Package: mispitools (via r-universe)

October 26, 2026

Title Missing Person Identification Tools

Version 2.0.1

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 pedtools, dplyr, tidyr, stats, reshape2, patchwork, graphics, ggplot2, shiny, Rcpp, RcppArmadillo

LinkingTo Rcpp, RcppArmadillo

Suggests forrel, Familias, pedprobr, pedmut, fbnet, DirichletReg, shinythemes, knitr, rmarkdown, testthat (>= 3.0.0)

SystemRequirements GNU make, C++17

VignetteBuilder knitr

RoxygenNote 7.3.3

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

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

NeedsCompilation yes

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

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

Config/pak/sysreqs make

Repository https://cran.r-universe.dev

Date/Publication 2026-08-26 21:10:07 UTC

RemoteUrl https://github.com/cran/mispitools

RemoteRef HEAD

RemoteSha 446744dbee4c78c85df0d419198a37f4d973b9c7

Contents

mispitools-package . . . 3
Argentina . . . 5
as_lr_dist . . . 7
Asia . . . 8
Austria . . . 9
belief_trajectory . . . 10
binary_belief_trajectory . . . 11
BosniaHerz . . . 12
calibrate_concentration_cutoff . . . 13
China . . . 15
compute_conditioned_prop . . . 16
compute_reference_prop . . . 17
concentration_index . . . 18
concentration_index_positive . . . 19
cpt_missing_person . . . 20
cpt_population . . . 22
decision_threshold . . . 24
entropy_log10 . . . 26
error_matrix_hair . . . 26
Europe . . . 29
familias_trajectory . . . 30
fragility_report . . . 32
get_allele_freqs . . . 33
herfindahl_index . . . 34
Japan . . . 35
kl_divergence_log10 . . . 36
leave_one_out . . . 37
lr_age . . . 38
lr_birthdate . . . 40
lr_combine . . . 43
lr_compute_pigmentation . . . 44
lr_distribution . . . 46
lr_hair_color . . . 47
lr_pigmentation . . . 49
lr_sensitivity . . . 51
lr_sex . . . 52

lr_to_dataframe 54
marker_model 56
mispitools_app 57
nongenetic_feature 58
per_marker_kl 61
per_marker_kl_profile 62
plot.lr_dist 63
plot_cpt 64
plot_decision_curve 66
plot_lr_distribution 67
print.marker_model 69
print.nongenetic_feature 69
quantile.lr_dist 70
shannon_concentration 71
sim_lr_genetic 72
sim_lr_prelim 74
sim_mp_prelim 76
sim_poi_prelim 78
sim_reference_pop 80
summary.lr_dist 81
threshold_rates 82
trajectory_metrics 84
USA 85

Index **87**

`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_reference_pop`: Simulate a reference population with traits
- `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

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)

Authors:

- Suisei Nakagawa <suisei.nakagawa@uni.minerva.edu>

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)
```

`as_lr_dist` *Coerce to a Likelihood-Ratio Distribution Object*

Description

Wraps a sparse $(\log_{10}\mathrm{LR}, P(\cdot \mid H_1), P(\cdot \mid H_2))$ table in an `lr_dist` object so the decision-theoretic `summary()` and `plot()` methods can be applied. This is the low-level constructor; the model-aware `lr_distribution()` wrapper is added in a later release and produces the same class.

Usage

```
as_lr_dist(x)
```

Arguments

`x`	A data.frame or list with numeric components `log10_lr`, `p_h1` and `p_h2`, all of the same length. Typically the output of the per-marker LR distribution engine.

Details

Each row is an atom of the discrete LR distribution: `log10_lr` is the log10 likelihood ratio, `p_h1` its probability under H_1 (the person of interest is the missing person) and `p_h2` its probability under H_2 (unrelated). `+Inf` / `-Inf` atoms (which arise under `mutation = "none"`) are allowed and handled by the summaries with the limit $0 \log 0 = 0$.

Value

An object of class `lr_dist` (a data.frame with the three columns), suitable for `summary.lr_dist()` and `plot.lr_dist()`.

See Also

`summary.lr_dist()`, `plot.lr_dist()`

Examples

```
d <- data.frame(
  log10_lr = c(-1, 0, 2),
  p_h1 = c(0.1, 0.3, 0.6),
  p_h2 = c(0.6, 0.3, 0.1)
)
x <- as_lr_dist(d)
summary(x)
```

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
```

`belief_trajectory` *Compute a belief trajectory from sequential evidence*

Description

Given a prior distribution over hypotheses and a sequence of likelihood ratio vectors (one per evidence step), computes the full Bayesian belief trajectory by sequential multiplicative updating.

Usage

```
belief_trajectory(prior, lr_list)
```

Arguments

`prior`	Numeric vector. Prior distribution over the n hypotheses (must sum to 1, entries in $[0, 1]$).
`lr_list`	List of numeric vectors. Each element is the likelihood ratio vector for an evidence step, with one entry per hypothesis (same length as `prior`).

Details

The update rule is the classical multiplicative Bayesian update:

$$P_t(H_i) \propto P_{t-1}(H_i) \cdot \mathrm{LR}_t(H_i)$$

with renormalization after each step. In the two-hypothesis case ($n = 2$), the ratio of the components of `lr_list[[t]]` recovers the classical forensic likelihood ratio.

Value

A matrix with $(T + 1)$ rows and n columns, where row 1 is the prior and row $t + 1$ is the posterior distribution after the first t evidence steps. Each row is a probability distribution.

References

Marsico, F. L. & Egeland, T. (in preparation). Belief dynamics during the investigative process.

See Also

`binary_belief_trajectory` for the two-hypothesis special case, `trajectory_metrics` for summary statistics.

Examples

```
# Two hypotheses, three evidence steps
prior <- c(0.5, 0.5)
lr_list <- list(c(3, 1), c(1, 2), c(2, 1))
traj <- belief_trajectory(prior, lr_list)
print(traj)
```

`binary_belief_trajectory` *Compute a binary belief trajectory from per-marker likelihood ratios*

Description

For the two-hypothesis forensic case, computes the trajectory of $P(H_1)$ as markers are added sequentially.

Usage

```
binary_belief_trajectory(per_marker_lrs, prior_odds = 1)
```

Arguments

`per_marker_lrs`	Named numeric vector. Per-marker likelihood ratios $\mathrm{LR}_k = P(\mathrm{profile}_k \mid H_1)/P(\mathrm{profile}_k \mid H_2)$.
`prior_odds`	Numeric scalar. Prior odds for H_1 vs H_2 (default 1, i.e., equal prior).

Value

A data frame with one row per step (including the prior at step 0) and the following columns:

`step` Integer, 0 for prior, 1 to K for markers.

`marker` Character, "Prior" or the marker name.

`log10_lr` Per-step $\log_{10} \mathrm{LR}_k$.

`cum_log10_lr` Cumulative $\log_{10} \mathrm{LR}$ up to and including step `step`.

`posterior_h1` $P(H_1)$ at step `step`.

`posterior_h2` $P(H_2)$ at step `step`.

References

Marsico, F. L. & Egeland, T. (in preparation). Belief dynamics during the investigative process.

See Also

`belief_trajectory` for the general multi-hypothesis case, `concentration_index` for fragility diagnostics.

Examples

```
lrs <- c(D3S1358 = 5.2, TH01 = 1.8, D21S11 = 12.0, D18S51 = 3.1)
binary_belief_trajectory(lrs)
```

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)
```

`calibrate_concentration_cutoff` *Calibrate pedigree-specific concentration cutoff*

Description

Simulates the distribution of the inclusion-fragility concentration index C_W^+ under the prosecution hypothesis H_p for a given pedigree and marker panel, and returns the chosen upper quantile as a case-flagging cutoff. Profiles whose observed C_W^+ exceeds the cutoff should trigger a leave-one-out review before reporting.

Usage

```
calibrate_concentration_cutoff(
  reference,
  missing,
  numsims = 1500,
  probs = 0.9,
  seed = 123,
  numCores = 1
)
```

Arguments

`reference`	A `pedtools::ped` object with markers attached and founder profiles simulated, as expected by `sim_lr_genetic`.
`missing`	Character or numeric. ID of the missing person in the pedigree.
`numsims`	Integer. Number of H_p simulations used to build the empirical C_W^+ distribution. Default 1500.
`probs`	Numeric in $(0, 1)$. Quantile of the H_p distribution to use as the cutoff. Default 0.90 (matches the paper convention).
`seed`	Integer. Random seed passed through to `sim_lr_genetic`.
`numCores`	Integer. Cores for the underlying `forrel::profileSim` call.

Details

The calibration uses only the H_p branch of `sim_lr_genetic`: for each simulated matching profile, the per-marker LR vector is extracted and passed to `concentration_index_positive`. Simulations with no positive per-marker support (empty numerator) are dropped before taking the quantile.

The default `probs = 0.90` matches the cutoff convention used in Marsico & Egeland (in preparation): a 10\ H_p is deemed acceptable in exchange for catching concentrated cases where the combined LR depends heavily on a single marker.

Value

A list with elements

`cutoff` The `probs`-quantile of C_W^+ under H_p.

`probs` The quantile level used.

`distribution` Numeric vector of simulated C_W^+ values, length `numsims`.

`total_log10_lr` Numeric vector of simulated $\log_{10}$ LR totals under H_p.

`numsims` The number of simulations actually used (profiles with non-positive total support are discarded).

References

Marsico, F. L. & Egeland, T. (in preparation). Belief dynamics during the investigative process.

See Also

`sim_lr_genetic`, `concentration_index_positive`, `fragility_report`.

Examples

```
## Not run:
  library(pedtools); library(forrel)
  x <- linearPed(2)
  x <- setMarkers(x, locusAttributes = NorwegianFrequencies[1:15])
  x <- profileSim(x, N = 1, ids = 2)
  cal <- calibrate_concentration_cutoff(x, missing = 5,
                                        numsims = 500, probs = 0.90)
  cal$cutoff

## End(Not run)
```

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, ]
```

`concentration_index` *Concentration index of per-step evidence contributions*

Description

Computes the fraction of the total absolute evidence weight carried by the single most impactful step. Ranges from $1/T$ (uniform contributions) to 1 (single-step dominance). Under the single-failure-mode axiomatization (Theorem T-001 in the supplementary material of Marsico & Egeland, in prep.), this is the canonical concentration measure for forensic fragility: it equals the fraction of the total weight of evidence lost under worst-case removal of a single piece of evidence.

Usage

```
concentration_index(weights)
```

Arguments

`weights`	Numeric vector. Per-step contributions, e.g., $\log_{10}(\mathrm{LR}_k)$ for the two-hypothesis forensic case or per-step Kullback-Leibler divergences for the general case.

Details

The formula is

$$C_W(w) = \frac{\max_k |w_k|}{\sum_k |w_k|}$$

with absolute values taken to handle both supporting ($w_k > 0$) and excluding ($w_k < 0$) evidence symmetrically. For inclusion-fragility specifically (restricting to positive weights), use `concentration_index_positive`. For robustness checks, compare against `herfindahl_index` and `shannon_concentration`.

Value

Numeric scalar in $[1/T, 1]$ where T is the length of `weights`. Returns 0 if all weights are zero.

References

Marsico, F. L. & Egeland, T. (in preparation). Belief dynamics during the investigative process.
Slooten, K. (2021). The analogy between DNA kinship and DNA mixture evaluation. *Forensic Science International: Genetics* 51, 102444.

See Also

`concentration_index_positive`, `herfindahl_index`, `shannon_concentration`, `leave_one_out`.

Examples

```
# Balanced: all markers contribute equally
concentration_index(rep(0.7, 15))  # -> 1/15

# Concentrated: one marker dominates
concentration_index(c(5, rep(0.1, 14)))  # -> ~0.78
```

---

concentration_index_positive

Inclusion-fragility concentration index (positive weights only)

Description

Signed variant of `concentration_index` that uses only the supporting ($w_k > 0$) contributions. For forensic inclusion cases, this measures the fragility of the *support* for the prosecution hypothesis specifically, ignoring any excluding markers.

Usage

```
concentration_index_positive(weights)
```

Arguments

`weights`	Numeric vector of per-step contributions (typically $\log_{10} \mathrm{LR}_k$).

Details

Defined as

$$C_W^+(w) = \frac{\max_k w_k^+}{\sum_k w_k^+}$$

with $w_k^+ = \max(w_k, 0)$. Under Theorem T-003.1 of the Belief Dynamics framework, C_W^+ equals the fraction of the total positive weight of evidence lost under adversarial removal of the single most impactful supporting marker.

Value

Numeric scalar in $[0, 1]$. Returns 0 if no weight is positive.

References

Marsico, F. L. & Egeland, T. (in preparation). Belief dynamics during the investigative process.

See Also

concentration_index, leave_one_out.

Examples

```
# Mostly supporting, one dominant
concentration_index_positive(c(2.5, 0.3, 0.4, 0.2, 0.1))

# Mixed with one strong exclusion: the exclusion does NOT contribute
concentration_index_positive(c(1.0, 0.8, 0.9, -5.0, 1.1))
```

---

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).

Deprecation

Soft-deprecated in mispitools 2.0. The per-feature H1 CPT it builds as an outer product is generalised by nongenetic_feature plus the unified per-feature engine (one feature per trait, combined downstream). The legacy function still works for the 2.0 release-candidate cycle and will be removed afterwards.

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

nongenetic_feature for the unified replacement, 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).

Deprecation

Soft-deprecated in mispitools 2.0. The per-feature H2 marginal it builds as an outer product is generalised by `nongenetic_feature` plus the unified per-feature engine (one feature per trait, combined downstream). The legacy function still works for the 2.0 release-candidate cycle and will be removed afterwards.

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

`nongenetic_feature` for the unified replacement, `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)
```

entropy_log10 *Shannon entropy in base-10 (bans)*

Description

Computes the Shannon entropy of a probability distribution, using base-10 logarithm so the result is expressed in *bans* (the forensic convention for weight-of-evidence units).

Usage

```
entropy_log10(p, epsilon = 1e-20)
```

Arguments

p	Numeric vector. Probability distribution (must sum to 1, entries in $[0, 1]$).
epsilon	Numeric scalar. Small positive constant to avoid $\log(0)$ for zero-probability outcomes. Default 10^{-20}.

Value

Numeric scalar. Shannon entropy $H(p) = -\sum_i p_i \log_{10} p_i$ in bans.

See Also

kl_divergence_log10, trajectory_metrics.

Examples

```
entropy_log10(c(0.5, 0.5))          # log10(2)
entropy_log10(rep(1/12, 12))        # log10(12)
entropy_log10(c(1, 0, 0, 0))        # ~0
```

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
```

`familias_trajectory` *Belief trajectory metrics from a* `Familias::FamiliasPosterior` *result*

Description

Convenience wrapper that extracts the per-marker likelihood ratios from the result list returned by `Familias::FamiliasPosterior` and computes the full belief-trajectory machinery of Marsico & Egeland (in preparation): the binary belief trajectory, the trajectory metrics (entropy, per-step KL divergence, total-variation path length, and the three concentration measures), the signed concentration index C_W^+ for fragility of inclusions, and the per-marker leave-one-out analysis.

Usage

```
familias_trajectory(familias_result, test_pedigree = 2, ref_pedigree = 1)
```

Arguments

`familias_result`	A list with the structure returned by `Familias::FamiliasPosterior`. Must contain a numeric matrix named `LRperMarker` with rows indexing markers and columns indexing candidate pedigrees. Column `ref_pedigree` is assumed to contain the reference hypothesis (typically all ones after normalization).
`test_pedigree`	Integer or character. Index or name of the alternative pedigree whose per-marker LRs (as a ratio against the reference pedigree) should be used to build the belief trajectory. Defaults to 2 (i.e., the first non-reference pedigree), which is the standard arrangement in two-pedigree comparisons.
`ref_pedigree`	Integer or character. Index or name of the reference pedigree. Defaults to 1. Used for sanity checks only; the actual LR-per-marker values for `test_pedigree` are taken from `familias_result$LRperMarker` directly because `FamiliasPosterior` already normalizes the matrix against the reference column.

Details

`Familias::FamiliasPosterior` computes posterior probabilities of candidate pedigrees given DNA evidence. Its return value includes a `LRperMarker` matrix of per-locus likelihood ratios already normalized against the reference pedigree. This wrapper simply selects the relevant column

and hands the resulting per-marker LR vector to the Belief Dynamics trajectory machinery, so that a user of Familias can obtain trajectory metrics and fragility diagnostics in a single function call without manually constructing the per-marker sequence.

If `Familias` is not installed, this function does not depend on it — the input is expected to be a list with the structure described above, which can equally well be constructed by hand for testing or from other sources (e.g., `forrel::missingPersonLR`).

Value

A list with components:

`lrs` Named numeric vector of per-marker likelihood ratios for the chosen test pedigree against the reference.

`trajectory` Data frame from `binary_belief_trajectory` with the posterior at each step, cumulative log-LR, and the per-step log-LR.

`metrics` List from `trajectory_metrics` with entropy, per-step KL divergence, cumulative KL from prior, per-step total-variation, path length, and the three concentration measures.

`concentration_positive` Numeric scalar. The signed concentration index C_W^+ restricted to positive (supporting) per-marker contributions; the primary fragility diagnostic of the Belief Dynamics framework.

`leave_one_out` Data frame from `leave_one_out` giving the per-marker fragility table.

References

Marsico, F. L. & Egeland, T. (in preparation). Belief dynamics during the investigative process. Egeland, T., Mostad, P. & Simonsson, I. (2015). Relationship inference with Familias and R. Academic Press.

See Also

`binary_belief_trajectory`, `trajectory_metrics`, `concentration_index_positive`, `leave_one_out`.

Examples

```
# Synthetic Familias-like result with 6 markers and 2 pedigrees
fam <- list(
  LRperMarker = matrix(
    c(1, 1, 1, 1, 1, 1,          # reference pedigree, col 1
      5.2, 1.8, 12.0, 3.1, 0.8, 2.7), # test pedigree,     col 2
    nrow = 6, ncol = 2,
    dimnames = list(c("D3S1358", "TH01", "D21S11",
                      "D18S51",  "CSF", "vWA"),
                    c("Unrelated", "GrandparentGrandchild"))
  )
)
result <- familias_trajectory(fam, test_pedigree = 2)
result$concentration_positive
result$leave_one_out
```

`fragility_report` *Per-case fragility report*

Description

Produces the case-level fragility summary described in §5 of Marsico & Egeland (in preparation): total weight of evidence, inclusion concentration index C_W^+, worst-case leave-one-out loss, residual support after removal of the most impactful marker, and an optional comparison against a pedigree-specific cutoff (typically produced by `calibrate_concentration_cutoff`).

Usage

```
fragility_report(per_marker_lrs, cutoff = NULL, probs = NULL)
```

Arguments

`per_marker_lrs`	Named numeric vector of per-marker likelihood ratios (strictly positive). Names become the marker labels in the report.
`cutoff`	Optional numeric scalar. If supplied, the report flags cases whose observed C_W^+ exceeds `cutoff` as requiring a leave-one-out review before final reporting.
`probs`	Optional numeric in $(0, 1)$. The quantile level at which `cutoff` was calibrated. Used only to enrich the reportable sentence (`"below the 90th-percentile cutoff"` etc.); ignored if `cutoff` is `NULL`.

Details

The report is a direct operationalization of Paper 1 §5:

- W is the combined weight of evidence in bans.
- C_W^+ answers the question "what fraction of the inclusion support comes from a single marker?"
- $(1 - C_W^+)W$ is the residual support after the single-marker worst case, i.e.\ what the analyst would be left with if a successful challenge to the top marker were sustained.
- The flag is raised when C_W^+ exceeds the pedigree-specific cutoff, which in turn is typically the 90th percentile of the H_p simulation distribution produced by `calibrate_concentration_cutoff`.

Value

A list with elements

`total_log10_lr` Total weight of evidence $W = \sum_k \log_{10} \mathrm{LR}_k$ (in bans).

`concentration` Inclusion concentration index $C_W^+ \in [0, 1]$.

`top_marker` Name of the single most impactful supporting marker.

`top_log10_lr` Log-LR of the top marker (in bans).

`residual_log10_lr` Total support that survives worst-case removal of the top marker, $(1-C_W^+)W$.

`flag` Logical. `TRUE` if `cutoff` was supplied and $C_W^+ >$ cutoff; `NA` otherwise.

`statement` Character scalar. A ready-to-paste natural-language fragility statement for the case report.

References

Marsico, F. L. & Egeland, T. (in preparation). Belief dynamics during the investigative process.

See Also

concentration_index_positive, leave_one_out, calibrate_concentration_cutoff.

Examples

```
# Balanced case: 15 markers, similar contributions
set.seed(1)
lrs <- setNames(runif(15, 1.5, 3.5), paste0("M", 1:15))
fragility_report(lrs)

# Concentrated case: one dominant marker
lrs2 <- setNames(c(1e5, rep(1.05, 14)), paste0("M", 1:15))
fragility_report(lrs2, cutoff = 0.20, probs = 0.90)
```

---

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])
```

---

herfindahl_index *Herfindahl-Hirschman concentration of evidence contributions*

Description

Computes the Herfindahl-Hirschman index on the normalized per-step contributions. A robustness check / alternative concentration measure to `concentration_index`. Ranges from $1/T$ (uniform) to 1 (single-step dominance).

Usage

```
herfindahl_index(weights)
```

Arguments

`weights`	Numeric vector of per-step contributions.

Details

$$H(w) = \sum_{k=1}^{T} p_k^2, \qquad p_k = \frac{|w_k|}{\sum_j |w_j|}$$

Unlike concentration_index (which depends only on the maximum and the sum), Herfindahl depends on the full distribution of weights and is therefore more sensitive to "second-tier" contributors.

Value

Numeric scalar in $[1/T, 1]$. Returns 0 if all weights are zero.

See Also

concentration_index, shannon_concentration.

Examples

```
herfindahl_index(rep(1, 10))           # 0.1
herfindahl_index(c(10, rep(0.1, 9)))  # ~0.92
```

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_divergence_log10 *Kullback-Leibler divergence in base-10 (bans)*

Description

Computes the Kullback-Leibler divergence $D_{\mathrm{KL}}(p\|q)$ between two distributions, in base-10 logarithm so the result is expressed in *bans*.

Usage

```
kl_divergence_log10(p, q, epsilon = 1e-20)
```

Arguments

p	Numeric vector. Target distribution.
q	Numeric vector. Reference distribution (same length as p).
epsilon	Numeric scalar. Small positive constant to avoid $\log(0)$. Default 10^{-20}.

Value

Numeric scalar. $D_{\mathrm{KL}}(p\|q) = \sum_i p_i \log_{10}(p_i/q_i)$ in bans.

See Also

entropy_log10, trajectory_metrics.

Examples

```
p <- c(0.7, 0.3)
q <- c(0.5, 0.5)
kl_divergence_log10(p, q)          # positive
kl_divergence_log10(p, p)          # 0
```

---

leave_one_out *Leave-one-out fragility analysis of evidence contributions*

Description

For each step, computes the total weight of evidence with that step removed. Identifies cases where a single step dominates: if fraction[k] is close to 1, the total weight hinges on that single step and the conclusion would be substantially weakened if it were excluded or challenged.

Usage

```
leave_one_out(per_marker_lrs)
```

Arguments

per_marker_lrs	Named numeric vector. Per-marker likelihood ratios (positive values).

Value

A data frame with columns

marker Marker name.

log10_lr $\log_{10} \mathrm{LR}_k$, this marker's contribution to the total.

total_without Total $\log_{10} \mathrm{LR}$ with this marker removed.

fraction $|\log_{10} \mathrm{LR}_k| / \sum_j |\log_{10} \mathrm{LR}_j|$, this marker's share of the absolute total.

References

Marsico, F. L. & Egeland, T. (in preparation). Belief dynamics during the investigative process.

See Also

concentration_index, concentration_index_positive.

Examples

```
lrs <- c(D3S1358 = 5.2, TH01 = 1.8, D21S11 = 120.0, D18S51 = 3.1)
leave_one_out(lrs)
```

---

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)

Deprecation

Soft-deprecated in mispitools 2.0. This feature is generalised by nongenetic_feature (a continuous feature); collapsing the population grid to two cells (within tolerance vs outside) reproduces the T1 / T0 LR exactly. The legacy function still works for the 2.0 release-candidate cycle and will be removed afterwards.

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

nongenetic_feature for the unified replacement, 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.

Deprecation

Soft-deprecated in mispitools 2.0. Generalised by `nongenetic_feature` (a date feature); the deterministic reference is the Dirichlet mean `alpha / sum(alpha)` over the discrepancy bins, which the legacy stochastic method-of-moments estimator converges to. The legacy function still works for the 2.0 release-candidate cycle and will be removed afterwards.

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

`nongenetic_feature` for the unified replacement, `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_distribution` *Composed likelihood-ratio distribution over a marker profile*

Description

Builds the exact distribution of the total $\log_{10}$ LR for a profile of independent markers. Each marker's per-marker LR distribution is computed from its joint H1 / H2 table, and the distributions are convolved (conditional independence: the total $\log_{10}$ LR is the sum of the per-marker $\log_{10}$ LRs). The result is an `as_lr_dist()` object, so `summary.lr_dist()`, `plot.lr_dist()` and `quantile.lr_dist()` apply directly.

Usage

```
lr_distribution(
  models,
  poi = NULL,
  method = c("exact", "grid"),
  grid_points = 512L
)
```

Arguments

`models`	A `marker_model` object or a `list` of them (see `marker_model()`). A single model is accepted and returns its own per-marker distribution unchanged. List names, when present and non-empty, are recorded in the `"markers"` attribute.
`poi`	Optional character scalar naming the person of interest, applied to every model. When `NULL` (default) the POI is resolved per model by the joint engine (see `per_marker_kl()`).
`method`	Composition method. `"exact"` (default) performs the exact sparse convolution and is bit-for-bit with the reference engine. `"grid"` projects each distribution onto a regular lattice before convolving (mass- and mean-preserving, with a shape error that shrinks with `grid_points`); it requires finite support and is useful for very long profiles.
`grid_points`	Integer number of lattice points used when `method = "grid"` (ignored for `"exact"`). Default `512`.

Details

Markers are assumed conditionally independent given the hypothesis (no linkage); linked-marker composition arrives with the F5 linkage work. `+Inf` / `-Inf` atoms (which arise under `mutation = "none"` when a hypothesis assigns zero mass to a state the other supports) propagate through the exact convolution and are reported by `summary.lr_dist()`; `method = "grid"` rejects infinite support.

Two atoms carry the same `log10` LR when they agree to a relative tolerance of 1e-12, rather than bit for bit. A single atom is a real number that the engine can reach by more than one arithmetic route, and on a platform that contracts `a * b + c` into a fused multiply-add the routes differ in the

last bits; grouping by exact equality would make the size of the support depend on the compiler. The tolerance sits four orders of magnitude above that rounding noise and four below the closest genuinely distinct pair observed, so it merges duplicates and nothing else.

Value

An object of class `lr_dist` (a data.frame with columns `log10_lr`, `p_h1`, `p_h2`), with attributes `"markers"` (the marker identifiers in input order), `"n_markers"`, `"method"` and `"poi"` (the resolved POI; a vector if it differs across models).

See Also

`marker_model()`, `as_lr_dist()`, `summary.lr_dist()`, `plot.lr_dist()`, `quantile.lr_dist()`, `per_marker_kl_profile()`.

Examples

```
if (requireNamespace("pedtools", quietly = TRUE)) {
  ped <- pedtools::nuclearPed(1)
  models <- list(
    D3 = marker_model(ped, "D3", c("15" = 0.4, "16" = 0.6),
                      mutation = list(model = "equal", rate = 1e-3)),
    vWA = marker_model(ped, "vWA", c("a" = 0.2, "b" = 0.3, "c" = 0.5),
                       mutation = list(model = "equal", rate = 1e-3))
  )
  d <- lr_distribution(models)
  summary(d)
  quantile(d)
}
```

---

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.

Deprecation

Soft-deprecated in mispitools 2.0. This feature is generalised by nongenetic_feature (a categorical feature with a full confusion matrix); the unified per-feature engine reproduces epc[MPc, o] / Pc[o] exactly. The legacy function still works for the 2.0 release-candidate cycle and will be removed afterwards.

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

nongenetic_feature for the unified replacement, 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)

Deprecation

Soft-deprecated in mispitools 2.0. Combined pigmentation is generalised by `nongenetic_feature` (a categorical feature over the joint pigmentation classes). The legacy function still works for the 2.0 release-candidate cycle and will be removed afterwards.

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

`nongenetic_feature` for the unified replacement, `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.

Deprecation

Soft-deprecated in mispitools 2.0. This feature is generalised by nongenetic_feature (a categorical feature); the deterministic LR it produces is reproduced exactly by the unified per-feature engine. The legacy function still works for the 2.0 release-candidate cycle and will be removed afterwards.

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

nongenetic_feature for the unified replacement, 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)
```

`marker_model` *Marker model (S3)*

Description

Constructor for a single-marker forensic genetic model. Bundles the pedigree, the marker identifier, the population allele frequencies, the mutation model, and (optionally) a linkage descriptor. The resulting object is the unit consumed by the per-marker engines (`per_marker_kl`, `per_marker_lr_dist`) that arrive in later milestones.

This is the public entry point for the F1 reference engine. The constructor only validates and stores its inputs; the actual joint CPT construction is done downstream.

Usage

```
marker_model(
  ped,
  marker_id,
  freqs,
  mutation = list(model = "none", rate = 0),
  linkage = NULL,
  tol = 1e-06
)
```

Arguments

`ped`	A `pedtools::ped` object describing the pedigree. Singletons and `pedList` objects are not accepted at this stage.
`marker_id`	Character scalar. Identifier used to label the marker (e.g. `"D3S1358"`). When `ped` already carries a marker with this name, downstream engines will read the genotypes of typed individuals from it; otherwise the model represents the prospective situation prior to typing.
`freqs`	Named numeric vector of population allele frequencies. Names are allele labels, values must lie in `[0, 1]` and sum to 1 within `tol`. At least two alleles are required.
`mutation`	List describing the mutation model. Required field `model` is one of `"none"`, `"equal"`, `"stepwise"`, `"asymmetric"`. For `"none"` no further fields are needed; for the other models a numeric `rate` in `[0, 1)` is required. `"stepwise"` additionally accepts `ratio` (geometric step ratio in `(0, 1)`); `"asymmetric"` additionally accepts `ratio` and `bias` (u parameter, in `[0, 1]`). Defaults to `list(model = "none", rate = 0)`.
`linkage`	Either `NULL` (default; unlinked marker) or a list with fields `partner` (character, the marker identifier of the linked partner) and `theta` (recombination fraction in `[0, 0.5]`). Linkage support is wired in milestone F5; F1 only validates the structure.
`tol`	Numeric tolerance used when checking that `freqs` sums to 1. Defaults to `1e-6`.

Value

An object of class "marker_model" carrying the validated inputs as components ped, marker_id, freqs, mutation, linkage, and alleles (names(freqs)).

Examples

```
if (requireNamespace("pedtools", quietly = TRUE)) {
  ped <- pedtools::nuclearPed(1)
  freqs <- c("12" = 0.2, "13" = 0.3, "14" = 0.5)
  mm <- marker_model(
    ped = ped,
    marker_id = "M1",
    freqs = freqs,
    mutation = list(model = "equal", rate = 1e-3)
  )
  print(mm)
}
```

---

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()
}
```

---

nongenetic_feature *Non-genetic feature model (S3)*

Description

Constructor for a single non-genetic forensic feature (sex, age, region, hair colour, eye colour, pigmentation, birth date, or a user-defined `"custom"` feature). It is the non-genetic counterpart of `marker_model()`: it bundles the recorded observation, the reference/observation sub-model, the population reference, and the observation-error model into a single object that the per-feature engines (arriving in milestones F6.3–F6.5) consume.

Like `marker_model()`, this constructor only validates and stores its inputs; no conditional probability table is built here. The point of F6.1 is to fix the structural contract so that the genetic and non-genetic paths are symmetric: every non-genetic feature reduces to a CPT under `H1` (missing person, with observation error) and a CPT under `H2` (population marginal), then `LR = P(D | H1) / P(D | H2)` — exactly the shape of the per-marker genetic path.

Usage

```
nongenetic_feature(
  type,
  observed,
  model = NULL,
  db_or_freqs = NULL,
  error = NULL,
  tol = 1e-06
)
```

Arguments

type	Character scalar. One of "sex", "age", "region", "hair", "eyes", "pigmentation", "birthdate", "custom".
observed	The recorded observation for the unidentified person. For categorical features a length-one value matching a category of db_or_freqs (numeric labels are matched as characters). For age a single finite numeric. For birthdate either a single Date / a "YYYY-MM-DD" string, or a single finite numeric day discrepancy.
model	Either NULL (a type-appropriate default is filled in) or a named list describing the reference/observation sub-model. For categorical features the field reference is one of "marginal" (use db_or_freqs as H2, the default) or "uniform". For age, reference is "uniform" (with a length-2 increasing numeric range, default c(1, 80)) or "empirical". For birthdate, search is "open" (default) or "closed", with a strictly increasing numeric cuts vector (default c(-120, -30, 30, 120, 240, 360)). For "custom", model must be a named list with a character class field in c("categorical", "continuous", "date").
db_or_freqs	The population reference. Categorical: a named numeric vector (>= 2 categories, entries in [0, 1], summing to 1 within tol). Continuous: NULL when reference = "uniform", otherwise a numeric sample (length >= 2, finite). Date: NULL for an open search, otherwise a non-negative numeric vector of bin frequencies of length length(cuts) + 1 or a data.frame of declared dates.
error	The observation-error model under H1. Categorical: a scalar in [0, 1) (symmetric misclassification) or a square row-stochastic numeric matrix whose dimension equals the number of categories. The matrix is positional: row i / column j are the i-th / j-th category in the order of db_or_freqs (any dimnames are informational only). Continuous: a scalar in [0, 1). Date: a numeric Dirichlet alpha vector, all strictly positive, of length length(cuts) + 1. Defaults are class-appropriate except for "custom", where error is required.
tol	Numeric tolerance for the sum-to-one and row-stochastic checks. Defaults to 1e-6.

Value

An object of class "nongenetic_feature": a list with components type, feature_class, observed, model, error, db_or_freqs, and categories (the category labels for categorical features, NULL otherwise).

Feature classes

Each `type` maps to one of three internal *feature classes* that determine how `db_or_freqs` and `error` are interpreted:

`categorical` `sex`, `region`, `hair`, `eyes`, `pigmentation`. `db_or_freqs` is a named numeric vector of population category frequencies (the `H2` marginal); `error` is either a scalar symmetric misclassification rate or a full row-stochastic confusion matrix `E` with `E[true, observed]`. Discrete support, so the categorical KL is identical to the genetic engine.

`continuous` `age`. The reference is either `"uniform"` over a numeric `range` or `"empirical"` from a sample passed in `db_or_freqs`; `error` is a scalar mis-binning rate.

`date` `birthdate`. A Dirichlet model over signed declared-minus-actual day discrepancy bins (`cuts`); `error` is the Dirichlet `alpha` vector. `search = "open"` uses a uniform `H2`; `search = "closed"` uses database bin frequencies.

`type = "custom"` requires `model` to declare its `class` (one of the three above); validation then follows that class.

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

`marker_model()` for the genetic counterpart; the legacy non-genetic functions `lr_sex()`, `lr_age()`, `lr_hair_color()`, `lr_birthdate()`, `lr_pigmentation()` whose behaviour this framework generalises and which serve as its regression oracles.

Examples

```
# Categorical: biological sex, missing person female, 5% error
f_sex <- nongenetic_feature(
  type = "sex",
  observed = "F",
  db_or_freqs = c(F = 0.5, M = 0.5),
  error = 0.05
)
print(f_sex)

# Categorical hair colour with a full confusion matrix
E <- error_matrix_hair()
f_hair <- nongenetic_feature(
  type = "hair",
  observed = 1,
  db_or_freqs = c("1" = 0.3, "2" = 0.2, "3" = 0.25, "4" = 0.15, "5" = 0.1),
  error = E
)

# Continuous: age, uniform reference over [1, 80]
f_age <- nongenetic_feature(
```

```
  type = "age",
  observed = 42,
  model = list(reference = "uniform", range = c(1, 80)),
  error = 0.05
)

# Date: birth-date discrepancy, open search, default Dirichlet
f_bd <- nongenetic_feature(
  type = "birthdate",
  observed = 45,
  error = c(1, 4, 60, 11, 6, 4, 4)
)
```

`per_marker_kl` *Per-marker bidirectional Kullback-Leibler divergence and expected log10 LR*

Description

Computes, for a single marker model, the expected `log10` likelihood ratio under both hypotheses and the bidirectional Kullback-Leibler divergence between the joint H1 and H2 distributions over pedigree typings. Drives the C++ engine introduced in F2 / F3.1; returns the same numerical content as the internal reference engine to bit-equivalent precision.

Boundary convention follows the reference engine:

- `P_H1(g) > 0` and `P_H2(g) = 0` contribute `+Inf` to `kl_h1_to_h2` and `e_log10_lr_h1`.
- `P_H1(g) = 0` and `P_H2(g) > 0` contribute `+Inf` to `kl_h2_to_h1` and `-Inf` to `e_log10_lr_h2`.
- States with `P_H1 = P_H2 = 0` are skipped (already filtered by the joint engine).

Under `mutation = "none"` the H2 distribution typically spans Mendelian-incompatible states with zero mass under H1, so `kl_h2_to_h1 = +Inf`. The `abs_cont_violations_*` and `mass_violations_*` columns quantify how much probability mass falls on violating states in each direction.

Usage

```
per_marker_kl(model, poi = NULL)
```

Arguments

`model`	A `marker_model` object (see `marker_model()`).
`poi`	Optional character scalar naming the person of interest in the pedigree. When `NULL` (default), the engine resolves the POI by picking the last untyped non-founder, falling back to the last non-founder, and finally to the last member.

Value

A one-row `data.frame` with columns

marker Marker identifier (`character`).

e_log10_lr_h1 Expected `log10(LR)` under H1.

e_log10_lr_h2 Expected `log10(LR)` under H2.

kl_h1_to_h2 KL(H1 ‖ H2) in nats.

kl_h2_to_h1 KL(H2 ‖ H1) in nats.

abs_cont_violations_h1 Number of joint states with positive H2 mass but zero H1 mass (integer).

abs_cont_violations_h2 Number of joint states with positive H1 mass but zero H2 mass (integer).

mass_violations_h1 Total H2 mass on states violating H1 absolute continuity (numeric).

mass_violations_h2 Total H1 mass on states violating H2 absolute continuity (numeric).

Attribute `"poi"` records the resolved POI identifier.

See Also

`marker_model()`, `per_marker_kl_profile()`.

Examples

```
if (requireNamespace("pedtools", quietly = TRUE)) {
  ped <- pedtools::nuclearPed(1)
  freqs <- c("a" = 0.3, "b" = 0.5, "c" = 0.2)
  mm <- marker_model(ped, "M1", freqs,
                     mutation = list(model = "equal", rate = 0.005))
  per_marker_kl(mm)
}
```

---

per_marker_kl_profile *Per-marker bidirectional KL across a marker profile*

Description

Vectorised wrapper around `per_marker_kl()` for a list of `marker_model` objects. Returns a `data.frame` with one row per input model in input order. When the profile shares one pedigree topology, uses no linkage, and every mutation model is wired to the C++ backend (`none` / `equal` / `stepwise`), evaluation routes through a single cross-marker C++ batch call with a shared mutation-matrix cache; otherwise it falls back to a per-marker R-level loop.

All models must share the same pedigree topology if `poi` is supplied as a scalar; otherwise the POI is resolved independently for each model.

Usage

```
per_marker_kl_profile(models, poi = NULL)
```

Arguments

models	A list of marker_model objects. Names of the list, when present and non-empty, override the per-model marker_id in the output marker column.
poi	Optional character scalar applied to every model, or NULL (default) for per-model resolution. Pass a vector by mapping the loop yourself if you need heterogeneous POIs.

Value

A data.frame with the same columns as per_marker_kl() and nrow(out) == length(models). When poi is a scalar, attribute "poi" carries that value; otherwise the resolved POIs appear as attribute "poi" of length length(models).

See Also

per_marker_kl().

Examples

```
if (requireNamespace("pedtools", quietly = TRUE)) {
  ped <- pedtools::nuclearPed(1)
  models <- list(
    M1 = marker_model(ped, "M1", c("a" = 0.4, "b" = 0.6),
                      mutation = list(model = "equal", rate = 1e-3)),
    M2 = marker_model(ped, "M2", c("a" = 0.2, "b" = 0.3, "c" = 0.5),
                      mutation = list(model = "equal", rate = 1e-3))
  )
  per_marker_kl_profile(models)
}
```

---

plot.lr_dist *Plot a Likelihood-Ratio Distribution*

Description

Draws the discrete $\log_{10}\mathrm{LR}$ distribution under both hypotheses as an overlaid lollipop plot: H_1 (related, blue) and H_2 (unrelated, red). Less overlap means stronger discrimination.

Usage

```
## S3 method for class 'lr_dist'
plot(x, ...)
```

Arguments

x	An lr_dist object (see as_lr_dist()).
...	Unused.

Details

`+Inf` / `-Inf` atoms cannot be placed on a finite axis and are dropped from the plot with a message; their mass is still reported by `summary.lr_dist()`. The x-axis is $\log_{10}$ LR (`0` = neutral, `> 0` favours H_1).

Value

A `ggplot2` object.

See Also

`summary.lr_dist()`, `as_lr_dist()`

Examples

```
d <- as_lr_dist(data.frame(
  log10_lr = c(-1, 0, 2),
  p_h1 = c(0.1, 0.3, 0.6),
  p_h2 = c(0.6, 0.3, 0.1)
))
plot(d)
```

---

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)
```

print.marker_model *Print method for marker_model*

Description

Print method for marker_model

Usage

```
## S3 method for class 'marker_model'
print(x, ...)
```

Arguments

x	A marker_model object.
...	Unused.

Value

Invisibly returns x.

print.nongenetic_feature *Print method for nongenetic_feature*

Description

Print method for nongenetic_feature

Usage

```
## S3 method for class 'nongenetic_feature'
print(x, ...)
```

Arguments

x	A nongenetic_feature object.
...	Unused.

Value

Invisibly returns x.

quantile.lr_dist *Quantiles of a likelihood-ratio distribution*

Description

Discrete inverse-CDF quantiles of $\log_{10}\mathrm{LR}$ under one of the two hypotheses, using R's quantile type-1 definition $Q(p) = \inf\{x : F(x) \geq p\}$ over the active support.

Usage

```
## S3 method for class 'lr_dist'
quantile(x, probs = c(0, 0.25, 0.5, 0.75, 1), under_h1 = TRUE, ...)
```

Arguments

x	An lr_dist object (see as_lr_dist() / lr_distribution()).
probs	Numeric vector of probabilities in $[0, 1]$. Defaults to the quartiles c(0, 0.25, 0.5, 0.75, 1).
under_h1	Logical. If TRUE (default) quantiles are taken under H_1 (weights p_h1); if FALSE, under H_2 (weights p_h2).
...	Unused.

Details

Atoms with zero mass under the requested hypothesis are dropped before the cumulative distribution is formed. A +Inf (or -Inf) atom is a legitimate quantile value when the requested probability falls in its mass. An error is raised if no atom has positive mass under the requested hypothesis.

Value

A named numeric vector of quantiles, one per probs entry.

See Also

lr_distribution(), summary.lr_dist(), plot.lr_dist()

Examples

```
d <- as_lr_dist(data.frame(
  log10_lr = c(-1, 0, 2),
  p_h1 = c(0.1, 0.3, 0.6),
  p_h2 = c(0.6, 0.3, 0.1)
))
quantile(d)
quantile(d, probs = c(0.05, 0.95), under_h1 = FALSE)
```

shannon_concentration *Shannon-entropy-based concentration of evidence contributions*

Description

Computes a concentration measure based on the Shannon entropy of the normalized per-step contributions. Robustness alternative to concentration_index and herfindahl_index.

Usage

```
shannon_concentration(weights)
```

Arguments

weights	Numeric vector of per-step contributions.

Details

Defined as

$$C_S(w) = 1 - \frac{H(p)}{\log T}, \qquad p_k = \frac{|w_k|}{\sum_j |w_j|},$$

where $H(p) = -\sum_k p_k \log p_k$ is the natural-log Shannon entropy of the normalized contributions. This is the complement of the normalized Shannon entropy: uniform weights give $C_S = 0$, single-step dominance gives $C_S \to 1$.

Value

Numeric scalar in $[0, 1]$. Returns 0 for uniform weights and approaches 1 for single-step dominance. Returns 0 if all weights are zero.

See Also

concentration_index, herfindahl_index.

Examples

```
shannon_concentration(rep(1, 10))           # 0
shannon_concentration(c(10, rep(0.01, 9)))  # ~0.95
```

`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_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_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)
```

`summary.lr_dist` *Summarise a Likelihood-Ratio Distribution*

Description

Computes the decision-theoretic summary of an `lr_dist` object: the mean, variance and standard deviation of $\log_{10}$ LR under each hypothesis, the median and quartiles, the area under the ROC curve, and the total probability mass (a sanity check on the input joint).

Usage

```
## S3 method for class 'lr_dist'
summary(object, ...)
```

Arguments

object	An lr_dist object (see as_lr_dist()).
...	Unused.

Details

mean_h1 is $E[\log_{10} \mathrm{LR} \mid H_1]$ and equals the per-marker KL-derived expectation. A +Inf atom drives the H1 moments to +Inf (and a -Inf atom the H2 moments to -Inf), matching the engine's 0\log 0 = 0 convention. auc is the exact concordance statistic $P(\mathrm{LR}_{H_1} > \mathrm{LR}_{H_2}) + \frac{1}{2}P(\mathrm{LR}_{H_1} = \mathrm{LR}_{H_2})$.

Value

A list of class summary.lr_dist with components mean_h1, mean_h2, var_h1, var_h2, sd_h1, sd_h2, mass_h1, mass_h2, auc, quantiles_h1, quantiles_h2, has_pos_inf and has_neg_inf. Printed in a compact table.

See Also

plot.lr_dist(), as_lr_dist()

Examples

```
d <- as_lr_dist(data.frame(
  log10_lr = c(-1, 0, 2),
  p_h1 = c(0.1, 0.3, 0.6),
  p_h2 = c(0.6, 0.3, 0.1)
))
s <- summary(d)
s$mean_h1
s$auc
```

---

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)
```

---

trajectory_metrics *Trajectory metrics from a belief trajectory matrix*

Description

Computes all trajectory-level metrics for a belief trajectory: entropy at each step, per-step Kullback-Leibler divergence (Bayesian surprise), cumulative divergence from the prior, per-step total-variation distance, total path length, and a family of concentration indices (max/sum, Herfindahl, Shannon-based).

Usage

```
trajectory_metrics(traj_matrix)
```

Arguments

traj_matrix	Matrix. Output from belief_trajectory with rows indexing time steps (row 1 = prior, row $T + 1$ = final posterior) and columns indexing hypotheses.

Value

A list with components:

- entropy Numeric vector of length $T + 1$: Shannon entropy at each step.
- kl_step Numeric vector of length T: per-step Kullback-Leibler divergence $D_{\mathrm{KL}}(P_t \| P_{t-1})$, equivalently Bayesian surprise (Itti & Baldi, 2009).
- kl_from_prior Numeric vector of length $T + 1$: cumulative Kullback-Leibler divergence from the prior, $D_{\mathrm{KL}}(P_t \| P_0)$.
- tv_step Numeric vector of length T: per-step total-variation distance $\mathrm{TV}(P_t, P_{t-1})$.
- path_length Numeric scalar. Total path length in total-variation distance, $\sum_t \mathrm{TV}(P_t, P_{t-1})$.

`concentration` Numeric scalar. Max/sum concentration index based on per-step information gains (see `concentration_index`).

`concentration_herfindahl` Numeric scalar. Herfindahl concentration (see `herfindahl_index`).

`concentration_shannon` Numeric scalar. Shannon-based concentration (see `shannon_concentration`).

References

Marsico, F. L. & Egeland, T. (in preparation). Belief dynamics during the investigative process. Itti, L. & Baldi, P. (2009). Bayesian surprise attracts human attention. *Vision Research* 49, 1295-1306.

See Also

`belief_trajectory`, `entropy_log10`, `kl_divergence_log10`, `concentration_index`.

Examples

```
prior <- c(0.5, 0.5)
lrs <- list(c(3, 1), c(1, 2), c(2, 1))
traj <- belief_trajectory(prior, lrs)
metrics <- trajectory_metrics(traj)
str(metrics)
```

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

∗ **datasets**
  Argentina, 5
  Asia, 8
  Austria, 9
  BosniaHerz, 12
  China, 15
  Europe, 29
  Japan, 35
  USA, 85
∗ **package**
  mispitools-package, 3

app_lr_comparison, *4*
app_mispitools, *4*
Argentina, 5, *5*, *8*, *29*, *33, 34*, *86*
as_lr_dist, 7
as_lr_dist(), *46, 47*, *63, 64*, *70*, *82*
Asia, *5, 6*, 8, *15*, *29*, *33*, *36*, *86*
Austria, *5, 6*, *8*, 9, *12*, *29*, *33*, *86*

belief_trajectory, 10, *11*, *84, 85*
binary_belief_trajectory, *10*, 11, *31*
BosniaHerz, *5, 6*, *8, 9*, 12, *29*, *33*, *86*

calibrate_concentration_cutoff, 13, *32, 33*
China, *5, 6*, *8*, 15, *29*, *33*, *36*, *86*
compute_conditioned_prop, 16, *18*, *45*, *81*
compute_reference_prop, 17, *17*, *45*, *81*
concentration_index, *11*, 18, *19, 20*, *34, 35*, *38*, *71*, *85*
concentration_index_positive, *14*, *18*, 19, *19*, *31*, *33*, *38*
cpt_missing_person, *4*, 20, *23*, *28*, *64, 65*
cpt_population, *4*, *21*, 22, *64, 65*

decision_threshold, *4*, 24, *58*, *67, 68*, *73*, *83*

entropy_log10, 26, *37*, *85*
error_matrix_hair, *4*, *21*, 26, *48, 49*
Europe, *5, 6*, *8, 9*, *12*, 29, *33, 34*, *86*

familias_trajectory, 30
fragility_report, *14*, 32

get_allele_freqs, *4*, *6*, *8, 9*, *12*, *15*, *29*, 33, *36*, *86*

herfindahl_index, *18, 19*, 34, *71*, *85*

Japan, *5, 6*, *8*, *15*, *29*, *33*, 35, *86*

kl_divergence_log10, *26*, 36, *85*

leave_one_out, *19, 20*, *31*, *33*, 37
lr_age, *4*, 38, *52*, *54*, *58*
lr_age(), *60*
lr_birthdate, *4*, 40, *58*
lr_birthdate(), *60*
lr_combine, *4*, *24*, 43, *55*, *58*, *66, 67*, *75*, *83*
lr_compute_pigmentation, *17, 18*, 44, *49, 50*
lr_distribution, 46
lr_distribution(), *70*
lr_hair_color, *4*, *28*, *40*, 47, *52*, *54*, *58*
lr_hair_color(), *60*
lr_pigmentation, *4*, *45*, 49, *49*, *81*
lr_pigmentation(), *60*
lr_sensitivity, 51
lr_sex, *4*, *40*, 52, *52*, *58*
lr_sex(), *60*
lr_to_dataframe, *4*, *24*, *43, 44*, 54, *66–68*, *73*, *83*

marker_model, 56
marker_model(), *46, 47*, *58*, *60–62*
mispitools *(*mispitools-package*)*, 3
mispitools-package, 3
mispitools_app, 57

nongenetic_feature, *21*, *23*, *40*, *42*, *48–50*, *54*, 58

per_marker_kl, 61
per_marker_kl(), *46*, *62, 63*
per_marker_kl_profile, 62
per_marker_kl_profile(), *47*, *62*
plot.lr_dist, 63
plot.lr_dist(), *7*, *46, 47*, *70*, *82*
plot_cpt, *4*, *21*, *23*, 64
plot_decision_curve, *4*, *25*, 66, *68*, *83*
plot_lr_distribution, *4*, *25*, *44*, *50*, *55*, 67, *67*, *73*
print.marker_model, 69
print.nongenetic_feature, 69

quantile.lr_dist, 70
quantile.lr_dist(), *46, 47*

shannon_concentration, *18, 19*, *35*, 71, *85*
sim_lr_genetic, *3*, *6*, *8, 9*, *12–15*, *24*, *29*, *34*, *36*, *43, 44*, *54, 55*, *66–68*, 72, *75*, *83*, *86*
sim_lr_prelim, *3*, *24*, *40*, *42*, *44*, *49*, *54*, *66–68*, *73*, 74, *77*, *79*, *83*
sim_mp_prelim, *3*, 76, *79*
sim_poi_prelim, *3*, *41, 42*, *75–77*, 78
sim_reference_pop, *3*, *16–18*, *50*, 80
summary.lr_dist, 81
summary.lr_dist(), *7*, *46, 47*, *64*, *70*

threshold_rates, *4*, *25*, *58*, *67*, 82
trajectory_metrics, *10*, *26*, *31*, *37*, 84

USA, *5, 6*, *8*, *29*, *33, 34*, 85