# Package 'diemr'

April 28, 2024

## R topics documented:

1

---

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 | 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` pro-
vides 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 = 0.99999,
  verbose = FALSE,
  nCores = parallel::detectCores() - 1,
  maxIterations = 50,
  ...
)
```

## Arguments

| | |
|---|---|
| files | character vector with paths to files with genotypes. |
| ploidy | list of length equal to length of `files`. Each element of the list contains a nu-meric vector with ploidy numbers for all individuals specified in the `ChosenInds`. |
| markerPolarity | FALSE or list of logical vectors. |
| ChosenInds | numeric vector of indices of individuals to be included in the analysis. |
| epsilon | numeric, specifying how much the hypothetical diagnostic markers should con-tribute to the likelihood calculations. Must be in [0,1), keeping tolerance set-ting 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 `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))`.

## 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.

## 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, 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 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(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 geneflow.

## 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 geneflow.

**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
```

---

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
```

---

plotPolarized        *Plots 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 colours, representing missing data, homozygots for genotype 0, heterozygots and homozygots for genotype 2. |
| ... | 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 default plot, purple and green 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)
```

## 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
```

---

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. chunk = 1 saves all markers into one file. |
| requireHomozygous | |
| | logical whether to require the marker to have at least one homozygous individual for each allele. |
| ... | additional arguments. |

### Details

Importing vcf files larger than 1GB, and those containing multiallelic genotypes is not recom-
mended. Instead, the path to the vcf file in SNP reads the file line by line, and might be a solution
for very large and complex genomic datasets.

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

  * When \code{chunk = 1}, one output file will be created.
 * Values of \code{chunk < 100} are interpreted as the number of files into which to
 split data in \code{SNP}. For \code{SNP} object of class \code{vcfR}, the number
 of markers per file is calculated from the dimensions of \code{SNP}. When class
 of \code{SNP} is \code{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 \code{chunk} greater
 than 100. \code{vcf2diem} will scan the whole input in the \code{SNP} file, creating
 additional output files until the last line in \code{SNP} is reached.
  * Values of \code{chunk >= 100} mean that each output file
 in diem format will contain \code{chunk} number of lines with the data in \code{SNP}.
```

When the vcf file contains markers non-informative for genome polarisation, those those are removed and listed in a file \*omittedLoci.txt\* in the working directory. The omitted loci are identified by their information in the CHROM and POS columns. The CHROM and POS information for loci included in the converted files are in \*includedLoci.txt\*.

**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")
myovcf <- vcfR::read.vcfR(myofile)

vcf2diem(SNP = myofile, filename = "test1")
vcf2diem(SNP = myofile, filename = "test2", chunk = 3)
vcf2diem(SNP = myovcf, filename = "test3")
vcf2diem(SNP = myovcf, filename = "test4", chunk = 3)

## End(Not run)
```

# Index