary data, and "uniform" uses only the prior probability of H1

eps	epsilon parameter sex
erRs	error parameter sex
epc	epsilon parameter hair color
erRc	error parameter hair color
MPc	Missing person hair color
epa	epsilon parameter age
erRa	error parameter age
MPa	Missing person age
MPr	Missing person age error range

Value

A value of posterior odds.

Examples

```
library(forrel)
x = linearPed(2)
plot(x)
x = setMarkers(x, locusAttributes = NorwegianFrequencies[1:5])
x = profileSim(x, N = 1, ids = 2)
datasim = simLRgen(x, missing = 5, 10, 123)
postSim(datasim)
```

refProp	<i>Generate a dataframe with hair colour, skin colour, eye colour and their specific combination frequencies</i>
---------	--

Description

This function creates a dataframe that lists every unique combination of hair colour, skin colour, and eye colour in the provided dataset, along with the proportion of occurrences of each combination.

Usage

```
refProp(data)
```

Arguments

data A data.frame containing the characteristics of individuals.

Value

A data.frame with columns for hair_colour, skin_colour, eye_colour, and f_h_s_y.

Examples

```
data <- simRef(1000)
refProp(data)
```

simLR2dataframe	<i>simLR2dataframe: A function for extracting LR distributions in a dataframe from simLRgen() output.</i>
-----------------	---

Description

simLR2dataframe: A function for extracting LR distributions in a dataframe from simLRgen() output.

Usage

```
simLR2dataframe(datasim)
```

Arguments

datasim	Input dataframe containing expected LRs for related and unrelated POIs. It should be the output from makeLRsims function.
---------	---

Value

A dataframe with LR values obtained from simulations.

simLRgen	<i>Simulate likelihoods ratio (LRs) based on genetic data: a function for obtaining expected LRs under relatedness and unrelatedness kinship hypothesis.</i>
----------	--

Description

Simulate likelihoods ratio (LRs) based on genetic data: a function for obtaining expected LRs under relatedness and unrelatedness kinship hypothesis.

Usage

```
simLRgen(reference, missing, numsims, seed, numCores = 1)
```

Arguments

reference	Reference pedigree. It could be an input from read_fam() function or a pedigree built with pedtools.
missing	Missing person ID/label indicated in the pedigree.
numsims	Number of simulations performed.
seed	Select a seed for simulations. If it is defined, results will be reproducible. Suggested, seed = 123
numCores	Enables parallelization

Value

An object of class data.frame with LRs obtained for both hypothesis, Unrelated where POI is not MP or Related where POI is MP.

Examples

```
library(forrel)
x = linearPed(2)
plot(x)
x = setMarkers(x, locusAttributes = NorwegianFrequencies[1:5])
x = profileSim(x, N = 1, ids = 2)
datasim = simLRgen(x, missing = 5, 10, 123)
```

simLRprelim	<i>Simulate likelihoods ratio (LRs) based on preliminary investigation data: a function for obtaining expected LRs under relatedness and unrelatedness kinship hypothesis.</i>
-------------	--

Description

Simulate likelihoods ratio (LRs) based on preliminary investigation data: a function for obtaining expected LRs under relatedness and unrelatedness kinship hypothesis.

Usage

```
simLRprelim(
  vartype,
  numsims = 1000,
  seed = 123,
  int = 5,
  ErrorRate = 0.05,
  alphaBdate = c(1, 4, 60, 11, 6, 4, 4),
  numReg = 6,
  MP = NULL,
  database,
  cuts = c(-120, -30, 30, 120, 240, 360)
)
```

Arguments

vartype	Indicates type of preliminary investigation variable. Options are: sex, region, age, birthDate and height.
numsims	Number of simulations performed.
seed	Seed for simulations.
int	Interval parameter, used for height and age vartypes. It defines the estimation range, for example, if MP age is 55, and int is 10, the estimated age range will be between 45 and 65.

ErrorRate	Error rate for sex, region, age and Height LR calculations.
alphaBdate	Vector containing alpha parameters for Dirichlet distribution. Usually they are the frequencies of the solved cases in each category.
numReg	Number of regions present in the case.
MP	Introduce the preliminary data of the selected variable (vartype) of the MP. If it is null, open search is carried out. If it is not NULL, close search LR is computed. Variables values must be named as those presented in makePOIprelim function.
database	It is used when the close search (MP not NULL), is carried out. It could be the output from makePOIprelim or a database with the same structure.
cuts	Value of differences between DBD and ABD used for category definition. They must be the same as the ones selected for alphaBdate vector.

Value

An object of class data.frame with LRs obtained for both hypothesis, Unrelated where POI/UHR is not MP or Related where POI/UHR is MP.

Examples

```
library(mispitools)
simLRprelim("sex")
```

simRef	<i>Generate Reference Properties for a Hypothetical Population</i>
--------	--

Description

This function simulates a dataset representing physical characteristics (hair color, skin color, eye color) of a hypothetical population, based on conditional probability distributions. The size of the simulated population can be adjusted by the user.

Usage

```
simRef(n = 1000, seed = 1234)
```

Arguments

n	The number of individuals in the simulated population.
seed	Selected seed for simulations.

Value

A data.frame with three columns: hair_colour, skin_colour, and eye_colour, each representing the respective characteristics of each individual in the sample population. The hair color is simulated based on predefined probabilities, and skin and eye colors are generated conditionally based on the hair color.

Examples

```
simRef(1000) # Generates a data frame with 1000 entries based on the defined distributions.
```

Trates	<i>Threshold rates: a function for computing error rates and Matthews correlation coefficient of a specific LR threshold.</i>
--------	---

Description

Threshold rates: a function for computing error rates and Matthews correlation coefficient of a specific LR threshold.

Usage

```
Trates(datasim, threshold)
```

Arguments

datasim	Input dataframe containing expected LRs for related and unrelated POIs. It should be the output from makeLRsims function.
threshold	Likelihood ratio threshold selected for error rates calculation.

Value

Values of false positive and false negative rates and MCC for a specific LR threshold.

Examples

```
library(forrel)
x = linearPed(2)
x = setMarkers(x, locusAttributes = NorwegianFrequencies[1:5])
x = profileSim(x, N = 1, ids = 2)
datasim = simLRgen(x, missing = 5, 10, 123)
Trates(datasim, 10)
```

USA	<i>STRs allelic frequencies from specified country.</i>
-----	---

Description

STRs allelic frequencies from specified country.

Usage

```
USA
```

Format

A data frame allele frequencies

Index

* datasets

Argentina, [3](#)
Asia, [3](#)
Austria, [3](#)
BosniaHerz, [4](#)
China, [5](#)
Europe, [12](#)
Japan, [13](#)
USA, [28](#)

Argentina, [3](#)
Asia, [3](#)
Austria, [3](#)

bidirectionalKL, [4](#)
BosniaHerz, [4](#)

China, [5](#)
Cmodel, [5](#)
combLR, [6](#)
compute_LRs_colors, [7](#)
conditionedProp, [8](#)
CondPlot, [8](#)
CPT_MP, [9](#)
CPT_POP, [10](#)

deplot, [10](#)
DeT, [11](#)

Europe, [12](#)

getfreqs, [12](#)

Japan, [13](#)

k1PIE, [13](#)

LRage, [14](#)
LRcol, [15](#)
LRcolors, [16](#)
LRdate, [16](#)

LRdist, [17](#)
LRsex, [18](#)

makeMPprelim, [19](#)
makePOIgen, [20](#)
makePOIprelim, [21](#)
mispiApp, [22](#)
multi_kl_divergence, [22](#)

postSim, [23](#)

refProp, [24](#)

simLR2dataframe, [25](#)
simLRgen, [25](#)
simLRprelim, [26](#)
simRef, [27](#)

Trates, [28](#)

USA, [28](#)