

Package ‘ibdfindr’

October 4, 2026

Title HMM Toolkit for Inferring IBD Segments from SNP Data

Version 0.5.0

Description Implements continuous-time hidden Markov models (HMMs) to infer identity-by-descent (IBD) segments shared by two individuals. Supports two- and three-state models using single-nucleotide polymorphism (SNP) genotypes or genotype likelihoods. Provides posterior probabilities at each marker (forward-backward algorithm), prediction of IBD segments (Viterbi algorithm), and functions for visualising results. Supports both autosomal data and X-chromosomal data. The methodology and package are described in Vigeland et al. (2026) <[doi:10.1016/j.fsigen.2025.103409](https://doi.org/10.1016/j.fsigen.2025.103409)>.

License GPL (>= 3)

URL <https://github.com/magnusdv/ibdfindr>

BugReports <https://github.com/magnusdv/ibdfindr/issues>

Depends R (>= 4.4)

Imports forrel, ggplot2, ibdsim2, pedtools, ribd

Suggests testthat (>= 3.0.0)

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData true

NeedsCompilation no

Author Magnus Dehli Vigeland [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-9134-4962>>)

Maintainer Magnus Dehli Vigeland <m.d.vigeland@medisin.uio.no>

Repository CRAN

Date/Publication 2026-10-04 16:00:02 UTC

Contents

brothersX	2
computePR	3
cousinsDemo	4
estimateDropout	5
findIBD	5
findSegments	7
fitHMM	8
ibdPosteriors	10
plotIBD	11
problemMarkers	12
readGL	13
sibsGL	14
totalLoglik	15
Index	16

brothersX	<i>Dataset with X-chromosomal SNP genotypes for two brothers</i>
-----------	--

Description

Simulated genotypes for two brothers at the X-chromosomal SNPs included in the FORCE panel (Tillmar et al., 2021). The data was generated with the ibdsim2 package.

Usage

brothersX

Format

A tibble with 246 rows and 9 variables:

- CHROM: Chromosome label
- MARKER: SNP identifier
- MB: Physical position in megabases
- CM: Map position in centimorgans
- A1: First SNP allele
- A2: Second SNP allele
- FREQ1: Population frequency of A1
- ID1: Genotype of individual 1
- ID2: Genotype of individual 2

References

Tillmar et al. *The FORCE Panel: An All-in-One SNP Marker Set for Confirming Investigative Genetic Genealogy Leads and for General Forensic Applications*. Genes. (2021)

Examples

brothersX

computePR	<i>Precision and Recall for IBD segment calls</i>
-----------	---

Description

Computes the precision and recall of IBD segment calls (typically from `findIBD()`) against a truth set of IBD segments.

Usage

`computePR(call, truth, details = FALSE)`

Arguments

- `call, truth` Data frames with IBD segments, each with columns `chrom`, `startCM` and `endCM`.
- `details` A logical indicating if additional details should be included in the output.

Value

A data frame with columns `Precision` and `Recall`. If `details = TRUE`, additional columns `F1`, `CallTotal` (total length of called segments) and `TruthTotal` (total length of truth segments) are included.

Examples

```
# Built-in X example
ibd = findIBD(brothersX)

# True segments (see code in `data-raw/brothersX.R`)
truth = data.frame(chrom = 23,
                   startCM = c(0, 66.841, 138.834),
                   endCM = c(10.867, 120.835, 164.398))

computePR(ibd$segments, truth)
plotIBD(ibd, refSegs = truth)
```

`cousinsDemo`*Dataset with autosomal SNP genotypes for two cousins*

Description

Simulated genotypes for two individuals at the autosomal kinship SNPs from the FORCE panel (Tillmar et al., 2021). The data was generated with the `ibdsim2` package, assuming a relationship of first cousins.

Usage

```
cousinsDemo
```

Format

A tibble with 3,915 rows and 9 variables:

- CHROM: Chromosome label
- MARKER: SNP identifier
- MB: Physical position in megabases
- CM: Map position in centimorgans
- A1: First SNP allele
- A2: Second SNP allele
- FREQ1: Population frequency of A1
- ID1: Genotype of individual 1
- ID2: Genotype of individual 2

References

Tillmar et al. *The FORCE Panel: An All-in-One SNP Marker Set for Confirming Investigative Genetic Genealogy Leads and for General Forensic Applications*. Genes. (2021)

Examples

```
cousinsDemo
```

estimateDropout	<i>Estimate allelic dropout</i>
-----------------	---------------------------------

Description

Estimates dropout probabilities for each sample, based on the SNP genotypes and allele frequencies. Two methods are implemented: a maximum likelihood estimator and a simple method of moments estimator.

Usage

```
estimateDropout(data, ids = NULL, method = c("likelihood", "moment"))
```

Arguments

data	SNP data with columns a1 and freq1 (case insensitive).
ids	Genotype columns (default: last 2 columns).
method	Either "likelihood" (default) or "moment".

Details

For a SNP with allele frequencies p and $1-p$, and dropout probability d , the probabilities of observing genotypes 1/1, 1/2, 2/2 are, respectively, $p^2 + p(1-p)d$, $2p(1-p)(1-d)$ and $(1-p)^2 + p(1-p)d$.

Value

A named numeric vector with one dropout estimate per individual.

Examples

```
estimateDropout(cousinsDemo)
```

findIBD	<i>All-in-one workflow for finding IBD segments</i>
---------	---

Description

This function conveniently wraps the key steps of the package. It first fits a continuous-time HMM to the data (`fitHMM()`), then identifies IBD segments (`findSegments()`), and finally computes the marker-wise posterior IBD probability at each marker locus (`ibdPosteriors()`). The result can be passed straight to `plotIBD()` for visualisation.

Usage

```
findIBD(
  data,
  ids = NULL,
  k1 = NULL,
  a = NULL,
  dropout = 0,
  err = 0,
  method = NULL,
  thompson = FALSE,
  input = c("GT", "GL"),
  verbose = TRUE,
  model = c("uni", "fullsib")
)
```

Arguments

data	SNP genotype data as described in fithMM() , or genotype likelihood data as returned by readGL() . Alternatively a <code>pedtools::ped</code> object or list of such, from which SNP data can be extracted.
ids	Two sample IDs or pedigree member IDs. By default, the last two samples are used.
k1, a	HMM parameters passed on to fithMM() . Supplying a value fixes the parameter; if NULL (default), the parameter is estimated.
dropout	Dropout parameter(s) passed on to fithMM() . Default: 0.
err	Error parameter passed on to fithMM() . Default: 0.
method	Optimisation method.
thompson	A logical passed on to fithMM() . Default: FALSE.
input	Either "GT" (genotype calls, default) or "GL" (genotype likelihoods). GL is currently implemented only for autosomal markers.
verbose	A logical, by default TRUE.
model	Either "uni" (unilineal, default) or "fullsib" (full siblings).

Details

Supports genotype calls and genotype likelihoods, with two- and three-state HMMs for different relationship types. See [fithMM\(\)](#) for details.

Value

A list with the following elements:

- `ids`: The two analysed individuals
- `k1`: HMM parameter (estimated or provided)
- `a`: HMM parameter (estimated or provided)

- dropout: Dropout probabilities (estimated or provided)
- loglik: Log-likelihood at the fitted (or provided) parameters
- segments: Data frame with IBD segments
- posteriors: Data frame with posterior IBD probabilities at each marker

See Also

[fitHMM\(\)](#), [findSegments\(\)](#), [ibdPosteriors\(\)](#), [plotIBD\(\)](#)

Examples

```
ibd = findIBD(brothersX)
plotIBD(ibd)

# Example with genotype likelihood input
gldat = readGL(sibsGL$data, annot = sibsGL$annot)
r = findIBD(gldat, input = "GL", model = "fullsib")
plotIBD(r)
```

findSegments	<i>Identify IBD segments</i>
--------------	------------------------------

Description

Identifies genomic segments shared identical-by-descent (IBD) between two individuals. The Viterbi algorithm is used to infer the most likely sequence of IBD states along each chromosome.

Usage

```
findSegments(
  data,
  ids = NULL,
  k1,
  a,
  dropout = 0,
  err = 0,
  prepped = FALSE,
  verbose = FALSE
)
```

Arguments

data	SNP data as described in fitHMM() .
ids	Genotype columns (default: last 2 columns).
k1, a	HMM parameters. See fitHMM() for how to estimate these.

dropout	Dropout parameter(s) passed on to <code>fitHMM()</code> . Default: 0.
err	Error parameter passed on to <code>fitHMM()</code> . Default: 0.
prepped	A logical indicating if the input data has been internally processed. Can be ignored by most users.
verbose	A logical.

Value

A data frame with columns `chrom`, `startCM`, `endCM` and `n` (number of markers). Three-state results also contain an `ibd` column. The columns `startCM` and `endCM` give the positions of the first and last markers assigned to each segment. If no segments are found, the data frame has zero rows.

See Also

`plotIBD()`

Examples

```
findSegments(cousinsDemo, k1 = 0.2, a = 5)
```

fitHMM

Fit a Hidden Markov Model to genotype data

Description

This function fits a continuous-time HMM to the provided genotype data. The parameter `k1` is the stationary probability of IBD1, while `a` is the transition rate per Morgan.

Usage

```
fitHMM(
  data,
  ids = NULL,
  k1 = NULL,
  a = NULL,
  dropout = 0,
  err = 0,
  method = "L-BFGS-B",
  thompson = FALSE,
  prepped = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

data	A data frame with columns chrom, a1, freq1, and either cm or mb (case insensitive), together with two genotype columns. Markers must be biallelic with single-character allele labels. Diploid genotypes must be consistently written as AB or A/B; male X genotypes as single alleles. Missing genotypes may be NA, "", "-" or "-/-"; markers missing in either individual are omitted.
ids	Genotype columns (default: last 2 columns).
k1, a	HMM parameters: k1 is the stationary IBD1 probability and a is the transition rate per Morgan. Supplying a value fixes the parameter; if NULL, it is estimated.
dropout	A numeric of length 1 or 2 with the dropout probability for each sample. A single value is used for both samples. Dropout must be 0 for X-chromosomal males. If NA, the dropouts are estimated with <code>estimateDropout()</code> . Default: 0.
err	IBD emission error parameter. With probability err, the IBD-state emission is replaced by the non-IBD emission. Default: 0.
method	Optimisation method passed to <code>stats::optim()</code> when k1 and a are estimated jointly.
thompson	A logical; if TRUE and k1 is not supplied, estimate k1 with <code>forrel::ibdEstimate()</code> before estimating a conditionally.
prepped	A logical indicating if the input data has been internally processed. Can be ignored by most users.
verbose	A logical indicating whether to print information during the optimisation.
...	Additional arguments passed to the control parameter of <code>stats::optim()</code> .

Details

The default two-state model is intended for unilineal relationships. A three-state model also allows IBD2 sharing. Use `findIBD()` to select the model and input type.

Markers with equal cM positions are treated as completely linked, with no state transition between them. Uninformative or unusable markers are reported and removed before analysis. This includes monomorphic markers (freq1 equal to 0 or 1), markers with missing annotation, markers with missing genotypes, and markers with impossible genotype/frequency combinations.

By default both parameters are optimised jointly using `stats::optim()`.

If `thompson = TRUE`, k1 is first estimated by `forrel::ibdEstimate()`, using the approach of Thompson (1975). This step uses marker-wise likelihoods and does not model linkage between markers. The parameter a is subsequently estimated conditional on k1.

Value

A list containing model parameters, dropout probabilities and the log-likelihood.

References

Thompson, E. A. (1975). *The estimation of pairwise relationships*. *Annals of Human Genetics* 39.

See Also

`totalLoglik()`, `forrel::ibdEstimate()`

Examples

```
fitHMM(cousinsDemo)
```

ibdPosteriors

IBD posteriors

Description

Computes the posterior probability of identity-by-descent (IBD) at each marker locus via the HMM forward-backward algorithm.

Usage

```
ibdPosteriors(
  data,
  ids = NULL,
  k1,
  a,
  dropout = 0,
  err = 0,
  prepped = FALSE,
  verbose = FALSE
)
```

Arguments

<code>data</code>	SNP data as described in <code>fitHMM()</code> .
<code>ids</code>	Genotype columns (default: last 2 columns).
<code>k1, a</code>	HMM parameters. See <code>fitHMM()</code> for how to estimate these.
<code>dropout</code>	Dropout parameter(s) passed on to <code>fitHMM()</code> . Default: 0.
<code>err</code>	Error parameter passed on to <code>fitHMM()</code> . Default: 0.
<code>prepped</code>	A logical indicating if the input data has been internally processed. Can be ignored by most users.
<code>verbose</code>	A logical.

Value

A data frame with the processed marker data and posterior IBD probabilities. The probability columns depend on the selected model.

See Also[plotIBD\(\)](#)**Examples**

```
ibdPosteriors(cousinsDemo, k1 = 0.2, a = 5)
```

plotIBD	<i>Plot IBD segments and posteriors</i>
---------	---

Description

Plot IBD segments and posteriors

Usage

```
plotIBD(
  x,
  segments = NULL,
  chrom = NULL,
  ncol = NULL,
  title = NA,
  base_size = 12,
  refSegs = NULL
)
```

Arguments

<code>x</code>	A list, typically produced with findIBD() , containing data frames named posteriors and segments. Alternatively, <code>x</code> may be just the output of ibdPosteriors() .
<code>segments</code>	A data frame with IBD segments, typically produced by findSegments() .
<code>chrom</code>	A vector of chromosomes to plot (default: all).
<code>ncol</code>	Number of columns in the plot. By default a suitable layout is chosen automatically.
<code>title</code>	Plot title. Generated automatically if NA (default); use NULL for no title.
<code>base_size</code>	Base font size.
<code>refSegs</code>	(Optional) A data frame with true IBD segments, mostly for testing and validation purposes. If provided, these segments are plotted in blue.

Value

A ggplot2 plot.

See Also

[findIBD\(\)](#), [findSegments\(\)](#), [ibdPosteriors\(\)](#)

Examples

```
x = subset(cousinsDemo, CHROM %in% 3:4)
ibd = findIBD(x, k1 = 0.2, a = 5)
plotIBD(ibd)
```

problemMarkers

Identify problematic markers

Description

Identifies markers with invalid or impossible HMM emission probabilities. Note that monomorphic markers (freq1 equal to 0 or 1) are not included.

Usage

```
problemMarkers(
  data,
  ids = NULL,
  dropout = 0,
  err = 0,
  prepped = FALSE,
  input = c("GT", "GL"),
  model = c("uni", "fullsib")
)
```

Arguments

data	SNP data as described in fithMM() .
ids	Genotype columns (default: last 2 columns).
dropout	Dropout parameter(s) passed on to fithMM() . Default: 0.
err	Error parameter passed on to fithMM() . Default: 0.
prepped	A logical for internal use; if TRUE, data is assumed to contain emission0 and emission1.
input	Either "GT" (genotype calls, default) or "GL" (genotype likelihoods).
model	Either "uni" (unilineal, default) or "fullsib" (full siblings).

Value

A data frame containing the problematic markers, or NULL if none are found. If prepped = TRUE, an integer vector of row indices is returned instead.

readGL	<i>Read genotype likelihood data</i>
--------	--------------------------------------

Description

Reads one file per sample and combines the genotype likelihoods with marker annotation. The result can be analysed with `findIBD(..., input = "GL")`.

Usage

```
readGL(files, annot, eps = NULL, sep = ",", header = TRUE, ...)
```

Arguments

<code>files</code>	A list of data frames, or a character vector of file paths, one for each sample. If unnamed, file names are used as sample IDs.
<code>annot</code>	A data frame with columns <code>chrom</code> , <code>marker</code> , <code>a1</code> , <code>a2</code> , <code>freq1</code> , and either <code>cm</code> or <code>mb</code> (all names case insensitive). Alleles must be A, C, G or T.
<code>eps</code>	Optional base error probability, strictly between 0 and 1. If <code>NULL</code> (default), the likelihoods in the files are used unchanged.
<code>sep, header</code>	Arguments passed to <code>utils::read.table()</code> , with defaults adapted to CSV files.
<code>...</code>	Additional arguments passed to <code>utils::read.table()</code> .

Details

Currently, each file must contain a column with marker names (accepted names include `rsID`, `marker`, `snp`, `name`, case insensitive). In addition the function expects a column `Coverage`, and the ten genotype likelihoods `AA`, `AC`, `AG`, `AT`, `CC`, `CG`, `CT`, `GG`, `GT`, `TT`. Likelihoods must be on the linear scale, not log-likelihoods or posterior probabilities. If `eps` is supplied, the likelihoods are instead computed from the base-count columns A, C, G, T.

Markers are matched by name to `annot$marker`. All annotation rows are retained, in their original order. Missing markers, zero coverage and all-zero selected likelihoods give three NA values for that sample.

Value

A data frame containing the annotation (with lower-case column names) and three numeric columns per sample: `ID_11`, `ID_12`, `ID_22`. These contain likelihoods for `a1/a1`, `a1/a2`, `a2/a2`, respectively.

See Also

[findIBD\(\)](#)

sibsGL

*Dataset with genotype likelihoods for two full siblings***Description**

Simulated genotype likelihoods (GLs) mimicking low-pass sequencing data for two full siblings. To keep the dataset small, this dataset only includes 3930 SNPs; real datasets typically contain many more.

Usage

sibsGL

Format

A list with four elements:

- **data:** A named list of two data frames, containing the simulated data for the two siblings. Each data frame contains SNP identifiers, nucleotide read counts, coverage and GLs (AA, AC, ..., TT).
- **annot:** SNP annotation including chromosome, physical and genetic position, alleles and allele frequency.
- **genotypes:** The true simulated genotypes of the two siblings.
- **ibd:** The simulated IBD pattern underlying the genotypes.

Details

The underlying IBD pattern and SNP genotypes were simulated with the `ibdsim2` package. Read counts for bases A, C, G, T were then simulated assuming Poisson coverage with mean 5 and a sequencing error rate of 0.001. Genotype likelihoods for all ten genotypes (AA, AC, ..., TT) were calculated from the read counts and normalised to have maximum 1 at each SNP.

Examples

sibsGL

totalLoglik	<i>Total log-likelihood for observed data</i>
-------------	---

Description

This function computes the total log-likelihood of the observed data, under the hidden Markov model. It is mainly for internal use, especially [fithMM\(\)](#).

Usage

```
totalLoglik(data, ids = NULL, k1, a, dropout = 0, err = 0, prepped = FALSE)
```

Arguments

data	SNP data as described in fithMM() .
ids	Genotype columns. Ignored if prepped = TRUE.
k1, a	HMM parameters.
dropout	Dropout parameter(s) passed on to fithMM() . Default: 0.
err	Error parameter passed on to fithMM() . Default: 0.
prepped	A logical indicating if the input data has been internally processed. Can be ignored by most users.

Value

A number: The total log-likelihood of the data under the HMM model.

Examples

```
totalLoglik(cousinsDemo, k1 = 0.2, a = 5)
```

Index

* datasets

- brothersX, [2](#)
- cousinsDemo, [4](#)
- sibsGL, [14](#)

brothersX, [2](#)

computePR, [3](#)
cousinsDemo, [4](#)

estimateDropout, [5](#)
estimateDropout(), [9](#)

findIBD, [5](#)
findIBD(), [3](#), [9](#), [11–13](#)
findSegments, [7](#)
findSegments(), [5](#), [7](#), [11](#), [12](#)
fitHMM, [8](#)
fitHMM(), [5–8](#), [10](#), [12](#), [15](#)
forrel::ibdEstimate(), [9](#), [10](#)

ibdPosteriors, [10](#)
ibdPosteriors(), [5](#), [7](#), [11](#), [12](#)

plotIBD, [11](#)
plotIBD(), [5](#), [7](#), [8](#), [11](#)
problemMarkers, [12](#)

readGL, [13](#)
readGL(), [6](#)

sibsGL, [14](#)
stats::optim(), [9](#)

totalLoglik, [15](#)
totalLoglik(), [10](#)

utils::read.table(), [13](#)