Package 'mispitools'

May 8, 2026

Title Missing Person Identification Tools

Version 1.4.0

Author Franco Marsico [aut, cre] (ORCID:
<https://orcid.org/0000-0002-0740-5516>)

Maintainer Franco Marsico <franco.lmarsico@gmail.com>

Description A comprehensive toolkit for missing person identification combining genetic and non-genetic evidence within a Bayesian framework. Computes likelihood ratios (LRs) for DNA profiles, biological sex, age, hair color, and birthdate evidence. Provides decision analysis tools including optimal LR thresholds, error rate calculations, and ROC curve visualization. Includes interactive Shiny applications for exploring evidence combinations. For methodological details see Marsico et al. (2023) <doi:10.1016/j.fsigen.2023.102891> and Marsico, Vigeland et al. (2021) <doi:10.1016/j.fsigen.2021.102519>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports forrel, pedtools, dplyr, tidyr, stats, reshape2, patchwork, graphics, ggplot2, DirichletReg, pROC, shiny, shinythemes

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

RoxygenNote 7.3.3

URL https://github.com/MarsicoFL/mispitools

BugReports https://github.com/MarsicoFL/mispitools/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2026-01-09 09:40:15 UTC

Contents

mispitools-package *mispitools: Missing Person Identification Tools*

Description

The mispitools package provides a comprehensive suite of statistical tools for missing person identification, combining both genetic and non-genetic evidence. It enables forensic geneticists and investigators to compute likelihood ratios (LRs), determine optimal decision thresholds, and assess error rates in database searches.

The package implements Bayesian methodology for evaluating evidence in kinship testing, particularly useful in humanitarian contexts such as identifying victims of enforced disappearances or natural disasters.

Simulation Functions

Functions for simulating LR distributions under different hypotheses:

- `sim_lr_genetic`: Simulate LRs from genetic (DNA) data
- `sim_lr_prelim`: Simulate LRs from preliminary investigation data
- `sim_posterior`: Simulate posterior odds combining evidence
- `sim_reference_pop`: Simulate a reference population with traits
- `sim_poi_genetic`: Generate genetic profiles for persons of interest
- `sim_poi_prelim`: Generate preliminary data for persons of interest
- `sim_mp_prelim`: Generate preliminary data for missing persons

LR Calculation Functions

Functions for computing likelihood ratios from different types of evidence:

- `lr_sex`: LR based on biological sex
- `lr_age`: LR based on age
- `lr_hair_color`: LR based on hair color
- `lr_pigmentation`: LR for combined pigmentation traits
- `lr_birthdate`: LR based on birth date discrepancies
- `lr_combine`: Combine LRs from independent sources
- `lr_to_dataframe`: Convert genetic LR simulations to dataframe

Conditional Probability Tables

Functions for computing conditional probability tables (CPTs):

- `cpt_population`: CPT based on population frequencies (H2)
- `cpt_missing_person`: CPT conditioned on MP characteristics (H1)
- `error_matrix_hair`: Create error/confusion matrix for hair color

Visualization Functions

Functions for visualizing results:

- `plot_lr_distribution`: Plot LR distributions under H1 and H2
- `plot_decision_curve`: Plot FPR vs FNR for different thresholds
- `plot_cpt`: Visualize conditional probability tables

Decision Analysis

Functions for determining optimal thresholds and error rates:

- `decision_threshold`: Compute optimal LR threshold
- `threshold_rates`: Compute error rates (FPR, FNR, MCC)

Population Genetics

Functions for working with allele frequency databases:

- `get_allele_freqs`: Retrieve allele frequencies for a population
- `kl_bidirectional`: Compute KL divergence between populations
- `kl_multi`: Compare multiple populations

Interactive Applications

Shiny applications for interactive analysis:

- `app_mispitools`: Comprehensive analysis application
- `app_lr_comparison`: LR comparison and ROC analysis

Datasets

The package includes STR allele frequency databases for multiple populations: `Argentina`, `Asia`, `Austria`, `BosniaHerz`, `China`, `Europe`, `Japan`, `USA`.

Core Dependencies

The genetic simulation functionality relies on the **forrel** and **pedtools** packages for pedigree handling and likelihood calculations.

Author(s)

Maintainer: Franco Marsico <franco.lmarsico@gmail.com> (ORCID)

References

Marsico FL, Iudica CE, Herrera Pinero F, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

Marsico FL, Vigeland MD, Egeland T, Herrera Pinero F (2021). "Making decisions in missing person identification cases with low statistical power." *Forensic Science International: Genetics*, 52, 102519. doi:10.1016/j.fsigen.2021.102519

See Also

Useful links:

- https://github.com/MarsicoFL/mispitools
- Report bugs at https://github.com/MarsicoFL/mispitools/issues

---

`Argentina` *STR Allele Frequencies from Argentina*

Description

Population allele frequency data for 24 autosomal Short Tandem Repeat (STR) markers from Argentina. These frequencies are used for calculating likelihood ratios in forensic genetics and missing person identification.

Usage

```
data(Argentina)
```

Format

A data frame with 93 rows (alleles) and 25 columns. First column is Allele (repeat number), remaining columns are allele frequencies for each STR marker.

Details

This dataset contains allele frequencies for the following 24 STR markers: D8S1179, D21S11, D7S820, CSF1PO, D3S1358, THO1, D13S317, D16S539, D2S1338, D19S433, VWA, TPOX, D18S51, D5S818, FGA, PENTAE, PENTAD, D12S391, D1S1656, D6S1043, D10S1248, D22S1045, D2S441, SE33.

The markers are compatible with common forensic STR kits including GlobalFiler, PowerPlex Fusion, and Investigator 24plex.

Source

Allele frequency data compiled from published Argentinian population studies. Format compatible with **pedtools** and **forrel** packages.

References

Marino M, et al. (2009). "Population genetic data for 15 STR loci in the Argentinian population." *Forensic Science International: Genetics Supplement Series*, 2(1), 369-370. doi:10.1016/j.fsigss.2009.08.178

See Also

get_allele_freqs for extracting frequencies, sim_lr_genetic for LR simulations using these frequencies.

Other frequency databases: Europe, Asia, USA, Austria, BosniaHerz, China, Japan

Examples

```
# Load the dataset
data(Argentina)

# View structure
head(Argentina)
dim(Argentina)

# List available markers
names(Argentina)[-1]

# Use with pedtools
library(forrel)
freqs <- get_allele_freqs(Argentina)
```

Asia *STR Allele Frequencies from Asian Populations*

Description

Population allele frequency data for 38 autosomal Short Tandem Repeat (STR) markers from Asian populations. This comprehensive dataset includes extended markers beyond core forensic loci.

Usage

```
data(Asia)
```

Format

A data frame with 98 rows (alleles) and 39 columns. First column is Allele (repeat number), remaining columns are allele frequencies for each STR marker.

Details

This dataset contains allele frequencies for 38 STR markers including both standard forensic core loci and extended markers: D1S1656, D2S1338, D2S441, D3S1358, D3S1744, D4S2366, D5S818, D5S2800, D6S474, D7S820, D7S3048, D8S1132, D8S1179, D9S1122, D10S1248, D11S2368, D12S391, D13S317, D13S325, D14S1434, D15S659, D16S539, D17S1301, D18S51, D18S1364, D19S253, D19S433, D20S482, D21S11, D21S2055, D22GATA198B05, D22S1045, CSF1PO, FGA, SE33, TH01, TPOX, VWA.

Source

Allele frequency data compiled from Asian population studies. Format compatible with **pedtools** and **forrel** packages.

References

Phillips C, et al. (2011). "Building a forensic STR allele frequency database." *Forensic Science International: Genetics Supplement Series*, 3(1), e69-e70. doi:10.1016/j.fsigss.2011.08.034

See Also

`get_allele_freqs` for extracting frequencies, `sim_lr_genetic` for LR simulations.

Other frequency databases: `Argentina`, `Europe`, `USA`, `Austria`, `BosniaHerz`, `China`, `Japan`

Examples

```
# Load the dataset
data(Asia)

# This dataset has more markers than others
ncol(Asia) - 1  # 38 markers
```

Austria *STR Allele Frequencies from Austria*

Description

Population allele frequency data for 16 autosomal Short Tandem Repeat (STR) markers from the Austrian population. Focused on core forensic markers used in European laboratories.

Usage

```
data(Austria)
```

Format

A data frame with 66 rows (alleles) and 17 columns. First column is Allele (repeat number), remaining columns are allele frequencies for each STR marker.

Details

This dataset contains allele frequencies for the following 16 STR markers: D1S1656, D2S1338, D2S441, D3S1358, D8S1179, D10S1248, D12S391, D16S539, D18S51, D19S433, D21S11, D22S1045, FGA, SE33, TH01, VWA.

These markers correspond to the European Standard Set (ESS) of forensic STR loci plus commonly used additional markers.

Source

Austrian population frequency data. Format compatible with **pedtools** and **forrel** packages.

References

Parson W, et al. (2008). "The EDNAP standardization of the NGM amplification kit." *Forensic Science International: Genetics Supplement Series*, 1(1), 183-184. doi:10.1016/j.fsigss.2007.10.062

See Also

`get_allele_freqs` for extracting frequencies, `sim_lr_genetic` for LR simulations.

Other European databases: `Europe`, `BosniaHerz`

Examples

```
# Load the dataset
data(Austria)

# View structure
head(Austria)

# Compare with Bosnia-Herzegovina (same marker set)
data(BosniaHerz)
identical(names(Austria), names(BosniaHerz))  # TRUE
```

BosniaHerz *STR Allele Frequencies from Bosnia and Herzegovina*

Description

Population allele frequency data for 16 autosomal Short Tandem Repeat (STR) markers from Bosnia and Herzegovina. These frequencies are particularly relevant for identification of missing persons from the Balkan conflicts.

Usage

```
data(BosniaHerz)
```

Format

A data frame with 63 rows (alleles) and 17 columns. First column is Allele (repeat number), remaining columns are allele frequencies for each STR marker.

Details

This dataset contains allele frequencies for the following 16 STR markers: D1S1656, D2S1338, D2S441, D3S1358, D8S1179, D10S1248, D12S391, D16S539, D18S51, D19S433, D21S11, D22S1045, FGA, SE33, TH01, VWA.

These markers correspond to the European Standard Set (ESS) of forensic STR loci. This database is particularly important for the ongoing identification efforts related to the Balkan wars.

Source

Population data from Bosnia and Herzegovina. Format compatible with **pedtools** and **forrel** packages.

References

Marjanovic D, et al. (2006). "Population data at 15 STR loci in the population of Bosnia and Herzegovina." *Journal of Forensic Sciences*, 51(5), 1190-1192. doi:10.1111/j.15564029.2006.00239.x

See Also

`get_allele_freqs` for extracting frequencies, `sim_lr_genetic` for LR simulations.

Other European databases: `Europe`, `Austria`

Examples

```
# Load the dataset
data(BosniaHerz)

# View structure
head(BosniaHerz)

# Calculate KL divergence with Austria
# kl_bidirectional(BosniaHerz, Austria)
```

China *STR Allele Frequencies from China*

Description

Comprehensive population allele frequency data for 70 autosomal Short Tandem Repeat (STR) markers from Chinese populations. This is one of the most extensive STR frequency databases available.

Usage

```
data(China)
```

Format

A data frame with 67 rows (alleles) and 71 columns. First column is Allele (repeat number), remaining columns are allele frequencies for each STR marker.

Details

This comprehensive dataset contains allele frequencies for 70 STR markers, including all standard forensic core loci plus an extensive set of additional markers. This enables very high discrimination power for identification purposes.

Core forensic markers included: CSF1PO, D1S1656, D2S441, D2S1338, D3S1358, D5S818, D7S820, D8S1179, D10S1248, D12S391, D13S317, D16S539, D18S51, D19S433, D21S11, D22S1045, FGA, TH01, TPOX, vWA, SE33.

Extended markers include: PENTA D, PENTA E, D6S1043, D4S2408, D9S1122, and many others for enhanced discrimination.

Source

Chinese population frequency data. Format compatible with **pedtools** and **forrel** packages.

References

Hu S, et al. (2015). "Population genetics of 17 Y-STR loci in the Han ethnic minority from Henan Province, Central China." *Forensic Science International: Genetics*, 19, e1-e2. doi:10.1016/j.fsigen.2015.05.005

See Also

`get_allele_freqs` for extracting frequencies, `sim_lr_genetic` for LR simulations.

Other Asian databases: `Asia`, `Japan`

Examples

```
# Load the dataset
data(China)

# This is one of the most comprehensive databases
ncol(China) - 1  # 70 markers

# Check common markers with other databases
common <- intersect(names(China), names(Japan))
length(common)  # Many shared markers
```

`compute_conditioned_prop` *Compute Conditioned Proportions for Pigmentation Traits*

Description

Calculates the conditioned proportions (numerator probabilities) for pigmentation trait combinations given the missing person's characteristics. These proportions represent P(observed traits | H1), accounting for observation errors.

Usage

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

Arguments

`data`	A data.frame with columns `hair_colour`, `skin_colour`, and `eye_colour`, typically output from `sim_reference_pop`.
`h`	Integer (1-5). Missing person's hair color category.
`s`	Integer (1-5). Missing person's skin color category.
`y`	Integer (1-5). Missing person's eye color category.
`eh`	Numeric (0-1). Error rate for observing hair color.
`es`	Numeric (0-1). Error rate for observing skin color.
`ey`	Numeric (0-1). Error rate for observing eye color.

Details

The function calculates the probability of observing each trait combination given the MP's true characteristics and the error rates. Higher probabilities are assigned to combinations matching the MP's traits, while combinations with mismatches have probabilities proportional to the error rates.

Value

A data.frame with the original trait columns plus:

- `numerators`: Conditioned probability for each combination, normalized to sum to 1

Only unique combinations are returned.

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

`sim_reference_pop` for generating the input data, `compute_reference_prop` for reference proportions, `lr_compute_pigmentation` for computing LRs.

Examples

```
# Generate reference population
pop_data <- sim_reference_pop(n = 500, seed = 123)

# Compute conditioned proportions for MP with traits (1,1,1)
cond_prop <- compute_conditioned_prop(
  pop_data,
  h = 1, s = 1, y = 1,
  eh = 0.01, es = 0.01, ey = 0.01
)
head(cond_prop)
```

`compute_reference_prop` *Compute Reference Population Proportions for Pigmentation Traits*

Description

Computes the frequency of each unique combination of hair color, skin color, and eye color in the reference population. These proportions serve as the denominator (H2 probabilities) in LR calculations for pigmentation traits.

Usage

```
compute_reference_prop(data)
```

Arguments

`data`	A data.frame with columns `hair_colour`, `skin_colour`, and `eye_colour`, typically output from `sim_reference_pop`.

Value

A data.frame with columns:

- `hair_colour`: Hair color category
- `skin_colour`: Skin color category
- `eye_colour`: Eye color category
- `f_h_s_y`: Population frequency of this combination

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

sim_reference_pop for generating the input data, compute_conditioned_prop for conditioned proportions, lr_compute_pigmentation for computing LRs.

Examples

```
# Generate reference population
pop_data <- sim_reference_pop(n = 500, seed = 123)

# Compute proportions
ref_prop <- compute_reference_prop(pop_data)
head(ref_prop)

# Most common combinations
ref_prop[order(-ref_prop$f_h_s_y), ][1:5, ]
```

---

cpt_missing_person *Missing Person-Based Conditional Probability Table*

Description

Computes a conditional probability table (CPT) representing the probability of observing evidence given the hypothesis that the unidentified person IS the missing person. This table represents P(D|H1), accounting for potential observation errors in sex, age, and hair color.

The function incorporates error rates (epsilon values) that model the probability of misclassifying the true characteristics of the missing person during observation.

Usage

```
cpt_missing_person(
  MPs = "F",
  MPc = 1,
  eps = 0.05,
  epa = 0.05,
  epc = error_matrix_hair()
)
```

Arguments

MPs	Character. Missing person's biological sex: "F" for female, "M" for male. Default: "F".
MPc	Integer (1-5). Missing person's hair color category: 1=Black, 2=Brown, 3=Blonde, 4=Red, 5=Gray/White. Default: 1.
eps	Numeric (0-1). Error rate for sex observation. The probability of incorrectly recording the sex. Default: 0.05.
epa	Numeric (0-1). Error rate for age categorization. The probability of classifying a person in the wrong age group (T0 instead of T1). Default: 0.05.

epc — Matrix. Hair color error/confusion matrix, typically created with error_matrix_hair. Rows represent true colors, columns represent observed colors. Default: error_matrix_hair().

Details

For a female MP (MPs = "F"), the joint probabilities are:

- P(F-T1) = (1 - eps) * (1 - epa): Correctly observed sex and age
- P(F-T0) = (1 - eps) * epa: Correct sex, wrong age group
- P(M-T1) = eps * (1 - epa): Wrong sex, correct age
- P(M-T0) = eps * epa: Wrong sex and age

The hair color probabilities come from the error matrix row corresponding to the MP's true hair color.

Value

A 4x5 numeric matrix representing conditional probabilities under H1. Rows correspond to observed sex-age group combinations:

- F-T1: Observed as Female, age within range
- F-T0: Observed as Female, age outside range
- M-T1: Observed as Male, age within range
- M-T0: Observed as Male, age outside range

Columns correspond to observed hair colors 1-5. Each cell contains P(Observed Sex, Observed Age, Observed Color | H1, MP characteristics).

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

cpt_population for the H2 conditional probability table, error_matrix_hair for creating the color error matrix, plot_cpt for visualization of CPTs.

Examples

```
# Default: Female MP with black hair
cpt_h1 <- cpt_missing_person()
print(cpt_h1)

# Male MP with brown hair, higher error rates
cpt_h1_male <- cpt_missing_person(
  MPs = "M",
  MPc = 2,
  eps = 0.10,
  epa = 0.10
```

```
)

# Compare H1 and H2 to compute LR
cpt_h2 <- cpt_population()
lr_matrix <- cpt_h1 / cpt_h2
print(log10(lr_matrix))
```

`cpt_population` *Population-Based Conditional Probability Table*

Description

Computes a conditional probability table (CPT) representing the joint probability distribution of sex, age group, and hair color in the reference population. This table represents P(D|H2), the probability of observing the evidence under the hypothesis that the unidentified person is NOT the missing person.

The function assumes a uniform age distribution across the population and computes the probability of falling within or outside the specified age range.

Usage

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

Arguments

`propS`	Numeric vector of length 2. Sex proportions in the population, specified as c(proportion_female, proportion_male). Must sum to 1. Default: c(0.5, 0.5) for equal proportions.
`MPa`	Numeric. Missing person's estimated age in years. Used to define the center of the age matching interval. Default: 40.
`MPr`	Numeric. Age range tolerance in years (plus/minus). The matching age interval is `(MPa - MPr)` to `(MPa + MPr)`. Individuals within this range are classified as T1 (age match), others as T0 (age mismatch). Default: 6.
`propC`	Numeric vector of length 5. Hair color proportions in the population for colors 1 through 5. Must sum to 1. Default values represent a typical distribution. Colors are: 1=Black, 2=Brown, 3=Blonde, 4=Red, 5=Gray/White.

Details

The probability of age match (T1) is calculated assuming a uniform distribution over ages 1-80:

$$P(T1) = (MPa + MPr - (MPa - MPr))/80 = 2 \times MPr/80$$

The joint probability for each cell is:

$$P(Sex, Age, Color) = P(Sex) \times P(AgeGroup) \times P(Color)$$

Value

A 4x5 numeric matrix representing joint probabilities. Rows correspond to sex-age group combinations:

- F-T1: Female, age within range
- F-T0: Female, age outside range
- M-T1: Male, age within range
- M-T0: Male, age outside range

Columns correspond to hair colors 1-5. Each cell contains P(Sex, AgeGroup, HairColor | H2).

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

`cpt_missing_person` for the H1 conditional probability table, `plot_cpt` for visualization of CPTs.

Examples

```
# Default parameters: equal sex proportions, MP age 40 +/- 6 years
cpt_h2 <- cpt_population()
print(cpt_h2)

# Custom population: 60% female, narrower age range, different hair colors
cpt_custom <- cpt_population(
  propS = c(0.6, 0.4),
  MPa = 35,
  MPr = 3,
  propC = c(0.4, 0.25, 0.2, 0.1, 0.05)
)

# Verify rows sum correctly (should sum to hair color proportions)
colSums(cpt_h2)
```

`decision_threshold` *Compute Optimal Decision Threshold*

Description

Calculates the optimal likelihood ratio (LR) threshold for classifying matches versus non-matches, based on the trade-off between false positive and false negative rates.

The optimal threshold minimizes a weighted Euclidean distance that balances the costs of different types of errors.

Usage

```
decision_threshold(datasim, weight = 10)
```

Arguments

`datasim`	A data.frame with columns `Related` and `Unrelated` containing LR values. Can be output from `sim_lr_genetic`, `sim_lr_prelim`, `lr_to_dataframe`, or `lr_combine`.
`weight`	Numeric. The relative weight of false positives compared to false negatives. A value > 1 penalizes false positives more heavily. Default: 10 (false positives are 10x worse than false negatives).

Details

If the input is a list (output from `sim_lr_genetic`), it is automatically converted to a data.frame using `lr_to_dataframe`.

Algorithm: The function computes the weighted Euclidean distance for each potential threshold value:

$$D = \sqrt{FNR^2 + (weight \times FPR)^2}$$

The threshold that minimizes this distance is returned as optimal.

Weight interpretation:

- weight = 1: Equal importance to FPR and FNR
- weight = 10: FPR is 10x more costly than FNR
- weight > 10: Very conservative (minimizes false positives)
- weight < 1: Aggressive (minimizes false negatives)

In missing person cases, false positives (wrongly identifying someone as the missing person) are typically considered more serious than false negatives (failing to identify a true match), justifying weight > 1.

Value

Prints and invisibly returns the suggested LR threshold value.

References

Marsico FL, Vigeland MD, Egeland T, Herrera Pinero F (2021). "Making decisions in missing person identification cases with low statistical power." *Forensic Science International: Genetics*, 52, 102519. doi:10.1016/j.fsigen.2021.102519

See Also

`threshold_rates` for computing error rates at a given threshold, `plot_decision_curve` for visualizing the FPR/FNR trade-off, `plot_lr_distribution` for LR distribution visualization.

Examples

```
# Simulate LRs
lr_sims <- sim_lr_prelim("sex", numsims = 500, seed = 123)

# Find optimal threshold (FP 10x worse than FN)
threshold <- decision_threshold(lr_sims, weight = 10)

# Check error rates at this threshold
threshold_rates(lr_sims, threshold)

# More conservative threshold (FP 20x worse)
decision_threshold(lr_sims, weight = 20)
```

---

error_matrix_hair *Hair Color Error/Confusion Matrix*

Description

Creates a 5x5 error matrix (also known as confusion matrix) that models the probability of observing each hair color given the true hair color. This accounts for observation errors in hair color classification.

The matrix rows represent the true hair color of the missing person, and columns represent the observed hair color. Each row sums to 1, indicating that some color must be observed.

Usage

```
error_matrix_hair(
  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	Character. Type of error model to use: • "custom": Use specific error rates for each color pair (default) • "uniform": Use a single error rate for all color pairs
ep	Numeric (0-1). Base error rate used when errorModel = "uniform". Represents the probability of confusing any two different colors. Default: 0.01.
ep12	Numeric. Error rate between colors 1 (Black) and 2 (Brown). Default: 0.01.
ep13	Numeric. Error rate between colors 1 (Black) and 3 (Blonde). Default: 0.005.
ep14	Numeric. Error rate between colors 1 (Black) and 4 (Red). Default: 0.01.
ep15	Numeric. Error rate between colors 1 (Black) and 5 (Gray/White). Default: 0.003.
ep23	Numeric. Error rate between colors 2 (Brown) and 3 (Blonde). Default: 0.01.
ep24	Numeric. Error rate between colors 2 (Brown) and 4 (Red). Default: 0.003.
ep25	Numeric. Error rate between colors 2 (Brown) and 5 (Gray/White). Default: 0.01.
ep34	Numeric. Error rate between colors 3 (Blonde) and 4 (Red). Default: 0.003.
ep35	Numeric. Error rate between colors 3 (Blonde) and 5 (Gray/White). Default: 0.003.
ep45	Numeric. Error rate between colors 4 (Red) and 5 (Gray/White). Default: 0.01.

Details

The error rates are symmetric: the probability of confusing color A with color B equals the probability of confusing B with A.

The diagonal elements (correct observations) are calculated to ensure each row sums to 1:

$$P(i|i) = 1/(1 + \sum_{j \neq i} ep_{ij})$$

Lower error rates between dissimilar colors (e.g., black and blonde) and higher rates between similar colors (e.g., brown and red) reflect realistic observation patterns.

Value

A 5x5 numeric matrix where:

- Rows represent true hair colors (1-5)
- Columns represent observed hair colors (1-5)
- Cell (i,j) = P(observed color j | true color i)

- Each row sums to 1
- Diagonal elements are highest (correct observations)

Hair color codes: 1=Black, 2=Brown, 3=Blonde, 4=Red, 5=Gray/White.

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

`cpt_missing_person` which uses this matrix, `lr_hair_color` for hair color LR calculations.

Examples

```
# Default custom error model
emat <- error_matrix_hair()
print(round(emat, 4))

# Verify rows sum to 1
rowSums(emat)

# Uniform error model with 2% error rate
emat_uniform <- error_matrix_hair(errorModel = "uniform", ep = 0.02)
print(round(emat_uniform, 4))

# Higher error rates for similar colors
emat_custom <- error_matrix_hair(
  errorModel = "custom",
  ep12 = 0.05,  # Black-Brown confusion more likely
  ep23 = 0.05,  # Brown-Blonde confusion more likely
  ep34 = 0.05   # Blonde-Red confusion more likely
)
```

Europe *STR Allele Frequencies from Europe*

Description

Population allele frequency data for 23 autosomal Short Tandem Repeat (STR) markers from European populations. These pan-European frequencies are useful for general European cases or when country-specific data is unavailable.

Usage

```
data(Europe)
```

Format

A data frame with 97 rows (alleles) and 24 columns. First column is Allele (repeat number), remaining columns are allele frequencies for each STR marker.

Details

This dataset contains allele frequencies for the following 23 STR markers: D1S1656, D2S1338, D2S441, D3S1358, D5S818, D7S820, D8S1179, D10S1248, D12S391, D13S317, D16S539, D18S51, D19S433, D21S11, D22S1045, CSF1PO, FGA, Penta D, Penta E, SE33, TH01, TPOX, VWA.

These frequencies represent a pooled European dataset suitable for general forensic applications across Europe.

Source

Allele frequency data compiled from European population studies. Format compatible with **pedtools** and **forrel** packages.

References

Butler JM (2006). "Genetics and Genomics of Core Short Tandem Repeat Loci Used in Human Identity Testing." *Journal of Forensic Sciences*, 51(2), 253-265. doi:10.1111/j.15564029.2006.00046.x

See Also

`get_allele_freqs` for extracting frequencies, `sim_lr_genetic` for LR simulations.

Other frequency databases: `Argentina`, `Asia`, `USA`, `Austria`, `BosniaHerz`, `China`, `Japan`

Examples

```
# Load the dataset
data(Europe)

# View structure
head(Europe)

# Compare number of markers with other databases
ncol(Europe) - 1  # 23 markers
```

`get_allele_freqs` *Get Allele Frequencies in pedtools Format*

Description

Converts allele frequency data from a data frame to a list format compatible with the **pedtools** package for pedigree analysis.

Usage

```
get_allele_freqs(region)
```

Arguments

region	A data frame containing allele frequencies. The first column should be "Allele" with allele designations, and subsequent columns should be marker names with frequency values. Available built-in databases: Argentina, Asia, Austria, BosniaHerz, China, Europe, Japan, USA.

Details

The function transforms the data frame format (rows = alleles, columns = markers) into the list format required by pedtools (one element per marker, named by allele). This enables seamless integration with pedigree likelihood calculations.

Value

A named list where each element represents a genetic marker. Each marker element is a named numeric vector with allele names and their corresponding frequencies. This format is directly compatible with `pedtools::setMarkers()`.

Source

doi:10.1016/j.fsigss.2009.08.178 doi:10.1016/j.fsigen.2016.06.008 doi:10.1016/j.fsigen.2018.07.013

References

Marino M, et al. (2009). "Allele frequencies of 15 STRs in an Argentine population sample." Forensic Science International: Genetics Supplement Series. doi:10.1016/j.fsigss.2009.08.178

See Also

Argentina, Europe, USA for available frequency databases, sim_lr_genetic for using these frequencies in simulations.

Examples

```
# Convert Argentina database to pedtools format
freqs <- get_allele_freqs(Argentina)

# Check available markers
names(freqs)

# Use with pedtools
library(pedtools)
library(forrel)
x <- linearPed(2)
x <- setMarkers(x, locusAttributes = freqs[1:5])
```

Japan *STR Allele Frequencies from Japan*

Description

Comprehensive population allele frequency data for 70 autosomal Short Tandem Repeat (STR) markers from the Japanese population. One of the most extensive STR frequency databases available for East Asian populations.

Usage

```
data(Japan)
```

Format

A data frame with 82 rows (alleles) and 71 columns. First column is Allele (repeat number), remaining columns are allele frequencies for each STR marker.

Details

This comprehensive dataset contains allele frequencies for 70 STR markers, matching the Chinese database in marker coverage. This enables high-powered comparisons and discrimination for Japanese individuals.

Core forensic markers included: CSF1PO, D1S1656, D2S441, D2S1338, D3S1358, D5S818, D7S820, D8S1179, D10S1248, D12S391, D13S317, D16S539, D18S51, D19S433, D21S11, D22S1045, FGA, TH01, TPOX, vWA, SE33.

Extended markers include: PENTA D, PENTA E, D6S1043, D4S2408, D9S1122, and many others for enhanced discrimination.

Source

Japanese population frequency data. Format compatible with **pedtools** and **forrel** packages.

References

Fujii K, et al. (2019). "Allele frequencies for 21 autosomal STR loci in the Japanese population." *Legal Medicine*, 36, 86-87. doi:10.1016/j.legalmed.2018.11.002

See Also

`get_allele_freqs` for extracting frequencies, `sim_lr_genetic` for LR simulations.

Other Asian databases: `Asia`, `China`

Examples

```
# Load the dataset
data(Japan)

# Compare with China database
data(China)
ncol(Japan) == ncol(China)  # TRUE - same number of columns

# Different number of alleles observed
nrow(Japan)  # 82 alleles
nrow(China)  # 67 alleles
```

`kl_bidirectional` *Bidirectional Kullback-Leibler Divergence for Genetic Markers*

Description

Computes the Kullback-Leibler (KL) divergence between allele frequency distributions of two populations. Calculates divergence in both directions to assess asymmetric differences between populations.

Usage

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

Arguments

`data1`	A data frame with allele frequencies for the first population. First column should be "Allele", subsequent columns are marker frequencies.
`data2`	A data frame with allele frequencies for the second population. Same format as `data1`.
`minFreq`	Numeric. Minimum frequency value to replace zeros (to avoid undefined logarithms). Default: 1e-10.

Details

The KL divergence measures how one probability distribution differs from another. It is asymmetric: KL(P || Q) != KL(Q || P).

Higher values indicate greater divergence between populations, which may affect the reliability of LR calculations when using frequency data from one population to analyze individuals from another.

The function:

1. Finds markers common to both populations
2. Merges allele frequency tables
3. Replaces missing/zero frequencies with `minFreq`
4. Normalizes frequencies to sum to 1
5. Computes KL divergence in both directions

Value

A named list with two elements:

- "KL from data1 to data2": KL(P || Q) - divergence from population 1 to population 2
- "KL from data2 to data1": KL(Q || P) - divergence from population 2 to population 1

Values are computed using log base 10.

References

Kullback S, Leibler RA (1951). "On Information and Sufficiency." *The Annals of Mathematical Statistics*, 22(1), 79-86.

See Also

kl_multi for comparing multiple populations, kl_pie for matrix-based KL divergence.

Examples

```
# Compare Argentina and Bosnia-Herzegovina populations
result <- kl_bidirectional(Argentina, BosniaHerz)
print(result)

# Compare Argentina and Europe
kl_bidirectional(Argentina, Europe)
```

---

kl_multi *Multi-Population Kullback-Leibler Divergence Matrix*

Description

Computes pairwise Kullback-Leibler (KL) divergence between multiple population allele frequency databases. Returns a matrix of divergence values useful for assessing population differentiation.

Usage

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

Arguments

datasets	A list of data frames, each containing allele frequencies for a different population. Each data frame should have "Allele" as the first column and marker frequencies in subsequent columns.
minFreq	Numeric. Minimum frequency to replace zeros. Default: 1e-10.

Details

This function is useful for:

- Comparing multiple reference populations
- Selecting the most appropriate frequency database for a case
- Assessing potential bias from population mismatch

Higher values indicate greater divergence between populations.

Value

A square numeric matrix where element (i,j) contains the KL divergence from population i to population j. Diagonal elements are 0.

References

Kullback S, Leibler RA (1951). "On Information and Sufficiency." *The Annals of Mathematical Statistics*, 22(1), 79-86.

See Also

`kl_bidirectional` for pairwise comparison of two populations.

Examples

```
# Compare three populations
kl_matrix <- kl_multi(list(Argentina, BosniaHerz, Europe))
print(kl_matrix)

# Visualize as heatmap
# heatmap(kl_matrix, main = "KL Divergence Between Populations")
```

---

kl_pie *Kullback-Leibler Divergence for Probability Matrices*

Description

Computes the Kullback-Leibler (KL) divergence between two probability matrices using base-10 logarithm. Calculates divergence in both directions.

This is useful for comparing conditional probability tables (CPTs) or other matrix representations of probability distributions.

Usage

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

Arguments

P	A numeric matrix representing the first probability distribution. Should sum to 1 (or will be treated as unnormalized).
Q	A numeric matrix representing the second probability distribution. Must have the same dimensions as P.
min_value	Numeric. Minimum value to replace zeros (to avoid undefined logarithms). Default: 1e-12.

Details

The KL divergence is computed as:

$$KL(P||Q) = \sum_i P_i \log_{10}(P_i/Q_i)$$

Zero values in P or Q are replaced with `min_value` to avoid undefined operations.

Value

A named numeric vector with two elements:

- `"P || Q"`: KL divergence from P to Q
- `"Q || P"`: KL divergence from Q to P

References

Kullback S, Leibler RA (1951). "On Information and Sufficiency." *The Annals of Mathematical Statistics*, 22(1), 79-86.

See Also

`kl_bidirectional` for allele frequency comparisons, `cpt_population`, `cpt_missing_person` for creating probability matrices.

Examples

```
# Compare two CPTs
cpt1 <- cpt_population()
cpt2 <- cpt_population(propS = c(0.6, 0.4))

kl_pie(cpt1, cpt2)
```

lr_age *Likelihood Ratio for Age Variable*

Description

Simulates age observations and optionally computes likelihood ratios (LRs) under either H1 (unidentified person is the missing person) or H2 (unidentified person is not the missing person).

Ages are categorized into two groups based on whether they fall within the missing person's estimated age range:

- T1: Age within range (MPa - MPr) to (MPa + MPr)
- T0: Age outside range

Usage

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

Arguments

MPa	Numeric. Missing person's estimated age in years. Default: 40.
MPr	Numeric. Age range tolerance (plus/minus years). The matching interval is (MPa - MPr) to (MPa + MPr). Default: 6.
UHRr	Numeric. Additional uncertainty range for the unidentified person's age estimation. Default: 1.
gam	Numeric. Gamma parameter for age uncertainty scaling. The uncertainty interval is age +/- (gam * age + UHRr). Default: 0.07.
numsims	Integer. Number of simulations to perform. Default: 1000.
epa	Numeric (0-1). Error rate for age categorization. Default: 0.05.
erRa	Numeric (0-1). Error rate in the reference/database. Defaults to epa.
H	Integer (1 or 2). Hypothesis to simulate under: • 1: H1 (Related) - Unidentified person IS the missing person

- 2: H2 (Unrelated) - Unidentified person is NOT the missing person

Default: 1.

`modelA` Character. Reference age distribution model:

- "uniform": Assumes uniform age distribution (default)
- "custom": Uses empirical frequencies from simulations

`LR` Logical. If TRUE, compute and return LR values. Default: FALSE.

`seed` Integer. Random seed for reproducibility. Default: 1234.

`nsims` Deprecated. Use `numsims` instead.

Details

Under H1 (Related): Age is sampled to fall within the MP's range with probability (1 - erRa), outside with probability erRa.

Under H2 (Unrelated): Age is sampled uniformly from 1-80, then categorized.

LR Calculation (uniform model):

- LR(T1) = (1 - epa) / P(T1), where P(T1) = 2*MPr/80
- LR(T0) = epa / P(T0), where P(T0) = 1 - P(T1)

Value

A data.frame with columns:

- `group`: Age group classification ("T1" or "T0")
- `age`: Simulated age value
- `UHRmin`: Lower bound of uncertainty interval
- `UHRmax`: Upper bound of uncertainty interval
- `LRa`: Likelihood ratio (only if LR = TRUE)

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

`sim_lr_prelim` for unified preliminary LR simulations, `lr_sex`, `lr_hair_color` for other variables.

Examples

```
# Simulate under H1 (related)
sim_h1 <- lr_age(MPa = 40, MPr = 6, H = 1, numsims = 100)
table(sim_h1$group)

# Simulate under H2 with LR values
sim_h2 <- lr_age(MPa = 40, MPr = 6, H = 2, numsims = 100, LR = TRUE)
```

```
head(sim_h2)

# Narrower age range (more discriminating)
sim_narrow <- lr_age(MPa = 35, MPr = 3, numsims = 500, LR = TRUE)
summary(sim_narrow$LRa)
```

`lr_birthdate` *Likelihood Ratio for Birth Date*

Description

Computes likelihood ratios (LRs) based on the discrepancy between the actual birth date (ABD) of the missing person and the declared birth date (DBD) of the person of interest. Uses Dirichlet distribution to model category probabilities.

Usage

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

Arguments

`ABD`	Character or Date. Actual birth date of the missing person in "YYYY-MM-DD" format. Default: "1976-05-31".
`DBD`	Character or Date. Declared birth date of the person of interest in "YYYY-MM-DD" format. Default: "1976-07-15".
`alpha`	Numeric vector. Alpha parameters for the Dirichlet distribution, typically representing frequencies of solved cases in each discrepancy category. Length should be one more than length of `cuts`. Default: c(1, 4, 60, 11, 6, 4, 4).
`cuts`	Numeric vector. Cutoff values (in days) for categorizing the difference between DBD and ABD. Creates length(cuts)+1 categories. Default: c(-120, -30, 30, 120, 240, 360).
`type`	Integer (1 or 2). Type of search scenario: • 1: Open search - MP may not be in database (uses uniform H2) • 2: Closed search - MP is in database (uses database frequencies) Default: 1.
`PrelimData`	Data.frame. Required when type = 2. Contains DBD column for persons of interest in the database. Can be output from `sim_poi_prelim`.

`draw`	Integer. Number of Dirichlet samples for probability estimation. Default: 500.
`seed`	Integer. Random seed for reproducibility. Default: 123.

Details

Categories: The difference between DBD and ABD (in days) is categorized using the `cuts` vector. Default categories are:

1. < -120 days (DBD more than 4 months before ABD)
2. -120 to -30 days
3. -30 to 30 days (close match)
4. 30 to 120 days
5. 120 to 240 days
6. 240 to 360 days
7. > 360 days (DBD more than 1 year after ABD)

Dirichlet Model: Uses method of moments to estimate category probabilities from Dirichlet samples. The `alpha` parameter reflects prior knowledge from solved cases about the distribution of birth date discrepancies.

LR Calculation:

- Type 1: LR = P(category | H1) / (1/n_categories)
- Type 2: LR = P(category | H1) / P(category in database)

Value

Numeric. The likelihood ratio for the given birth date discrepancy. Also printed to console.

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

`sim_lr_prelim` for simulating LR distributions, `sim_poi_prelim` for generating preliminary databases.

Examples

```
# Type 1: Open search - close match (45 days difference)
lr1 <- lr_birthdate(
  ABD = "1976-05-31",
  DBD = "1976-07-15",
  type = 1,
  seed = 123
)

# Type 1: Open search - larger discrepancy
lr2 <- lr_birthdate(
```

```
  ABD = "1976-05-31",
  DBD = "1977-03-15",
  type = 1,
  seed = 123
)

## Not run:
# Type 2: Closed search with database
# Requires a large database with varied birth dates
db <- sim_poi_prelim(numsims = 1000, seed = 456)
lr3 <- lr_birthdate(
  ABD = "1976-05-31",
  DBD = "1976-07-15",
  type = 2,
  PrelimData = db,
  seed = 123
)

## End(Not run)
```

`lr_combine` *Combine Likelihood Ratios from Multiple Sources*

Description

Combines (multiplies) likelihood ratios from two independent evidence sources using the Bayesian multiplication principle. This is used to integrate genetic and non-genetic evidence, or multiple non-genetic variables.

Usage

```
lr_combine(LRdatasim1, LRdatasim2)
```

Arguments

`LRdatasim1`	A data.frame with columns `Unrelated` and `Related` containing LR values from the first evidence source. Must be a data.frame (use `lr_to_dataframe` to convert genetic simulation output first).
`LRdatasim2`	A data.frame with columns `Unrelated` and `Related` containing LR values from the second evidence source.

Details

Under the assumption of conditional independence of evidence given each hypothesis, the combined LR is the product of individual LRs:

$$LR_{combined} = LR_1 \times LR_2$$

This follows from Bayes' theorem and is valid when the evidence sources are conditionally independent given the hypothesis.

Important: Both inputs must be data.frames with the same structure. If using output from `sim_lr_genetic`, first convert it using `lr_to_dataframe`.

Value

A data.frame with columns:

- `Unrelated`: Product of LR values under H2
- `Related`: Product of LR values under H1

The number of rows equals the minimum of the input data frames.

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

`sim_lr_genetic` for genetic LR simulations, `sim_lr_prelim` for non-genetic LR simulations, `lr_to_dataframe` for converting genetic simulations, `plot_lr_distribution` for visualizing combined distributions.

Examples

```
# Simulate LRs from two different variables
lr_sex <- sim_lr_prelim("sex", numsims = 500, seed = 123)
lr_age <- sim_lr_prelim("age", numsims = 500, seed = 456)

# Combine the evidence
lr_combined <- lr_combine(lr_sex, lr_age)
head(lr_combined)

# Compare distributions
summary(log10(lr_sex$Related))
summary(log10(lr_combined$Related))

# Visualize combined distribution
plot_lr_distribution(lr_combined)

# Combining genetic and non-genetic evidence
library(forrel)
x <- linearPed(2)
x <- setMarkers(x, locusAttributes = NorwegianFrequencies[1:5])
x <- profileSim(x, N = 1, ids = 2)

# Simulate genetic LRs and convert to dataframe
lr_genetic <- sim_lr_genetic(x, missing = 5, numsims = 100, seed = 123)
lr_genetic_df <- lr_to_dataframe(lr_genetic)
```

```
# Simulate non-genetic LRs
lr_prelim <- sim_lr_prelim("sex", numsims = 100, seed = 123)

# Combine both sources
lr_total <- lr_combine(lr_genetic_df, lr_prelim)
```

`lr_compute_pigmentation` *Compute Likelihood Ratios for Pigmentation Traits*

Description

Computes likelihood ratios (LRs) for each unique combination of hair color, skin color, and eye color by dividing conditioned proportions (numerators) by reference proportions (denominators).

Usage

```
lr_compute_pigmentation(conditioned, unconditioned)
```

Arguments

`conditioned`	A data.frame with columns `hair_colour`, `skin_colour`, `eye_colour`, and `numerators`. Typically output from `compute_conditioned_prop`.
`unconditioned`	A data.frame with columns `hair_colour`, `skin_colour`, `eye_colour`, and `f_h_s_y`. Typically output from `compute_reference_prop`.

Value

A data.frame with:

- `hair_colour`, `skin_colour`, `eye_colour`: Trait combination
- `f_h_s_y`: Population frequency (denominator)
- `numerators`: Conditioned probability (numerator)
- `LR`: Likelihood ratio = numerators / f_h_s_y

Combinations not present in both inputs are excluded.

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

`compute_conditioned_prop` for computing numerators, `compute_reference_prop` for computing denominators, `lr_pigmentation` for simulating LR distributions.

Examples

```
# Generate population data
pop_data <- sim_reference_pop(n = 500, seed = 123)

# Compute proportions
conditioned <- compute_conditioned_prop(pop_data, 1, 1, 1, 0.01, 0.01, 0.01)
unconditioned <- compute_reference_prop(pop_data)

# Compute LRs
lrs <- lr_compute_pigmentation(conditioned, unconditioned)
head(lrs)

# Highest LRs (most discriminating combinations)
lrs[order(-lrs$LR), ][1:5, ]
```

---

lr_hair_color *Likelihood Ratio for Hair Color*

Description

Simulates hair color observations and optionally computes likelihood ratios (LRs) under either H1 (unidentified person is the missing person) or H2 (unidentified person is not the missing person).

Hair color is categorized into 5 groups: 1=Black, 2=Brown, 3=Blonde, 4=Red, 5=Gray/White

Usage

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

Arguments

MPc	Integer (1-5). Missing person's hair color category. Default: 1.
epc	Matrix. Hair color error/confusion matrix, typically created with error_matrix_hair. Rows represent true colors, columns represent observed colors. Default: error_matrix_hair().
erRc	Matrix. Error matrix for the reference/database. Defaults to epc.
numsims	Integer. Number of simulations to perform. Default: 1000.

Pc	Numeric vector of length 5. Hair color proportions in the population. Must sum to 1. Default: c(0.3, 0.2, 0.25, 0.15, 0.1).
H	Integer (1 or 2). Hypothesis to simulate under: • 1: H1 (Related) - Unidentified person IS the missing person • 2: H2 (Unrelated) - Unidentified person is NOT the missing person Default: 1.
Qprop	Integer. Query color for testing. Defaults to MPc.
LR	Logical. If TRUE, compute and return LR values. Default: FALSE.
seed	Integer. Random seed for reproducibility. Default: 1234.
nsims	Deprecated. Use numsims instead.

Details

Under H1 (Related): Observed color is sampled using the row of the error matrix corresponding to the MP's true hair color. This accounts for observation errors.

Under H2 (Unrelated): Color is sampled from the population proportions Pc.

LR Calculation: LR = P(observed color | true color is MPc) / P(observed color in population) LR = epc(MPc, observed) / Pc(observed)

Value

A data.frame with column Col containing simulated color observations (1-5). If LR = TRUE, also includes column LRc with the likelihood ratio for each observation.

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

error_matrix_hair for creating the error matrix, lr_pigmentation for combined pigmentation traits, sim_lr_prelim for unified preliminary LR simulations.

Examples

```
# Simulate under H1 (related) with black hair MP
sim_h1 <- lr_hair_color(MPc = 1, H = 1, numsims = 100)
table(sim_h1$Col)

# Simulate under H2 with LR values
sim_h2 <- lr_hair_color(MPc = 2, H = 2, numsims = 100, LR = TRUE)
head(sim_h2)
summary(sim_h2$LRc)

# Custom population proportions
sim_custom <- lr_hair_color(
  MPc = 3,  # Blonde
```

```
  Pc = c(0.1, 0.4, 0.3, 0.1, 0.1),  # Different population
  numsims = 500,
  LR = TRUE
)
```

`lr_pigmentation` *Simulate LR Distributions for Pigmentation Traits*

Description

Simulates likelihood ratio (LR) distributions for combined pigmentation traits (hair, skin, and eye color) under both hypotheses. Uses pre-computed LRs from `lr_compute_pigmentation`.

Usage

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

Arguments

`df`	A data.frame with columns `numerators`, `f_h_s_y`, and `LR`. Typically output from `lr_compute_pigmentation`.
`seed`	Integer. Random seed for reproducibility. Default: 1234.
`nsim`	Integer. Number of LR values to simulate per hypothesis. Default: 500.

Details

The function samples LR values with probabilities proportional to:

- *H2 (Unrelated)*: Population frequencies (`f_h_s_y`)
- *H1 (Related)*: Conditioned probabilities (`numerators`)

This simulates the expected distribution of LRs when comparing the MP's traits against either random individuals (H2) or the true match (H1).

Value

A data.frame with two columns:

- `Unrelated`: LR values simulated under H2 (sampling proportional to population frequencies)
- `Related`: LR values simulated under H1 (sampling proportional to conditioned probabilities)

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

sim_reference_pop for generating population data, lr_compute_pigmentation for computing input LRs, plot_lr_distribution for visualization.

Examples

```
# Full workflow for pigmentation LRs
pop_data <- sim_reference_pop(n = 500, seed = 123)
conditioned <- compute_conditioned_prop(pop_data, 1, 1, 1, 0.01, 0.01, 0.01)
unconditioned <- compute_reference_prop(pop_data)
lrs <- lr_compute_pigmentation(conditioned, unconditioned)

# Simulate LR distribution
lr_dist <- lr_pigmentation(lrs, nsim = 500, seed = 456)
head(lr_dist)

# Visualize
plot_lr_distribution(lr_dist)
```

---

lr_sensitivity *Sensitivity Analysis for Likelihood Ratios*

Description

Evaluates how the likelihood ratio changes when model parameters vary. This is essential for understanding the robustness of forensic conclusions and for communicating uncertainty to decision-makers.

Usage

```
lr_sensitivity(
  evidence_type,
  param,
  range = NULL,
  steps = 20,
  match = TRUE,
  baseline = NULL
)
```

Arguments

evidence_type	Character. Type of evidence to analyze. Options: "sex", "age", "hair", "region".
param	Character. Parameter to vary. Options depend on evidence_type: • "sex": "eps" (error rate), "freq" (population frequency) • "age": "eps" (error rate), "range" (age interval) • "hair": "eps" (error rate), "freq" (population frequency) • "region": "eps" (error rate), "nreg" (number of regions)

`range`	Numeric vector of length 2. Range of parameter values to test. Default depends on param type.
`steps`	Integer. Number of steps in the range. Default: 20.
`match`	Logical. TRUE for matching evidence (same sex/age in range/etc), FALSE for mismatching. Default: TRUE.
`baseline`	List. Baseline parameter values. If NULL, uses defaults.

Details

Sensitivity analysis is critical in forensic science because:

1. Parameters (error rates, population frequencies) are often estimated with uncertainty
2. Different reference populations may have different frequencies
3. The analysis reveals which parameters most affect conclusions

Interpretation:

- Steep curves indicate high sensitivity (conclusions depend strongly on parameter choice)
- Flat curves indicate robustness (conclusions stable across reasonable parameter values)

Value

A data.frame with columns:

- `param_value`: Parameter value tested
- `LR`: Resulting likelihood ratio
- `log10_LR`: Log10 of LR (useful for plotting)

References

Kling D, Tillmar AO, Egeland T (2014). "Familias 3-Extensions and new functionality." *Forensic Science International: Genetics*, 13, 121-127.

See Also

`lr_sex`, `lr_age`, `lr_hair_color` for individual LR calculations.

Examples

```
# How does sex LR change with error rate?
sens_eps <- lr_sensitivity("sex", param = "eps", range = c(0.01, 0.20))
plot(sens_eps$param_value, sens_eps$log10_LR, type = "l",
     xlab = "Error rate", ylab = "log10(LR)",
     main = "Sex LR sensitivity to error rate")
abline(h = 0, lty = 2)

# How does sex LR change with population frequency?
sens_freq <- lr_sensitivity("sex", param = "freq", range = c(0.3, 0.7))
plot(sens_freq$param_value, sens_freq$log10_LR, type = "l",
```

```
     xlab = "Female frequency", ylab = "log10(LR)",
     main = "Sex LR sensitivity to population frequency")

# Age LR sensitivity to range parameter
sens_range <- lr_sensitivity("age", param = "range", range = c(2, 15))
plot(sens_range$param_value, sens_range$log10_LR, type = "l",
     xlab = "Age range (+/- years)", ylab = "log10(LR)",
     main = "Age LR sensitivity to range")
```

`lr_sex` *Likelihood Ratio for Biological Sex*

Description

Simulates observations of biological sex and optionally computes likelihood ratios (LRs) under either H1 (unidentified person is the missing person) or H2 (unidentified person is not the missing person).

Usage

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

Arguments

`MPs`	Character. Missing person's biological sex: "F" for female, "M" for male. Default: "F".
`eps`	Numeric (0-1). Error rate (epsilon) for sex observation. Probability of misclassifying sex when recording. Default: 0.05.
`erRs`	Numeric (0-1). Error rate in the database/reference. Defaults to `eps` if not specified.
`numsims`	Integer. Number of simulations to perform. Default: 1000.
`Ps`	Numeric vector of length 2. Sex proportions in the population, c(proportion_female, proportion_male). Must sum to 1. Default: c(0.5, 0.5).
`H`	Integer (1 or 2). Hypothesis to simulate under: • 1: H1 (Related) - Unidentified person IS the missing person • 2: H2 (Unrelated) - Unidentified person is NOT the missing person

	Default: 1.
LR	Logical. If TRUE, compute and return LR values for each simulated observation. Default: FALSE.
seed	Integer. Random seed for reproducibility. Default: 1234.
nsims	Deprecated. Use numsims instead.

Details

Under H1 (Related): The observed sex matches the MP's true sex with probability (1 - erRs), and is incorrectly recorded with probability erRs.

Under H2 (Unrelated): Sex is sampled from the population proportions Ps.

LR Calculation: For a matching observation: `LR = (1 - eps) / Ps_MP` For a non-matching observation: `LR = eps / Ps_other`

Value

A data.frame with column `Sexo` containing simulated sex observations ("F" or "M"). If `LR = TRUE`, also includes column `LRs` with the likelihood ratio for each observation.

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

`sim_lr_prelim` for unified preliminary LR simulations, `lr_age`, `lr_hair_color` for other variables.

Examples

```
# Simulate under H1 (related)
sim_h1 <- lr_sex(MPs = "F", H = 1, numsims = 100)
table(sim_h1$Sexo)

# Simulate under H2 (unrelated) with LR values
sim_h2 <- lr_sex(MPs = "F", H = 2, numsims = 100, LR = TRUE)
head(sim_h2)

# Different population proportions
sim_custom <- lr_sex(
  MPs = "M",
  Ps = c(0.52, 0.48),  # 52% female population
  numsims = 500,
  LR = TRUE
)
summary(sim_custom$LRs)
```

`lr_to_dataframe` *Convert Genetic LR Simulations to Data Frame*

Description

Converts the list output from `sim_lr_genetic` into a tidy data frame suitable for analysis and visualization. Extracts the total LR values from each simulation.

Usage

```
lr_to_dataframe(datasim)
```

Arguments

`datasim`	A list object returned by `sim_lr_genetic`, containing `Unrelated` and `Related` components with LR objects.

Details

The function extracts `LRtotal[["H1:H2"]]` from each LR object in the simulation lists. This represents the overall likelihood ratio across all genetic markers.

Value

A data.frame with two columns:

- `Unrelated`: Numeric LR values from H2 simulations
- `Related`: Numeric LR values from H1 simulations

The number of rows equals the number of simulations.

References

Marsico FL, Vigeland MD, Egeland T, Herrera Pinero F (2021). "Making decisions in missing person identification cases with low statistical power." *Forensic Science International: Genetics*, 52, 102519. doi:10.1016/j.fsigen.2021.102519

See Also

`sim_lr_genetic` for generating the input, `plot_lr_distribution` for visualization, `lr_combine` for combining with other LR sources.

Examples

```
library(forrel)

# Create pedigree and simulate
x <- linearPed(2)
x <- setMarkers(x, locusAttributes = NorwegianFrequencies[1:5])
x <- profileSim(x, N = 1, ids = 2)

# Simulate LRs
lr_sims <- sim_lr_genetic(x, missing = 5, numsims = 50, seed = 123)

# Convert to dataframe
lr_df <- lr_to_dataframe(lr_sims)
head(lr_df)

# Now can use with other functions
summary(log10(lr_df$Related))
plot_lr_distribution(lr_df)
```

`mispitools_app` *Comprehensive Shiny App for Missing Person Identification*

Description

Launches a comprehensive interactive Shiny application for calculating likelihood ratios (LRs) from non-genetic evidence in missing person cases. This unified app integrates all evidence types (sex, age, hair color, birthdate) with tutorials, visualizations, and decision analysis tools.

Usage

```
mispitools_app()
```

Details

This app provides a complete workflow for forensic identification using non-genetic evidence. It implements the Bayesian framework where:

- H1: The unidentified person IS the missing person
- H2: The unidentified person is NOT the missing person
- LR = P(Evidence | H1) / P(Evidence | H2)

Evidence types supported:

- Biological sex (male/female)
- Age (within expected range)
- Hair color (5 categories)
- Birth date (discrepancy analysis)

Value

A Shiny app object. When run interactively, launches a multi-tab web interface with:

- **Welcome**: Introduction to LR concepts
- **Individual Evidence**: Calculate LR for each evidence type
- **CPT Analysis**: Visualize conditional probability tables
- **Distribution**: Simulate and visualize LR distributions
- **Combine Evidence**: Combine multiple evidence types
- **Decision Analysis**: Threshold selection and error metrics
- **Tutorial**: Step-by-step educational content

References

Marsico FL, Caridi I (2023). "Incorporating non-genetic evidence in large scale missing person searches: A general approach beyond filtering." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

Marsico FL, Vigeland MD, et al. (2021). "Making decisions in missing person identification cases with low statistical power." *Forensic Science International: Genetics*, 52, 102519. doi:10.1016/j.fsigen.2021.102519

See Also

`lr_sex`, `lr_age`, `lr_hair_color`, `lr_birthdate` for individual LR calculations, `lr_combine` for combining evidence, `decision_threshold`, `threshold_rates` for decision analysis.

Examples

```
if (interactive()) {
  mispitools_app()
}
```

---

plot_cpt *Plot Conditional Probability Tables Comparison*

Description

Creates a three-panel visualization comparing conditional probability tables (CPTs) and their resulting likelihood ratios:

- Panel A: P(D|H2) - Population-based probabilities
- Panel B: P(D|H1) - Missing person-based probabilities
- Panel C: log10(LR) - Likelihood ratios for each combination

This visualization helps understand how different combinations of sex, age group, and hair color contribute to the likelihood ratio.

Usage

```
plot_cpt(CPT_POP, CPT_MP)
```

Arguments

CPT_POP	Matrix. Population-based conditional probability table, typically output from cpt_population.
CPT_MP	Matrix. Missing person-based conditional probability table, typically output from cpt_missing_person.

Details

The heatmaps use a blue gradient where darker colors indicate higher values (higher probabilities or higher LRs).

Each cell is labeled with its value rounded to 2 decimal places.

The LR panel (C) shows log10(LR), where:

- Positive values (blue) favor H1 (related)
- Negative values favor H2 (unrelated)
- Zero indicates neutral evidence

Row labels indicate sex and age group combinations:

- F-T1: Female, age within MP range
- F-T0: Female, age outside MP range
- M-T1: Male, age within MP range
- M-T0: Male, age outside MP range

Column labels indicate hair color categories (1-5).

Value

A ggplot2 object with three panels arranged horizontally, showing heatmaps with cell values annotated.

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

cpt_population for creating the H2 table, cpt_missing_person for creating the H1 table.

Examples

```
# Create both CPTs
cpt_h2 <- cpt_population()
cpt_h1 <- cpt_missing_person(MPs = "F", MPc = 1)

# Visualize comparison
plot_cpt(cpt_h2, cpt_h1)

# Different MP characteristics
cpt_h1_male <- cpt_missing_person(MPs = "M", MPc = 3)
plot_cpt(cpt_h2, cpt_h1_male)
```

---

plot_decision_curve *Plot Decision Curve (FPR vs FNR)*

Description

Creates a scatter plot showing the trade-off between false positive rate (FPR) and false negative rate (FNR) across different LR threshold values. This visualization helps identify optimal decision thresholds based on the relative costs of different types of errors.

Usage

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

Arguments

`datasim`	A data.frame with columns `Related` and `Unrelated` containing LR values. Can be output from `sim_lr_genetic`, `sim_lr_prelim`, `lr_to_dataframe`, or `lr_combine`.
`LRmax`	Numeric. Maximum LR value to use as threshold. Points are generated for thresholds from 1 to LRmax. Default: 1000.

Details

If the input is a list (output from `sim_lr_genetic`), it is automatically converted to a data.frame using `lr_to_dataframe`.

Error Rate Definitions:

- *FPR*: Proportion of unrelated (H2) cases with LR > threshold
- *FNR*: Proportion of related (H1) cases with LR < threshold

Ideal point: The origin (0,0) represents perfect discrimination. Points closer to the origin indicate better thresholds.

Trade-off: Moving along the curve, decreasing FNR typically increases FPR and vice versa. The optimal point depends on the relative costs of false positives vs false negatives.

Value

A ggplot2 scatter plot where:

- X-axis: False Negative Rate (FNR) - proportion of true matches missed
- Y-axis: False Positive Rate (FPR) - proportion of non-matches incorrectly identified
- Each point represents a different LR threshold

The first and last threshold values are labeled on the plot.

References

Marsico FL, Vigeland MD, Egeland T, Herrera Pinero F (2021). "Making decisions in missing person identification cases with low statistical power." *Forensic Science International: Genetics*, 52, 102519. doi:10.1016/j.fsigen.2021.102519

See Also

plot_lr_distribution for LR distribution visualization, decision_threshold for computing optimal threshold, threshold_rates for error rates at a specific threshold.

Examples

```
# Using preliminary data
lr_sims <- sim_lr_prelim("sex", numsims = 500, seed = 123)
plot_decision_curve(lr_sims)

# With lower maximum threshold for finer resolution
plot_decision_curve(lr_sims, LRmax = 100)
```

---

plot_lr_distribution *Plot Likelihood Ratio Distributions*

Description

Creates a density plot showing the expected log10(LR) distributions under both hypotheses:

- H1 (Related/Blue): Distribution when POI is the missing person
- H2 (Unrelated/Red): Distribution when POI is unrelated

This visualization helps assess the discriminatory power of the evidence and identify potential overlap between the two hypotheses.

Usage

```
plot_lr_distribution(datasim)
```

Arguments

datasim	A data.frame with columns Related and Unrelated containing LR values. Can be output from sim_lr_genetic, sim_lr_prelim, lr_to_dataframe, or lr_combine.

Details

If the input is a list (output from sim_lr_genetic), it is automatically converted to a data.frame using lr_to_dataframe.

The x-axis shows log10(LR), which is more interpretable than raw LR values:

- log10(LR) = 0 means LR = 1 (neutral evidence)
- log10(LR) > 0 means evidence favors H1 (related)
- log10(LR) < 0 means evidence favors H2 (unrelated)

Less overlap between distributions indicates better discrimination.

Value

A ggplot2 object showing overlaid density curves. Blue area represents H1 (Related), red area represents H2 (Unrelated).

References

Marsico FL, Vigeland MD, Egeland T, Herrera Pinero F (2021). "Making decisions in missing person identification cases with low statistical power." *Forensic Science International: Genetics*, 52, 102519. doi:10.1016/j.fsigen.2021.102519

See Also

plot_decision_curve for FPR/FNR trade-off visualization, decision_threshold for computing optimal thresholds, sim_lr_genetic, sim_lr_prelim for generating input.

Examples

```
# Using preliminary data
lr_sims <- sim_lr_prelim("sex", numsims = 500, seed = 123)
plot_lr_distribution(lr_sims)

# Using genetic data
library(forrel)
x <- linearPed(2)
x <- setMarkers(x, locusAttributes = NorwegianFrequencies[1:5])
x <- profileSim(x, N = 1, ids = 2)
lr_genetic <- sim_lr_genetic(x, missing = 5, numsims = 50, seed = 123)
plot_lr_distribution(lr_genetic)
```

`sim_lr_genetic` *Simulate Likelihood Ratios from Genetic Data*

Description

Simulates likelihood ratio (LR) distributions based on genetic (DNA) marker data. This function generates expected LR distributions under two hypotheses:

- H1 (Related): The unidentified person IS the missing person
- H2 (Unrelated): The unidentified person is NOT the missing person

This function wraps functionality from the **forrel** package to perform missing person LR calculations using pedigree structures.

Usage

```
sim_lr_genetic(reference, missing, numsims = 100, seed = 123, numCores = 1)
```

Arguments

`reference`	A pedigree object with attached genetic markers. Can be created using **pedtools** functions like `linearPed()`, `nuclearPed()`, etc., with markers attached via `setMarkers()`.
`missing`	Character or numeric. The ID/label of the missing person in the pedigree.
`numsims`	Integer. Number of simulations to perform. Default: 100.
`seed`	Integer. Random seed for reproducibility. Default: 123.
`numCores`	Integer. Number of CPU cores for parallel processing. Default: 1 (no parallelization).

Details

The function performs two types of simulations:

1. **H2 (Unrelated)**: Generates random genetic profiles for unrelated individuals using population allele frequencies, then calculates the LR for each profile.
2. **H1 (Related)**: Simulates genetic profiles for the actual missing person based on the pedigree structure, then calculates the LR for each profile.

The LR is computed using `forrel::missingPersonLR()`, which calculates the ratio of likelihoods: P(data | POI is MP) / P(data | POI is unrelated).

Value

A list with two components:

- `Unrelated`: List of LR objects from simulations where POI is unrelated to the pedigree (H2 simulations)
- `Related`: List of LR objects from simulations where POI is the actual missing person (H1 simulations)

Use `lr_to_dataframe` to convert this to a data.frame for further analysis.

References

Marsico FL, Vigeland MD, Egeland T, Herrera Pinero F (2021). "Making decisions in missing person identification cases with low statistical power." *Forensic Science International: Genetics*, 52, 102519. doi:10.1016/j.fsigen.2021.102519

Vigeland MD, Egeland T (2021). "Joint DNA-based disaster victim identification." *Forensic Science International: Genetics*, 52, 102465.

See Also

`lr_to_dataframe` for converting output to dataframe, `sim_lr_prelim` for non-genetic LR simulations, `plot_lr_distribution` for visualizing LR distributions, `decision_threshold` for computing optimal thresholds.

Examples

```
library(forrel)
library(pedtools)

# Create a simple pedigree: grandparent-parent-child
x <- linearPed(2)
plot(x)

# Add genetic markers (using Norwegian frequencies as example)
x <- setMarkers(x, locusAttributes = NorwegianFrequencies[1:5])

# Simulate a profile for the reference person (ID 2)
x <- profileSim(x, N = 1, ids = 2)

# Simulate LRs (person 5 is missing)
lr_sims <- sim_lr_genetic(x, missing = 5, numsims = 50, seed = 123)

# Convert to dataframe for analysis
lr_df <- lr_to_dataframe(lr_sims)
head(lr_df)

# Visualize distributions
plot_lr_distribution(lr_df)
```

`sim_lr_prelim` *Simulate Likelihood Ratios from Preliminary Investigation Data*

Description

Simulates likelihood ratio (LR) distributions based on non-genetic (preliminary investigation) data such as sex, age, region, height, or birth date. This function generates expected LR distributions under both hypotheses:

- H1 (Related): The unidentified person IS the missing person
- H2 (Unrelated): The unidentified person is NOT the missing person

Usage

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

Arguments

`vartype`	Character. Type of preliminary investigation variable. Options: "sex", "region", "age", "height", "birthdate".
`numsims`	Integer. Number of simulations to perform. Default: 1000.
`seed`	Integer. Random seed for reproducibility. Default: 123.
`int`	Numeric. Interval parameter for "age" and "height" variables. Defines the estimation range (e.g., if MP age is 55 and int is 10, the range is 45-65). Default: 5.
`ErrorRate`	Numeric (0-1). Error rate for observations. Default: 0.05.
`alphaBdate`	Numeric vector. Alpha parameters for Dirichlet distribution used in birthdate LR calculations. Usually frequencies of solved cases in each category. Default: c(1, 4, 60, 11, 6, 4, 4).
`numReg`	Integer. Number of regions in the case (for "region" variable). Default: 6.
`MP`	Value of the MP's characteristic for closed search. If NULL, open search is performed. For "sex": "F" or "M"; for "age"/"height": numeric; for "birthdate": date string; for "region": region ID. Default: NULL.

`database`	Data frame. Database of POIs for closed search (when MP is not NULL). Should have columns matching the variable type (e.g., "Sex", "Age", "Height", "Region", "DBD"). Can be output from `sim_poi_prelim`.
`cuts`	Numeric vector. Cutoff values for birthdate categories. Days difference between declared and actual birth dates. Default: c(-120, -30, 30, 120, 240, 360).

Details

Open Search (MP = NULL): Used when it's unknown whether the MP is in the database. LR is computed using general population frequencies as the denominator.

Closed Search (MP specified): Used when comparing a specific MP against a database. LR denominator uses frequencies from the actual database.

Variable-specific calculations:

- *sex*: Binary match/mismatch with error rate
- *region*: Match against numReg possible regions
- *age/height*: Match if within +/- int of MP value
- *birthdate*: Dirichlet-based probability for date discrepancy

Value

A data.frame with two columns:

- `Unrelated`: LR values simulated under H2 (POI is not MP)
- `Related`: LR values simulated under H1 (POI is MP)

Each column contains `numsims` values.

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

`sim_lr_genetic` for genetic LR simulations, `lr_combine` for combining LRs from different sources, `sim_poi_prelim` for creating preliminary databases.

Examples

```
# Open search for sex variable
lr_sex <- sim_lr_prelim("sex", numsims = 500, seed = 123)
head(lr_sex)

# Check distribution
summary(log10(lr_sex$Related))
summary(log10(lr_sex$Unrelated))

# Visualize
```

```
plot_lr_distribution(lr_sex)

# Closed search with database
db <- sim_poi_prelim(numsims = 100, seed = 456)
lr_sex_closed <- sim_lr_prelim(
  "sex",
  numsims = 500,
  MP = "F",
  database = db
)

# Age variable
lr_age <- sim_lr_prelim("age", numsims = 500, int = 10)
```

`sim_mp_prelim` *Simulate Preliminary Investigation Data for Missing Persons*

Description

Generates a simulated database of preliminary investigation data for missing persons (MPs). This complements `sim_poi_prelim` which generates data for persons of interest. Supports two case types: missing children and missing migrants.

Usage

```
sim_mp_prelim(
  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`	Character. Type of missing person case: • "children": Generates birth date, sex, birth month, and birth place • "migrants": Generates age, sex, height, and region Default: "children".
`dateinit`	Character. Minimum birth date for simulated MPs in "YYYY/MM/DD" format. Only for casetype = "children". Default: "1975/01/01".
`scenario`	Integer (1 or 2). Birth date distribution scenario: • 1: Non-uniform (gamma distribution)

	• 2: Uniform distribution Only for casetype = "children". Default: 1.
`femaleprop`	Numeric (0-1). Proportion of females. Default: 0.5.
`ext`	Numeric. Extension parameter for date simulation. Default: 100.
`numsims`	Integer. Number of MPs to simulate. Default: 10000.
`seed`	Integer. Random seed for reproducibility. Default: 123.
`region`	Character vector. Names of regions/locations. Default: c("North America", "South America", "Africa", "Asia", "Europe", "Oceania").
`regionprob`	Numeric vector. Probabilities for each region. Default: c(0.2, 0.2, 0.2, 0.1, 0.2, 0.1).

Details

This function generates the "ground truth" characteristics of missing persons, while `sim_poi_prelim` generates the observed/recorded characteristics of persons of interest (which may include observation errors or falsified data).

Value

A data.frame with columns depending on casetype:

- **children**: POI-ID, DBD, Sex, Month, Birth place
- **migrants**: UHR-ID, Age, Sex, Height, Region

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

`sim_poi_prelim` for simulating POI data, `sim_lr_prelim` for using this data in LR calculations.

Examples

```
# Simulate missing children data
mp_children <- sim_mp_prelim(casetype = "children", numsims = 100, seed = 123)
head(mp_children)

# Simulate missing migrants data
mp_migrants <- sim_mp_prelim(casetype = "migrants", numsims = 100, seed = 456)
head(mp_migrants)
```

`sim_poi_genetic` *Simulate Genetic Profiles for Persons of Interest*

Description

Generates a database of simulated genetic profiles for persons of interest (POIs) or unidentified human remains (UHRs). Profiles are randomly sampled from population allele frequencies.

Usage

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

Arguments

`numsims`	Integer. Number of genetic profiles to simulate. Default: 100.
`reference`	A named list of allele frequencies in pedtools format. Can be created using `get_allele_freqs`.
`seed`	Integer. Random seed for reproducibility. Default: 123.

Details

This function uses **pedtools** and **forrel** to generate random genetic profiles based on the provided allele frequency database. The profiles represent unrelated individuals sampled from the population.

This is useful for:

- Creating test databases for simulation studies
- Generating H2 (unrelated) profiles for LR calculations
- Educational demonstrations of genetic variation

Value

A data.frame where:

- First column `id`: POI identifier (1 to numsims)
- Subsequent columns: Genetic marker data with allele pairs

Each row represents one simulated individual.

References

Marsico FL, Vigeland MD, Egeland T, Herrera Pinero F (2021). "Making decisions in missing person identification cases with low statistical power." *Forensic Science International: Genetics*, 52, 102519. doi:10.1016/j.fsigen.2021.102519

See Also

`get_allele_freqs` for preparing frequency data, `sim_lr_genetic` for genetic LR simulations.

Examples

```
library(forrel)

# Get frequency data
freqdata <- get_allele_freqs(Argentina)

# Simulate 50 POI profiles
poi_db <- sim_poi_genetic(numsims = 50, reference = freqdata, seed = 123)
head(poi_db)

# Check available markers
names(poi_db)
```

`sim_poi_prelim` *Simulate Preliminary Investigation Data for Persons of Interest*

Description

Generates a simulated database of preliminary investigation data for persons of interest (POIs) or unidentified human remains (UHRs). Supports two case types: missing children and missing migrants.

Usage

```
sim_poi_prelim(
  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`	Character. Type of missing person case: • "children": Generates birth date, sex, birth type, and region • "migrants": Generates age, sex, height, and region Default: "children".
`dateinit`	Character. Minimum birth date for simulated POIs in "YYYY/MM/DD" format. Only used for casetype = "children". Default: "1975/01/01".
`scenario`	Integer (1 or 2). Birth date distribution scenario:

- 1: Non-uniform (gamma distribution, more realistic)
- 2: Uniform distribution

Only used for casetype = "children". Default: 1.

`femaleprop` Numeric (0-1). Proportion of females in the simulated population. Default: 0.5.

`ext` Numeric. Extension parameter:

- Scenario 1: Scale factor for gamma distribution
- Scenario 2: Number of days range

Default: 100.

`numsims` Integer. Number of POIs/UHRs to simulate. Default: 10000.

`seed` Integer. Random seed for reproducibility. Default: 123.

`birthprob` Numeric vector of length 3. Probabilities for birth type: c(home_birth, hospital_birth, unknown/adoption). Only for "children". Default: c(0.09, 0.9, 0.01).

`region` Character vector. Names of regions/locations. Default: c("North America", "South America", "Africa", "Asia", "Europe", "Oceania").

`regionprob` Numeric vector. Probabilities for each region. Must sum to 1 and have same length as `region`. Default: c(0.2, 0.2, 0.2, 0.1, 0.2, 0.1).

Details

For missing children cases, this function simulates characteristics of children who may have been taken during periods of conflict or human rights violations, with their identity documents potentially falsified.

For missing migrants cases, this simulates characteristics of unidentified human remains that may correspond to missing migrants.

The birth date distribution in scenario 1 uses a gamma distribution with shape=12, which creates a more realistic non-uniform pattern of births.

Value

A data.frame with columns depending on casetype:

- **children**: POI-ID, DBD (declared birth date), Sex, Birth-type, Region
- **migrants**: UHR-ID, Age, Sex, Height, Region

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

`sim_mp_prelim` for simulating missing person data, `sim_lr_prelim` for using this data in LR calculations.

Examples

```
# Simulate children case database
db_children <- sim_poi_prelim(
  casetype = "children",
  dateinit = "1975/01/01",
  scenario = 1,
  numsims = 100,
  seed = 123
)
head(db_children)

# Simulate migrants case database
db_migrants <- sim_poi_prelim(
  casetype = "migrants",
  numsims = 100,
  seed = 456
)
head(db_migrants)
summary(db_migrants$Age)
```

`sim_posterior` *Simulate Posterior Odds Combining Genetic and Non-Genetic Evidence*

Description

Simulates posterior odds distributions by combining prior probabilities with likelihood ratios from both genetic and non-genetic (preliminary investigation) evidence. This implements a full Bayesian integration of multiple evidence sources.

Usage

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

Arguments

`datasim`	Output from `sim_lr_genetic` containing genetic LR simulations.
`Prior`	Numeric (0-1). Prior probability for H1 (that POI is MP). Default: 0.01.
`PriorModel`	Character. How to incorporate preliminary evidence: • "prelim": Combines prior with preliminary data LRs (sex, age, color) • "uniform": Uses only the prior probability without preliminary LRs Default: "prelim".
`eps`	Numeric (0-1). Error rate for sex observation. Default: 0.05.
`erRs`	Numeric (0-1). Error rate for sex in database. Default: 0.01.
`epc`	Matrix. Hair color error matrix from `error_matrix_hair`.
`erRc`	Matrix. Hair color error matrix for database.
`MPc`	Integer (1-5). Missing person's hair color. Default: 1.
`epa`	Numeric (0-1). Error rate for age. Default: 0.05.
`erRa`	Numeric (0-1). Error rate for age in database. Default: 0.01.
`MPa`	Numeric. Missing person's age. Default: 10.
`MPr`	Numeric. Age range tolerance. Default: 2.

Details

Posterior odds are calculated as:

$$Posterior = Prior \times LR_{prelim} \times LR_{genetic}$$

Where:

- Prior = P(H1) / P(H2) = Prior / (1 - Prior)
- LR_prelim = LR_sex * LR_age * LR_color
- LR_genetic = from genetic simulation

Value

A data.frame with two columns:

- `Unrelated`: Posterior odds under H2
- `Related`: Posterior odds under H1

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

`sim_lr_genetic` for genetic simulations, `lr_sex`, `lr_age`, `lr_hair_color` for preliminary evidence LRs.

Examples

```
library(forrel)

# Setup pedigree
x <- linearPed(2)
x <- setMarkers(x, locusAttributes = NorwegianFrequencies[1:5])
x <- profileSim(x, N = 1, ids = 2)

# Simulate genetic LRs
datasim <- sim_lr_genetic(x, missing = 5, numsims = 50, seed = 123)

# Compute posterior odds with preliminary evidence
post <- sim_posterior(datasim, Prior = 0.01, PriorModel = "prelim")
head(post)

# Visualize
plot_lr_distribution(post)
```

---

sim_reference_pop *Simulate Reference Population with Pigmentation Traits*

Description

Generates a simulated population dataset with correlated pigmentation characteristics (hair color, skin color, eye color). The traits are simulated using conditional probability distributions that reflect realistic correlations between these characteristics.

Usage

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

Arguments

n	Integer. Number of individuals to simulate. Default: 1000.
seed	Integer. Random seed for reproducibility. Default: 1234.

Details

Hair color categories:

1. Blonde/Light
2. Light brown
3. Medium brown
4. Dark brown
5. Black

The simulation uses conditional probability distributions where:

- Hair color is sampled first from population frequencies
- Skin color is sampled conditional on hair color
- Eye color is sampled conditional on both hair and skin color

This captures realistic correlations (e.g., darker hair tends to co-occur with darker skin and eyes).

Value

A data.frame with three columns:

- `hair_colour`: Hair color category (1-5)
- `skin_colour`: Skin color category (1-5)
- `eye_colour`: Eye color category (1-5)

Categories are numbered 1 (lightest) to 5 (darkest).

References

Marsico FL, et al. (2023). "Likelihood ratios for non-genetic evidence in missing person cases." *Forensic Science International: Genetics*, 66, 102891. doi:10.1016/j.fsigen.2023.102891

See Also

`compute_conditioned_prop` for computing proportions, `compute_reference_prop` for reference frequencies, `lr_pigmentation` for pigmentation LR calculations.

Examples

```
# Simulate a population of 500 individuals
pop_data <- sim_reference_pop(n = 500, seed = 123)
head(pop_data)

# Check trait distributions
table(pop_data$hair_colour)
table(pop_data$skin_colour)
table(pop_data$eye_colour)

# Use for LR calculations
conditioned <- compute_conditioned_prop(pop_data, h = 1, s = 1, y = 1,
                                        eh = 0.01, es = 0.01, ey = 0.01)
unconditioned <- compute_reference_prop(pop_data)
```

`threshold_rates` *Compute Error Rates at a Specific Threshold*

Description

Calculates error rates and performance metrics for a given likelihood ratio (LR) threshold, including:

- False Positive Rate (FPR)
- False Negative Rate (FNR)
- Matthews Correlation Coefficient (MCC)

Usage

```
threshold_rates(datasim, threshold)
```

Arguments

`datasim`	A data.frame with columns `Related` and `Unrelated` containing LR values. Can be output from `sim_lr_genetic`, `sim_lr_prelim`, `lr_to_dataframe`, or `lr_combine`.
`threshold`	Numeric. The LR threshold value for which to compute error rates. Cases with LR > threshold are classified as matches.

Details

If the input is a list (output from `sim_lr_genetic`), it is automatically converted to a data.frame using `lr_to_dataframe`.

Metrics:

- *FPR*: Proportion of unrelated cases incorrectly classified as matches (LR > threshold when H2 is true)
- *FNR*: Proportion of related cases incorrectly classified as non-matches (LR < threshold when H1 is true)
- *TPR*: 1 - FNR (sensitivity, recall)
- *TNR*: 1 - FPR (specificity)
- *MCC*: Matthews Correlation Coefficient, ranges from -1 to +1:
  - +1: Perfect classification
  - 0: Random classification
  - -1: Completely wrong classification

Value

Prints the error rates and MCC, and invisibly returns a named list with components:

- `FNR`: False Negative Rate
- `FPR`: False Positive Rate
- `TPR`: True Positive Rate
- `TNR`: True Negative Rate
- `MCC`: Matthews Correlation Coefficient

References

Marsico FL, Vigeland MD, Egeland T, Herrera Pinero F (2021). "Making decisions in missing person identification cases with low statistical power." *Forensic Science International: Genetics*, 52, 102519. doi:10.1016/j.fsigen.2021.102519

Matthews BW (1975). "Comparison of the predicted and observed secondary structure of T4 phage lysozyme." *Biochimica et Biophysica Acta*, 405(2), 442-451.

See Also

`decision_threshold` for finding optimal threshold, `plot_decision_curve` for visualizing the FPR/FNR trade-off.

Examples

```
# Simulate LRs
lr_sims <- sim_lr_prelim("sex", numsims = 500, seed = 123)

# Check error rates at threshold = 10
rates <- threshold_rates(lr_sims, threshold = 10)

# Access individual metrics
rates$FPR
rates$MCC

# Compare different thresholds
threshold_rates(lr_sims, threshold = 5)
threshold_rates(lr_sims, threshold = 50)
threshold_rates(lr_sims, threshold = 100)
```

USA *STR Allele Frequencies from United States*

Description

Population allele frequency data for 29 autosomal Short Tandem Repeat (STR) markers from the United States population. Includes both core CODIS loci and extended markers.

Usage

```
data(USA)
```

Format

A data frame with 97 rows (alleles) and 30 columns. First column is Allele (repeat number), remaining columns are allele frequencies for each STR marker.

Details

This dataset contains allele frequencies for 29 STR markers: CSF1PO, D10S1248, D12S391, D13S317, D16S539, D18S51, D19S433, D1S1656, D21S11, D22S1045, D2S1338, D2S441, D3S1358, D5S818, D6S1043, D7S820, D8S1179, F13A01, F13B, FESFPS, FGA, LPL, Penta_C, Penta_D, Penta_E, SE33, TH01, TPOX, vWA.

This dataset is compatible with the expanded CODIS core loci and includes additional markers for higher discrimination power.

Source

NIST Population Data. Format compatible with **pedtools** and **forrel** packages.

References

Hill CR, et al. (2013). "U.S. population data for 29 autosomal STR loci." *Forensic Science International: Genetics*, 7(3), e82-e83. doi:10.1016/j.fsigen.2012.12.004

See Also

`get_allele_freqs` for extracting frequencies, `sim_lr_genetic` for LR simulations.

Other frequency databases: `Argentina`, `Europe`, `Asia`, `Austria`, `BosniaHerz`, `China`, `Japan`

Examples

```
# Load the dataset
data(USA)

# Check CODIS core loci are present
codis <- c("CSF1PO", "D3S1358", "D5S818", "D7S820", "D8S1179",
           "D13S317", "D16S539", "D18S51", "D21S11", "FGA",
           "TH01", "TPOX", "vWA")
all(codis %in% names(USA))
```

Index