Package ‘APIS’

August 19, 2019

Type Package

Title Auto-Adaptive Parentage Inference Software Tolerant to Missing Parents

Version 0.1.0

Author Ronan Griot, François Allal, Romain Morvezen, Sophie Brard-Fudulea, Pierrick Haffray, Florence Phocas and Marc Vandeputte

Maintainer Ronan Griot <ronan.griot@gmail.com>

Description Parentage assignment package.

Parentage assignment is performed based on observed average Mendelian transmission probability distributions.

The main function of this package is the function APIS(), which is the parentage assignment function.

License GPL

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Depends R (>= 3.4.0)

Imports foreach, parallel, doParallel, ggplot2, gridExtra

NeedsCompilation yes

Suggests knitr, rmarkdown

VignetteBuilder knitr

Repository CRAN

Date/Publication 2019-08-19 09:50:05 UTC

R topics documented:

allFreq ................................................................. 2
APIS ................................................................. 2
APIS_dam ............................................................ 3
APIS_offspring ..................................................... 4
allFreq

Estimate the allele frequencies

Description
This function estimates allele frequencies

Usage

allFreq(genotype)

Arguments

genotype A matrix of genotypes (n*p) n = number of individuals p = number of markers (coded as "All1/All2", ex: "A/A" or "NA/NA" for missing genotype)

Value

allele frequencies

Examples

data("APIS_offspring")
freq <- allFreq(APIS_offspring)

APIIS

APIS function that assigns with observed data

Description
This function performs the APIS procedure

Usage

APIIS(off.genotype, sire.genotype, dam.genotype, error = 0,
exclusion.threshold = ncol(off.genotype), preselect.Parent = FALSE,
 nb.cores = 2, verbose = TRUE)
Arguments

off.genotype  Offspring genotypes | Matrix (n*p) where n = number of individuals p = number of markers rownames(offspring) = labels of offspring marker coding = "All1/All2" example: "A/A", "A/B", "NA/NA" (for missing genotype)

sire.genotype  Sire genotypes | Matrix (n*p) where n = number of individuals p = number of markers rownames(sire) = labels of sires marker coding = "All1/All2" example: "A/A", "A/B", "NA/NA" (for missing genotype)

dam.genotype  Dam genotypes | Matrix (n*p) where n = number of individuals p = number of markers rownames(dam) = labels of dams marker coding = "All1/All2" example: "A/A", "A/B", "NA/NA" (for missing genotype)

error (default: 0) The assignment error rate accepted by the user

exclusion.threshold (default: ncol(off.genotype)) Threshold for exclusion (number of mismatches allowed)

preselect.Parent (default: FALSE) Preselection of parents. Can be FALSE, an integer or a vector of two integers (number of sires, numbers of dams)

nb.cores (default: 2) Number of cores to use. If you have more than 2 cores, you can use the "parallel" function detectCores()

verbose (default : TRUE) Display the process of the function on the console.

Value

pedigree

a log file

Examples

data("APIS_offspring")
data("APIS_sire")
data("APIS_dam")

result <- APIS(off.genotype = APIS_offspring[1:50, ],
sire.genotype = APIS_sire,
dam.genotype = APIS_dam,
error = 0.05,
verbose = FALSE)

---

Example dam genotypes

Description

Example dam genotypes
**Usage**  
APIS_sire

**Format**  
A matrix with 39 rows (one row = one sire) and 100 columns (one column = one marker)

---

**Example sire genotypes**

---

**Usage**  
APIS_offspring

**Format**  
A matrix with 1068 rows (one row = one offspring) and 100 columns (one column = one marker)

---

**Example offspring genotypes**

---

**Usage**  
APIS_dam

**Format**  
A matrix with 14 rows (one row = one dam) and 100 columns (one column = one marker)
### assignmentFortran

**Assignment function to obtain the average Mendelian transmission probabilities using a Fortran library**

#### Description

This function calculates the average Mendelian transmission probabilities.

#### Usage

```r
assignmentFortran(offspring, sire, dam, thresh = ncol(offspring),
                   preselect.Parent = FALSE, nb.cores = 2, verbose = TRUE)
```

#### Arguments

- **offspring**: Offspring genotypes | Matrix (n*p) where n = number of individuals, p = number of markers rownames(offspring) = labels of offspring marker coding = "All1/All2" example: "A/A", "A/B", "NA/NA" (for missing genotype)
- **sire**: Sire genotypes | Matrix (n*p) where n = number of individuals, p = number of markers rownames(sire) = labels of sires marker coding = "All1/All2" example: "A/A", "A/B", "NA/NA" (for missing genotype)
- **dam**: Dam genotypes | Matrix (n*p) where n = number of individuals, p = number of markers rownames(dam) = labels of dams marker coding = "All1/All2" example: "A/A", "A/B", "NA/NA" (for missing genotype)
- **thresh**: (default: ncol(offspring)) Threshold for exclusion (number of mismatches allowed)
- **preselect.Parent**: (default: FALSE) Preselection of parents. Can be FALSE, an integer or a vector of two integers (number of sires, numbers of dams)
- **nb.cores**: (default: 2) Number of cores to use. If you have more than 2 cores, you can use the "parallel" function detectCores()
- **verbose**: (default: TRUE) Display the process of the function on the console.

#### Value

- intermediate pedigree
- log file for Mendelian transmission probabilities
- log file for exclusion

#### Examples

```r
data("APIS_offspring")
data("APIS_sire")
data("APIS_dam")

assignment <- assignmentFortran(APIS_offspring[1:50, ], APIS_sire, APIS_dam, verbose = FALSE)
```
**assignmentPower**  
*calculate the theoretical assignment power*

**Description**  
This function calculates the theoretical assignment power of the marker set.

**Usage**  
```r
assignmentPower(sire, dam)
```

**Arguments**

- `sire`  
  Sire genotypes | Matrix (n*p) where n = number of individuals, p = number of markers  
  rownames(sire) = labels of sires  
  marker coding = "All1/All2" example: "A/A", "A/B", "NA/NA" (for missing genotype)

- `dam`  
  Dam genotypes | Matrix (n*p) where n = number of individuals, p = number of markers  
  rownames(dam) = labels of dams  
  marker coding = "All1/All2" example: "A/A", "A/B", "NA/NA" (for missing genotype)

**Value**  
Theoretical assignment power of the marker set.

**Examples**
```r
data("APIS_sire")
data("APIS_dam")
assignmentPower(APIS_sire, APIS_dam)
```

---

**personalThreshold**  
*Establish personal threshold*

**Description**  
This function allows the user to set up his own threshold.

**Usage**  
```r
personalThreshold(APIS.result, method, threshold = NULL, verbose = TRUE)
```
selectParents

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APIS.result</td>
<td>API function output</td>
</tr>
<tr>
<td>method</td>
<td>the method for the new threshold</td>
</tr>
<tr>
<td>threshold</td>
<td>personal threshold</td>
</tr>
<tr>
<td>verbose</td>
<td>(default: TRUE) Display the process of the function on the console.</td>
</tr>
</tbody>
</table>

Value

new pedigree from the new threshold

Examples

```r
data("APIS_offspring")
data("APIS_sire")
data("APIS_dam")

result <- APIS(off.genotype = APIS_offspring[1:50, ],
  sire.genotype = APIS_sire,
  dam.genotype = APIS_dam,
  error = 0.05,
  verbose = FALSE)

new.result <- personalThreshold(result, method = "exclusion", threshold = 2, verbose = FALSE)
```

selectParents

Select most likely parents for potent parent pairs tests

Description

This function allows the selection of the most likely parents for assignment, reducing computation time

Usage

```r
selectParents(off.genotype, parent.genotype, parent.sex, n.Parent)
```

Arguments

- `off.genotype` genotype of one offspring
- `parent.genotype` genotype matrix of parent genotypes
- `parent.sex` vector of parents sex
- `n.Parent` vector of number of sires and dams to select
Value

list of potential sires and dams

---

**setThreshold**

*Set the APIS threshold*

**Description**

This function calculates the threshold for APIS

**Usage**

```r
setThreshold(ped.log, ped.exclu, nb.mrk, error = NULL, verbose = TRUE)
```

**Arguments**

- `ped.log` log like from assignment function
- `ped.exclu` log exclu from assignment function
- `nb.mrk` Number of markers
- `error` (default: NULL) The assignment error rate accepted by the user
- `verbose` (default: TRUE) Display the process of the function on the console.

**Value**

- `pedigree`
- `log file`
Index

+Topic **APIs**
  - APIs, 2
  - personalThreshold, 6
  - selectParents, 7
+Topic **allele**
  - allFreq, 2
+Topic **assignment**
  - APIs, 2
    - assignmentFortran, 5
    - assignmentPower, 6
    - personalThreshold, 6
    - selectParents, 7
    - setThreshold, 8
+Topic **datasets**
  - APIS_dam, 3
  - APIS_offspring, 4
  - APIS_sire, 4
+Topic **exclusion**
  - assignmentPower, 6
+Topic **frequencies**
  - allFreq, 2
+Topic **power**
  - assignmentPower, 6
+Topic **threshold**
  - personalThreshold, 6
  - selectParents, 7

  allFreq, 2
  APIS, 2
  APIS_dam, 3
  APIS_offspring, 4
  APIS_sire, 4
  assignmentFortran, 5
  assignmentPower, 6

  personalThreshold, 6
  selectParents, 7
  setThreshold, 8