

Package ‘diemr’

March 29, 2022

Title Diagnostic Index Expectation Maximisation in R

Version 1.0

Description Likelihood-based genome polarisation finds which alleles of genomic markers belong to which side of the barrier.
Co-estimates which individuals belong to either side of the barrier and barrier strength.
Uses expectation maximisation in likelihood framework.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.1.2

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

Imports zoo

VignetteBuilder knitr

NeedsCompilation no

Author Natalia Martinkova [aut, cre] (<<https://orcid.org/0000-0003-4556-4363>>),
Stuart Baird [aut] (<<https://orcid.org/0000-0002-7144-9919>>)

Maintainer Natalia Martinkova <martinkova@ivb.cz>

Repository CRAN

Date/Publication 2022-03-29 07:50:06 UTC

R topics documented:

CheckDiemFormat	2
diem	3
emPolarise	5
importPolarized	6
ModelOfDiagnostic	7
pHetErrOnStateCount	8
plotPolarized	9
sStateCount	10
Index	11

CheckDiemFormat	<i>Diadem input file checker</i>
-----------------	----------------------------------

Description

Checks format of files with genotype data.

Usage

```
CheckDiemFormat(files, ChosenInds, ploidy)
```

Arguments

files	character vector with paths to files with genotypes.
ChosenInds	numeric vector of indices of individuals to be included in the analysis.
ploidy	list of length equal to length of files. Each element of the list contains a numeric vector with ploidy numbers for all individuals specified in the ChosenInds.

Details

The input file must have genotypes of one marker for all individuals on one line. The line must start with a letter "S" and contain only characters "_" or "U" for unknown genotypes or a third/fourth allele, "0" for homozygots for allele 1, "1" for heterozygots, and "2" for homozygots for allele 2. Check the vignette with `browseVignettes(package = "diemr")` for the example of the input format.

Ploidies must be given as a list with each element corresponding to a genomic compartment (aka a file). For each compartment, the numeric vector specifying ploidies of all individuals chosen for the specific analysis must be given.

Value

Returns invisible TRUE if all files are executable by diem. Exits with informative error messages otherwise, specifying file names and lines with potential problems. When too many lines contain problems, the first six are given.

Examples

```
# set up input genotypes file names, ploidies and selection of individual samples
inputFile <- system.file("extdata", "data6x3.txt", package = "diemr")
ploidies <- list(c(2, 1, 2, 2, 2, 1))
inds <- 1:6

# check input data
CheckDiemFormat(files = inputFile, ploidy = ploidies, ChosenInds = inds)
# File check passed: TRUE
# Ploidy check passed: TRUE
```

diem

*Diagnostic Index Expectation Maximisation***Description**

Estimates how to assign alleles in a genome to maximise the distinction between two unknown groups of individuals. Using expectation maximisation (EM) in likelihood framework, `diem` provides marker polarities for importing data, their likelihood-based diagnostic index and its support for all markers, and hybrid indices for all individuals.

Usage

```
diem(
  files,
  ploidy = list(2),
  markerPolarity = FALSE,
  ChosenInds,
  epsilon = 0.99,
  verbose = FALSE,
  nCores = parallel::detectCores() - 1,
  maxIterations = 50,
  ...
)
```

Arguments

<code>files</code>	character vector with paths to files with genotypes.
<code>ploidy</code>	list of length equal to length of <code>files</code> . Each element of the list contains a numeric vector with ploidy numbers for all individuals specified in the <code>ChosenInds</code> .
<code>markerPolarity</code>	FALSE or list of logical vectors.
<code>ChosenInds</code>	numeric vector of indices of individuals to be included in the analysis.
<code>epsilon</code>	numeric, specifying how much the hypothetical diagnostic markers should contribute to the likelihood calculations. Must be in $[0, 1)$, keeping tolerance setting of the R session in mind.
<code>verbose</code>	logical or character with path to directory where run diagnostics will be saved.
<code>nCores</code>	numeric. Number of cores to be used for parallelisation. Must be <code>nCores = 1</code> on Windows.
<code>maxIterations</code>	numeric.
<code>...</code>	additional arguments.

Details

Given two alleles of a marker, one allele can belong to one side of a barrier to geneflow and the other to the other side. Which allele belongs where is a non-trivial matter. A marker state in an individual can be encoded as 0 if the individual is homozygous for the first allele, and 2 if the

individual is homozygous for the second allele. Marker polarity determines how the marker will be imported. Marker polarity equal to FALSE means that the marker will be imported as-is. A marker with polarity equal to TRUE will be imported with states 0 mapped as 2 and states 2 mapped as 0, in effect switching which allele belongs to which side of a barrier to gene flow.

When `markerPolarity = FALSE`, `diem` uses random null polarities to initiate the EM algorithm. To fix the null polarities, `markerPolarity` must be a list of length equal to the length of the `files` argument, where each element in the list is a logical vector of length equal to the number of markers (rows) in the specific file.

Ploidy needs to be given for each compartment and for each individual. For example, for a dataset of three diploid mammal males consisting of an autosomal compartment, an X chromosome compartment and a Y chromosome compartment, the ploidy list would be `ploidy = list(rep(2, 3), rep(1, 3), rep(1, 3))`. If the dataset consisted of one male and two females, ploidy for the sex chromosomes should be vectors reflecting that females have two X chromosomes, but males only one, and females have no Y chromosomes: `ploidy = list(rep(2, 3), c(1, 2, 2), c(1, 0, 0))`.

Value

A list including suggested marker polarities, diagnostic indices and support for all markers, four-state counts matrix for all individuals, and polarity changes for the EM iterations.

Note

To ensure that the data input format of the genotype files, ploidies and individual selection are readable for `diem`, first use [CheckDiemFormat](#). Fix all errors, and run `diem` only once the checks all passed.

See Also

[CheckDiemFormat](#)

Examples

```
# set up input genotypes file names, ploidies and selection of individual samples
inputFile <- system.file("extdata", "data6x3.txt", package = "diemr")
ploidies <- list(c(2, 1, 2, 2, 1))
inds <- 1:6

# check input data
CheckDiemFormat(files = inputFile, ploidy = ploidies, ChosenInds = inds)
# File check passed: TRUE
# Ploidy check passed: TRUE

# run diem
## Not run:
# diem will store temporal files during EM iterations
# prior to running diem, set the working directory to a location with write permission
fit <- diem(files = inputFile, ChosenInds = inds, ploidy = ploidies, nCores = 1)

# run diem with fixed null polarities
fit2 <- diem(
```

```

files = inputFile, ChosenInds = inds, ploidy = ploidies, nCores = 1,
markerPolarity = list(c(TRUE, FALSE, TRUE))
)

## End(Not run)

```

emPolarise

Polarises a marker

Description

Changes encodings of genomic markers according to user specification.

Usage

```
emPolarise(origM, changePolarity = TRUE)
```

Arguments

`origM` character vector of genotypes comprising of `_012` encodings.
`changePolarity` logical scalar, indicating whether to leave the marker as is (FALSE) or whether to change its polarity (TRUE).

Value

Returns a character vector with polarised markers.

Note

Note that [diem](#) and [importPolarized](#) accept also a U encoding for an unknown or third allele, but `emPolarise` requires all U to be replaced with `_`.

See Also

[diem](#) for determining appropriate marker polarity with respect to a barrier to gene flow.

Examples

```

emPolarise(c("0", "0", "1", "2", "2"), TRUE)
# [1] "2" "2" "1" "0" "0"

emPolarise(c("0", "_", "2", "2", "1"), FALSE)
# [1] "0" "_" "2" "2" "1"

```

importPolarized	<i>Imports genomic data polarized according to the the specification</i>
-----------------	--

Description

Reads genotypes from a file and changes marker polarity.

Usage

```
importPolarized(file, changePolarity, ChosenInds)
```

Arguments

`file` character vector with a single path to a file with genotypes.
`changePolarity` logical vector with length equal to the number of markers.
`ChosenInds` numeric vector of indices of individuals to be included in the analysis.

Details

For details on the input data format, check the file with `CheckDiemFormat`.

The `changePolarity` argument influences how each marker is imported. Value `FALSE` means that the marker will be imported as it is saved in the file. Value `TRUE` means that the genotypes encoded as 0 will be imported as 2, and genotypes encoded in the file as 2 will be imported as 0.

Value

Returns a character matrix with rows containing individual genotypes and columns containing markers.

See Also

[diem](#) for determining appropriate marker polarity with respect to a barrier to gene flow.

Examples

```
dat <- importPolarized(
  file = system.file("extdata", "data6x3.txt", package = "diemr"),
  changePolarity = c(FALSE, TRUE, TRUE),
  ChosenInds = 1:6
)
dat
#   m1 m2 m3
# 1 "0" "1" "2"
# 2 "0" "0" "0"
# 3 "1" "1" "0"
# 4 "1" "2" "0"
# 5 "2" "2" "1"
# 6 "2" "2" "_"
```

ModelOfDiagnostic *Model of Diagnostic Marker Based on All Individual State Counts*

Description

Estimates a diagnostic marker for the state counts of all genomic markers for all individuals. Using the hypothetical, diagnostic marker, calculates individual state counts with respect to their weighted similarity to the diagnostic marker states.

Usage

```
ModelOfDiagnostic(I4, OriginalHI, epsilon = 0.99, verbose = FALSE, ...)
```

Arguments

I4	a matrix or data.frame with 4 numeric columns representing character state counts for missing data, homozygots for allele 1, heterozygots, and homozygots for allele 2. Individuals in rows.
OriginalHI	numeric vector of length equal to number of rows in I4, representing hybrid indices of individuals.
epsilon	numeric, specifying how much the hypothetical diagnostic markers should contribute to the likelihood calculations. Must be in $[0, 1)$, keeping tolerance setting of the R session in mind.
verbose	logical or character with path to directory where run diagnostics will be saved.
...	parameters to be passed to other functions.

Details

The OriginalHI can be calculated with [pHetErrOnStateCount](#).

Value

Matrix with dimensions of I4.

See Also

[diem](#) for utilising the model to determine appropriate marker polarisation in estimating barriers to gene flow.

Examples

```
# state count matrix
dat <- matrix(c(0, 0, 1, 3, 1, 2, 2, 0, 2, 1, 4, 1), ncol = 4)

# hybrid index calculation, assuming diploid markers
HI <- apply(dat * 2, MARGIN = 1, FUN = pHetErrOnStateCount)[1, ]
```

```
# run model of diagnostics, with the weight of the ideal diagnostic marker being 0.8
ModelOfDiagnostic(I4 = dat, OriginalHI = HI, epsilon = 0.8)
#      [,1] [,2] [,3] [,4]
# [1,] 0.0  5.4  0.4  0.2
# [2,] 0.0  0.2  0.0  5.6
# [3,] 0.4  4.4  0.8  0.4
```

`pHetErrOnStateCount` *Hybrid index, heterozygosity, error rate*

Description

Using genotype allele counts, calculates the hybrid index, heterozygosity and error rate in a single individual.

Usage

```
pHetErrOnStateCount(sCount)
```

Arguments

`sCount` a numeric vector of length 4 with allele counts for missing data, homozygots for allele 1, heterozygots, and homozygots for allele 2.

Details

Allele counts are genomic state counts multiplied by ploidy. As different compartments might have different ploidies (e.g. autosomal markers, sex chromosomes, mitochondrial markers), allele counts should be calculated per compartment and then summarised to obtain the correct genomic allele counts. When all individuals in each compartment have the same ploidy, state counts do not need to be corrected.

Value

Returns a named numeric vector with three values: `p` - hybrid index, `Het` - heterozygosity, `Err` - error rate.

Examples

```
pHetErrOnStateCount(sCount = c(2, 4, 2, 6))
#      p      Het      Err
# 0.5833333 0.1666667 0.1428571
```

plotPolarized *Plots polarized genotypes*

Description

Plots genotypes that can be optionally polarized.

Usage

```
plotPolarized(genotypes, HI, ...)
```

Arguments

genotypes	character matrix comprising of _012 encodings.
HI	numeric vector of individual hybrid indices with length equal to number of rows in genotypes.
...	additional arguments.

Details

To import and polarize genotypes, use the function [importPolarized](#).

When using [diem](#) with argument `verbose = TRUE`, hybrid indices, HI, can be found in file 'HIwith-OptimalPolarities.txt' in folder 'diagnostics' in the working directory.

Value

No return value, called for side effects. In the plot, red and blue represent side of the barrier to geneflow encoded as 0 and 2, respectively, yellow shows heterozygots and white missing or undetermined genotypes. Individuals are ordered according to the HI.

Examples

```
gen <- importPolarized(  
  file = system.file("extdata", "data7x10.txt", package = "diemr"),  
  changePolarity = c(TRUE, FALSE, TRUE, TRUE, FALSE, FALSE, TRUE, FALSE, FALSE, TRUE),  
  ChosenInds = 1:7  
)  
  
h <- c(0.625, 0.5, 0.455, 0.455, 0.227, 0.818, 0.292)  
  
plotPolarized(genotypes = gen, HI = h)
```

sStateCount	<i>Count states in a vector</i>
-------------	---------------------------------

Description

Counts genomic states in one sample.

Usage

```
sStateCount(s)
```

Arguments

s character vector with elements "_", "0", "1", "2" representing missing data, homozygots for allele 1, heterozygots, and homozygots for allele 2. The vector should represent a single individual.

Details

Summarizes the number of markers that are fixed for an allele in the genome of one individual. This is used to construct the I4 matrix in [diem](#).

Value

Numeric vector of length 4 with counts of "_", "0", "1", "2" respectively.

See Also

[emPolarise](#) for changing marker polarity.

Examples

```
genotype <- c("0", "0", "_", "2", "1", "0", "1")
sStateCount(genotype)
# [1] 1 3 2 1

# calculate state counts for a polarised genotype
sStateCount(emPolarise(genotype, TRUE))
# [1] 1 1 2 3
```

Index

CheckDiemFormat, [2](#), [4](#)

diem, [3](#), [5–7](#), [9](#), [10](#)

emPolarise, [5](#), [10](#)

importPolarized, [5](#), [6](#), [9](#)

ModelOfDiagnostic, [7](#)

pHetErrOnStateCount, [7](#), [8](#)

plotPolarized, [9](#)

sStateCount, [10](#)