Package 'diemr'

July 16, 2024

Title Diagnostic Index Expectation Maximisation in R

Version 1.4

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. The method is described in Baird et al. (2023) <doi:10.1111/2041-210X.14010>.

BugReports https://github.com/StuartJEBaird/diem/issues

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.1

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

Imports zoo, vcfR

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 2024-07-16 09:50:05 UTC

Contents

`brenthis` *Dataset of butterfly genotypes*

Description

A subset of single nucleotide polymorphisms in butterflies of the genus *Brenthis*.

Format

vcf file with 13 individuals and 4 markers.

Details

The data is used to test conversion of genotype data from vcf to diem format with function `vcf2diem`.

Examples

```
filename <- system.file("extdata", "brenthis.vcf", package = "diemr")
```

`CheckDiemFormat` *diem input file checker*

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`	logical or 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 `files`.

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", "data7x3.txt", package = "diemr")
ploidies <- list(c(2, 1, 2, 2, 2, 1, 2))
inds <- 1:7

# 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 = FALSE,
  markerPolarity = FALSE,
  ChosenInds,
  ChosenSites = "all",
  epsilon = 0.99999,
  verbose = FALSE,
  nCores = parallel::detectCores() - 1,
  maxIterations = 50,
  ...
)
```

Arguments

`files` character vector with paths to files with genotypes.

`ploidy` logical or 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 `files`.

`markerPolarity` `FALSE` or list of logical vectors.

`ChosenInds` numeric vector of indices of individuals to be included in the analysis.

`ChosenSites` logical vector indicating which sites are to be included in the analysis.

`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.

`nCores` numeric. Number of cores to be used for parallelisation. Must be at most equal to the number of files in the `files` argument, and `nCores = 1` on Windows.

`maxIterations` numeric.

`...` 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 geneflow.

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))`.

When `verbose = TRUE`, `diem` will output multiple files with information on the iterations of the EM algorithm, including tracking marker polarities and the respective likelihood-based diagnostics. See vignette `vignette("Understanding-genome-polarisation-output-files", package = "diemr")` for a detailed explanation of the individual output files.

Value

A list including suggested marker polarities, diagnostic indices and support for all markers, four genomic 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.

The working directory or a folder optionally specified in the `verbose` argument must have write permissions. `diem` will store temporary files in the location and output results files.

See Also

CheckDiemFormat

Examples

```
# set up input genotypes file names, ploidies and selection of individual samples
inputFile <- system.file("extdata", "data7x3.txt", package = "diemr")
ploidies <- list(c(2, 1, 2, 2, 2, 1, 2))
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 write 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 geneflow.

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` *Imports genomic data polarized according to the specification*

Description

Reads genotypes from a file and changes marker polarity.

Usage

```
importPolarized(
  files,
  changePolarity,
  ChosenInds,
  ChosenSites = "all",
  nCores = 1,
  verbose = FALSE,
  ...
)
```

Arguments

files character vector with paths to files 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.

ChosenSites logical vector indicating which sites are to be included in the analysis.

nCores numeric. Number of cores to be used for parallelisation. Must be at most equal to the number of files in the files argument, and nCores = 1 on Windows.

verbose logical whether to show messages on import progress.

... optional numeric vector of compartmentSizes.

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 geneflow.

Examples

```
dat <- importPolarized(
  files = system.file("extdata", "data7x3.txt", package = "diemr"),
  changePolarity = c(FALSE, TRUE, TRUE),
  ChosenInds = 1:6,
  ChosenSites = "all"
)
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,
  folder = "likelihood",
  ...
)
```

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.

`folder` character specifying path to a folder for the verbose output.

`...` additional arguments.

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 geneflow.

myotis *Dataset of modified genotypes of bats*

Description

A subset of single nucleotide polymorphisms in *Myotis myotis* from Harazim et al. (2021). The genotypes were modified for testing purposes in such a way that markers 15 and 17 now include additional indel and substitution alleles. Eight markers used in the dataset are monomorphic.

Format

vcf file with 14 individuals and 20 markers.

Details

The data is used to test conversion of genotype data from vcf to diem format with function `vcf2diem`.

Source

Harazim M., Pialek L., Pikula J., Seidlova V., Zukal J., Bachorec E., Bartonicka T., Kokurewicz T., Martinkova N. (2021) Associating physiological functions with genomic variability in hibernating bats. *Evolutionary Ecology*, 35, 291-308, doi: 10.1007/s10682-020-10096-4.

Examples

```
filename <- system.file("extdata", "myotis.vcf", package = "diemr")
```

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 compartmenst 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
```

---

plotDeFinetti *Plot the De Finetti Diagram for Polarized Genotypes*

Description

This function calculates genotype frequencies from polarized genotypes, ideally imported using the `importPolarized` function. It plots individuals onto a ternary De Finetti diagram and includes a curve indicating Hardy-Weinberg equilibrium if specified.

Usage

```
plotDeFinetti(
  genotypes,
  cols,
  HWE = TRUE,
  tipLabels = c("Homozygous 0", "Heterozygous 1", "Homozygous 2"),
  verbose = FALSE,
  ...
)
```

Arguments

`genotypes`	character matrix comprising of _012 encodings.
`cols`	character, vector of colors with a length equal to the number of individuals (rows) in `genotypes`.
`HWE`	logical indicating whether to plot the curve for Hardy-Weinberg Equilibrium.
`tipLabels`	character vector of length 3 with names for the ternary plot vertices.
`verbose`	logical whether to show messages on import progress.
`...`	additional graphical parameters (see plot.default).

Details

To import and polarize genotypes, use the importPolarized function.

Value

No return value; the function is called for its side effects.

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
)

plotDeFinetti(genotypes = gen, cols = palette.colors(nrow(gen), "Accent"), pch = 19)
```

`plotMarkerAxis` *Add a Marker Axis with Chromosome Names to a Plot of Polarized Genotypes*

Description

This function adds a marker axis with chromosome names to an existing plot of polarized genotypes. It requires that the plot is already created using plotPolarized.

Usage

```
plotMarkerAxis(
  includedSites,
  ChosenSites,
  tickDist = 1e+06,
  axisInfo = NULL,
  ...
)
```

Arguments

`includedSites`	character. Path to a file with columns `CHROM` and `POS`.
`ChosenSites`	logical vector indicating which sites are to be included in the analysis.
`tickDist`	numeric. Indicates the spacing of physical tick marks along a chromosome.
`axisInfo`	list with user-defined tick positions and labels for marker axis. See Details.
`...`	additional arguments.

Details

The includedSites file should ideally be generated by vcf2diem to ensure congruence between the plotted genotypes and the respective metadata.

Tick mark distances within a chromosome are located at tickDist and formated to multiples of millions.

The optional axisInfo argument must have five named elements with the following information:

- CHROMbreaks: Numeric vector with positions defining ticks separating chromosomes. The metric for all positions is in the number of markers.
- CHROMnamesPos: Numeric vector with positions to place the chromosome labels.
- CHROMnames: Character vector with the names of the chromosomes. Must be the same length as CHROMnamesPos.
- ticksPos: Numeric vector with positions of ticks within chromosomes.
- ticksNames: Character vector with the names to be displayed at ticksPos.

When axisInfo = NULL, the function extracts the necessary information from the includedSites file.

Value

Returns an invisible axisInfo list with the tick positions and labels for the marker axis.

Examples

```
## Not run:
# Run this example in a working directory with write permissions
myo <- system.file("extdata", "myotis.vcf", package = "diemr")
vcf2diem(myo, "myo")
inds <- 1:14
fit <- diem("myo-001.txt", ChosenInds = inds, ploidy = FALSE)
gen <- importPolarized("myo-001.txt", fit$markerPolarity, inds)
h <- apply(gen, 1, function(x) pHetErrOnStateCount(sStateCount(x)))[1, ]
plotPolarized(gen, h, xlab = "")
plotMarkerAxis("myo-includedSites.txt", rep(TRUE, 11), tickDist = 100)

## End(Not run)
```

---

plotPolarized *Plot Polarized Genotypes*

Description

Plots genotypes that can be optionally polarized.

Usage

```
plotPolarized(
  genotypes,
  HI,
  cols = c("#FFFFFF", "#800080", "#FFE500", "#008080"),
  ...
)
```

Arguments

`genotypes` character matrix comprising of _012 encodings.

`HI` numeric vector of individual hybrid indices with length equal to number of rows in `genotypes`.

`cols` vector of four colors, representing missing data, homozygotes for genotype 0, heterozygotes and homozygotes for genotype 2.

`...` additional selected arguments passed to image and axis.

Details

To import and polarize genotypes, use the importPolarized function.

When using diem, hybrid indices, `HI`, can be found in the file 'HIwithOptimalPolarities.txt'. Alternatively, calculate `HI` from the polarized genotypes as shown in the examples.

By default, the function plots colored tick marks for individuals, changing the color at the steepest change in sorted `HI`. The second and fourth colors in `cols` are used for the tick marks.

- To turn off this feature, use the argument `tick = FALSE`.
- To use custom tick mark colors, provide a vector of colors for all individuals (equal to the number of rows in `genotypes`). The **vector of colors must be ordered** according to `order(HI)`.
- To include individual `labels` (e.g., accession numbers), provide a character vector with the **names in the same order as they are** in the `genotypes`.

Value

No return value, called for side effects. In the default plot, purple and green represent sides of the barrier to gene flow encoded as `0` and `2`, respectively, yellow shows heterozygotes and white represents missing or undetermined genotypes. Individuals are ordered according to the sorted `HI`.

See Also

plotMarkerAxis to add chromosome information to the x axis.

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 <- apply(gen, 1, FUN = function(x) pHetErrOnStateCount(sStateCount(x)))[1, ]

plotPolarized(genotypes = gen, HI = h)

# Incorrect tick color order
plotPolarized(gen, h, col.ticks = c(rep("purple", 5), "green", "purple"), lwd = 3)

# Correct tick color order
plotPolarized(gen, h, col.ticks = c(rep("purple", 5), "green", "purple")[order(h)], lwd = 3)

# Correct individual label order
plotPolarized(gen, h, labels = c(paste("purple", 1:5), "green 1", "purple 6"), ylab = "")
```

`sStateCount` *Count states in a vector*

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
```

testdata *Dataset of fish genotypes*

Description

A subset of single nucleotide polymorphisms in fish for testing purposes of multiallelic markers.

Format

vcf file with 92 individuals and 6 markers.

Details

The data is used to test conversion of genotype data from vcf to diem format with the function `vcf2diem`.

Examples

```
filename <- system.file("extdata", "testdata.vcf", package = "diemr")
```

vcf2diem *Convert vcf files to diem format*

Description

Reads vcf files and writes genotypes of the most frequent alleles based on chromosome positions to diem format.

Usage

```
vcf2diem(SNP, filename, chunk = 1L, requireHomozygous = TRUE)
```

Arguments

`SNP` character vector with a path to the '.vcf' or '.vcf.gz' file, or an `vcfR` object. Diploid data are currently supported.

`filename` character vector with a path where to save the converted genotypes.

`chunk` numeric indicating by how many markers should the result be split into separate files.

`requireHomozygous` logical whether to require the marker to have at least one homozygous individual for each allele.

Details

Importing vcf files larger than 1GB, and those containing multiallelic genotypes is not recommended. Instead, use the path to the vcf file in `SNP`. `vcf2diem` then reads the file line by line, which is a preferred solution for data conversion, especially for very large and complex genomic datasets.

The number of files `vcf2diem` creates depends on the `chunk` argument and class of the `SNP` object.

- Values of `chunk < 100` are interpreted as the number of files into which to split data in `SNP`. For `SNP` object of class `vcfR`, the number of markers per file is calculated from the dimensions of `SNP`. When class of `SNP` is `character`, the number of markers per file is approximated from a model with a message. If this number of markers per file is inappropriate for the expected output, provide the intended number of markers per file in `chunk` greater than 100 (values greater than 10000 are recommended for genomic data). `vcf2diem` will scan the whole input specified in the `SNP` file, creating additional output files until the last line in `SNP` is reached.
- Values of `chunk >= 100` mean that each output file in diem format will contain `chunk` number of lines with the data in `SNP`.

When the vcf file contains markers not informative for genome polarisation, those are removed and listed in a file ending with *omittedSites.txt* in the directory specified in the `SNP` argument or in the working directory. The omitted loci are identified by their information in the CHROM and POS columns, and include the QUAL column data. The last column is an integer specifying the reason why the respective marker was omitted. The reasons why markers are not informative for genome polarisation using `diem` are:

1. Marker has fewer than 2 alleles representing substitutions.
2. Required homozygous individuals for the 2 most frequent alleles are not present (optional, controlled by the `requireHomozygous` argument).
3. The second most frequent allele is found only in one heterozygous individual.
4. Dataset is invariant for the most frequent allele.
5. Dataset is invariant for the allele listed as the first ALT in the vcf input.

The CHROM, POS, and QUAL information for loci included in the converted files are listed in the file ending with *includedSites.txt*. Additional columns show which allele is encoded as 0 in its homozygous state and which is encoded as 2.

Value

No value returned, called for side effects.

Author(s)

Natalia Martinkova

Filip Jagos 521160@mail.muni.cz

Jachym Postulka 506194@mail.muni.cz

Examples

```
## Not run:
# vcf2diem will write files to a working directory or a specified folder
# make sure the working directory or the folder are at a location with write permission
myofile <- system.file("extdata", "myotis.vcf", package = "diemr")

vcf2diem(SNP = myofile, filename = "test1")
vcf2diem(SNP = myofile, filename = "test2", chunk = 3)

## End(Not run)
```

Index