# Package 'pedSimulate'

August 18, 2021

**Title** Pedigree, Genetic Merit and Phenotype Simulation

**Version** 1.1.1

**Description** Simulate pedigree, genetic merits and phenotypes with random/non-random matings followed by random/non-random selection with different intensities and patterns in males and females. Bijma, P. & Rutten, M. (2002) <https://www.wur.nl/en/Research-Results/Chair-groups/Animal-Sciences/Animal-Breeding-and-Genomics-Group/Research/Software.htm>.

**License** GPL-3

**LazyData** true

**URL** https://github.com/nilforooshan/pedSimulate

**BugReports** https://github.com/nilforooshan/pedSimulate/issues

**RoxygenNote** 7.1.1

**Encoding** UTF-8

**Repository** CRAN

## R topics documented:



---

`pedSimulte-package` *Pedigree, genetic merit and phenotype simulation*

---

### Description

An R package for simulating a pedigree with genetic merits and phenotypes, starting from a base population (generation 0) or an existing pedigree. The pedigree depth and design can be chosen by the values provided to the arguments of the simulation function.

## Details

Starting from a base population with a user-defined size and equal number of males and females, next generations are simulated for the user-defined littersize and number of generations. No selection (natural or artificial) and non-random mating is applied to this population. Alternatively, the simulation can be started from an existing pedigree. Natural (mortality) and artificial selection are applied to the next generations. Different generation overlap, selection intensities and selection patterns can be applied to males and females. Selected males and females are oredered similarly/diffrently to simulate various random, assortative or disassortative mating scenarios. Performance and genetic merit of individuals are simulated using the basic rules of quantitative genetics. The performance (P) of an individual is influenced by genetic (A) and environmental (E) effects. Thus, P = A + E, and Var(P) = Var(A) + Var(E). The additive genetic merit (A) of an individual is the average of its parents' additive genetic merits (PA = ($A_{sire}$ + $A_{dam}$)/2) plus the Mendelian Sampling term due to the sampling of alleles passed from the parent to the offspring. The Mendelian Sampling variance is half of Var(A) in the base population. Because there is no provided information for environmental effects, the environment effect is assigned to individuals from a normal distribution of random numbers (E ~ N(0, **I**Var(E))). The package also provides functions to identify halfsib, fullsib and parent-progeny matings in the pedigree.

## Author(s)

Mohammad Ali Nilforooshan <m.a.nilforooshan@gmail.com>

---

`appendPed` *Simulate new generations from an existing pedigree*

---

## Description

Simulate pedigree, genetic merits and phenotypes with random/assortative/disassortative matings followed by random/non-random selection of males and females with similar/different selection patterns in males and females, starting from an existing pedigree.

## Usage

```
appendPed(
  ped,
  Va0,
  Ve,
  littersize = 1,
  ngen,
  mort.rate = 0,
  overlap.s = 0,
  overlap.d = 0,
  f.rate = 1,
  m.rate = 1,
  fsel = "R",
  msel = "R",
  f.order = "fsel",
  m.order = "msel"
)
```

## Arguments

| | |
|---|---|
| ped | : The input pedigree data.frame with 9 columns: ID, SIRE, DAM, SEX, GEN (generation), PA (parent average), MS (Mendelian Sampling), E (environment and residuals), and P (phenotype). Note that PA + MS + E = P - $\mu$, where $\mu$ is the population mean, and PA + MS = BV (genetic merit or breeding value). If MS and E are unknown, those can be set to 0. PA should be equal to the average of sire BV (SBV) and dam BV (DBV). If this condition is not met, PA - (SBV + DBV)/2 is added to MS and (SBV + DBV)/2 replaces PA. |
| Va0 | : Additive genetic variance in the base generation (i.e., F0). |
| Ve | : Residual variance, constant across generations. |
| littersize | : Litter size, default = 1. |
| ngen | : Number of generations to simulate after the founder generation. |
| mort.rate | : Mortality rate per generation, after the availability of phenotype (e.g., birth weight, weaning weight) and before the age of maturity (i.e., before mating), default = 0. Maximum mort.rate = 0.5. |
| overlap.s | : Number of overlapping generations for sires, default = 0 for no generation overlap. |
| overlap.d | : Number of overlapping generations for dams, default = 0 for no generation overlap. |
| f.rate | : Proportion of females selected as dams, default = 1. |
| m.rate | : Proportion of males (<= f.rate) selected as sires, default = 1. |
| fsel | : If "R" (default), random selection on females; if "P", selection on phenotypes or true breeding values if Ve = 0; if "PA", selection on true parent averages. "-P" and "-PA" work in opposite direction of "P" and "PA", respectively. |
| msel | : If "R" (default), random selection on males; if "P", selection on phenotypes or true breeding values if Ve = 0; if "PA", selection on true parent averages. "-P" and "-PA" work in opposite direction of "P" and "PA", respectively. |
| f.order | : Ordering selected females for mating; if "fsel" (default), same as the selection order; if "R" random ordering; if "P", ordering based on phenotypes or true breeding values if Ve = 0; if "PA", ordering based on true parent averages. "-P" and "-PA" work in opposite direction of "P" and "PA", respectively. |
| m.order | : Ordering selected males for mating; if "msel" (default), same as the selection order; if "R" random ordering; if "P", ordering based on phenotypes or true breeding values if Ve = 0; if "PA", ordering based on true parent averages. "-P" and "-PA" work in opposite direction of "P" and "PA", respectively. |

## Value

ped2 : The output pedigree data.frame with the same format as the input pedigree data.frame.

## Examples

```
ped = simulatePed(
    F0size = 100,
    Va0 = 9,
    Ve = 36,
    littersize = 2,
    ngen = 4,
    mort.rate = 0.05,
```

```
    overlap.s = 1,
    overlap.d = 0,
    f.rate = 0.8,
    m.rate = 0.5,
    fsel = "P",
    msel = "PA"
)
ped2 = appendPed(
    ped = ped,
    Va0 = 9,
    Ve = 36,
    littersize = 2,
    ngen = 2,
    mort.rate = 0.05,
    overlap.s = 1,
    overlap.d = 0,
    f.rate = 0.8,
    m.rate = 0.5,
    fsel = "R",
    msel = "R",
    f.order = "P",
    m.order = "PA"
)
```

---

`fs_mate_finder` *Find fullsib mates*

---

## Description

Find fullsib matings in the pedigree

## Usage

```
fs_mate_finder(ped)
```

## Arguments

| | |
|---|---|
| `ped` | : A pedigree `data.frame`. The first three columns (ID, SIRE, DAM) are used. |

## Value

fs_mates : A `data.frame` with two columns (SIRE, DAM) representing fullsib mates.

## Examples

```
ped = data.frame(ID=1:7, SIRE=c(0,0,1,0,3,3,5), DAM=c(0,0,0,2,4,4,6))
fs_mate_finder(ped)
```

## hs_mate_finder *Find halfsib mates*

### Description

Find halfsib matings in the pedigree

### Usage

```
hs_mate_finder(ped)
```

### Arguments

| | |
|---|---|
| ped | : A pedigree data.frame. The first three columns (ID, SIRE, DAM) are used. |

### Value

hs_mates : A data.frame with two columns (SIRE, DAM) representing halfsib mates.

### Examples

```
ped = data.frame(ID=1:7, SIRE=c(0,0,1,1,0,3,5), DAM=c(0,0,2,2,2,4,4))
hs_mate_finder(ped)
```

## pp_mate_finder *Find parent-progeny mates*

### Description

Find parent-progeny matings in the pedigree

### Usage

```
pp_mate_finder(ped)
```

### Arguments

| | |
|---|---|
| ped | : A pedigree data.frame. The first three columns (ID, SIRE, DAM) are used. |

### Value

pp_mates : A data.frame with two columns (SIRE, DAM) representing parent-progeny mates.

### Examples

```
ped = data.frame(ID=1:4, SIRE=c(0,0,1,1), DAM=c(0,0,2,3))
pp_mate_finder(ped)
```

`simulatePed` *Simulate pedigree, genetic merits and phenotypes*

## Description

Simulate pedigree, genetic merits and phenotypes with random/assortative/disassortative matings followed by random/non-random selection of males and females with similar/different selection patterns in males and females.

## Usage

```
simulatePed(
  F0size,
  Va0,
  Ve,
  littersize = 1,
  ngen,
  mort.rate = 0,
  overlap.s = 0,
  overlap.d = 0,
  f.rate = 1,
  m.rate = 1,
  fsel = "R",
  msel = "R",
  f.order = "fsel",
  m.order = "msel"
)
```

## Arguments

| | |
|---|---|
| `F0size` | : Even number of founder animals. No mortality, selection and non-random mating in this generation. |
| `Va0` | : Additive genetic variance in the base generation (i.e., F0). |
| `Ve` | : Residual variance, constant across generations. |
| `littersize` | : Litter size, default = 1. |
| `ngen` | : Number of generations to simulate after the founder generation. |
| `mort.rate` | : Mortality rate per generation, after the availability of phenotype (e.g., birth weight, weaning weight) and before the age of maturity (i.e., before mating), default = 0. Maximum `mort.rate` = 0.5. |
| `overlap.s` | : Number of overlapping generations for sires, default = 0 for no generation overlap. |
| `overlap.d` | : Number of overlapping generations for dams, default = 0 for no generation overlap. |
| `f.rate` | : Proportion of females selected as dams, default = 1. |
| `m.rate` | : Proportion of males (<= `f.rate`) selected as sires, default = 1. |
| `fsel` | : If `"R"` (default), random selection on females; if `"P"`, selection on phenotypes or true breeding values if `Ve` = 0; if `"PA"`, selection on true parent averages. `"-P"` and `"-PA"` work in opposite direction of `"P"` and `"PA"`, respectively. |

`msel` : If `"R"` (default), random selection on males; if `"P"`, selection on phenotypes or true breeding values if `Ve` = 0; if `"PA"`, selection on true parent averages. `"-P"` and `"-PA"` work in opposite direction of `"P"` and `"PA"`, respectively.

`f.order` : Ordering selected females for mating; if `"fsel"` (default), same as the selection order; if `"R"` random ordering; if `"P"`, ordering based on phenotypes or true breeding values if `Ve` = 0; if `"PA"`, ordering based on true parent averages. `"-P"` and `"-PA"` work in opposite direction of `"P"` and `"PA"`, respectively.

`m.order` : Ordering selected males for mating; if `"msel"` (default), same as the selection order; if `"R"` random ordering; if `"P"`, ordering based on phenotypes or true breeding values if `Ve` = 0; if `"PA"`, ordering based on true parent averages. `"-P"` and `"-PA"` work in opposite direction of `"P"` and `"PA"`, respectively.

## Details

The output pedigree `data.frame` (`ped`) has 9 columns: ID, SIRE, DAM, SEX, GEN (generation number starting with 0 for the base generation), PA (parent average), MS (Mendelian Sampling), E (environment and residuals), and P (phenotype).

Random, assortative, and disassortative matings can be simulated with different combinations of `fsel`, `msel`, `f.order`, and `m.order`.

## Value

ped : The output pedigree `data.frame`. Further information provided in **Details**.

## Examples

```
ped = simulatePed(
    F0size = 100,
    Va0 = 9,
    Ve = 36,
    littersize = 2,
    ngen = 4,
    mort.rate = 0.05,
    overlap.s = 1,
    overlap.d = 0,
    f.rate = 0.8,
    m.rate = 0.5,
    fsel = "P",
    msel = "PA",
    f.order = "fsel",
    m.order = "msel"
)
```

# Index