Package 'penetrance'

September 8, 2026

Type Package

Title Methods for Penetrance Estimation in Family-Based Studies

Version 0.1.4

Depends R (>= 3.5.0)

Imports clipp, stats, parallel, MASS, kinship2 (>= 1.8.5), graphics, grDevices

Description Implements statistical methods for estimating disease penetrance in family-based studies. Penetrance refers to the probability of disease manifestation in individuals carrying specific genetic variants. The package provides tools for age-specific penetrance estimation, handling missing data, and accounting for ascertainment bias in family studies.
Cite as: Kubista, N., Braun, D. & Parmigiani, G. (2025) <doi:10.1093/bioadv/vbaf154>.

License GPL-3

URL https://github.com/bayesmendel/penetrance

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.3

VignetteBuilder knitr

Suggests knitr, rmarkdown, scales, ggplot2, roxygen2, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Sol Rosito [cre],
Nicolas Kubista [aut],
BayesMendel Lab [aut],
Giovanni Parmigiani [aut],
Danielle Braun [aut],
Zaid Al-Ississ [aut],
Alice Zhang [aut]

Maintainer Sol Rosito <bmendel@jimmy.harvard.edu>

Repository CRAN

Date/Publication 2026-09-08 19:00:02 UTC

Contents

absValue . . . 3
apply_burn_in . . . 3
apply_thinning . . . 4
baseline_data_default . . . 4
calculateBaseline . . . 5
calculateEmpiricalDensity . . . 5
calculateNCPen . . . 6
calculate_weibull_parameters . . . 7
combine_chains . . . 8
combine_chains_noSex . . . 9
distribution_data_default . . . 9
drawBaseline . . . 10
drawEmpirical . . . 10
generate_density_plots . . . 11
generate_summary . . . 11
generate_summary_noSex . . . 12
imputeAges . . . 12
imputeAgesInit . . . 15
imputeUnaffectedAges . . . 16
lik.fn . . . 17
lik_noSex . . . 18
makePriors . . . 19
mhChain . . . 20
mhLogLikelihood_clipp . . . 23
mhLogLikelihood_clipp_noSex . . . 25
out_sim . . . 26
penetrance . . . 27
plot_acf . . . 32
plot_loglikelihood . . . 33
plot_pdf . . . 33
plot_penetrance . . . 34
plot_trace . . . 34
printRejectionRates . . . 35
prior_params_default . . . 36
risk_proportion_default . . . 37
simulated_families . . . 37
test_fam2 . . . 38
transformDF . . . 39
validate_allele_freq . . . 40
validate_baseline_data . . . 42
validate_weibull_parameters . . . 43

Index 45

`absValue` *Function to return absolute values*

Description

Function to return absolute values

Usage

```
absValue(x)
```

Arguments

`x`	Numeric, the input value.

Value

Numeric, the absolute value of the input.

`apply_burn_in` *Apply Burn-In*

Description

Apply Burn-In

Usage

```
apply_burn_in(results, burn_in)
```

Arguments

`results`	A list of MCMC chain results.
`burn_in`	The fraction or proportion of results to discard as burn-in (0 to 1). The default is no burn-in, burn_in=0.

Value

A list of results with burn-in applied.

apply_thinning *Apply Thinning*

Description

Apply Thinning

Usage

```
apply_thinning(results, thinning_factor)
```

Arguments

results	A list of MCMC chain results.
thinning_factor	The factor by which to thin the results (positive integer). The default thinning factor is 1, which implies no thinning.

Value

A list of results with thinning applied.

baseline_data_default *Default Baseline Data*

Description

This dataset contains age-specific cancer penetrance rates for both females and males. As an example, the data is derived from the SEER program for colorectal cancer females and males.

Usage

```
baseline_data_default

baseline_data_default
```

Format

A data frame with 94 rows and 3 variables:

Age Age in years

Female Female baseline risk

Male Male baseline risk

A data frame with 94 rows and 3 variables:

Age Age in years (1 to 94)

Female Penetrance rate for females

Male Penetrance rate for males

`calculateBaseline` *Calculate Baseline Risk*

Description

This function extracts the penetrance data for a specified cancer type, gene, race, and penetrance type from the provided database.

Usage

```
calculateBaseline(cancer_type, gene, race, type, db)
```

Arguments

`cancer_type`	The type of cancer for which the risk is being calculated.
`gene`	The gene of interest for which the risk is being calculated.
`race`	The race of the individual.
`type`	The type of penetrance calculation.
`db`	The dataset used for the calculation, containing penetrance data.

Value

A matrix of penetrance data for the specified parameters.

`calculateEmpiricalDensity`
Calculate Empirical Age Density

Description

Calculates empirical age density distributions for different subgroups in the data, separated by sex and genetic testing status.

Usage

```
calculateEmpiricalDensity(
  data,
  aff_column = "aff",
  age_column = "age",
  sex_column = "sex",
  geno_column = "geno",
  n_points = 10000,
  sex_specific = TRUE
)
```

Arguments

data	A data frame containing the family data
aff_column	Name of the affection status column
age_column	Name of the age column
sex_column	Name of the sex column
geno_column	Name of the genotype column
n_points	Number of points to use in density estimation
sex_specific	Logical; whether to calculate sex-specific densities

Value

A list of density objects for different subgroups (tested/untested, male/female)

Examples

```
# Create sample data
data <- data.frame(
  sex = c(1, 1, 2, 2),
  aff = c(0, 0, 0, 0),
  age = c(45, 50, 35, 40),
  geno = c("1/2", NA, "1/2", NA)
)

# Calculate density with sex-specific parameters
density_sex <- calculateEmpiricalDensity(data, sex_specific = TRUE)

# Calculate density without sex-specific parameters
density_nosex <- calculateEmpiricalDensity(data, sex_specific = FALSE)
```

---

calculateNCPen *Calculate Age-Specific Non-Carrier Penetrance*

Description

This function calculates the age-specific non-carrier penetrance based on SEER baseline data, penetrances for carriers, and allele frequencies. It adjusts penetrance estimates for genetic testing by incorporating the genetic risk attributable to specified alleles.

Usage

```
calculateNCPen(SEER_baseline, alpha, beta, delta, gamma, prev, max_age)
```

Arguments

SEER_baseline	Numeric, the baseline penetrance derived from SEER data for the general population without considering genetic risk factors.
alpha	Numeric, shape parameter for the Weibull distribution used to model carrier risk.
beta	Numeric, scale parameter for the Weibull distribution used to model carrier risk.
delta	Numeric, location parameter for the Weibull distribution used to model carrier risk.
gamma	Numeric, scaling factor applied to the Weibull distribution to adjust carrier risk.
prev	Numeric, the carrier prevalence (heterozygote frequency) in the population. This should be approximately 2p where p is the allele frequency when the allele is rare.
max_age	Integer, the maximum age up to which the calculations are performed.

Value

A list containing:

weightedCarrierRisk	Numeric vector, the weighted risk for carriers at each age based on prevalence.
yearlyProb	Numeric vector, the yearly probability of not getting the disease at each age.
cumulativeProb	Numeric vector, the cumulative probability of not getting the disease up to each age.

---

calculate_weibull_parameters

Calculate Weibull Parameters

Description

This function calculates the shape (alpha) and scale (beta) parameters of a Weibull distribution given the median, first quartile, and delta values.

Usage

```
calculate_weibull_parameters(given_median, given_first_quartile, delta)
```

Arguments

given_median	The median of the data.
given_first_quartile	The first quartile of the data.
delta	A constant offset value.

Value

A list containing the calculated Weibull parameters:

alpha	The shape parameter of the Weibull distribution
beta	The scale parameter of the Weibull distribution

Examples

```
# Calculate Weibull parameters
params <- calculate_weibull_parameters(
  given_median = 50,
  given_first_quartile = 30,
  delta = 15
)
print(params)
```

---

combine_chains *Combine Chains Function to combine the posterior samples from the multiple chains.*

Description

Combine Chains Function to combine the posterior samples from the multiple chains.

Usage

```
combine_chains(results)
```

Arguments

results	A list of MCMC chain results.

Value

A list with combined results, including median, threshold, first quartile, and asymptote values.

combine_chains_noSex *Combine Chains for Non-Sex-Specific Estimation*

Description

Combines the posterior samples from multiple MCMC chains for non-sex-specific estimations.

Usage

```
combine_chains_noSex(results)
```

Arguments

results	A list of MCMC chain results, where each element contains posterior samples of parameters.

Value

A list with combined results, including samples for median, threshold, first quartile, asymptote values, log-likelihoods, and log-acceptance ratios.

distribution_data_default *Default Distribution Data*

Description

Default data frame structure with row names for use in the `makePriors` function.

Usage

```
distribution_data_default

distribution_data_default
```

Format

A data frame for prior distribution parameters

A data frame with the following columns:

age Age values (NA for default).

at_risk Proportion of people at risk (NA for default).

drawBaseline *Draw Ages Using the Inverse CDF Method from the baseline data*

Description

This function draws ages using the inverse CDF method from baseline data.

Usage

```
drawBaseline(baseline_data)
```

Arguments

baseline_data	A data frame containing baseline data with columns 'cum_prob' and 'age'.

Value

A single age value drawn from the baseline data.

drawEmpirical *Draw Ages Using the Inverse CDF Method from Empirical Density*

Description

This function draws ages using the inverse CDF method from empirical density data, based on sex and whether the individual was tested.

Usage

```
drawEmpirical(empirical_density, sex, tested, sex_specific = TRUE)
```

Arguments

empirical_density	A list of density objects containing the empirical density of ages for different groups.
sex	Numeric, the sex of the individual (1 for male, 2 for female).
tested	Logical, indicating whether the individual was tested (has a non-NA 'geno' value).
sex_specific	Logical, indicating whether the imputation should be sex-specific. Default is TRUE.

Value

A single age value drawn from the appropriate empirical density data.

generate_density_plots *Generate Posterior Density Plots*

Description

Generates histograms of the posterior samples for the different parameters

Usage

```
generate_density_plots(data)
```

Arguments

data	A list with combined results.

Value

No return value, called for side effects. Creates density plots for each parameter.

Examples

```
# Create example data
data <- list(
  median_male_results = rnorm(1000, 50, 5),
  median_female_results = rnorm(1000, 45, 5),
  threshold_male_results = runif(1000, 20, 30),
  threshold_female_results = runif(1000, 25, 35),
  asymptote_male_results = rbeta(1000, 2, 2),
  asymptote_female_results = rbeta(1000, 2, 2)
)

# Generate density plots
old_par <- par(no.readonly = TRUE)  # Save old par settings
generate_density_plots(data)
par(old_par)  # Restore old par settings
```

generate_summary *Generate Summary*

Description

Function to generate summary statistics

Usage

```
generate_summary(data, verbose = FALSE)
```

Arguments

data	A list with combined results.
verbose	Logical, whether to print summary to console. Default is FALSE.

Value

A data.frame containing summary statistics (min, 1st quartile, median, mean, 3rd quartile, max) for each parameter.

---

generate_summary_noSex *Generate Summary for Non-Sex-Specific Estimation*

Description

Generates summary statistics for the combined MCMC results for non-sex-specific estimations.

Usage

```
generate_summary_noSex(data, verbose = FALSE)
```

Arguments

data	A list containing combined results of MCMC chains, typically the output of `combine_chains_noSex`.
verbose	Logical, whether to print summary to console. Default is FALSE.

Value

A data.frame containing summary statistics (min, 1st quartile, median, mean, 3rd quartile, max) for median, threshold, first quartile, and asymptote values.

---

imputeAges *Impute Missing Ages in Family-Based Data*

Description

Imputes missing ages in family-based data using a combination of Weibull distributions for affected individuals and empirical distributions for unaffected individuals. The function can perform both sex-specific and non-sex-specific imputations.

Usage

```
imputeAges(
  data,
  na_indices,
  baseline_male = NULL,
  baseline_female = NULL,
  alpha_male = NULL,
  beta_male = NULL,
  delta_male = NULL,
  alpha_female = NULL,
  beta_female = NULL,
  delta_female = NULL,
  baseline = NULL,
  alpha = NULL,
  beta = NULL,
  delta = NULL,
  max_age,
  sex_specific = TRUE,
  max_attempts = 100,
  geno_freq,
  trans,
  lik
)
```

Arguments

`data`	A data frame containing family-based data with columns: family, individual, father, mother, sex, aff, age, geno, and isProband
`na_indices`	Vector of indices where ages need to be imputed
`baseline_male`, `baseline_female`	Data frames containing baseline age distributions for males/females
`alpha_male`, `alpha_female`	Shape parameters for male/female Weibull distributions
`beta_male`, `beta_female`	Scale parameters for male/female Weibull distributions
`delta_male`, `delta_female`	Location parameters for male/female Weibull distributions
`baseline`	Data frame containing overall baseline age distribution (non-sex-specific)
`alpha`, `beta`, `delta`	Overall Weibull parameters (non-sex-specific)
`max_age`	Maximum allowable age
`sex_specific`	Logical; whether to use sex-specific parameters
`max_attempts`	Maximum number of attempts for generating valid ages
`geno_freq`	Vector of genotype frequencies
`trans`	Transmission probabilities
`lik`	Likelihood matrix

Value

A data frame with the following modifications:

age	Updated with imputed ages for previously missing values

The rest of the data frame remains unchanged.

Examples

```
# Create sample data with the same structure as used in mhChain
data <- data.frame(
  family = rep(1:2, each=5),
  individual = rep(1:5, 2),
  father = c(NA,1,1,1,1, NA,6,6,6,6),
  mother = c(NA,2,2,2,2, NA,7,7,7,7),
  sex = c(1,2,1,2,1, 1,2,1,2,1),
  aff = c(1,0,1,0,NA, 1,0,1,0,NA),
  age = c(45,NA,25,NA,20, 50,NA,30,NA,22),
  geno = c("1/2",NA,"1/2",NA,NA, "1/2",NA,"1/2",NA,NA),
  isProband = c(1,0,0,0,0, 1,0,0,0,0)
)

# Initialize parameters
na_indices <- which(is.na(data$age))
geno_freq <- c(0.999, 0.001)  # Frequency of normal and risk alleles
trans <- matrix(c(1,0,0.5,0.5), nrow=2)  # Transmission matrix
lik <- matrix(1, nrow=nrow(data), ncol=2)  # Likelihood matrix

# Create baseline data for both sex-specific and non-sex-specific cases
age_range <- 20:94
n_ages <- length(age_range)

# Sex-specific baseline data
baseline_male <- data.frame(
  age = age_range,
  cum_prob = (1:n_ages)/n_ages * 0.8  # Male cumulative probabilities
)

baseline_female <- data.frame(
  age = age_range,
  cum_prob = (1:n_ages)/n_ages * 0.9  # Female cumulative probabilities
)

# Non-sex-specific baseline data
baseline <- data.frame(
  age = age_range,
  cum_prob = (1:n_ages)/n_ages * 0.85  # Overall cumulative probabilities
)

# Example with sex-specific imputation
imputed_data_sex <- imputeAges(
  data = data,
  na_indices = na_indices,
```

```
  baseline_male = baseline_male,
  baseline_female = baseline_female,
  alpha_male = 3.5,
  beta_male = 20,
  delta_male = 20,
  alpha_female = 3.2,
  beta_female = 18,
  delta_female = 18,
  max_age = 94,
  sex_specific = TRUE,
  geno_freq = geno_freq,
  trans = trans,
  lik = lik
)

# Example with non-sex-specific imputation
imputed_data_nosex <- imputeAges(
  data = data,
  na_indices = na_indices,
  baseline = baseline,
  alpha = 3.3,
  beta = 19,
  delta = 19,
  max_age = 94,
  sex_specific = FALSE,
  geno_freq = geno_freq,
  trans = trans,
  lik = lik
)
```

`imputeAgesInit` *Initialize Age Imputation*

Description

Initializes the age imputation process by filling missing ages with random values between a threshold and maximum age.

Usage

```
imputeAgesInit(data, threshold, max_age)
```

Arguments

`data`	A data frame containing family-based data
`threshold`	Minimum age value for initialization
`max_age`	Maximum age value for initialization

Value

A list containing:

data	The data frame with initialized ages
na_indices	Indices of missing age values

Examples

```
# Create sample data
data <- data.frame(
  family = c(1, 1),
  individual = c(1, 2),
  father = c(NA, 1),
  mother = c(NA, NA),
  sex = c(1, 2),
  aff = c(1, 0),
  age = c(NA, NA),
  geno = c("1/2", NA),
  isProband = c(1, 0)
)

# Initialize ages with random values between 20 and 94
result <- imputeAgesInit(data, threshold = 20, max_age = 94)

# Access the results
imputed_data <- result$data
missing_indices <- result$na_indices
```

---

imputeUnaffectedAges *Impute Ages for Unaffected Individuals*

Description

This function imputes ages for unaffected individuals in a dataset based on their sex and whether they were tested, using empirical age distributions.

Usage

```
imputeUnaffectedAges(data, na_indices, empirical_density, max_age)
```

Arguments

data	A data frame containing the individual data, including columns for age, sex, and geno.
na_indices	A vector of indices indicating the rows in the data where ages need to be imputed.
empirical_density	A list of density objects containing the empirical density of ages for different groups.
max_age	Integer, the maximum age considered in the analysis.

Value

The data frame with imputed ages for unaffected individuals.

`lik.fn` *Penetrance Function*

Description

Calculates the penetrance for an individual based on Weibull distribution parameters. This function estimates the probability of developing cancer given the individual's genetic and demographic information.

Usage

```
lik.fn(
  i,
  data,
  alpha_male,
  alpha_female,
  beta_male,
  beta_female,
  delta_male,
  delta_female,
  gamma_male,
  gamma_female,
  max_age,
  baselineRisk,
  BaselineNC,
  prev
)
```

Arguments

`i`	Integer, index of the individual in the data set.
`data`	Data frame, containing individual demographic and genetic information. Must include columns for 'sex', 'age', 'aff' (affection status), and 'geno' (genotype).
`alpha_male`	Numeric, Weibull distribution shape parameter for males.
`alpha_female`	Numeric, Weibull distribution shape parameter for females.
`beta_male`	Numeric, Weibull distribution scale parameter for males.
`beta_female`	Numeric, Weibull distribution scale parameter for females.
`delta_male`	Numeric, shift parameter for the Weibull function for males.
`delta_female`	Numeric, shift parameter for the Weibull function for females.
`gamma_male`	Numeric, asymptote parameter for males (only scales the entire distribution).
`gamma_female`	Numeric, asymptote parameter for females (only scales the entire distribution).

max_age	Integer, maximum age considered in the analysis.
baselineRisk	Numeric matrix, baseline risk for each age by sex. Columns correspond to sex (1 for male, 2 for female) and rows to age.
BaselineNC	Logical, indicates if non-carrier penetrance should be based on SEER data.
prev	Numeric, the carrier prevalence (heterozygote frequency) in the population. This should be approximately 2p where p is the allele frequency when the allele is rare.

Value

Numeric vector, containing penetrance values for unaffected and affected individuals.

---

lik_noSex *Likelihood Calculation without Sex Differentiation*

Description

This function calculates the likelihood for an individual based on Weibull distribution parameters without considering sex differentiation.

Usage

```
lik_noSex(
  i,
  data,
  alpha,
  beta,
  delta,
  gamma,
  max_age,
  baselineRisk,
  BaselineNC,
  prev
)
```

Arguments

i	Integer, index of the individual in the data set.
data	Data frame, containing individual demographic and genetic information. Must include columns for 'age', 'aff' (affection status), and 'geno' (genotype).
alpha	Numeric, Weibull distribution shape parameter.
beta	Numeric, Weibull distribution scale parameter.
delta	Numeric, shift parameter for the Weibull function.
gamma	Numeric, asymptote parameter (only scales the entire distribution).
max_age	Integer, maximum age considered in the analysis.

baselineRisk	Numeric vector, baseline risk for each age.
BaselineNC	Logical, indicates if non-carrier penetrance should be based on SEER data or the calculated non-carrier penetrance.
prev	Numeric, the carrier prevalence (heterozygote frequency) in the population. This should be approximately 2p where p is the allele frequency when the allele is rare.

Value

Numeric vector, containing likelihood values for unaffected and affected individuals.

`makePriors` *Make Priors*

Description

This function generates prior distributions based on user input or default parameters. It is designed to aid in the statistical analysis of risk proportions in populations, particularly in the context of cancer research. The distributions are calculated for various statistical metrics such as asymptote, threshold, median, and first quartile.

Usage

```
makePriors(
  data,
  sample_size,
  ratio,
  prior_params,
  risk_proportion,
  baseline_data
)
```

Arguments

data	A data frame containing age and risk data. If NULL or entirely NA, default parameters are used. If provided, all entries in the age column must be numeric and non-NA.
sample_size	Numeric, the total sample size used for risk proportion calculations.
ratio	Numeric, the odds ratio (OR) or relative risk (RR) used in asymptote parameter calculations.
prior_params	List, containing prior parameters for the beta distributions. If NULL, default parameters are used.
risk_proportion	Data frame, with default proportions of people at risk.
baseline_data	Data frame with the baseline risk data.

Details

The function includes internal helper functions for normalizing median and first quartile values, and for computing beta distribution parameters. The function handles various settings: using default parameters, applying user inputs, and calculating parameters based on sample size and risk proportions.

If the OR/RR ratio is provided, the asymptote parameters are computed based on this ratio, overriding other inputs for the asymptote.

The function returns a list of distribution functions for the asymptote, threshold, median, and first quartile, which can be used for further statistical analysis.

Value

A list of functions representing the prior distributions for asymptote, threshold, median, and first quartile.

See Also

qbeta, runif

`mhChain` *Execution of a Single Chain in Metropolis-Hastings for Cancer Risk Estimation*

Description

Performs a single chain execution in the Metropolis-Hastings algorithm for Bayesian inference, specifically tailored for cancer risk estimation. This function can handle both sex-specific and non-sex-specific scenarios.

Usage

```
mhChain(
  seed,
  n_iter,
  burn_in,
  chain_id,
  ncores,
  data,
  twins,
  max_age,
  baseline_data,
  prior_distributions,
  prev,
  median_max,
  BaselineNC,
  var,
```

```
  age_imputation,
  imp_interval,
  remove_proband,
  sex_specific
)
```

Arguments

seed	Integer, the seed for the random number generator to ensure reproducibility.
n_iter	Integer, the number of iterations to perform in the Metropolis-Hastings algorithm.
burn_in	Integer, the number of initial iterations to discard (burn-in period).
chain_id	Integer, the identifier for the chain being executed.
ncores	Integer, the number of cores to use for parallel computation.
data	Data frame, containing family and genetic information used in the analysis.
twins	Information on monozygous twins or triplets in the pedigrees.
max_age	Integer, the maximum age considered in the analysis.
baseline_data	Numeric matrix or vector, containing baseline risk estimates for different ages and sexes.
prior_distributions	List, containing prior distributions for the parameters being estimated.
prev	Numeric, the carrier prevalence (heterozygote frequency) in the population. Note: This is automatically calculated from allele frequency in the main penetrance() function as approximately 2p for rare variants.
median_max	Logical, indicates if the maximum median age should be used for the Weibull distribution.
BaselineNC	Logical, indicates if non-carrier penetrance should be based on SEER data.
var	Numeric, the variance for the proposal distribution in the Metropolis-Hastings algorithm.
age_imputation	Logical, indicates if age imputation should be performed.
imp_interval	Integer, the interval at which age imputation should be performed when age_imputation = TRUE.
remove_proband	Logical, indicates if the proband should be removed from the analysis.
sex_specific	Logical, indicates if the analysis should differentiate by sex.

Value

A list containing samples, log likelihoods, log-acceptance ratio, and rejection rate for each iteration.

Examples

```
# Create sample data in FamPRO format
data <- data.frame(
  ID = 1:10,
```

```
  PedigreeID = rep(1, 10),
  Sex = c(0, 1, 0, 1, 0, 1, 0, 1, 0, 1), # 0=female, 1=male
  MotherID = c(NA, NA, 1, 1, 3, 3, 5, 5, 7, 7),
  FatherID = c(NA, NA, 2, 2, 4, 4, 6, 6, 8, 8),
  isProband = c(1, rep(0, 9)),
  CurAge = c(45, 35, 55, 40, 50, 45, 60, 38, 52, 42),
  isAff = c(1, 0, 1, 0, 1, 0, 1, 0, 1, 0),
  Age = c(40, NA, 50, NA, 45, NA, 55, NA, 48, NA),
  Geno = c(1, NA, 1, 0, 1, 0, NA, NA, 1, NA)
)

# Transform data into required format
data <- transformDF(data)

# Set parameters for the chain
seed <- 123
n_iter <- 10
burn_in <- 0.1 # 10% burn-in
chain_id <- 1
ncores <- 1
max_age <- 100

# Create baseline data (simplified example)
baseline_data <- matrix(
  c(rep(0.005, max_age), rep(0.008, max_age)), # Increased baseline risks
  ncol = 2,
  dimnames = list(NULL, c("Male", "Female"))
)

# Set prior distributions with carefully chosen bounds
prior_distributions <- list(
  prior_params = list(
    asymptote = list(g1 = 2, g2 = 3), # Mode around 0.4
    threshold = list(min = 20, max = 30), # Narrower range for threshold
    median = list(m1 = 3, m2 = 2), # Mode around 0.6
    first_quartile = list(q1 = 2, q2 = 3) # Mode around 0.4
  )
)

# Create variance vector for all 8 parameters in sex-specific case
# Using very small variances for initial stability
var <- c(
  0.005, 0.005, # asymptotes (smaller variance since between 0-1)
  1, 1, # thresholds
  1, 1, # medians
  1, 1
) # first quartiles

# Run the chain
results <- mhChain(
  seed = seed,
  n_iter = n_iter,
  burn_in = burn_in,
```

```
  chain_id = chain_id,
  ncores = ncores,
  data = data,
  twins = NULL,
  max_age = max_age,
  baseline_data = baseline_data,
  prior_distributions = prior_distributions,
  prev = 0.05, # Increased prevalence
  median_max = FALSE, # Changed to FALSE for simpler median constraints
  BaselineNC = TRUE,
  var = var,
  age_imputation = FALSE,
  imp_interval = 10,
  remove_proband = TRUE,
  sex_specific = TRUE
)
```

`mhLogLikelihood_clipp` *Calculate Log Likelihood using clipp Package*

Description

Calculate Log Likelihood using clipp Package

Usage

```
mhLogLikelihood_clipp(
  paras,
  families,
  twins,
  max_age,
  baseline_data,
  prev,
  geno_freq,
  trans,
  BaselineNC,
  ncores
)
```

Arguments

`paras`	Numeric vector of parameters
`families`	Data frame of pedigree information
`twins`	Information on monozygous twins
`max_age`	Integer, maximum age
`baseline_data`	Numeric matrix of baseline risk data

prev	Numeric, the carrier prevalence (heterozygote frequency) in the population. This should be approximately 2p where p is the allele frequency when the allele is rare.
geno_freq	Numeric vector of frequencies
trans	Numeric matrix of transmission probabilities
BaselineNC	Logical for baseline choice
ncores	Integer for parallel computation

Value

Numeric value representing the calculated log likelihood.

Examples

```
# Create example parameters and data
paras <- c(0.8, 0.7, 20, 25, 50, 45, 30, 35)  # Example parameters

# Create sample data in Fam3PRO format
families <- data.frame(
  ID = 1:10,
  PedigreeID = rep(1, 10),
  Sex = c(0, 1, 0, 1, 0, 1, 0, 1, 0, 1),  # 0=female, 1=male
  MotherID = c(NA, NA, 1, 1, 3, 3, 5, 5, 7, 7),
  FatherID = c(NA, NA, 2, 2, 4, 4, 6, 6, 8, 8),
  isProband = c(1, rep(0, 9)),
  CurAge = c(45, 35, 55, 40, 50, 45, 60, 38, 52, 42),
  isAff = c(1, 0, 1, 0, 1, 0, 1, 0, 1, 0),
  Age = c(40, NA, 50, NA, 45, NA, 55, NA, 48, NA),
  Geno = c(1, NA, 1, 0, 1, 0, NA, NA, 1, NA)
)

# Transform data into required format
families <- transformDF(families)

trans <- matrix(
  c(
    1, 0, # both parents are wild type
    0.5, 0.5, # mother is wildtype and father is a heterozygous carrier
    0.5, 0.5, # father is wildtype and mother is a heterozygous carrier
    1 / 3, 2 / 3 # both parents are heterozygous carriers
  ),
 nrow = 4, ncol = 2, byrow = TRUE
)

# Calculate log likelihood
loglik <- mhLogLikelihood_clipp(
  paras = paras,
  families = families,
  twins = NULL,
  max_age = 94,
  baseline_data = baseline_data_default,
```

```
  prev = 0.001,
  geno_freq = c(0.999, 0.001),
  trans = trans,
  BaselineNC = TRUE,
  ncores = 1
)
```

`mhLogLikelihood_clipp_noSex` *Calculate Log Likelihood without Sex Differentiation*

Description

This function calculates the log likelihood for a set of parameters and data without considering sex differentiation using the clipp package.

Usage

```
mhLogLikelihood_clipp_noSex(
  paras,
  families,
  twins,
  max_age,
  baseline_data,
  prev,
  geno_freq,
  trans,
  BaselineNC,
  ncores
)
```

Arguments

`paras`	Numeric vector, the parameters for the Weibull distribution and scaling factors. Should contain in order: gamma, delta, given_median, given_first_quartile.
`families`	Data frame, containing pedigree information with columns for 'age', 'aff' (affection status), and 'geno' (genotype).
`twins`	Information on monozygous twins or triplets in the pedigrees.
`max_age`	Integer, maximum age considered in the analysis.
`baseline_data`	Numeric vector, baseline risk data for each age.
`prev`	Numeric, the carrier prevalence (heterozygote frequency) in the population. This should be approximately 2p where p is the allele frequency when the allele is rare.
`geno_freq`	Numeric vector, represents the frequency of the risk type and its complement in the population.

`trans`	Numeric matrix, transition matrix that defines the probabilities of allele transmission from parents to offspring.
`BaselineNC`	Logical, indicates if non-carrier penetrance should be based on the baseline data or the calculated non-carrier penetrance.
`ncores`	Integer, number of cores to use for parallel computation.

Value

Numeric, the calculated log likelihood.

References

Details about the clipp package and methods can be found in the package documentation.

`out_sim` *Simulated Output Data*

Description

This dataset contains the simulated output data for the penetrance package.

Usage

```
data(out_sim)
```

Format

A list with the following components:

summary_stats A data frame with 18000 observations of 8 variables:

- **Median_Male** numeric, Median value for males
- **Median_Female** numeric, Median value for females
- **Threshold_Male** numeric, Threshold value for males
- **Threshold_Female** numeric, Threshold value for females
- **First_Quartile_Male** numeric, First quartile value for males
- **First_Quartile_Female** numeric, First quartile value for females
- **Asymptote_Male** numeric, Asymptote value for males
- **Asymptote_Female** numeric, Asymptote value for females

density_plots A list of 1 element, mfrow: integer vector of length 2

trace_plots A list of 1 element, mfrow: integer vector of length 2

penetrance_plot A list of 2 elements: rect and text

pdf_plots A list of 2 elements: rect and text

combined_chains A list of 19 numeric vectors with 18000 elements each

results A list of 1 element which is a list of 24 elements, each with 18000 elements

data A data frame with 4727 observations of 9 variables:

individual integer, Individual ID

isProband numeric, Indicator if the individual is a proband

family integer, Family ID

mother numeric, Mother's ID

father numeric, Father's ID

aff numeric, Affected status

sex numeric, Sex of the individual

age numeric, Age of the individual

geno character, Genotype

Examples

```
data(out_sim)
head(out_sim$summary_stats)
```

`penetrance` *penetrance: A Package for Penetrance Estimation*

Description

A comprehensive package for penetrance estimation in family-based studies. This package implements Bayesian methods using Metropolis-Hastings algorithm for estimating age-specific penetrance of genetic variants. It supports both sex-specific and non-sex-specific analyses, and provides various visualization tools for examining MCMC results.

This function implements the Independent Metropolis-Hastings algorithm for Bayesian penetrance estimation of cancer risk. It utilizes parallel computing to run multiple chains and provides various options for analyzing and visualizing the results.

Usage

```
penetrance(
  pedigree,
  twins = NULL,
  n_chains = 1,
  n_iter_per_chain = 10000,
  ncores = 6,
  max_age = 94,
  baseline_data = baseline_data_default,
  remove_proband = FALSE,
  age_imputation = FALSE,
  median_max = TRUE,
  BaselineNC = TRUE,
  var = c(0.1, 0.1, 2, 2, 5, 5, 5, 5),
  burn_in = 0,
```

```
  thinning_factor = 1,
  imp_interval = 100,
  distribution_data = distribution_data_default,
  allele_freq = 1e-04,
  sample_size = NULL,
  ratio = NULL,
  prior_params = prior_params_default,
  risk_proportion = risk_proportion_default,
  summary_stats = TRUE,
  rejection_rates = TRUE,
  density_plots = TRUE,
  plot_trace = TRUE,
  penetrance_plot = TRUE,
  penetrance_plot_pdf = TRUE,
  plot_loglikelihood = TRUE,
  plot_acf = TRUE,
  probCI = 0.95,
  sex_specific = TRUE
)
```

Arguments

`pedigree` A list of data frames, where each data frame represents a single pedigree and contains the following columns:

- `PedigreeID`: A numeric or character identifier for the family/pedigree. Must be consistent for all members of the same family within a data frame.
- `ID`: A unique numeric or character identifier for each individual within their respective pedigree data frame.
- `Sex`: An integer representing biological sex: `0` for female, `1` for male. Use `NA` for unknown sex.
- `MotherID`: The `ID` of the individual's mother. Should correspond to an `ID` within the same pedigree data frame or be `NA` if the mother is not in the pedigree (founder).
- `FatherID`: The `ID` of the individual's father. Should correspond to an `ID` within the same pedigree data frame or be `NA` if the father is not in the pedigree (founder).
- `isProband`: An integer indicating if the individual is a proband: `1` for proband, `0` otherwise.
- `CurAge`: An integer representing the age of censoring. This is the current age if the individual is alive, or the age at death if deceased. Must be between `1` and `max_age`. Use `NA` for unknown ages (but note this may affect analysis or require imputation).
- `isAff`: An integer indicating the affection status for the cancer of interest: `1` if diagnosed, `0` if unaffected. Use `NA` for unknown status.
- `Age`: An integer representing the age at cancer diagnosis. Should be `NA` if `isAff` is `0` or `NA`. Must be between `1` and `max_age`, and less than or equal to `CurAge`. Use `NA` for unknown diagnosis age (but note this may affect analysis or require imputation).

- Geno: An integer representing the germline genetic test result: 1 for carrier (positive), 0 for non-carrier (negative). Use NA for unknown or untested individuals.

twins A list specifying identical twins or triplets in the family. Each element of the list should be a vector containing the IDs of the identical siblings within a pedigree. For example: list(c("ID1", "ID2"), c("ID3", "ID4", "ID5")). Default is NULL.

n_chains Integer, the number of chains for parallel computation. Default is 1.

n_iter_per_chain
Integer, the number of iterations for each chain. Default is 10000.

ncores Integer, the number of cores for parallel computation. Default is 6.

max_age Integer, the maximum age considered for analysis. Default is 94.

baseline_data Data providing the absolute age-specific baseline risk (probability) of developing the cancer in the general population (e.g., from SEER database). All probability values must be between 0 and 1. IMPORTANT: This should be AGE-SPECIFIC risk, NOT cumulative risk. The function will warn if the data appears to be cumulative (monotonically increasing or sum > 1). - If sex_specific = TRUE (default): A data frame with columns 'Male' and 'Female', where each column contains the age-specific probabilities for that sex. The number of rows should ideally correspond tomax_age. - If sex_specific = FALSE: A numeric vector or a single-column data frame containing the age-specific probabilities for the combined population. The length (or number of rows) should ideally correspond to max_age. Default data is provided for Colorectal cancer from SEER (up to age 94). If the number of rows/length does not match max_age, the data will be truncated or extended with the last value.

remove_proband Logical, indicating whether to remove probands from the analysis. Default is FALSE.

age_imputation Logical, indicating whether to perform age imputation. Default is FALSE.

median_max Logical, indicating whether to use the baseline median age or max_age as an upper bound for the median proposal. Default is TRUE.

BaselineNC Logical, indicating that the non-carrier penetrance is assumed to be the baseline penetrance. Currently only TRUE is supported. Setting FALSE will throw an error.

var Numeric vector, variances for the proposal distribution in the Metropolis-Hastings algorithm. Default is c(0.1, 0.1, 2, 2, 5, 5, 5, 5).

burn_in Numeric, the fraction of results to discard as burn-in (0 to 1). Default is 0 (no burn-in).

thinning_factor
Integer, the factor by which to thin the results. Default is 1 (no thinning).

imp_interval Integer, the interval at which age imputation should be performed when age_imputation = TRUE.

distribution_data
Data for generating prior distributions.

`allele_freq` Numeric, the population allele frequency of the risk variant (p). This will be automatically converted to carrier prevalence (approximately 2p for rare alleles) for internal Bayesian calculations. Default is 0.0001. Must be between 0 and 1. The function will warn if the value seems unusually high (> 1%), which may indicate confusion with carrier prevalence.

`sample_size` Optional numeric, sample size for distribution generation.

`ratio` Optional numeric, ratio parameter for distribution generation.

`prior_params` List, parameters for prior distributions.

`risk_proportion` Numeric, proportion of risk for distribution generation.

`summary_stats` Logical, indicating whether to include summary statistics in the output. Default is TRUE.

`rejection_rates` Logical, indicating whether to include rejection rates in the output. Default is TRUE.

`density_plots` Logical, indicating whether to include density plots in the output. Default is TRUE.

`plot_trace` Logical, indicating whether to include trace plots in the output. Default is TRUE.

`penetrance_plot` Logical, indicating whether to include penetrance plots in the output. Default is TRUE.

`penetrance_plot_pdf` Logical, indicating whether to include PDF plots in the output. Default is TRUE.

`plot_loglikelihood` Logical, indicating whether to include log-likelihood plots in the output. Default is TRUE.

`plot_acf` Logical, indicating whether to include autocorrelation function (ACF) plots for posterior samples. Default is TRUE.

`probCI` Numeric, probability level for credible intervals in penetrance plots. Must be between 0 and 1. Default is 0.95.

`sex_specific` Logical, indicating whether to use sex-specific parameters in the analysis. Default is TRUE.

Details

Key features:

- Bayesian estimation of penetrance using family-based data
- Support for sex-specific and non-sex-specific analyses
- Age imputation for missing data
- Visualization tools for MCMC diagnostics
- Integration with the clipp package for likelihood calculations

Value

A list containing combined results from all chains, including optional statistics and plots.

Author(s)

Maintainer: Sol Rosito <bmendel@jimmy.harvard.edu>

Authors:

- Nicolas Kubista
- BayesMendel Lab
- Giovanni Parmigiani
- Danielle Braun
- Zaid Al-Ississ
- Alice Zhang

See Also

Useful links:

- https://github.com/bayesmendel/penetrance

Examples

```
# Create example baseline data (simplified for demonstration)
baseline_data_default <- data.frame(
  Age = 1:94,
  Female = rep(0.01, 94),
  Male = rep(0.01, 94)
)

# Create example distribution data
distribution_data_default <- data.frame(
  Age = 1:94,
  Risk = rep(0.01, 94)
)

# Create example prior parameters
prior_params_default <- list(
  shape = 2,
  scale = 50
)

# Create example risk proportion
risk_proportion_default <- 0.5

# Create a simple example pedigree
example_pedigree <- data.frame(
  PedigreeID = rep(1, 4),
  ID = 1:4,
  Sex = c(1, 0, 1, 0),  # 1 for male, 0 for female
```

```
  MotherID = c(NA, NA, 2, 2),
  FatherID = c(NA, NA, 1, 1),
  isProband = c(0, 0, 1, 0),
  CurAge = c(70, 68, 45, 42),
  isAff = c(0, 0, 1, 0),
  Age = c(NA, NA, 40, NA),
  Geno = c(NA, NA, 1, NA)
)

# Basic usage with minimal iterations
result <- penetrance(
  pedigree = list(example_pedigree),
  n_chains = 1,
  n_iter_per_chain = 10,  # Very small number for example
  ncores = 1,             # Single core for example
  summary_stats = TRUE,
  plot_trace = FALSE,     # Disable plots for quick example
  density_plots = FALSE,
  penetrance_plot = FALSE,
  penetrance_plot_pdf = FALSE,
  plot_loglikelihood = FALSE,
  plot_acf = FALSE
)

# View basic results
head(result$summary_stats)
```

`plot_acf` *Plot Autocorrelation for Multiple MCMC Chains (Posterior Samples)*

Description

This function plots the autocorrelation for sex-specific or non-sex-specific posterior samples across multiple MCMC chains. It defaults to key parameters like `asymptote_male_samples`, `asymptote_female_samples`, etc.

Usage

```
plot_acf(results, n_chains, max_lag = 50)
```

Arguments

`results`	A list of MCMC chain results.
`n_chains`	The number of chains.
`max_lag`	Integer, the maximum lag to be considered for the autocorrelation plot. Default is 50.

Value

A series of autocorrelation plots for each chain.

`plot_loglikelihood` *Plot Log-Likelihood for Multiple MCMC Chains*

Description

This function plots the log-likelihood values across iterations for multiple MCMC chains. It helps visualize the convergence of the chains based on the log-likelihood values.

Usage

```
plot_loglikelihood(results, n_chains)
```

Arguments

`results`	A list of MCMC chain results, each containing the `loglikelihood_current` values.
`n_chains`	The number of chains.

Value

A series of log-likelihood plots for each chain.

`plot_pdf` *Plot Weibull Probability Density Function with Credible Intervals*

Description

This function plots the Weibull PDF with credible intervals for the given MCMC results. It allows for visualization of density curves based on the posterior samples.

Usage

```
plot_pdf(combined_chains, prob, max_age, sex = "NA")
```

Arguments

`combined_chains`	List of combined MCMC chain results containing posterior samples for penetrance parameters.
`prob`	Numeric, probability level for confidence intervals (between 0 and 1).
`max_age`	Integer, maximum age to plot.
`sex`	Character, one of "Male", "Female", or "NA" for non-sex-specific. Default is "NA".

Value

A plot showing the Weibull PDF with credible intervals.

`plot_penetrance` *Plot Weibull Distribution with Credible Intervals*

Description

This function plots the Weibull distribution with credible intervals for the given MCMC results. It allows for visualization of penetrance curves based on the posterior samples.

Usage

```
plot_penetrance(combined_chains, prob, max_age, sex = "NA")
```

Arguments

`combined_chains`	List of combined MCMC chain results containing posterior samples for penetrance parameters.
`prob`	Numeric, probability level for confidence intervals (between 0 and 1).
`max_age`	Integer, maximum age to plot.
`sex`	Character, one of "Male", "Female", or "NA" for non-sex-specific. Default is "NA".

Value

A plot showing the Weibull distribution with credible intervals.

`plot_trace` *Plot MCMC Trace Plots*

Description

Plot MCMC Trace Plots

Usage

```
plot_trace(results, n_chains, verbose = FALSE)
```

Arguments

`results`	A list of MCMC chain results.
`n_chains`	Integer, the number of chains.
`verbose`	Logical, whether to print progress messages. Default is FALSE.

Value

No return value, called for side effects. Creates trace plots for each parameter.

Examples

```
# Create example results list
results <- list(
  list(
    median_samples = rnorm(100),
    threshold_samples = runif(100),
    first_quartile_samples = rgamma(100, 2, 2),
    asymptote_samples = rbeta(100, 2, 2)
  )
)

# Generate trace plots
plot_trace(results, n_chains = 1)
```

---

printRejectionRates *Print MCMC Rejection Rates*

Description

Print MCMC Rejection Rates

Usage

```
printRejectionRates(results, verbose = TRUE)
```

Arguments

results	A list of MCMC chain results.
verbose	Logical, whether to print rates to console. Default is TRUE.

Details

Extracts and prints the rejection rates from MCMC chain results.

Value

A named numeric vector containing the rejection rate (between 0 and 1) for each MCMC chain. Names are of the form "Chain X" where X is the chain number.

Examples

```
# Create example results list with two chains
results <- list(
  list(rejection_rate = 0.3),
  list(rejection_rate = 0.4)
)

# Get rejection rates without printing
rates <- printRejectionRates(results, verbose = FALSE)

# Print rejection rates
rates <- printRejectionRates(results)
```

---

prior_params_default *Default Prior Parameters*

Description

Default parameters for the prior distributions used in the `makePriors` function.

Usage

```
prior_params_default

prior_params_default
```

Format

A list of prior distribution parameters

A list with the following components:

asymptote A list with components `g1` and `g2`, default values for the asymptote parameters.

threshold A list with components `min` and `max`, default values for the threshold parameters.

median A list with components `m1` and `m2`, default values for the median parameters.

first_quartile A list with components `q1` and `q2`, default values for the first quartile parameters.

risk_proportion_default *Default Risk Proportions*

Description

Default proportions of people at risk used in the `makePriors` function.

Usage

```
risk_proportion_default

risk_proportion_default
```

Format

A data frame of risk proportions

A data frame with the following columns:

median Proportion of people at risk at the median age.

first_quartile Proportion of people at risk at the first quartile age.

max_age Proportion of people at risk at the maximum age.

simulated_families *Processed Family Data*

Description

A dataset containing processed information about the first simulated 130 families. These families are referenced in the vigniette simulation_study.Rmd and using_penetrance.Rmd The user must specify the `pedigree` argument as a data frame which contains the family data (see `test_fam`). The family data must be in the correct format with the following columns:

Usage

```
simulated_families
```

Format

A list of processed family data.

Details

`ID` A numeric value representing the unique identifier for each individual. There should be no duplicated entries.

`Sex` A numeric value where `0` indicates female and `1` indicates male. Missing entries are not currently supported.

`MotherID` A numeric value representing the unique identifier for an individual's mother.

`FatherID` A numeric value representing the unique identifier for an individual's father.

`isProband` A numeric value where `1` indicates the individual is a proband and `0` otherwise.

`CurAge` A numeric value indicating the age of censoring (current age if the person is alive or age at death if the person is deceased). Allowed ages range from `1` to `94`.

`isAff` A numeric value indicating the affection status of cancer, with `1` for diagnosed individuals and `0` otherwise. Missing entries are not supported.

`Age` A numeric value indicating the age of cancer diagnosis, encoded as `NA` if the individual was not diagnosed. Allowed ages range from `1` to `94`.

`Geno` A column for germline testing or tumor marker testing results. Positive results should be coded as `1`, negative results as `0`, and unknown results as `NA` or left empty.

Source

Generated for package example

`test_fam2` *Processed Family Data*

Description

A dataset containing processed information about the first simulated 130 families. These families are referenced in the vigniette simulation_study_real.Rmd The user must specify the `pedigree` argument as a data frame which contains the family data (see `test_fam`). The family data must be in the correct format with the following columns:

Usage

```
test_fam2
```

Format

A list of processed family data.

Details

`ID` A numeric value representing the unique identifier for each individual. There should be no duplicated entries.

`Sex` A numeric value where `0` indicates female and `1` indicates male. Missing entries are not currently supported.

`MotherID` A numeric value representing the unique identifier for an individual's mother.

`FatherID` A numeric value representing the unique identifier for an individual's father.

`isProband` A numeric value where `1` indicates the individual is a proband and `0` otherwise.

`CurAge` A numeric value indicating the age of censoring (current age if the person is alive or age at death if the person is deceased). Allowed ages range from `1` to `94`.

`isAff` A numeric value indicating the affection status of cancer, with `1` for diagnosed individuals and `0` otherwise. Missing entries are not supported.

`Age` A numeric value indicating the age of cancer diagnosis, encoded as `NA` if the individual was not diagnosed. Allowed ages range from `1` to `94`.

`Geno` A column for germline testing or tumor marker testing results. Positive results should be coded as `1`, negative results as `0`, and unknown results as `NA` or left empty.

Source

Generated for package example

`transformDF` *Transform Data Frame*

Description

This function transforms a data frame from the standard format used in Fam3PRO into the required format which conforms to the requirements of penetrance (and clipp).

Usage

```
transformDF(df)
```

Arguments

`df`	The input data frame in the usual Fam3PRO format.

Details

This function implements a two-tier naming convention:

- User-facing input: uppercase 'Geno' (values 0 or 1)
- Internal processing: lowercase 'geno' (values "1/1" or "1/2")

The transformation converts 'Geno' = 1 (carrier) to 'geno' = "1/2", and 'Geno' = 0 (non-carrier) to 'geno' = "1/1". This separation provides clear distinction between user interface and internal implementation.

Value

A data frame in the format required for clipp with the following columns:

individual	ID of the individual
isProband	Indicator if the individual is a proband
family	Family ID
mother	Mother's ID
father	Father's ID
aff	Affection status
sex	Sex (2 for female, 1 for male)
age	Age at diagnosis or current age
geno	Genotype information (internal format)

Examples

```
# Create example data frame
df <- data.frame(
  ID = 1:2,
  PedigreeID = c(1,1),
  Sex = c(0,1),
  MotherID = c(NA,1),
  FatherID = c(NA,NA),
  isProband = c(1,0),
  CurAge = c(45,20),
  isAff = c(1,0),
  Age = c(40,NA),
  Geno = c(1,0)
)

# Transform the data frame
transformed_df <- transformDF(df)
```

---

validate_allele_freq *Validate Allele Frequency Input*

Description

This function validates whether the provided input is a valid allele frequency. Allele frequencies must be numeric, scalar, and within the range `[0, 1]`. The function also provides warnings for edge cases and common mistakes.

Usage

```
validate_allele_freq(
  allele_freq,
  param_name = "allele_freq",
  warn_threshold = 0.01
)
```

Arguments

allele_freq	The allele frequency value to validate. Should be a numeric value between 0 and 1.
param_name	Character string specifying the parameter name (for error messages). Default is "allele_freq".
warn_threshold	Numeric value above which to issue a warning about unusually high allele frequency. Default is 0.01 (1%).

Details

The function checks:

- Whether the input is numeric
- Whether the input is a single value (not a vector)
- Whether the value is between 0 and 1 (inclusive)
- Whether the value is unusually high (> warn_threshold), which may indicate the user provided carrier prevalence instead of allele frequency
- Whether the value is exactly 0 or 1, which may not be biologically meaningful

Value

Logical value TRUE if the allele frequency is valid (with possible warnings), otherwise stops with an error message.

Examples

```
# Valid allele frequencies
validate_allele_freq(0.0001) # Common for rare variants
validate_allele_freq(0.001)
validate_allele_freq(0.05)

## Not run:
# Invalid inputs (will throw errors)
validate_allele_freq("0.001")          # Not numeric
validate_allele_freq(c(0.001, 0.002))  # Vector instead of scalar
validate_allele_freq(-0.001)           # Negative value
validate_allele_freq(1.5)              # Greater than 1
validate_allele_freq(NA)               # Missing value

# Valid but will trigger warnings
validate_allele_freq(0.02)         # Unusually high (>1%), warning
validate_allele_freq(0)            # Edge case, warning
validate_allele_freq(1)            # Edge case, warning

## End(Not run)
```

validate_baseline_data *Validate Baseline Penetrance Data*

Description

This function validates baseline penetrance data to ensure it represents age-specific probabilities rather than cumulative risk. It checks for monotonicity and whether the sum exceeds 1, both of which suggest the data may be cumulative rather than age-specific.

Usage

```
validate_baseline_data(
  baseline_data,
  sex_specific = TRUE,
  param_name = "baseline_data",
  tolerance = 1e-10
)
```

Arguments

baseline_data	The baseline data to validate. Can be: • A data frame with 'Male' and 'Female' columns (when sex_specific = TRUE) • A numeric vector (when sex_specific = FALSE) • A single-column data frame (when sex_specific = FALSE)
sex_specific	Logical, indicating whether the data is sex-specific. Default is TRUE.
param_name	Character string specifying the parameter name (for messages). Default is "baseline_data".
tolerance	Numeric value for checking strict monotonicity (to account for floating point precision). Default is 1e-10.

Details

The function performs the following checks:

- For monotonicity: If values are strictly non-decreasing (monotonically increasing), this suggests cumulative risk rather than age-specific probabilities. A warning is issued.
- For sum > 1: If the sum of all probabilities exceeds 1, this is problematic because these should be age-specific probabilities. A warning is issued.
- Individual values must be between 0 and 1 (probabilities)
- No NA or infinite values are allowed

Age-specific baseline risk represents the probability of developing disease at each specific age, while cumulative risk represents the total probability up to that age. For proper penetrance estimation, age-specific (not cumulative) risk should be used.

Value

Logical value TRUE if validation passes (with possible warnings), otherwise stops with an error.

Examples

```
# Valid age-specific data (varies, not monotone)
age_specific <- c(0.001, 0.002, 0.003, 0.002, 0.004, 0.003, 0.005)
validate_baseline_data(age_specific, sex_specific = FALSE)

# Valid sex-specific data
baseline_df <- data.frame(
  Male = c(0.001, 0.002, 0.001, 0.003),
  Female = c(0.002, 0.003, 0.002, 0.004)
)
validate_baseline_data(baseline_df, sex_specific = TRUE)

## Not run:
# Will trigger warnings
# Monotone increasing (suggests cumulative risk)
cumulative <- c(0.001, 0.002, 0.003, 0.004, 0.005)
validate_baseline_data(cumulative, sex_specific = FALSE)

# Sum greater than 1
high_values <- rep(0.1, 15)  # sum = 1.5
validate_baseline_data(high_values, sex_specific = FALSE)

# Invalid data
invalid_data <- c(0.001, -0.002, 0.003)  # Negative value
validate_baseline_data(invalid_data, sex_specific = FALSE)

## End(Not run)
```

---

validate_weibull_parameters

Validate Weibull Parameters

Description

This function validates the given parameters for calculating Weibull distribution.

Usage

```
validate_weibull_parameters(
  given_first_quartile,
  given_median,
  threshold,
  asymptote
)
```

Arguments

given_first_quartile	The first quartile of the data.
given_median	The median of the data.
threshold	A constant threshold value.
asymptote	A constant asymptote value (gamma).

Value

Logical value indicating whether the parameters are valid (TRUE) or not (FALSE)

Examples

```
# Validate parameters
is_valid <- validate_weibull_parameters(
  given_first_quartile = 30,
  given_median = 50,
  threshold = 15,
  asymptote = 0.8
)
print(is_valid)
```

Index

∗ **datasets**
  baseline_data_default, 4
  distribution_data_default, 9
  out_sim, 26
  prior_params_default, 36
  risk_proportion_default, 37
  simulated_families, 37
  test_fam2, 38

absValue, 3
apply_burn_in, 3
apply_thinning, 4

baseline_data_default, 4

calculate_weibull_parameters, 7
calculateBaseline, 5
calculateEmpiricalDensity, 5
calculateNCPen, 6
combine_chains, 8
combine_chains_noSex, 9

distribution_data_default, 9
drawBaseline, 10
drawEmpirical, 10

generate_density_plots, 11
generate_summary, 11
generate_summary_noSex, 12

imputeAges, 12
imputeAgesInit, 15
imputeUnaffectedAges, 16

lik.fn, 17
lik_noSex, 18

makePriors, 19
mhChain, 20
mhLogLikelihood_clipp, 23
mhLogLikelihood_clipp_noSex, 25

out_sim, 26

penetrance, 27
penetrance-package *(*penetrance*)*, 27
plot_acf, 32
plot_loglikelihood, 33
plot_pdf, 33
plot_penetrance, 34
plot_trace, 34
printRejectionRates, 35
prior_params_default, 36

qbeta, *20*

risk_proportion_default, 37
runif, *20*

simulated_families, 37

test_fam2, 38
transformDF, 39

validate_allele_freq, 40
validate_baseline_data, 42
validate_weibull_parameters, 43