Package 'ibdfindr'

August 18, 2025

Title HMM Toolkit for Inferring IBD Segments from SNP Genotypes

Version 0.3.1

Description Implements continuous-time hidden Markov models (HMMs) to infer identity-by-descent (IBD) segments shared by two individuals from their single-nucleotide polymorphism (SNP) genotypes. 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.

License GPL (>= 3)

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

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

Depends R (>= 4.2)

Imports forrel, ggplot2, ibdsim2, pedtools, ribd

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.3.2

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 2025-08-18 14:10:02 UTC

Contents

`brothersX` *Dataset with X-chromosomal SNP genotypes for two brothers*

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 centiMorgan
- `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 *Precision and Recall for IBD segment calls*

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`, `TP` (true positives), `FP` (false positives) and `FN` (false negatives) 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 centiMorgan
- `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
```

`findIBD` *All-in-one workflow for finding IBD segments*

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,
  err = 0,
```

```
  method = NULL,
  thompson = FALSE,
  verbose = TRUE
)
```

Arguments

data	Data frame with required columns chrom, cm, a1 and freq1 (case insensitive). Alternatively, a ped object, in which case the SNP data is extracted internally.
ids	Character vector indicating genotype columns of data (default: last 2 columns).
k1, a	HMM parameters passed on to fitHMM(). Supplying a value fixes the parameter; if NULL (default), the parameter is estimated.
err	Error rate; a single number in [0,1] (default: 0).
method	Optimisation method.
thompson	A logical passed on to fitHMM(). Default: FALSE.
verbose	A logical, by default TRUE.

Value

A list with the following elements:

- k1: HMM parameter (estimated or provided)
- a: HMM parameter (estimated or provided)
- 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)
```

---

findSegments *Identify IBD segments*

Description

Identifies genomic segments shared identical-by-descent (IBD) between two individuals from SNP marker data. The method applies a hidden Markov model (HMM) along each chromosome, with states 0 (non-IBD) and 1 (IBD), and uses the Viterbi algorithm to infer the most likely sequence of states.

Usage

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

Arguments

data	Data frame with required columns chrom, cm, a1 and freq1.
ids	Genotype columns (default: last 2 columns).
k1, a	HMM parameters. See fitHMM() for how to estimate these.
err	Error rate; a single number in [0,1] (default: 0).
prepped	A logical indicating if the input data has been internally processed. Can be ignored by most users.
verbose	A logical.

Value

Data frame with IBD segments, described with columns chrom, startCM, endCM and n (the number of markers in the segment).

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, by optimising the parameters k1 (the probability of being in an IBD state) and a (the transition rate) to maximise the total log-likelihood.

Usage

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

Arguments

data	Data frame with required columns chrom, cm, a1 and freq1 (case insensitive).
ids	Genotype columns (default: last 2 columns).
k1, a	Numeric HMM parameters. Supplying a value fixes the parameter; if NULL (default), the parameter is estimated.
err	Error rate; a single number in [0,1] (default: 0).
method	A character string indicating the optimisation method to use.
thompson	A logical indicating the optimisation method. (See Details.)
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 stats::optim().

Details

By default (thompson = FALSE) both parameters k1 and a are optimised together, using stats::optimise().

If thompson = TRUE, then k1 is estimated first, using the maximum-likelihood approach for pairwise relatedness coefficients described by Thompson (1975). (Note that, although this method was originally developed for unlinked markers, it yields unbiased estimates also with linked markers.) The estimation of k1 is performed internally by calling forrel::ibdEstimate(). Subsequently, the parameter a is estimated conditional on the k1 value.

Value

A list containing the fitted parameters k1 and a, and some additional information about the optimisation.

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,
  err = 0,
  prepped = FALSE,
  verbose = FALSE
)
```

Arguments

`data`	Data frame with required columns `chrom`, `cm`, `a1` and `freq1`.
`ids`	Genotype columns (default: last 2 columns).
`k1, a`	HMM parameters. See `fitHMM()` for how to estimate these.
`err`	Error rate; a single number in `[0,1]` (default: 0).
`prepped`	A logical indicating if the input data has been internally processed. Can be ignored by most users.
`verbose`	A logical.

Value

Data frame similar to `data`, with a column `post` containing the posterior IBD probability at each marker locus.

See Also

`plotIBD()`

Examples

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

plotIBD *Plot IBD segments and posteriors*

Description

Plot IBD segments and posteriors

Usage

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

Arguments

x	A list, typically produced with findIBD(), containing data frames named posteriors and segments. Alternatively, x may be just the output of ibdPosteriors().
segments	A data frame with IBD segments, typically produced by findSegments().
chrom	A vector of chromosomes to plot (default: all).
ncol	Number of columns in the plot. By default a suitable layout is chosen automatically.
title	Plot title. Generated automatically if NA (default).
base_size	Base font size.
refSegs	(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)
```

`totalLoglik` *Total log-likelihood for observed data*

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, err = 0, prepped = FALSE)
```

Arguments

`data`	Data frame with required columns `chrom`, `cm`, `a1` and `freq1`.
`ids`	Genotype columns (ignored unless `prep = TRUE`).
`k1, a`	HMM parameters.
`err`	Error rate; a single number in `[0,1]` (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