

Package: acoRn (via r-universe)

November 3, 2024

Title Exclusion-Based Parentage Assignment Using Multilocus Genotype Data

Version 0.1.0

Description Exclusion-based parentage assignment is essential for studies in biodiversity conservation and breeding programs - Kang Huang, Rui Mi, Derek W Dunn, Tongcheng Wang, Baoguo Li, (2018), <[doi:10.1534/genetics.118.301592](https://doi.org/10.1534/genetics.118.301592)>. The tool compares multilocus genotype data of potential parents and offspring, identifying likely parentage relationships while accounting for genotyping errors, missing data, and duplicate genotypes. 'acoRn' includes two algorithms: one generates synthetic genotype data based on user-defined parameters, while the other analyzes existing genotype data to identify parentage patterns. The package is versatile, applicable to diverse organisms, and offers clear visual outputs, making it a valuable resource for researchers.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports data.table, stringr, stringi

Depends R (>= 2.10)

LazyData true

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository <https://npechl.r-universe.dev>

RemoteUrl <https://github.com/npechl/acorn>

RemoteRef HEAD

RemoteSha 21c7fd3fbdc711aacba9573c5138f9fcf61ad77f

Contents

acoRn	2
clean_input	2
create_mock_parents	3
create_mock_progeny	3
exclude_duplicates	4
find_parents	4
identify_duplicates	5
offspring	5
parents	6

Index	7
--------------	----------

acoRn	<i>acoRn workflow</i>
-------	-----------------------

Description

acoRn workflow

Usage

acoRn(adults, progeny)

Arguments

adults	a data.frame
progeny	a data.frame

Value

a data.frame

clean_input	<i>Title</i>
-------------	--------------

Description

Title

Usage

clean_input(genotypes)

Arguments

genotypes data.table

Value

data.table

create_mock_parents *Title*

Description

Title

Usage

```
create_mock_parents(nmarkers = 10, ntrees = 100, nvariants = 4, maf = NULL)
```

Arguments

nmarkers number of markers
 ntrees number of trees
 nvariants number of trees
 maf minimum allele frequency

Value

a list

create_mock_progeny *Title*

Description

Title

Usage

```
create_mock_progeny(info, fparents, mparents, prog)
```

Arguments

info mock parents, as generated from create_mock_parents function
 fparents number of female parents
 mparents number of male parents
 prog number of progeny??

Value

a data table

exclude_duplicates	<i>Report duplicates</i>
--------------------	--------------------------

Description

Report duplicates

Usage

```
exclude_duplicates(parents, adults = NULL, progeny = NULL)
```

Arguments

parents	a data.frame
adults	a data.frame
progeny	a data.frame

Value

a data.frame

find_parents	<i>Identify relationships between parents and progenies</i>
--------------	---

Description

Identify relationships between parents and progenies

Usage

```
find_parents(adults, progeny)
```

Arguments

adults	a data.frame containing
progeny	a data.frame

Value

a data.frame

identify_duplicates *Identify duplicates in genotypes (i.e. parents or progenies)*

Description

Identify duplicates in genotypes (i.e. parents or progenies)

Usage

```
identify_duplicates(genotypes, abbr = NULL)
```

Arguments

genotypes	a data.frame with the genotypes
abbr	a string with abbreviation to use

Value

a data.frame

offspring *Tree progeny data set*

Description

An example of tree progeny data set

Usage

```
offspring
```

Format

offspring:
 A data frame with 7,240 rows and 60 columns:
country Country name
iso2, iso3 2 & 3 letter ISO country codes
year Year ...

Source

<https://www.who.int/teams/global-tuberculosis-programme/data>

parents

Tree parents data set

Description

An example of tree parents data set

Usage

parents

Format

parents:

A data frame with 7,240 rows and 60 columns:

country Country name

iso2, iso3 2 & 3 letter ISO country codes

year Year ...

Source

<https://www.who.int/teams/global-tuberculosis-programme/data>

Index

* datasets

offspring, [5](#)
parents, [6](#)

acoRn, [2](#)

clean_input, [2](#)

create_mock_parents, [3](#)

create_mock_progeny, [3](#)

exclude_duplicates, [4](#)

find_parents, [4](#)

identify_duplicates, [5](#)

offspring, [5](#)

parents, [6](#)