Package: RZooRoH (via r-universe)

October 26, 2024

Type Package

Title Partitioning of Individual Autozygosity into Multiple Homozygous-by-Descent Classes

Version 0.3.2.1

Author Tom Druet, Naveen Kumar Kadri, Amandine Bertrand and Mathieu Gautier

Maintainer Tom Druet <tom.druet@uliege.be>

Description Functions to identify Homozygous-by-Descent (HBD) segments associated with runs of homozygosity (ROH) and to estimate individual autozygosity (or inbreeding coefficient). HBD segments and autozygosity are assigned to multiple HBD classes with a model-based approach relying on a mixture of exponential distributions. The rate of the exponential distribution is distinct for each HBD class and defines the expected length of the HBD segments. These HBD classes are therefore related to the age of the segments (longer segments and smaller rates for recent autozygosity / recent common ancestor). The functions allow to estimate the parameters of the model (rates of the exponential distributions, mixing proportions), to estimate global and local autozygosity probabilities and to identify HBD segments with the Viterbi decoding. The method is fully described in Druet and Gautier (2017) <doi:10.1111/mec.14324>.

Depends R (>= 3.2.0), methods

License GPL-3

Encoding UTF-8

LazyData true

Imports foreach, doParallel, parallel, data.table, RColorBrewer, iterators

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation yes

Date/Publication 2023-10-27 08:40:02 UTC

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

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

RemoteRef HEAD

RemoteSha f53f113acc18c385a6d32db28f8b06558a3e168f

Contents

`BBB_NMP_ad_subset` *Example for "ad" format specification*

Description

A dataset containing real allele depth information for 1,000 SNPs. The data is available for ten individuals and the first 1,000 SNPs on chromosome 1. The data corresponds to a low-fold sequencing experiment. There are two columns per individuals (read counts for allele 1 and for allele 2).

Usage

```
BBB_NMP_ad_subset
```

Format

A data frame with 1,000 rows and 25 variables:

chr The chromosome number

marker_name The marker id

pos The position of the marker

allele1 The name of the first marker allele

allele2 The name of the second marker allele

id1_ad1 The read count for the first individual at the first markrer

id1_ad2 The read count for the first individual at the second markrer

id2_ad1 The read count for the second individual at the first markrer

id2_ad2 The read count for the second individual at the second markrer

id3_ad1, id3_ad2, id4_ad1, id4_ad2, id5_ad1, id5_ad2, id6_ad1, id6_ad2, id7_ad1, id7_ad2, id8_ad1, id8_ad2, id9_ad
The read counts for the other individuals

`BBB_NMP_GP_subset` *Example for "gp" format specification*

Description

A dataset containing real genotype probabilities for 1,000 SNPs. The data is available for ten individuals and 1,000 SNPs on chromosome 10. The data corresponds to a low-fold sequencing experiment. There are three columns per individuals (for genotypes 00, 01 and 11).

Usage

```
BBB_NMP_GP_subset
```

Format

A data frame with 1,000 rows and 35 variables:

chr The chromosome number

marker_name The marker id

pos The position of the marker

allele1 The name of the first marker allele

allele2 The name of the second marker allele

id1_gp1 The AA genotype probability for the first individual

id1_gp2 The AB genotype probability for the first individual

id1_gp3 The BB genotype probability for the first individual

id2_gp1 The AA genotype probability for the second individual

id2_gp2 The AB genotype probability for the second individual

id2_gp3 The BB genotype probability for the second individual

id3_gp1, id3_gp2, id3_gp3, id4_gp1, id4_gp2, id4_gp3, id5_gp1, id5_gp2, id5_gp3, id6_gp1, id6_gp2, id6_gp3, id7_gp
The genotype probabilities for the other individuals

`BBB_NMP_pl_subset` *Example for "pl" format specification*

Description

A dataset containing real genotype likelihoods in phred scores for 1,000 SNPs. The data is available for ten individuals and the first 1,000 SNPs on chromosome 1. The data corresponds to a low-fold sequencing experiment. There are three columns per individuals (for genotypes 00, 01 and 11).

Usage

```
BBB_NMP_pl_subset
```

Format

A data frame with 1,000 rows and 35 variables:

chr The chromosome number

marker_name The marker id

pos The position of the marker

allele1 The name of the first marker allele

allele2 The name of the second marker allele

id1_pl1 The AA phred likelihood for the first individual

id1_pl2 The AB phred likelihood for the first individual

id1_pl3 The BB phred likelihood for the first individual

id2_pl1 The AA phred likelihood for the second individual

id2_pl2 The AB phred likelihood for the second individual

id2_pl3 The BB phred likelihood for the second individual

id3_pl1, id3_pl2, id3_pl3, id4_pl1, id4_pl2, id4_pl3, id5_pl1, id5_pl2, id5_pl3, id6_pl1, id6_pl2, id6_pl3, id7_pl1, id7_
The phred likelihoods for the other individuals

`BBB_PE_gt_subset` *Example for "gt" format specification*

Description

A dataset containing real genotypes for 1,000 SNPs. The data is available for ten individuals and the first 1,000 SNPs on chromosome 1. The data corresponds to a WGS experiment. There is one columns per individual.

Usage

```
BBB_PE_gt_subset
```

Format

A data frame with 1,000 rows and 14 variables:

chr The chromosome number

pos The position of the marker

allele1 The name of the first marker allele

allele2 The name of the second marker allele

id1 The genotypes for the first individuals (id1)

id2 The genotypes for the second individuals (id2)

id3, id4, id5, id6, id7, id8, id9, id10 The genotypes of the remaining individuals

`BBB_samples` *A file with names or IDs for ten samples.*

Description

The names (or IDs) are provided for ten samples.

Usage

```
BBB_samples
```

Format

A data frame with 10 rows and one variable.

IDs The ids for ten individuals

cumhbd *Computes the realized inbreeding coefficient*

Description

Computes the realized inbreeding coefficient with respect to a base population by summing the autozygosity from all HBD class with a rate lower or equal to the threshold T. This amounts to set the base population approximately 0.5*T generations ago. HBD classes with a higher rate are then no longer considered as autozygous.

Usage

```
cumhbd(zres, T = NULL)
```

Arguments

`zres` The name of the zres object created by the zoorun function.

`T` The value chosen to define the base population. When T is not provided, all HBD classes are considered to estimate the inbreeding coefficient.

Value

An array with the compute inbreeding coefficients for all the individuals in the analysis.

genoex *Subset of a dataset with genotypes for 20 sheeps*

Description

A dataset containing real genotypes for 20 individuals. Genotypes are available for 14,831 SNPs from the first three chromosomes were selected. The twenty last columns correspond to the genotypes. Missing genotypes are set to 9.

Usage

```
genoex
```

Format

A data frame with 14,831 rows and 25 variables:

chr The chromosome number

marker_name The name of the marker

pos The position of the marker

allele1 The name of the first marker allele

allele2 The name of the second marker allele

id1 The genotype for the first individual

id2 The genotype for the second individual

id3, id4, id5, id6, id7, id8, id9, id10, id11, id12, id13, id14, id15, id16, id17, id18, id19, id20 The genotypes of the remaining individuals

`genosim` *Example from a small simulated data set*

Description

A dataset containing simulated genotypes for 20 individuals. Genotypes are available for 10,000 SNPs on 10 chromosomes. The ten last columns correspond to the genotypes.

Usage

```
genosim
```

Format

A data frame with 10,000 rows and 24 variables:

chr The chromosome number

pos The position of the marker

allele1 The name of the first marker allele

allele2 The name of the second marker allele

id1 The genotype for the first individual

id2 The genotype for the second individual

id3, id4, id5, id6, id7, id8, id9, id10, id11, id12, id13, id14, id15, id16, id17, id18, id19, id20 The genotypes of the remaining individuals

`probhbd` *Extracts the HBD probabilities from the zres object*

Description

Extracts the locus specific HBD probabilities for an individual. A specific chromosomal region can be specified. A threshold T can be used to determine which HBD classes are used in the computation of the HBD probability. This function requires that the option "localhbd" was set to TRUE when creating the zres object.

Usage

```
probhbd(
  zres,
  zooin,
  id,
  chrom = NULL,
  startPos = NULL,
  endPos = NULL,
  T = FALSE
)
```

Arguments

zres	The name of the zres object created by the zoorun function.
zooin	The name of the zdata object created by the zoodata function. See "zoodata" for more details.
id	The number of the individual to extract.
chrom	the number of the chromosome where we are looking for HBD segments. This chromosome number refers to the position of the chromosome in the list of all chromsomes present in the input genotype data.
startPos	The starting position (on the chromosome) of the interval from which we will extract HBD segments (1 by default).
endPos	The ending position (on the chromosome) of the interval from which we will extract HBD segments (last position by default).
T	The value chosen to define the base population (to determine which classes are used in estimated of HBD probability, which classes are considered autozygous). When T is not provided, all HBD classes are considered to estimate the local HBD probability.

Value

The function returns a vector of HBD probabilities for the specified individual and chromosomal region. The HBD probabilities are computed as the sum of the probabilities for each HBD class with a rate smaller or equal than the threshold (the sum from all the HBD classes when T is not specified).

rara_mix10r *The result of an analysis on 22 sheeps from Rasa Aragonesa population.*

Description

The results were obtained by running the default model (10 classes with pre-defined rates) on 22 individuals genotyped at 37465 SNPs.

Usage

```
rara_mix10r
```

Format

the results are a zres object.

---

realized *Extracts the realized autozygosity from the zres object*

Description

Extracts the realized autozygosity from the zres object. Extraction is performed for the indicated classes (all by default) and names are added to the columns. The function must be used with more than one individual is the zres object.

Usage

```
realized(zres, classNum = NULL)
```

Arguments

`zres`	The name of the zres object created by the zoorun function.
`classNum`	An array with the number of the classes to extract. All classes are extracted by default.

Value

The function returns a data frame with one row per individual and one column per extracted classes. In addition, it gives names to the columns. For a pre-defined model, the names of HBD classes are "R_X" where X is the rate of the corresponding class. For a model with rate estimation, the names of the HBD classes are "HBDclassX" where X is the number of the HBD class. For non-HBD classes, we use "NonHBD".

---

rohbd *Extracts the HBD segments from the zres object*

Description

Extracts the HBD segments (or RoH) from the zres object. Extraction is performed for the indicated individuals and the selected region (all by default).

Usage

```
rohbd(
  zres,
  ids = NULL,
  chrom = NULL,
  startPos = NULL,
  endPos = NULL,
  inside = TRUE
)
```

Arguments

`zres` The name of the zres object created by the zoorun function.

`ids` An array with the ids of the individuals to extract. All individuals are extracted by default.

`chrom` the number of the chromosome where we are looking for HBD segments. This chromosome number refers to the position of the chromosome in the list of all chromsomes present in the input genotype data.

`startPos` The starting position (on the chromosome) of the interval from which we will extract HBD segments (1 by default).

`endPos` The ending position (on the chromosome) of the interval from which we will extract HBD segments (last position by default).

`inside` A logical indicating whether we extract only segment within the interval (TRUE) or overlapping with the interval (FALSE). By 'within the interval', we mean that both starting and end position of the HBD segment should be in the interval. By 'overlapping', we mean that at least one part of the HBD segment should be located in the interval.

Value

The function returns a data frame with the HBD segments fitting the filtering rules (id and position). The data frame has one line per identified HBD segment and nine columns: id is the number of the individual in which the HBD segments is located, chrom is the chromosome of the HBD segments, start_snp is the number of the SNP at which the HBD segment starts (the SNP number within the chromosome), start_end is the number of the SNP at which the HBD segment ends (the SNP number within the chromosome), start_pos is the position at which the HBD segment starts (within the chromosome), start_end is the position at which the HBD segment ends (within the chromosome), number_snp is the number of consecutive SNPs in the HBD segment, length is the length of the HBD segment (for instance in bp or in cM/1000000) and HBDclass is the HBD class associated with the HBD segment.

`soay_mix10r` *The result of an analysis on 110 sheeps from the Soay population.*

Description

The results were obtained by running the default model (10 classes with pre-defined rates) on 110 individuals genotyped at 37465 SNPs.

Usage

```
soay_mix10r
```

Format

the results are a zres object.

`typs` *Subset of a dataset with genotypes for 6 individuals from a cattle population.*

Description

A dataset containing real genotypes for 6 individuals. Genotypes are available for a low density array with 6370 SNPs on 29 autosomes. The six last columns correspond to the genotypes. Missing genotypes are set to 9.

Usage

```
typs
```

Format

A data frame with 6370 rows and 10 variables:

chr The chromosome number

pos The position of the marker

allele1 The name of the first marker allele

allele2 The name of the second marker allele

id1 The genotypes for the first individuals (id1)

id2 The genotypes for the second individuals (id2)

id3 The genotypes for the third individuals (id3)

id4 The genotypes for the fourth individuals (id4)

id5 The genotypes for the fifth individuals (id5)

id6 The genotypes for the sixth individuals (id6)

`typsfrq` *A file with marker allele frequencies for the cattle population.*

Description

The allele frequencies of the first allele were computed in a larger sample.

Usage

```
typsfrq
```

Format

A data frame with 6370 rows and one variable.

frq The allele frequencies estimated for the first allele

`wilt_mix10r` *The result of an analysis on 23 sheeps from Wiltshire population.*

Description

The results were obtained by running the default model (10 classes with pre-defined rates) on 23 individuals genotyped at 37465 SNPs.

Usage

```
wilt_mix10r
```

Format

the results are a zres object.

zoodata *Read the genotype data file*

Description

Read a data file and convert it to the RZooRoH format in a 'zooin' object required for further analysis.

Usage

```
zoodata(
  genofile,
  min_maf = 0,
  zformat = "gt",
  supcol = 5,
  chrcol = 1,
  poscol = 3,
  allelefreq = NULL,
  samplefile = NA
)
```

Arguments

`genofile` The name of the input data file. Note that the model is designed for autosomes. Other chromosomes and additional filtering (e.g. call rate, missing, HWE, etc.) should be performed prior to run RZooRoH with tools such as PLINK or bcftools for instance. The model works on an ordered map and ignores SNPs with a null position.

`min_maf` The minimum allele frequency to keep variants in the analysis (optional / set to 0.00 by default to keep all markers). Values such as 0.01 allows to exclude monomorphic markers that are not informative and to reduce the size of the data and computational costs. There is no marker exclusion on call rate. However, we expect that data filtering is done prior to RZooRoH with tools such as PLINK or vcftools.

`zformat` The code corresponding to the format of the data file ("gt" for genotypes, "gp" for genotype probabilities, "gl" for genotype likelihoods in Phred scores, "ad" for allelic depths). For all these formats, markers are ordered per rows and individuals per columns. Variants should be ordered by chromosome and position. By default, the first five columns are chromosome identification (e.g, "1", "chr1"), the name of the marker, the position of the marker in base pairs or better in cM multiplied by 1,000,000 when genetic distances are known, the first marker allele and the second marker allele. Information per individual varies according to the format. With the "gt" format we have one column per individual with 0, 1 and 2 indicating the number of copies of the first allele (and 9 for missing). With the "gp" format we have three column per individual with the probabilities of genotype 11 (homozygous for the first allele), genotype 12 and genotype 22 (this corresponds to the oxford GEN format). Similarly, with the "gl" format, we have three column per individual with the likelihoods for genotypes 11, 12 and 22 in Phred scores. Finally, with the "ad" format, we expect two columns per individual: the number of reads for allele 1 and the number of reads for allele 2. For these three last formats, missing values must be indicated by setting all elements to 0. If one of the columns is non-null for one individual, the genotype will be considered non-missing. Note that the marker alleles specified in columns 4 and 5 are not used.

Conversion of a PLINK ped file or a VCF file to RZooRoH format can easily be performed using PLINK (version 1.9) or using bcftools.

For ped files, recode them to oxford gen format with plink –file myinput –recode oxford –autosome –out myoutput. The autosome option keeps only SNPs on autosomes as required by RZooRoH.

For vcf files, bcftools can be used to recode a vcf to the oxford gen format with the convert option: bcftools convert -t ^chrX,chrY,chrM -g outfile –chrom –tag GT myfile.vcf. The –chrom option is important to obtain chromosome number in the first column. The tag option allows to select which field from the vcf file (GT, PL, GL or GP) is used to generate the genotype probabilities exported in the oxford gen format. The -t option allows to exclude chromosomes (this is an example and chromosome names must be adapted if necessary). The needed output data is then outfile.gen.

If some genotype probabilities are missing, with a value of "-nan", you must replace them with "0" (triple 0 is considered as missing). This can be done with this command:

sed -e ’s/-nan/0/g’ file.gen > newfile.gen

`supcol` An optional argument that indicates the number of additional columns before the individuals genotypes (five columns are expected by default as described in the zformat argument description). Note that the function requires at least two information columns: the chromosome number and the marker position.

`chrcol` An optional argument that indicates the column number where the chromosome information is indicated (first column by default).

`poscol` An optional argument that indicates the column number where the marker position is indicated (third column by default).

`allelefreq` A vector with allele frequencies for the first marker allele (optional). By default, the allele frequencies are estimated from the data. The option allows to skip this computation or to provide external allele frequencies estimated by another method or on another data set.

`samplefile` A file with names of the samples (optional). It must match with the number of genotypes. If none is provided, the position in the genofile is used as ID.

Value

The function return a zooin object called containing the following elements: zooin@genos a matrix with the genotypes or genotype probabilities, zooin@bp an array with marker positions, zooin@chrbound a matrix with the first and last marker number for each chromosome, zooin@nind the number of individuals, zooin@nsnps the number of markers conserved after filtering for minor allele frequency, zooin@freqs an array with the marker allele frequencies, zooin@nchr the number of chromosomes, zooin@zformat the format of the data ("gt","gp","gl","ad") and zooin@sample_ids (the names of the samples).

Examples

```
# Get the name and location of example files

myfile1 <- system.file("exdata","genoex.txt",package="RZooRoH")
myfile2 <- system.file("exdata","genosim.txt",package="RZooRoH")

# Load your data with default format into a zooin object named "data1":

data1 <- zoodata(myfile1)

# Load the first data file with default format and filtering out markers with MAF < 0.02
# into a zooin object called "data1frq002":

data1frq002 <- zoodata(myfile1, min_maf = 0.02)

# Load the first data file with default format, with external allele frequencies
# (here a random set we create) and filtering out markers with MAF < 0.01:

myrandomfreq <- runif(14831)
data1c <- zoodata(myfile1, allelefreq = myrandomfreq, min_maf = 0.01)

# Load the second data file and indicate your own format (chromosome number in column 1,
# map position in column 2, 4 columns before genotypes) and filtering out markers with
# MAF < 0.01. The created zooin object is called "Sim5":

Sim5 <- zoodata(myfile2, chrcol = 1, poscol =2, supcol = 4, min_maf = 0.01)
```

zoodoris *Returns number of HBD segments (or genome proportion) per size bins for DoRIS.*

Description

The number of HBD segments in 1Mb bins is computed and the function returns a table that can be used with DoRIS to estimate paste effective population size. Alternatively, the proportion of the genome in the different 1 Mb bins is computed. The user must specify the range of the bins (the smallest and largest HBD segments considered). #'

Usage

```
zoodoris(zres, minv, maxv, glen, method = "counts")
```

Arguments

`zres`	The name of the zres object created by the zoorun function.
`minv`	The minimum length of HBD segments (in Mb) used for the DoRIS analysis.
`maxv`	The maximum length of HBD segments (in Mb) used for the DoRIS analysis.
`glen`	The length of the genome in Mb (including only the autosomes, the space where HBD segments have been searched for).
`method`	The argument is used to indicate that function will return counts of HBD segments (method = "counts") or proportion of the genome (method ="sharing") per 1 Mb bins.

Value

A data frame with three columns and one row per 1Mb bins. The first column indicates the the length of smallest HBD segments in the bin, the second column indicates the length of the largest HBD segments in the bin and the third column indicates the number of segments in the bin or the proportion of the genome in the bin.

`zoomodel` *Define the model for the RZooRoH*

Description

Help the user to create a model for RZooRoH, including default parameters. The output is a zmodel object necessary to run RZooRoH.

Usage

```
zoomodel(
  predefined = TRUE,
  K = 10,
  mix_coef = rep(0, K),
  base_rate = 2,
  krates = rep(0, K),
  err = 0.001,
  seqerr = 0.001,
  layers = TRUE
)
```

Arguments

`predefined`	Logical (TRUE or FALSE) to define whether rates of HBD and non-HBD-classes will be estimated by the model ("kr" model) or whether the rates of these classes are fixed and pre-defined by the user ("mixkr" model). The default value is "predefined = TRUE".

`K` The number of HBD and non-HBD classes. There are always K-1 HBD classes and one non-HBD class. By default, K is set to 10 but this is not optimal for all data sets. Hence, we recommend to the users to select their own value. If K is set to 2 and rates are estimated, RZooRoH will use the same rate for the HBD and the non-HBD class (so-called 1R model).

`mix_coef` The starting value for the mixing coefficients for all HBD and non-HBD classes. The mixing coefficients determine the frequency of the segments from different classes, they determine the probability to start a new segment in a given class when a segment ends. The mixing coefficients should sum to 1 and the function expects K mixing coefficients. The default values are 0.01 for HBD classes and the value for the non-HBD class is such that all mixing coefficients sum to 1. In case the parameters are not estimated (e.g. when running the forward-backward or the Viterbi algorithm alone), these are the mixing coefficients used by the RZooRoH model.

`base_rate` is a integer used to define the rates of successive HBD classes (see krates below). This parameter is most useful when using a mixkr model with predefined rates. The rate of each HBD class will be equal to the base_rate raised to the exponent k (the class number). The non-HBD class will have the same rate as the last HBD class. For instance, with a base_rate of 2 and five classes, we have the following rates: 2, 4, 8, 16 and 16. Similarly, with a base_rate of 10 and four classes, we have 10, 100, 1000 and 1000. With this method, more HBD classes are defined for more recent ancestors (for which we have more information to estimate R) and less for ancient HBD classes (it doesn't make sense to try to distinguish R = 1000 from R = 1010). In addition, since the expected length of HBD segments is expected to be approximately 1/R, the ratio between successive expected HBD lengths remains the same. This ratio also determines the ability of the model to distinguish segments from distinct classes. By keeping the ratio constant, the aptitude to discriminate between HBD classes is also constant. In addition, this method allows to cover a wide range of generations in the past, with more emphasis on recent ancestors. The default value for the base_rate is 2.

`krates` Is an array with a rate for each HBD and non-HBD class. The function expects K positive rates. These rates are parameters of the exponential distribution that together with the distance in centimorgans defines the probability to end a HBD segments between two markers. Each HBD class has a distinct rate. Therefore, the expected length of HBD classes is defined by the rates. The expected length if equal to 1/R. These krates are associated with the age of the common ancestor of the HBD segment. The rate is approximately equal to the size of the inbreeding loop (twice the number of generations to the common ancestor) when the map is given in Morgans. By default, the rates are defined by the base_rate parameter (2, 4, 8, 16, ...).

`err` Indicates the error term, the probability to observe an heterozygous genotype in a HBD segment. The genotype could be heterozygous due to a mutation occuring on the path to the common ancestor. It can also be associated with a genotype calling error or a technical error. In case GP or GL formats are used (with genotyped probabilities or phred scores) or when an AD format is used (based on read counts), this error term still represents the probability to observe an heterozygous genotype in a HBD segment. When an heterozygous genotype was called with a probability equal to 1.00, this heterozygosity in an

HBD track might be associated to a mutation or to errors not accounted for by the model used to estimate the genotype probabilities (e.g., GATK). The emission probability to observe a heterozygous genotype in an HBD class will never go below the error term. The default value is 0.001.

`seqerr` This parameters is used only with the AD format. In the AD format the user gives the number of reads for both alleles. A simple model is then used to estimate the genotype probabilities based on the read counts. In that model, the seqerr represents the probability to have a sequencing error in one read. The default value is 0.001.

`layers` Logical (TRUE or FALSE) - When true, this parameter indicates that the data is modeled as mosaic of HBD and non-HBD classes at different levels. At each level, HBD and non-HBD classes have the same rate (the same expected length). Non-HBD classes are subsequently modelled as mosaic of non-HBD segments and HBD segments from more ancient generations (from smaller sizes). At each level, the mixing coefficients can be interpreted as the inbreeding coefficient at that level (TRUE by default). This model corresponds to the Nested 1R model (N1R).

Value

The function return an object that defines a model for RZooRoH and incuding the following elements: zmodel@typeModel equal to "kr", "mixkr" or "mixkl" according to the selected model, zmodel@mix_coef an array with mixing coefficients, zmodel@krates an array with the rates of the HBD and non-HBD classes, zmodel@err the parameter defining the probability to observe an heterozygous genotype in an HBD class, and zmodel@seqerr the parameter defining the probability of sequencing error per read.

Examples

```
# To define a the default model, with 10 classes (9 HBD and 1 non-HBD class)
# and with pre-defined rates for HBD classes with a base of 2 (2, 4, 8, ...):

mix10R <- zoomodel()

# To see the parameters of the defined model, just type:

mix10R

# To define a model with pre-defined rates for 5 classes (4 HBD and 1 non-HBD
# class) and using a base of 10 to define rates (10, 100, 1000, ...):

mix5R <- zoomodel(K=5,base=10)

# To define a model with two classes, with estimation of rates for HBD classes
# and starting with a rate 10:

my.mod1R <- zoomodel(predefined=FALSE,K=2,krates=c(10,10))

# To define a model with four classes, with estimation of rates for HBD classes
# and choosing four initial rates:
```

```
my.mod4R <- zoomodel(predefined=FALSE,K=4,krates=c(16,64,256,256))
```

`zooplot_hbdseg` *Plot HBD segments identified with the ZooROH model*

Description

Plot HBD segments identified with the ZooRoH model for one or several populations.

Usage

```
zooplot_hbdseg(
  input,
  chr = NULL,
  coord = NULL,
  minlen = 0,
  cols = NULL,
  plotids = TRUE,
  toplot = NULL,
  randomids = FALSE,
  nrandom = (rep(10, length(input))),
  seed = 100
)
```

Arguments

`input`	a named list with one or several zres objects obtained after running zoorun. The zres objects are the output of the zoorun function. For instance, putting list(name1 = zres1, name2 = zres2). The function will then use the names in the plot (in case several zres objects are used).
`chr`	the number of the chromosome where we are looking for HBD segments. This chromosome number refers to the position of the chromosome in the list of all chromsomes present in the input genotype data.
`coord`	a vector with the start and end position (in bp) of the region to plot.
`minlen`	the minimal length (in cM or Mb) of HBD segments to be plotted (set to 0 by default).
`cols`	a vector with the colors to be used for each population or zres object.
`plotids`	a logical indicating whether the IDs of the individuals are plotted on the graph (TRUE by default).
`toplot`	a list of vectors indicating the zres@ids to be plotted. This option can be used to select the individuals to plot. The list must contain one vector per population or zres object. By default, all individuals are plotted.

`randomids` a logical indicating whether a randomset of individuals is plotted. This option allows to reduce the number of individuals in the plot. The option can not be used simultaneously with the toplot option. By default, randomids is FALSE.

`nrandom` a vector indicating the number of individuals to be randomly sampled per population or per zres object when randomids is TRUE. By default, we select 10 individuals per zres object. This vector must have the same length as the input list.

`seed` a value for the random seed used to sample individuals to plot (when the randomids option is TRUE).

Value

The function plots the HBD segments identified in the region, using different colors for different zres object. Each line represents a different individual.

`zooplot_individuals` *Plot individual curves with proportion of the genome in each HBD class or cumulated proportion in HBD classes with rates smaller than a threshold.*

Description

For each individual, the function plots the mean percentage of the genome in different HBD classes or the inbreeding coefficient obtained by summing autozygosity associated with HBD classes with a rate lower or equal to a threshold (e.g., including all HBD classes with longer and more recent HBD segments than a selected threshold).

Usage

```
zooplot_individuals(input, cumulative = TRUE, toplot = NULL, ncols = 2)
```

Arguments

`input` a named list with one or several zres objects obtained after running zoorun. The zres objects are the output of the zoorun function. For instance, putting list(name1 = zres1, name2 = zres2). The function will then use the names in the plot (in case several zres objects are used).

`cumulative` a logical indicating whether individual autozygosity is plotted per class (FALSE) or summed over all HBD class with a rate smaller than a value (these cumulated values are obtained for every rate defined in the model). By default, this value is TRUE. When FALSE, the percentages correspond to the individual genome-wide probabilities of belonging to each HBD-class or to the fraction of the genome in an autozygosity class. When TRUE, we obtain the probability of belonging to an HBD class with a rate smaller or equal than a threshold (here we use the pre-defined rates of the model as thresholds), averaged over the whole genome for each individual. This corresponds to report individual genomic inbreeding coefficients estimated with respect to different base populations obtained by selecting different thresholds T that determine which HBD classes are considered in the estimation of the genomic inbreeding coefficient (setting the base population approximately 0.5 * T generations ago).

`toplot` A list of vectors indicating the zres@ids to be plotted. This option can be used to select the individuals to plot. The list must contain one vector per population or zres object. By default, all individuals are plotted.

`ncols` when several populations are plotted, ncols determines how many results (graphs) are plotted per row.

Value

The function plots either the individual proportions of the genome associated with different HBD classes or individual genomic inbreeding coefficients estimated with respect to different base populations (from young to older). With both option, the average values are plotted in red.

zooplot_partitioning *Plot the partitioning of the genome in different HBD classes for each individual*

Description

Plot the partitioning of the genome in different HBD classes for each individual

Usage

```
zooplot_partitioning(
  input,
  cols = NULL,
  plotids = TRUE,
  toplot = NULL,
  randomids = FALSE,
  nrandom = NULL,
  seed = 100,
  ylim = c(0, 1),
  border = TRUE,
  nonhbd = TRUE,
  vertical = FALSE
)
```

Arguments

`input` a named list with one or several zres objects obtained after running zoorun. The zres objects are the output of the zoorun function. For instance, putting list(name1 = zres1, name2 = zres2). The function will then use the names in the plot (in case several zres objects are used).

`cols` A vector with the colors to be used for each class in the model.

`plotids` A logical indicating whether the IDs of the individuals are plotted on the graph (TRUE by default).

`toplot` A list of vectors indicating the zres@ids to be plotted. This option can be used to select the individuals to plot. The list must contain one vector per population or zres object. By default, all individuals are plotted.

`randomids` A logical indicating whether a randomset of individuals is plotted. This option allows to reduce the number of individuals in the plot. The option can not be used simultaneously with the toplot option. By default, randomids is FALSE.

`nrandom` A vector indicating the number of individuals to be randomly sampled per population or per zres object when randomids is TRUE. By default, we select 10 individuals per zres object. This vector must have the same length as the input list.

`seed` A value for the random seed used to sample individuals to plot (when the randomids option is TRUE).

`ylim` The limits of the y-axis.

`border` Whether a border is plotted around each block of the barplot or not. When set to FALSE, it allows to get a less dense plot when many individuals are plotted.

`nonhbd` Whether the a border is plotted around the non-hbd contribution. When set to FALSE, it allows to get a less dense plot when many individuals are plotted.

`vertical` Whether the populations or zres labels are printed vertically or not.

Value

Individuals are presented with stacked barplots. Each vertical stack of bars represents one individual. Each class is represented with a bar of a different color. The height of the bar represents the proportion associated with the corresponding class. The total height of the stack is the total autozygosity.

`zooplot_prophbd` *Plot proportion of the genome associated with different HBD classes*

Description

Plot the mean percentage of the genome in different HBD classes or the inbreeding coefficient obtained by summing autozygosity associated with HBD classes with a rate lower or equal to a threshold (e.g., including all HBD classes with longer and more recent HBD segments than a selected threshold).

Usage

```
zooplot_prophbd(input, cols = NULL, style = "barplot", cumulative = FALSE)
```

Arguments

`input` a named list with one or several zres objects obtained after running zoorun. The zres objects are the output of the zoorun function. For instance, putting list(name1 = zres1, name2 = zres2). The function will then use the names in the plot (in case several zres objects are used).

`cols` a vector with the colors to be used for each population or zres object.

`style` select "barplot", "lines" or "boxplot" for the graphic styles. Boxplot can be used with a single zres file or population.

`cumulative` a logical indicating whether mean autozygosity is estimated per class (FALSE) or summed over all HBD class with a rate smaller than a value (these cumulated values are obtained for every rate defined in the model). By default, this value is FALSE. When FALSE, the percentages correspond to the mean individual genome-wide probabilities of belonging to each HBD-class or to the fraction of the genome in an autozygosity class. When TRUE, we obtain the mean probability of belonging to an HBD class with a rate smaller or equal than a threshold (here we use the pre-defined rates of the model as thresholds), averaged over the whole genome and all individuals. This corresponds to report mean genomic inbreeding coefficients estimated with respect to different base populations obtained by selecting different thresholds T that determine which HBD classes are considered in the estimation of the genomic inbreeding coefficient (setting the base population approximately 0.5 * T generations ago).

Value

The function plots either the average proportion of the genome associated with different HBD classes or the average genomic inbreeding coefficient estimated with respect to different base populations (from young to older).

`zoorun` *Run the ZooRoH model*

Description

Apply the defined model on a group of individuals: parameter estimation, computation of realized autozygosity and homozygous-by-descent probabilities, and identification of HBD segments (decoding).

Usage

```
zoorun(
  zoomodel,
  zooin,
  ids = NULL,
  method = "opti",
  fb = TRUE,
  vit = TRUE,
  minr = 0,
  maxr = 1e+08,
  maxiter = 1000,
  convem = 1e-10,
  localhbd = FALSE,
  nT = 1
)
```

Arguments

`zoomodel` A valid zmodel object as defined by the zoomodel function. The model indicates whether rates of exponential distributions are estimated or predefined, the number of classes, the starting values for mixing coefficients and rates, the error probabilities. See "zoomodel" for more details.

`zooin` A valid zdata object as obtained by the zoodata function. See "zoodata" for more details.

`ids` An optional argument indicating the individual (its position in the data file) that must be proceeded. It can also be a vector containing the list of numbers that must be proceeded. By default, the model runs for all individuals.

`method` Specifies whether the parameters are estimated by optimization with the L-BFGS-B method from the optim function (option method="opti", default value) or with the EM-algorithm (option method="estem"). When the EM algorithm is used, the Forward-Backward algorithm is run automatically (no need to select the fb option - see below). We recommend to set minr to 1 when using the EM algorithm. If the user doesn't want to estimate the parameters he must set method="no".

`fb` A logical indicating whether the forward-backward algorithm is run (optional argument - true by default). The Forward-Backward algorithm estimates the local probabilities to belong to each HBD or non-HBD class. By default, the function returns only the HBD probabilities for each class, averaged genome-wide, and corresponding to the realized autozygosity associated with each class. To obtain HBD probabilities at every marker position, the option localhbd must be set to true (this generates larger outputs).

`vit` A logical indicating whether the Viterbi algorithm is run (optional argument - true by default). The Viterbi algorithm performs the decoding (determining the underlying class at every marker position). Whereas the Forward-Backward algorithms provide HBD probabilities (and how confident a region can be declared HBD), the Viterbi algorithm assigns every marker position to one of the defined classes (HBD or non-HBD). When informativity is high (many SNPs per HBD segments), results from the Forward-Backward and the Viterbi algorithm are very similar. The Viterbi algorithm is best suited to identify HBD segments. To estimate realized inbreeding and determine HBD status of a position, we recommend to use the Forward-Backward algorithm that better reflects uncertainty.

`minr` With optim and the reparametrized model (default), this indicates the minimum difference between rates of successive classes. With the EM algorithm, it indicates the minimum rate for a class. It is an optional argument set to 0. Adding such constraints might slow down the speed of convergence with optim and we recommend to run first optim without these constraints. With the EM algorithm, we recommend to use a value of 1.

`maxr` With optim and the reparametrized model (default), this indicates the maximum difference between rates of successive classes. With the EM algorithm, it indicates the maximum rate for a class. It is an optional argument set to an arbitrarily large value (100000000). Adding such constraints might slow down the speed of convergence with optim and we recommend to run first optim without these constraints.

`maxiter` Indicates the maximum number of iterations with the EM algorithm (optional argument - 1000 by default).

`convem` Indicates the convergence criteria for the EM algorithm (optional argument / 1e-10 by default).

`localhbd` A logical indicating whether the HBD probabilities for each individual at each marker are returned when using the EM or the Forward-Backward algorithm (fb option). This is an optional argument that is false by default.

`nT` Indicates the number of threads used when running RZooRoH in parallel (optional argument - one thread by default).

Value

The function return a zoores object with several slots accesses by the "@" symbol. The three main results are zoores@realized (the matrix with partitioning of the genome in different HBD classes for each individual), zoores@hbdseg (a data frame with identified HBD segments) and zoores@hbdp (a list of matrices with HBD probabilities per SNP and per class).

Here is a list with all the slots and their description:

1. zoores@nind the number of individuals in the analysis,
2. zoores@ids a vector containing the numbers of the analyzed individuals (their position in the data file),
3. zoores@mixc the (estimated) mixing coefficients per class for all individuals,
4. zoores@krates the (estimated) rates for the exponential distributions associated with each HBD or non-HBD class for all individuals,
5. zoores@niter the number of iterations for estimating the parameters (per individual),
6. zoores@modlik a vector containing the likelihood of the model for each individual,
7. zoores@modbic a vector containing the value of the BIC for each individual,
8. zoores@realized a matrix with estimated realized autozygosity per HBD class (columns) for each individual (rows). These values are obtained with the Forward-Backward algorithm - fb option),
9. zoores@hbdp a list of matrices with the local probabilities to belong to an underlying hidden state (computed for every class and every individual). Each matrix has one row per class and one column per snp. To access the matrix from individual i, use the brackets "[[]]", for instance zoores@hbdp[[i]],
10. zoores@hbdseg a data frame with the list of identified HBD segments with the Viterbi algorithm (the columns are the individual number, the chromosome number, the first and last SNP of the segment, the positions of the first and last SNP of the segment, the number of SNPs in the segment, the length of the segment, the HBD state of the segment),

11. zoores@optimerr a vector indicating whether optim ran with or without error (0/1),
12. zoores@sampleids is a vector with the names of the samples (when provided in the zooin object through the zoodata function).

Examples

```
# Start with a small data set with six individuals and external frequencies.
freqfile <- (system.file("exdata","typsfrq.txt",package="RZooRoH"))
typfile <- (system.file("exdata","typs.txt",package="RZooRoH"))
frq <- read.table(freqfile,header=FALSE)
typ <- zoodata(typfile,supcol=4,chrcol=1,poscol=2,allelefreq=frq$V1)
# Define a model with two HBD classes with rates equal to 10 and 100.
Mod3R <- zoomodel(K=3,base_rate=10)
# Run the model on all individuals.
typ.res <- zoorun(Mod3R, typ)
# Observe some results: likelihood, realized autozygosity in different
# HBD classes and identified HBD segments.
typ.res@modlik
typ.res@realized
typ.res@hbdseg
# Define a model with one HBD and one non-HBD class and run it.
Mod1R <- zoomodel(K=2,predefined=FALSE)
typ2.res <- zoorun(Mod1R, typ)
# Print the estimated rates and mixing coefficients.
typ2.res@krates
typ2.res@mixc
# Get the name and location of a second example file and load the data:
myfile <- (system.file("exdata","genoex.txt",package="RZooRoH"))
ex2 <- zoodata(myfile)
# Run RZooRoH to estimate parameters on your data with the 1 HBD and 1 non-HBD
# class (parameter estimation with optim).
my.mod1R <- zoomodel(predefined=FALSE,K=2,krates=c(10,10))
my.res <- zoorun(my.mod1R, ex2, fb = FALSE, vit = FALSE)
# The estimated rates and mixing coefficients:
my.res@mixc
my.res@krates
# Run the same model and run the Forward-Backward alogrithm to estimate
# realized autozygosity and the Viterbi algorithm to identify HBD segments:
my.res2 <- zoorun(my.mod1R, ex2)
# The table with estimated realized autozygosity:
my.res2@realized
# Run a model with 4 classes (3 HBD classes) and estimate the rates of HBD
# classes with one thread:
my.mod4R <- zoomodel(predefined=FALSE,K=4,krates=c(16,64,256,256))
my.res3 <- zoorun(my.mod4R, ex2, fb = FALSE, vit = FALSE, nT =1)
# The estimated rates for the 4 classes and the 20 individuals:
my.res3@krates
# Run a model with 5 classes (4 HBD classes) and predefined rates.
# The model is run only for a subset of four selected individuals.
# The parameters are estimated with the EM-algorithm, the Forward-Backward
# alogrithm is ued to estimate realized autozygosity and the Viterbi algorithm to
# identify HBD segments. One thread is used.
```

```
mix5R <- zoomodel(K=5,base=10)
my.res4 <- zoorun(mix5R,ex2,ids=c(7,12,16,18), method = "estem", nT = 1)
# The table with all identified HBD segments:
my.res4@hbdseg
```

zoosimd — *Realizes a simulation using a zooroh data set and a zooroh model*

Description

Performs a simulation under a model similar too ZooRoH. It simulates the genome as a mosaic of HBD and non-HBD segments. Several non-HBD classes can be simulated. Classes with high rates (short segments from ancient ancestors) are simulated first. Then, more recent classes are subsequently added. Recent HBD segments will mask more ancient HBD segments. See details in the method published in Molecular Ecology (Druet and Gautier, 2017). Genotypes are simulated using provided allele frequencies (from the sample), their genetic distances, the number of chromosomes and the number of SNPs. These simulations do not take into account linkage disequilibrium information. This simulation tool can for instance be used to check whether there is enough information in the data set to estimate HBD segments and their partitioning in multiple classes. Note that this simulation tool is not computationally efficient.

Usage

```
zoosimd(simdata, simmodel, nsim, fullout = FALSE)
```

Arguments

`simdata`	The name of the zdata object created by the zoodata function.
`simmodel`	The name of the zmodel object created by the zoomodel function. The simulation program uses only the number of classes, their rate and their mixing coefficient (it is the same for a pre-defined model or not).
`nsim`	The number of simulated individuals.
`fullout`	Indicates whether a more detailed output is requested.

Value

The function simulates genotypes with the properties of the data set (number of SNPs, genetic map, allele frequencies) and according to the specified model (number of HBD classes, rate of the classes and proportion of mixing). The simulation is created as in Druet and Gautier (2017) and new autozygosity masks more ancient autozygosity. The output is a new zoodata object that can be analyzed with the zoorun function.

If a more detailed output is requested with the fullout parameter, then the function returns list (for instance simres) containing the zoodata, a matrix with realized inbreeding per individual, estimated at SNP positions, in the different classes (with 1 being the most ancient class (non-HBD) and the highest number corresponds to the most recent class), and a matrix containing for each individual and at each marker position the simulated class (1 for non-HBD, 2 for most ancient HBD, etc). These elements can be accessed using the simres[[1]], simres[[2]] and simres[[3]], respectively.

Index