

Package ‘AquaBPsim’

October 12, 2022

Title Aquaculture Breeding Program Simulation

Version 0.0.1

Description Breeding programs can be simulated with this package. The functions are written to simulate production and reproduction systems encountered in aquaculture and are easy to combine with custom functions. Simulating breeding programs is useful to predict the expected genetic gain, rate of inbreeding and the effect of changes in the breeding program. AquaBPsim does not simulate genome wide-markers and QTLs, but it simulates estimated breeding values as values correlated to the true breeding values. The correlation equals the accuracy, which can be provided or calculated using deterministic formulas. For genomic selection, the accuracy can be calculate using the formula of Deatwyler et al. (2010) <[doi:10.1534/genetics.110.116855](https://doi.org/10.1534/genetics.110.116855)>. Without genomic selection, accuracy can be calculated with the selection index method (Mrode, 2014. ISBN:978-1-84593-981-6).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

Imports matlab, MASS, pedigree, readxl, stats

Suggests rmarkdown, knitr

VignetteBuilder knitr

NeedsCompilation no

Author Chantal Roozeboom [aut, cre],
Benan Gulzari [ctb],
John W.M. Bastiaansen [ctb]

Maintainer Chantal Roozeboom <chantal.roozeboom@hotmail.nl>

Repository CRAN

Date/Publication 2022-03-01 08:40:02 UTC

R topics documented:

avail_selection	2
breeding_values	4

cor_var	7
deltaG_F	8
founderpopfam	9
founderpopgroup	12
gen_param	16
groupmating	16
offspringFSfam	18
offspringFSgroup	21
preselphen	23
preselrandom	26
preselcand	28
randommating	30
select	32
survive	34

Index	37
--------------	-----------

avail_selection	<i>Available as selection candidates</i>
-----------------	--

Description

Function to determine which fish are available for selection. Fish are randomly chosen.

Usage

```
avail_selection(gen, batch = 0, presel = 0, surv = NA, fish_per_FSfam = NA)
```

Arguments

gen	The generation of the fish.
batch	The batch of the fish. Default is 0. It is possible to provide a vector with multiple batches.
presel	Identifies which preselected fish are available for selection. If no pre-selection took place, then presel should be 0 (= default).
surv	Proportion of fish that is assumed to survive till the moment of selection. Either surv or fish_per_FSfam need to be provided.
fish_per_FSfam	The number of fish available for selection per full sib family. Fish are randomly selected within a full sib family if fish_per_FSfam is specified. Either surv or fish_per_FSfam need to be provided.

Value

This function will make changes to the data frame called 'ped'. Fish that become available as selection candidates will be assigned a 1 to their column 'selcand'.

Examples

```

ped <- founderpopfam(Nm = 60,
                    Nf = 60,
                    Nm2 = 0,
                    Nf2 = 0,
                    Ntraits = 2,
                    TraitsIndex = 2,
                    Rgen = matrix(c(1.00 , 0.48,
                                   0.48 , 1.00),
                                  nrow = 2),
                    Rcom = matrix(c(1.00 , 0.5,
                                   0.5 , 1.00),
                                  nrow = 2),
                    Rres = matrix(c(1.00 , 0.32,
                                   0.32 , 1.00),
                                  nrow = 2),
                    mean = c(50,500),
                    a_var = c(200,8000),
                    c_var = c(150,1000),
                    e_var = c(250,12000))

Mating <- randommating(gen = 0,
                      Nfam_FS = 120)

for(fam in 1: nrow(Mating)){
ped <- offspringFSfam(gen = 1,
                    No = 100,
                    probmale = 0.5,
                    sire = Mating$Sire[fam],
                    dam = Mating$Dam[fam],
                    Ntraits = 2,
                    TraitsIndex = 2,
                    Rgen = matrix(c(1.00 , 0.48,
                                   0.48 , 1.00),
                                  nrow = 2),
                    Rcom = matrix(c(1.00 , 0.5,
                                   0.5 , 1.00),
                                  nrow = 2),
                    Rres = matrix(c(1.00 , 0.32,
                                   0.32 , 1.00),
                                  nrow = 2),
                    a_var = c(200,8000),
                    c_var = c(150,1000),
                    e_var = c(250,12000))
}

ped <- preselectphen(gen = 1,
                    Nenv = 2,
                    Npreselect = c(25,15),
                    trait = 1,
                    Ntraits = 2)

```

```
ped <- avail_selection(gen = 1,
                      presel = 1,
                      surv = 0.9)
```

breeding_values

Simulating estimated breeding values

Description

This function can be used to simulate estimated breeding values (EBV) for the selection candidates.

Usage

```
breeding_values(
  gen,
  batch = 0,
  TraitsIndex = c(1:Ntraits),
  Ntraits = BPdata$Ntraits,
  EBV,
  accuracy = NA,
  GenomLength,
  Ne,
  SizeTraining,
  h2 = BPdata$h2,
  presel_sibs = c(rep(1, times = length(TraitsIndex))),
  c2 = BPdata$c2,
  indexweights = c(1),
  a_var = BPdata$a_var,
  method_indexweights = 1
)
```

Arguments

gen	The generation of the fish for which breeding values need to be simulated.
batch	The batch of the fish for which breeding values need to be simulated, default is 0.
TraitsIndex	A vector of traits that are in the index. For these traits, breeding values will be simulated.
Ntraits	Total number of simulated traits. Does not need to be specified if a list called 'BPdata' including the variable Ntraits is available.
EBV	A vector of methods for simulating EBVs for each trait in the selection index needs to be specified. Options: "pheno" for simulating an EBV that is equal to the phenotype of the trait; "GEBV" for simulating GEBVs as a value correlated to the true breeding value of the animal, this correlation equals the accuracy; "PEBV" for simulating EBVs as a value correlated to the true breeding value

	of the animal (correlation = accuracy) and the prediction errors are correlated to the common environmental effects; "sib_pheno" for simulating EBVs using only information of the sibs of the selection candidate. Accuracies can be specified in the parameter 'accuracy', or in the case of GEBVs, accuracies can be calculated with the formula of Deatwyler et al. (2010) when the effective population size, genome length and size of the training population for each trait are specified.
accuracy	Vector of accuracies with the length equal to the number of traits in the Index. For traits for which the accuracy is not needed or traits for which the accuracy needs to be calculated, NA needs to be specified in the vector.
GenomLength	Genome length in Morgan.
Ne	Effective population size.
SizeTraining	Vector with the size of the training population for each trait in the index.
h2	Vector with heritabilities. Does not need to be specified when the heritabilities are already provided in the list 'BPdata'.
presel_sibs	Vector to indicate which preselected fish can be used to calculate the EBV when EBV = "sibs_pheno". Needs to be specified for each trait in index (if not applicable, then NA can be specified).
c2	Vector with common environmental effects. Does not need to be specified when the common environmental effects are already provided in the list 'BPdata'.
indexweights	If traits need to be combined in an index, then desired gain indices need to be specified for each trait in the index.
a_var	Vector of genetic variances of all traits, does not need to be specified if already provided in the list 'BPdata'.
method_indexweights	Either 1 or 2, default is 1, see Details.

Details

Breeding values will be simulated only for fish with `ped$selcand == 1`. EBVs are not estimated but simulated as a value correlated to the true breeding value. The correlation is equal to the accuracy, which can be calculated or provided by the user.

There are three options for simulating the EBVs, namely:

- "pheno": EBV equal to phenotype.
- "PEBV": Pedigree estimated breeding values: either an accuracy needs to be provided or the accuracy will be calculated.
- "GEBV": Genomically estimated breeding values. Either an accuracy needs to be provided or the genome length and effective population size need to be provided.
- "sib_pheno": EBVs are calculated from the phenotypes of the full sibs and half sibs. If this option is chosen, the accuracy will be calculated.

If EBVs are simulated for more than one trait, then the EBVs are combined in an index. Combining the traits in an index using the desired gains can be done in two ways:

- Method 1 (default): For each trait, the desired gains are divided by the genetic standard deviations of the traits and then multiplied by the accuracy.
- Method 2: The EBVs are first standardized to a mean of 100 and a standard deviation of 10 and then multiplied by the desired gain.

Value

This function will change the data frame called 'ped'. Simulated breeding values will be added to the EBV columns and values will be added to the column 'Index'.

Examples

```
ped <- founderpopfam(Nm = 60,
  Nf = 60,
  Nm2 = 0,
  Nf2 = 0,
  Ntraits = 2,
  TraitsIndex = 2,
  Rgen = matrix(c(1.00 , 0.48,
    0.48 , 1.00),
    nrow = 2),
  Rcom = matrix(c(1.00 , 0.5,
    0.5 , 1.00),
    nrow = 2),
  Rres = matrix(c(1.00 , 0.32,
    0.32 , 1.00),
    nrow = 2),
  mean = c(50,500),
  a_var = c(200,8000),
  c_var = c(150,1000),
  e_var = c(250,12000))

Mating <- randommating(gen = 0,
  Nfam_FS = 120)

for(fam in 1: nrow(Mating)){
  ped <- offspringFSfam(gen = 1,
    No = 100,
    probmale = 0.5,
    sire = Mating$Sire[fam],
    dam = Mating$Dam[fam],
    Ntraits = 2,
    TraitsIndex = 2,
    Rgen = matrix(c(1.00 , 0.48,
      0.48 , 1.00),
      nrow = 2),
    Rcom = matrix(c(1.00 , 0.5,
      0.5 , 1.00),
      nrow = 2),
    Rres = matrix(c(1.00 , 0.32,
      0.32 , 1.00),
      nrow = 2),
    a_var = c(200,8000),
    c_var = c(150,1000),
    e_var = c(250,12000))
}
```

```
ped <- preselectphen(gen = 1,
                    Nenv = 2,
                    Npreselect = c(25,15),
                    trait = 1,
                    Ntraits = 2)

ped <- avail_selection(gen = 1,
                    preselect = 1,
                    surv = 0.9)

ped <- breeding_values(gen = 1,
                    TraitsIndex = 2,
                    EBV = "GEBV",
                    GenomLength = 11.3,
                    Ne = 100,
                    SizeTraining = nrow(ped[ped$preselected ==2,]),
                    Ntraits = 2,
                    a_var = c(200,8000),
                    h2 = c(0.33,0.38))
```

cor_var	<i>simulating correlated variable</i>
---------	---------------------------------------

Description

simulating correlated variable

Usage

```
cor_var(x, cor)
```

Arguments

x	Vector of values to which the other vector needs to be correlated
cor	correlation

Value

Returns a vector with values that are correlated to vector x.

Examples

```
cor_var(c(2,4,2,2,6,7,5,6,6,7,9,4,5), 0.5)
```

`deltaG_F`*Calculating genetic gain and rate of inbreeding*

Description

This function can be used to calculate the genetic gain per generation for each trait and the rate of inbreeding from the output of the breeding program simulations (pedigree needs to be called ped). Rate of inbreeding is calculated from the first simulated generation till the last generation, unless a different end generation is specified. Genetic gain is an average genetic gain per generation, calculated over multiple generations.

Usage

```
deltaG_F(  
  Ntraits = BPdata$Ntraits,  
  startgen = c(rep(2, times = Ntraits)),  
  endgen = c(rep(max(ped$generation), times = Ntraits)),  
  endgenF = max(ped$generation)  
)
```

Arguments

<code>Ntraits</code>	The number of simulated traits. Does not need to be specified if <code>Ntraits</code> is in the list 'BPdata'.
<code>startgen</code>	The first generation from which the (average) genetic gain per generation needs to be calculated. By default, this is generation 2 for each trait.
<code>endgen</code>	The last generation that needs to be included in the calculation of the (average) genetic gain per generation. By default, this is the last simulated generation for each trait.
<code>endgenF</code>	The last generation that needs to be included in the calculateion of the rate of inbreeding. This is by default the last simulated generation for each trait.

Value

A data frame with the rate of inbreeding and the genetic gain for each trait.

Examples

```
## Not run:  
deltaG_F()  
  
## End(Not run)
```

founderpopfam *Founder population family design*

Description

This function can be used to create a founder population of unrelated animals for a breeding program with a family design.

Usage

```
founderpopfam(
  Nm = BPdata$Nm,
  Nf = BPdata$Nf,
  Nm2 = BPdata$Nm2,
  Nf2 = BPdata$Nf2,
  Nbatch = NA,
  Nbatch2 = NA,
  batch = NA,
  batch2 = NA,
  Ntraits = BPdata$Ntraits,
  TraitsIndex = c(1:Ntraits),
  Rgen = BPdata$Rgen,
  Rres = BPdata$Rres,
  Rcom = BPdata$Rcom,
  mean = BPdata$mean,
  a_var = BPdata$a_var,
  c_var = BPdata$c_var,
  e_var = BPdata$e_var,
  est_EBV = FALSE,
  EBV,
  accuracy,
  indexweights = c(rep(1, Ntraits))
)
```

Arguments

Nm	Number of males that will be used to breed the first generation of offspring. Males are listed first in the output data frame and coded with 1 for sex.
Nf	Number of females that will be used to breed the first generation of offspring. Females are listed second in the output data frame and coded with 2 for sex.
Nm2	Additional males that will not be used to breed the first generation of offspring, but can be used as additional selection candidates in further generations. Default is zero.
Nf2	Additional females that will not be used to breed the first generation of offspring, but can be used as additional selection candidates in further generations. Default is zero.

Nbatch	Number of batches over which the founder animals are divided. The number of founder animals that need to be simulated should be a multiple of the number of batches. If Nbatch is not specified, the parameter batch can be used to divide the founder animals over batches. If both Nbatch and batch are not specified, then all founder animals will be assigned to batch 0.
Nbatch2	Number of batches over which the additional founder animals ($N_{m2} + N_{f2}$) are divided. The number of additional founder animals that need to be simulated should be a multiple of the number of batches. If Nbatch2 is not specified, the parameter batch2 can be used to divide the additional founder animals over batches. If both Nbatch2 and batch2 are both not specified, then the additional founder animals are divided over batches in the same way as the other founder animals ($N_m + N_f$).
batch	A vector with names of the batches. The number of founder animals that need to be simulated should be the same or a multiple of the length of the vector. If batch is not specified, the parameter Nbatch can be used to divide the founder animals over batches. If both Nbatch and batch are not specified, then all founder animals will be assigned to batch 0.
batch2	A vector with names of the batches for the additional founder animals ($N_{m2} + N_{f2}$). The number of additional founder animals that are simulated should be the same or a multiple of the length of the vector. If batch2 is not specified, the parameter Nbatch2 can be used to divide the additional founder animals over batches. If both Nbatch2 and batch2 are not specified, then the additional founder animals are divided over batches in the same way as the other founder animals ($N_m + N_f$).
Ntraits	Number of traits to be simulated. Does not need to be specified if Ntraits is in the list 'BPdata'.
TraitsIndex	Vector of traits that are in the index. By default, all traits are in the selection index.
Rgen	Matrix of all genetic correlations between all Ntraits. Only needs to be specified if there is no matrix of genetic correlations named Rgen in the list called 'BPdata'.
Rres	Matrix of all residual correlations between all Ntraits. Only needs to be specified if there is no matrix of residual correlations named Rres in the list called 'BPdata'.
Rcom	Matrix of all common environmental correlations between all Ntraits. Only needs to be specified if there is no matrix of common environmental correlations named Rcom in the list called 'BPdata'.
mean	Vector of means of all traits. Only needs to be specified if there is no vector of means named mean in the list 'BPdata'.
a_var	Vector of genetic variances of all traits. Only needs to be specified if there is no vector of genetic variances named a_var in the list 'BPdata'.
c_var	Vector of common environmental variances of all traits. Only needs to be specified if there is no vector of common environmental variances named c_var in the list 'BPdata'. If there is no common environmental effect, provide a vector of zero's.

e_var	Vector of residual variances of all traits. Only needs to be specified if there is no vector of residual variances named e_var in the list 'BPdata'.
est_EBV	TRUE or FALSE for estimating breeding values for all founder animals. The default is FALSE.
EBV	If est_EBV is TRUE, a vector of methods for simulating EBVs for each trait in the selection index need to be specified. There are four options: 0 for giving each animal a breeding values of 0 for the specific trait; "mean_pop" for simulating EBVs for all founder populations that are equal to the mean of the population, the mean of the population is provided in the vector 'mean'; "pheno" for simulating EBVs that are equal to the phenotype of the animal for the specific trait and "EBV" for simulating EBVs as a value correlated to the true breeding value of the animal. This correlation equals the accuracy. Accuracies need to be specified in the parameter 'accuracy'.
accuracy	If for one of the traits the EBVs need to be calculated with the method "EBV", specified in the parameter 'EBV', then accuracies need to be provided. A value need to be added for each trait, however this value can be NA for traits from which the EBVs are not simulated using method "EBV".
indexweights	If traits need to be combined in an index, then desired gain indices need to be specified for each trait in the index.

Value

A data frame is returned, which should be called 'ped' in order to be able to use it in the other functions. The 'ped' data frame consist of all the simulated base animals, including their sex, generation, batch number, inbreeding level, phenotypes and genetic, common environmental and residual effects of each trait.

Examples

```
ped <- founderpopfam(Nm=60, Nm2=0,
  Nf=60, Nf2=0,
  batch = c(0,1,2),
  Ntraits=2,
  TraitsIndex = 2,
  Rgen= matrix(c(1.00 , 0.48,
                0.48 , 1.00),
              nrow = 2),
  Rcom= matrix(c(1.00 , 0.5,
                0.5 , 1.00),
              nrow = 2),
  Rres= matrix(c(1.00 , 0.32,
                0.32 , 1.00),
              nrow = 2),
  mean=c(50,500),
  a_var=c(200,8000),
  c_var=c(150,1000),
  e_var= c(250,12000))
```

```

ped <- founderpopfam(Nm=60,
                    Nf=60,
                    Nm2=120,
                    Nf2=120,
                    Nbatch = 4,
                    batch2 = c(-3,-2,-1,0),
                    Ntraits=2,
                    TraitsIndex = c(1,2),
                    Rgen= matrix(c(1.00 , 0.48,
                                   0.48 , 1.00),
                                nrow = 2),
                    Rcom= matrix(c(1.00 , 0,
                                   0 , 1.00),
                                nrow = 2),
                    Rres= matrix(c(1.00 , 0.32,
                                   0.32 , 1.00),
                                nrow = 2),
                    mean=c(50,500),
                    a_var=c(200,8000),
                    c_var=c(0,0),
                    e_var= c(250,12000),
                    est_EBV = TRUE,
                    EBV= c("pheno", "EBV"),
                    accuracy= c(NA,0.78),
                    indexweight= c(1,5))

```

founderpopgroup

Founder population group mating design

Description

This function can be used to create a founder population of unrelated animals for a breeding program with group mating.

Usage

```

founderpopgroup(
  Nm = BPdata$Nm,
  Nf = BPdata$Nf,
  Nm2 = BPdata$Nm2,
  Nf2 = BPdata$Nf2,
  Nbatch = NA,
  Nbatch2 = NA,
  batch = NA,
  batch2 = NA,
  Ntraits = BPdata$Ntraits,
  TraitsIndex = c(1:Ntraits),

```

```

Rgen = BPdata$Rgen,
Rres = BPdata$Rres,
Rcom = BPdata$Rcom,
mean = BPdata$mean,
a_var = BPdata$a_var,
c_var = BPdata$c_var,
e_var = BPdata$e_var,
est_EBV = FALSE,
EBV,
accuracy,
indexweights = c(1)
)

```

Arguments

Nm	Number of males that will be used to breed the first generation of offspring. Males are listed first in the output data frame and coded with 1 for sex.
Nf	Number of females that will be used to breed the first generation of offspring. Females are listed second in the output data frame and coded with 2 for sex.
Nm2	Additional males that will not be used to breed the first generation of offspring, but can be used as additional selection candidates in further generations. Default is zero.
Nf2	Additional females that will not be used to breed the first generation of offspring, but can be used as additional selection candidates in further generations. Default is zero.
Nbatch	Number of batches over which the founder animals are divided. The number of founder animals that need to be simulated should be a multiple of the number of batches. If Nbatch is not specified, the parameter batch can be used to divide the founder animals over batches. If both Nbatch and batch are not specified, then all founder animals will be assigned to batch 0.
Nbatch2	Number of batches over which the additional founder animals (Nm2 + Nf2) are divided. The number of additional founder animals that need to be simulated should be a multiple of the number of batches. If Nbatch2 is not specified, the parameter batch2 can be used to divide the additional founder animals over batches. If both Nbatch2 and batch2 are both not specified, then the additional founder animals are divided over batches in the same way as the other founder animals (Nm+Nf).
batch	A vector with names of the batches. The number of founder animals that need to be simulated should be the same or a multiple of the length of the vector. If batch is not specified, the parameter Nbatch can be used to divide the founder animals over batches. If both Nbatch and batch are not specified, then all founder animals will be assigned to batch 0.
batch2	A vector with names of the batches for the additional founder animals (Nm2 + Nf2). The number of additional founder animals that are simulated should be the same or a multiple of the length of the vector. If batch2 is not specified, the parameter Nbatch2 can be used to divide the additional founder animals over batches. If both Nbatch2 and batch2 are not specified, then the additional

	founder animals are divided over batches in the same way as the other founder animals (Nm+Nf).
Ntraits	Number of traits to be simulated. Does not need to be specified if Ntraits is in the list 'BPdata'.
TraitsIndex	Vector of traits that are in the index. By default, all traits are in the selection index.
Rgen	Matrix of all genetic correlations between all Ntraits. Only needs to be specified if there is no matrix of genetic correlations named Rgen in the list called 'BPdata'.
Rres	Matrix of all residual correlations between all Ntraits. Only needs to be specified if there is no matrix of residual correlations named Rres in the list called 'BPdata'.
Rcom	Matrix of all common environmental correlations between all Ntraits. Only needs to be specified if there is no matrix of common environmental correlations named Rcom in the list called 'BPdata'.
mean	Vector of means of all traits. Only needs to be specified if there is no vector of means named mean in the list 'BPdata'.
a_var	Vector of genetic variances of all traits. Only needs to be specified if there is no vector of genetic variances named a_var in the list 'BPdata'.
c_var	Vector of common environmental variances of all traits. Only needs to be specified if there is no vector of common environmental variances named c_var in the list 'BPdata'. If there is no common environmental effect, provide a vector of zero's.
e_var	Vector of residual variances of all traits. Only needs to be specified if there is no vector of residual variances named e_var in the list 'BPdata'.
est_EBV	TRUE or FALSE for estimating breeding values for all founder animals. The default is FALSE.
EBV	If est_EBV is TRUE, a vector of methods for simulating EBVs for each trait in the selection index need to be specified. There are four options: 0 for giving each animal a breeding values of 0 for the specific trait; "mean_pop" for simulating EBVs for all founder populations that are equal to the mean of the population, the mean of the population is provided in the vector 'mean'; "pheno" for simulating EBVs that are equal to the phenotype of the animal for the specific trait and "EBV" for simulating EBVs as a value correlated to the true breeding value of the animal. This correlation equals the accuracy. Accuracies need to be specified in the parameter 'accuracy'.
accuracy	If for one of the traits the EBVs need to be calculated with the method "EBV" specified in the parameter 'EBV', then accuracies need to be provided. A value need to be added for each trait, however this value can be zero for traits from which the EBVs are not simulated using method "EBV".
indexweights	If traits need to be combined in an index, then desired gain indices need to be specified for each trait in the index.

Value

A data frame is returned, which should be called 'ped' in order to be able to use it in the other functions. The 'ped' data frame consist of all the simulated base animals, including their sex, generation, batch number, inbreeding level, phenotypes and genetic, common environmental and residual effects of each trait.

Examples

```
ped <- founderpopgroup(Nm=60, Nm2=0,
  Nf=60, Nf2 = 0,
  batch = c(0,1,2),
  Ntraits=2,
  TraitsIndex = 2,
  Rgen= matrix(c(1.00 , 0.48,
    0.48 , 1.00),
    nrow = 2),
  Rcom= matrix(c(1.00 , 0,
    0 , 1.00),
    nrow = 2),
  Rres= matrix(c(1.00 , 0.32,
    0.32 , 1.00),
    nrow = 2),
  mean=c(50,500),
  a_var=c(200,8000),
  c_var=c(0,0),
  e_var= c(250,12000))
```

```
ped <- founderpopgroup(Nm=60,
  Nf=60,
  Nm2=120,
  Nf2=120,
  Nbatch = 4,
  batch2 = c(-3,-2,-1,0),
  Ntraits=2,
  TraitsIndex = c(1,2),
  Rgen= matrix(c(1.00 , 0.48,
    0.48 , 1.00),
    nrow = 2),
  Rcom= matrix(c(1.00 , 0,
    0 , 1.00),
    nrow = 2),
  Rres= matrix(c(1.00 , 0.32,
    0.32 , 1.00),
    nrow = 2),
  mean=c(50,500),
  a_var=c(200,8000),
  c_var=c(0,0),
  e_var= c(250,12000),
  est_EBV = TRUE,
  EBV= c("pheno", "EBV"),
  accuracy= c(NA,0.78),
```

```
indexweight= c(1,5)
```

gen_param

Importing genetic parameters from excel file

Description

This function can be used to import data (heritabilities, variances, correlations) from a excel file with a specific format, and it produces a list. If this list is called 'BPdata', then the genetic parameters do not need to be specified anymore in the other functions.

Usage

```
gen_param(nameexcelfile)
```

Arguments

nameexcelfile Name or path to the excel file.

Value

A list with parameters that can be used in the other functions of this package.

Examples

```
## Not run:  
BPdata <- gen_param("example.xlsx")  
  
## End(Not run)
```

groupmating

Group mating

Description

Function to determine which full sib families are produced in a group mating design.

Usage

```
groupmating(
  gen,
  batch = 0,
  batch_m = NA,
  batch_f = NA,
  No,
  contr_m,
  contr_f,
  distribution = "Gamma",
  shape = 0.75,
  scale = 0.11,
  selected = 1
)
```

Arguments

gen	The generations of the sires and dams
batch	The batches of the sires and dams. Default is 0.
batch_m	The batch of the sires. Default is NA. If batch_m is specified, batch_f also needs to be specified and the parameter batch is not used.
batch_f	The batch of the dams. Default is NA. If batch_f is specified, batch_m also needs to be specified and the parameter batch is not used.
No	The total number of offspring of all matings.
contr_m	Proportion of sires that contribute to the offspring.
contr_f	Proportion of dams that contribute to the offspring.
distribution	The distribution from which the contributions are drawn. Options are: "Gamma" (default) and "Uniform"
shape	The shape of the gamma distribution. Default is 0.75.
scale	The scale of the gamma distribution. Default is 0.11.
selected	The value in ped\$selected of the selected sires and dams. Default is 1.

Details

By default, the contribution of the sires and dams that do reproduce come from a gamma distribution. The default shape and scale of the gamma distribution are 0.75 and 0.11, respectively. A uniform distribution can also be specified for the contributions of the sires and dams. If not all sires and dams should contribute to the offspring, then the sires and dams that are going to reproduce are randomly chosen. The output is a dataframe called Mating with the sire, dam and the size of each full sib family.

The sires and dams can come from multiple batches or generations. In that case, a vector of batches or generations need to be provided.

Value

The output is a data frame with for each full sib family the sire and dam and the number of offspring per full sib family.

Examples

```

{ped <- founderpopgroup(Nm=60,
                        Nf=60,
                        Nm2=120,
                        Nf2=120,
                        Nbatch = 4,
                        batch2 = c(-3,-2,-1,0),
                        Ntraits=2,
                        TraitsIndex = 2,
                        Rgen= matrix(c(1.00 , 0.48,
                                       0.48 , 1.00),
                                    nrow = 2),
                        Rcom= matrix(c(1.00 , 0,
                                       0 , 1.00),
                                    nrow = 2),
                        Res= matrix(c(1.00 , 0.32,
                                       0.32 , 1.00),
                                    nrow = 2),
                        mean=c(50,500),
                        a_var=c(200,8000),
                        c_var=c(0,0),
                        e_var= c(250,12000))

Mating <- groupmating(gen = 0,
                      batch=-3,
                      No=1000,
                      contr_m = 0.5,
                      contr_f = 0.5)
}

```

offspringFSfam

Creating offspring for family design

Description

This function can be used to create the offspring of one full sib family in a family design, with genetic, common environmental and residual effects for each offspring. Offspring are added to the ped file.

Usage

```

offspringFSfam(
  gen,
  No,
  sire,
  dam,
  batch = 0,
  probmale = BPdata$prob_male,

```

```

Ntraits = BPdata$Ntraits,
TraitsIndex = c(1:Ntraits),
Rgen = BPdata$Rgen,
Rres = BPdata$Rres,
Rcom = BPdata$Rcom,
a_var = BPdata$a_var,
c_var = BPdata$c_var,
e_var = BPdata$e_var,
inbreeding = TRUE
)

```

Arguments

gen	The generation of the offspring.
No	The number of offspring per fullsib family.
sire	The sire of the full sib family. Sire should also be in the ped file with information on the inbreeding level and true breeding values for each trait.
dam	The dam of the full sib family. Dam should also be in the ped file with information on the inbreeding level and the true breeding values for each trait.
batch	The batch of the offspring. Default is 0.
probmale	The probability that a offspring is male. The probability that the offspring is female is calculated as 1 - probmale. Probmale does not need to be specified if it is in the list called 'BPdata'. The default is 0.5.
Ntraits	Number of traits to be simulated. Does not need to be specified if Ntraits is in the list 'BPdata'.
TraitsIndex	Vector of traits that are in the selection index. By default, all traits are in the index.
Rgen	Matrix of all genetic correlations between all Ntraits. Only needs to be specified if there is no matrix of genetic correlations named Rgen in the list called 'BPdata'.
Rres	Matrix of all residual correlations between all Ntraits. Only needs to be specified if there is no matrix of residual correlations named Rres in the list called 'BPdata'.
Rcom	Matrix of all common environmental correlations between all Ntraits. Only needs to be specified if there is no matrix of common environmental correlations named Rcom in the list called 'BPdata'.
a_var	Vector of genetic variances of all traits. Only needs to be specified if there is no vector of genetic variances named a_var in the list called 'BPdata'.
c_var	Vector of common environmental variances of all traits. Only needs to be specified if there is no vector of common environmental variances named c_var in the list called 'BPdata'.
e_var	Vector of residual variances of all traits. Only needs to be specified if there is no vector of residual variances named e_var in the list called 'BPdata'.
inbreeding	If TRUE (default), then the inbreeding level is calculated for each offspring.

Value

This function returns the 'ped' data frame: the new offspring are added to this data frame.

Examples

```
ped <- founderpopfam(Nm = 60,
                    Nf = 60,
                    Nm2 = 0,
                    Nf2 = 0,
                    Ntraits = 2,
                    TraitsIndex = 2,
                    Rgen = matrix(c(1.00 , 0.48,
                                   0.48 , 1.00),
                                  nrow = 2),
                    Rcom = matrix(c(1.00 , 0.5,
                                   0.5 , 1.00),
                                  nrow = 2),
                    Rres = matrix(c(1.00 , 0.32,
                                   0.32 , 1.00),
                                  nrow = 2),
                    mean = c(50,500),
                    a_var = c(200,8000),
                    c_var = c(150,1000),
                    e_var = c(250,12000))

Mating <- randommating(gen = 0,
                      Nfam_FS = 120)

for(fam in 1: nrow(Mating)){
ped <- offspringFSfam(gen = 1,
                    No = 100,
                    probmale = 0.5,
                    sire = Mating$Sire[fam],
                    dam = Mating$Dam[fam],
                    Ntraits = 2,
                    TraitsIndex = 2,
                    Rgen = matrix(c(1.00 , 0.48,
                                   0.48 , 1.00),
                                  nrow = 2),
                    Rcom = matrix(c(1.00 , 0.5,
                                   0.5 , 1.00),
                                  nrow = 2),
                    Rres = matrix(c(1.00 , 0.32,
                                   0.32 , 1.00),
                                  nrow = 2),
                    a_var = c(200,8000),
                    c_var = c(150,1000),
                    e_var = c(250,12000))
}
```

offspringFSgroup *Creating offspring for a group mating design*

Description

This function can be used to create the offspring of one full sib family in a group mating design, with genetic, common environmental and residual effects for each offspring. Offspring are added to the ped file.

Usage

```
offspringFSgroup(
  gen,
  No,
  sire,
  dam,
  batch = 0,
  probmale = BPdata$prob_male,
  Ntraits = BPdata$Ntraits,
  TraitsIndex = c(1:Ntraits),
  Rgen = BPdata$Rgen,
  Rres = BPdata$Rres,
  Rcom = BPdata$Rcom,
  a_var = BPdata$a_var,
  c_var = BPdata$c_var,
  e_var = BPdata$e_var,
  inbreeding = TRUE
)
```

Arguments

gen	The generation of the offspring.
No	The number of offspring per fullsib family.
sire	The sire of the full sib family. Sire should also be in the ped file with information on the inbreeding level and true breeding values for each trait.
dam	The dam of the full sib family. Dam should also be in the ped file with information on the inbreeding level and the true breeding values for each trait.
batch	The batch of the offspring. Default is 0.
probmale	The probability that a offspring is male. The probability that the offspring is female is calculated as 1 - probmale. Probmale does not need to be specified if it is in the list called 'BPdata'. The default is 0.5.
Ntraits	Number of traits to be simulated. Does not need to be specified if Ntraits is in the list 'BPdata'.
TraitsIndex	Vector of traits that are in the selection index. By default, all traits are in the index.

Rgen	Matrix of all genetic correlations between all Ntraits. Only needs to be specified if there is no matrix of genetic correlations named Rgen in the list called 'BPdata'.
Rres	Matrix of all residual correlations between all Ntraits. Only needs to be specified if there is no matrix of residual correlations named Rres in the list called 'BPdata'.
Rcom	Matrix of all common environmental correlations between all Ntraits. Only needs to be specified if there is no matrix of common environmental correlations named Rcom in the list called 'BPdata'.
a_var	Vector of genetic variances of all traits. Only needs to be specified if there is no vector of genetic variances named a_var in the list called 'BPdata'.
c_var	Vector of common environmental variances of all traits. Only needs to be specified if there is no vector of common environmental variances named c_var in the list called 'BPdata'.
e_var	Vector of residual variances of all traits. Only needs to be specified if there is no vector of residual variances named e_var in the list called 'BPdata'.
inbreeding	If True (default), then the inbreeding level is calculated for each offspring.

Value

This function returns the 'ped' data frame: the new offspring are added to this data frame.

Examples

```
ped <- founderpopgroup(Nm = 60,
  Nf = 60,
  Nm2 = 0,
  Nf2 = 0,
  batch = c(-3,-2,-1,0),
  Ntraits = 2,
  TraitsIndex = 2,
  Rgen = matrix(c(1.00 , 0.48,
                 0.48 , 1.00),
               nrow = 2),
  Rcom = matrix(c(1.00 , 0,
                 0 , 1.00),
               nrow = 2),
  Rres = matrix(c(1.00 , 0.32,
                 0.32 , 1.00),
               nrow = 2),
  mean = c(50,500),
  a_var = c(200,8000),
  c_var = c(0,0),
  e_var = c(250,12000))

Mating <- groupmating(gen = 0,
  batch = -3,
  No = 1000,
```

```

        contr_m = 0.5,
        contr_f = 0.5)

for(fam in 1: nrow(Mating)){
ped <- offspringFSgroup(gen = 1,
    No = Mating$No[fam],
    probmale = 0.5,
    sire = Mating$Sire[fam],
    dam = Mating$Dam[fam],
    batch = 1,
    Ntraits =2,
    TraitsIndex = 2,
    Rgen = matrix(c(1.00 , 0.48,
                    0.48 , 1.00),
                  nrow = 2),
    Rcom = matrix(c(1.00 , 0.5,
                    0.5 , 1.00),
                  nrow = 2),
    Res = matrix(c(1.00 , 0.32,
                    0.32 , 1.00),
                  nrow = 2),
    a_var = c(200,8000),
    c_var = c(150,1000),
    e_var = c(250,12000))
}

```

```
preselectphen
```

Preselecting offspring based on phenotype

Description

This function can be used to preselect offspring based on the phenotype of one trait.

Usage

```

preselectphen(
  gen,
  batch = 0,
  withinfam = TRUE,
  Nenv,
  Npreselect,
  trait,
  Ntraits = BPdata$Ntraits
)

```

Arguments

`gen` The generation of the offspring.

batch	The batch of the offspring. Default is 0.
withinfam	Preselection within a full sib family or not, default is TRUE.
Nenv	The number of environments the fish need to be preselected for, for example a certain number of fish need to be preselected for a nucleus and a certain number of fish need to be preselected for a production environment in order to measure sib traits. Preselection is based on the same trait.
Npresel	A vector of the number of fish that needs to be preselected for each environment.
trait	The trait on which the phenotypic preselection is based, for example trait = 2.
Ntraits	The total number of simulated traits. Needs to be specified if Ntraits is not specified in a list called 'BPdata'.

Value

This function will make changes to the data frame called 'ped'. Fish that are pre-selected will get a number assigned to their column 'preselected'.

Examples

```
ped <- founderpopfam(Nm = 60,
                    Nf = 60,
                    Nm2 = 0,
                    Nf2 = 0,
                    Ntraits = 2,
                    TraitsIndex = 2,
                    Rgen = matrix(c(1.00 , 0.48,
                                    0.48 , 1.00),
                                   nrow = 2),
                    Rcom = matrix(c(1.00 , 0.5,
                                    0.5 , 1.00),
                                   nrow = 2),
                    Rres = matrix(c(1.00 , 0.32,
                                    0.32 , 1.00),
                                   nrow = 2),
                    mean = c(50,500),
                    a_var = c(200,8000),
                    c_var = c(150,1000),
                    e_var = c(250,12000))

Mating <- randommating(gen = 0,
                      Nfam_FS = 120)

for(fam in 1: nrow(Mating)){
  ped <- offspringFSfam(gen = 1,
                      No = 100,
                      probmale = 0.5,
                      sire = Mating$Sire[fam],
                      dam = Mating$Dam[fam],
                      Ntraits = 2,
                      TraitsIndex = 2,
```



```

      Rgen = matrix(c(1.00 , 0.48,
                    0.48 , 1.00),
                  nrow = 2),
      Rcom = matrix(c(1.00 , 0.5,
                    0.5 , 1.00),
                  nrow = 2),
      Rres = matrix(c(1.00 , 0.32,
                    0.32 , 1.00),
                  nrow = 2),
      a_var = c(200,8000),
      c_var = c(150,1000),
      e_var = c(250,12000)
}

```

```

ped <- presephen(gen = 1,
               Nenv = 2,
               Nprese1 = c(25,15),
               trait = 1,
               Ntraits = 2)

```

```

ped <- founderpopgroup(Nm = 60,
                    Nf = 60,
                    Nm2 = 0,
                    Nf2 = 0,
                    batch = c(-3,-2,-1,0),
                    Ntraits = 2,
                    TraitsIndex = 2,
                    Rgen = matrix(c(1.00 , 0.48,
                                    0.48 , 1.00),
                                  nrow = 2),
                    Rcom = matrix(c(1.00 , 0,
                                    0 , 1.00),
                                  nrow = 2),
                    Rres = matrix(c(1.00 , 0.32,
                                    0.32 , 1.00),
                                  nrow = 2),
                    mean = c(50,500),
                    a_var = c(200,8000),
                    c_var = c(0,0),
                    e_var = c(250,12000))

```

```

Mating <- groupmating(gen = 0,
                    batch = -3,
                    No = 1000,
                    contr_m = 0.5,
                    contr_f = 0.5)

```

```

for(fam in 1: nrow(Mating)){
ped <- offspringFSgroup(gen = 1,
                    No = Mating$No[fam],
                    probmale = 0.5,

```

```

    sire = Mating$Sire[fam],
    dam = Mating$Dam[fam],
    batch = 1,
    Ntraits = 2,
    TraitsIndex = 2,
    Rgen = matrix(c(1.00 , 0.48,
                   0.48 , 1.00),
                 nrow = 2),
    Rcom = matrix(c(1.00 , 0.5,
                   0.5 , 1.00),
                 nrow = 2),
    Rres = matrix(c(1.00 , 0.32,
                   0.32 , 1.00),
                 nrow = 2),
    a_var = c(200,8000),
    c_var = c(150,1000),
    e_var = c(250,12000))
}

ped <- preselectphen(gen = 1,
                    batch = 1,
                    withinfam = FALSE,
                    Nenv = 2,
                    Npreselect = c(400,150),
                    trait = 1,
                    Ntraits = 2)

```

```
preselectrandom
```

```
Randomly preselecting offspring
```

Description

This function can be used to randomly preselect offspring or randomly allocate offspring to for example the nucleus and production environment.

Usage

```
preselectrandom(gen, batch = 0, withinfam = TRUE, Nenv, Npreselect)
```

Arguments

gen	The generation of the offspring.
batch	The batch of the offspring that need to be randomly preselected. Default is 0.
withinfam	Preselection within a full sib family or not, default is TRUE.
Nenv	The number of environments the fish need to be preselected for, for example a certain number of fish need to be preselected for a nucleus and a certain number of fish need to be preselected for a production environment in order to measure sib traits.
Npreselect	The number of fish that needs to be preselected for each environment.

Value

This function will make changes to the data frame called 'ped'. Fish that are pre-selected will get a number assigned to their column 'preselected'.

Examples

```
ped <- founderpopfam(Nm = 60,
                    Nf = 60,
                    Nm2 = 0,
                    Nf2 = 0,
                    Ntraits = 2,
                    TraitsIndex = 2,
                    Rgen = matrix(c(1.00 , 0.48,
                                   0.48 , 1.00),
                                  nrow = 2),
                    Rcom = matrix(c(1.00 , 0.5,
                                   0.5 , 1.00),
                                  nrow = 2),
                    Rres = matrix(c(1.00 , 0.32,
                                   0.32 , 1.00),
                                  nrow = 2),
                    mean = c(50,500),
                    a_var = c(200,8000),
                    c_var = c(150,1000),
                    e_var = c(250,12000))

Mating <- randommating(gen = 0,
                      Nfam_FS = 120)

for(fam in 1: nrow(Mating)){
ped <- offspringFSfam(gen = 1,
                    No = 100,
                    probmale = 0.5,
                    sire = Mating$Sire[fam],
                    dam = Mating$Dam[fam],
                    Ntraits = 2,
                    TraitsIndex = 2,
                    Rgen = matrix(c(1.00 , 0.48,
                                   0.48 , 1.00),
                                  nrow = 2),
                    Rcom = matrix(c(1.00 , 0.5,
                                   0.5 , 1.00),
                                  nrow = 2),
                    Rres = matrix(c(1.00 , 0.32,
                                   0.32 , 1.00),
                                  nrow = 2),
                    a_var = c(200,8000),
                    c_var = c(150,1000),
                    e_var = c(250,12000))
}

ped <- preselrandom(gen = 1,
```

```
Nenv = 2,
Npresel = c(25,15))
```

preselcand

Preselection of selection candidates

Description

Function to preselect selection candidates based on their Index/EBV/phenotype. Fish are preselected from the fish that are available for selection (`ped$selcand==1`). Fish that are not preselected will be assigned a value of 2 for `ped$selcand`. These fish will not be used in the function `select`.

Usage

```
preselcand(
  gen,
  batch = 0,
  select_on = "Index",
  trait,
  Ntraits = BPdata$Ntraits,
  Nm,
  Nf,
  N,
  within_FSfam = FALSE,
  max_FSfam = "all"
)
```

Arguments

<code>gen</code>	The generation of the selection candidates. A vector of generations can be provided.
<code>batch</code>	The batch of the selection candidates. Default is 0. It is possible to provide a vector with multiple batches.
<code>select_on</code>	Options: "Index", "EBV" or "Phenotype". For "EBV" and "Phenotype", also the trait need to be specified in 'trait'. Default is Index.
<code>trait</code>	Which trait or EBV the selection is based on when option "EBV" or "Phenotype" is chosen in 'select on'.
<code>Ntraits</code>	Number of simulated traits. Does not need to be specified if <code>Ntraits</code> is specified in a list called 'BPdata'.
<code>Nm</code>	Number of males to preselect (In total or per full sib family, depending on <code>within_FSfam</code>).
<code>Nf</code>	Number of females to preselect (In total or per full sib family, depending on <code>within_FSfam</code>).
<code>N</code>	Total number of pre-selected fish. Does not need to be specified if <code>Nm</code> and <code>Nf</code> are specified.

- `within_FSfam` If True, pre-selection takes place within a full sib family. Default is False. Only use within family selection in a family design. Each full sib family must have at least `Nm` male sibs and `Nf` female sibs (or `N` sibs).
- `max_FSfam` Maximum number of sibs that can be selected per full sib familie, in case selection does not take place within a full sib family. Default is 'all'.

Value

This function will change the data frame called 'ped'. Fish that are not preselected will be assigned a value of 2 for `ped$selcand`.

Examples

```
ped <- founderpopfam(Nm = 60,
                    Nf = 60,
                    Nm2 = 0,
                    Nf2 = 0,
                    Ntraits = 2,
                    TraitsIndex = 2,
                    Rgen = matrix(c(1.00 , 0.48,
                                   0.48 , 1.00),
                                  nrow = 2),
                    Rcom = matrix(c(1.00 , 0.5,
                                   0.5 , 1.00),
                                  nrow = 2),
                    Rres = matrix(c(1.00 , 0.32,
                                   0.32 , 1.00),
                                  nrow = 2),
                    mean = c(50,500),
                    a_var = c(200,8000),
                    c_var = c(150,1000),
                    e_var = c(250,12000))

Mating <- randommating(gen = 0,
                      Nfam_FS = 120)

for(fam in 1: nrow(Mating)){
ped <- offspringFSfam(gen = 1,
                    No = 100,
                    probmale = 0.5,
                    sire = Mating$Sire[fam],
                    dam = Mating$Dam[fam],
                    Ntraits = 2,
                    TraitsIndex = 2,
                    Rgen = matrix(c(1.00 , 0.48,
                                   0.48 , 1.00),
                                  nrow = 2),
                    Rcom = matrix(c(1.00 , 0.5,
                                   0.5 , 1.00),
                                  nrow = 2),
                    Rres = matrix(c(1.00 , 0.32,
                                   0.32 , 1.00),
```

```

        nrow = 2),
    a_var = c(200,8000),
    c_var = c(150,1000),
    e_var = c(250,12000))
}

ped <- preselphen(gen = 1,
  Nenv = 2,
  Npresel = c(25,15),
  trait = 1,
  Ntraits = 2)

ped <- avail_selection(gen = 1,
  presel = 1,
  surv = 0.9)

ped <- breeding_values(gen = 1,
  TraitsIndex = 2,
  EBV = "GEBV",
  GenomLength = 11.3,
  Ne = 100,
  SizeTraining = nrow(ped[ped$preselected ==2,]),
  Ntraits = 2,
  a_var = c(200,8000),
  h2 = c(0.33,0.38))

ped <- preselcand(gen = 1,
  Nm = 300,
  Nf = 300,
  max_FSfam = 15,
  Ntraits = 2)

```

randommating

Random mating family design

Description

Function to randomly allocate sires to dams.

Usage

```

randommating(
  gen,
  batch = 0,
  batch_m = NA,
  batch_f = NA,
  Nfam_FS,
  No = NA,
  No_FSfam = NA,

```

```

    selected = 1
  )

```

Arguments

gen	The generations of the sires and dams.
batch	The batch of the sires and dams. Default is 0.
batch_m	The batch of the sires. Default is NA. If batch_m is specified, batch_f also needs to be specified and the parameter batch is not used.
batch_f	The batch of the dams. Default is NA. If batch_f is specified, batch_m also needs to be specified and the parameter batch is not used.
Nfam_FS	The number of full sib families.
No	The total number of offspring of all matings. Default is NA.
No_FScam	The number of offspring in each full sib family. Default is NA.
selected	The value in ped\$selected of the selected sires and dams. Default is 1.

Details

A dataframe called ped needs to be present in the data. Ped needs to contain all the sires and dams that need to be allocated to each other and the columns sex, selected, generation, batch and id (first column).

The sires and dams can come from multiple batches or generations. In that case, a vector of batches or generations need to be provided.

Optionally, a column with the number of offspring per full sib family can be added to the dataframe. To do this, either the argument No_FScam or No needs to be provided. In case No_FScam is provided, each mating will have the same number of offspring (namely the value provided with No_FScam). When No is provided, then the total number of offspring will be evenly divided among each mating if possible.

Value

The output is a data frame with for each full sib family the sire and dam and (optional) the number of offspring per full sib family.

Examples

```

{ ped <- founderpopfam(Nm=60,
  Nf=60,
  Nm2=0,
  Nf2=0,
  Ntraits=2,
  TraitsIndex = 2,
  Rgen= matrix(c(1.00 , 0.48,
                0.48 , 1.00),
              nrow = 2),
  Rcom= matrix(c(1.00 , 0.5,
                0.5 , 1.00),
              nrow = 2),

```

```

Rres= matrix(c(1.00  , 0.32,
              0.32  , 1.00),
            nrow = 2),
            mean=c(50,500),
            a_var=c(200,8000),
            c_var=c(150,1000),
            e_var= c(250,12000))

Mating <- randommating(gen = 0,
                      Nfam_FS = 120)
}

```

select

Selection

Description

Function to select selection candidates based on their Index, EBV or phenotype. `ped$selcand` should be 1 for the selection candidates.

Usage

```

select(
  gen,
  batch = 0,
  select_on = "Index",
  trait,
  Ntraits = BPdata$Ntraits,
  Nm = BPdata$Nm,
  Nf = BPdata$Nf,
  max_FSfam = "all",
  mature_m = 1,
  mature_f = 1,
  selected = 1
)

```

Arguments

<code>gen</code>	The generation of the selection candidates. A vector with multiple generations can be provided.
<code>batch</code>	The batch of the selection candidates. Default is 0. It is possible to provide a vector with multiple batches.
<code>select_on</code>	Options: "Index", "EBV" or "Phenotype". For "EBV" and "Phenotype", also the trait need to be specified in 'trait'. Default is Index.
<code>trait</code>	Which trait or EBV the selection is base on when option "EBV" or "Phenotype" is choosen in 'select_on'.

Ntraits	Number of simulated traits. Does not need to be specified if Ntraits is specified in a list called 'BPdata'.
Nm	Number of males to select.
Nf	Number of females to select.
max_FSfam	Maximum number of sibs that can be selected per full sib familie. Default is 'all'.
mature_m	Proportion of male selection candidates that is assumed to be mature and available at the moment of selection. Default is 1
mature_f	Proportion of female selection candidates that is assumed to be mature and available at the moment of selection. Default is 1
selected	The value assigned to ped\$selected for the selected animals, default is 1.

Value

This function will change the data frame called 'ped'. Fish that are selected will be assigned a value to their column 'selected'.

Examples

```
ped <- founderpopfam(Nm = 60,
                    Nf = 60,
                    Nm2 = 0,
                    Nf2 = 0,
                    Ntraits = 2,
                    TraitsIndex = 2,
                    Rgen = matrix(c(1.00 , 0.48,
                                   0.48 , 1.00),
                                  nrow = 2),
                    Rcom = matrix(c(1.00 , 0.5,
                                   0.5 , 1.00),
                                  nrow = 2),
                    Rres = matrix(c(1.00 , 0.32,
                                   0.32 , 1.00),
                                  nrow = 2),
                    mean = c(50,500),
                    a_var = c(200,8000),
                    c_var = c(150,1000),
                    e_var = c(250,12000))

Mating <- randommating(gen = 0,
                      Nfam_FS = 120)

for(fam in 1: nrow(Mating)){
  ped <- offspringFSfam(gen = 1,
                       No = 100,
                       probmale = 0.5,
                       sire = Mating$Sire[fam],
                       dam = Mating$Dam[fam],
                       Ntraits = 2,
                       TraitsIndex = 2,
```

```

      Rgen = matrix(c(1.00 , 0.48,
                    0.48 , 1.00),
                  nrow = 2),
      Rcom = matrix(c(1.00 , 0.5,
                    0.5 , 1.00),
                  nrow = 2),
      Rres = matrix(c(1.00 , 0.32,
                    0.32 , 1.00),
                  nrow = 2),
      a_var = c(200,8000),
      c_var = c(150,1000),
      e_var = c(250,12000)
}

ped <- preselphen(gen = 1,
                 Nenv = 2,
                 Npresel = c(25,15),
                 trait = 1,
                 Ntraits = 2)

ped <- avail_selection(gen = 1,
                      presel = 1,
                      surv = 0.9)

ped <- breeding_values(gen = 1,
                      TraitsIndex = 2,
                      EBV = "GEBV",
                      GenomLength = 11.3,
                      Ne = 100,
                      SizeTraining = nrow(ped[ped$preselected ==2,]),
                      Ntraits = 2,
                      a_var = c(200,8000),
                      h2 = c(0.33,0.38))

ped <- select(gen=1,
             Nm = 60,
             Nf = 60,
             mature_m = 0.5,
             mature_f = 0.4,
             Ntraits = 2)

```

survive

Survive

Description

Function to randomly select which fish from an environment 'survive': fish that don't are not pre-selected anymore for that environment. This is done by assigning 0 to the 'presel' variable of the fish.

Usage

```
survive(gen, batch = 0, preselect, surv = NA, fish_per_FSfam = NA)
```

Arguments

gen	The generation of the fish
batch	The batch of the fish. Default is 0. It is possible to provide a vector with multiple batches.
preselect	Identifies the environment. Fish first need to be pre-selected.
surv	Proportion of fish that is assumed to survive. Either surv or fish_per_FSfam need to be provided.
fish_per_FSfam	The number of fish that survives per full sib family. Fish are randomly selected within a full sib family if fish_per_FSfam is specified. Either surv or fish_per_FSfam need to be provided.

Value

This function will make changes to the data frame 'ped'. Pre-selected fish that do not survive will get a zero in their column 'preselected'.

Examples

```
ped <- founderpopfam(Nm = 60,
                    Nf = 60,
                    Nm2 = 0,
                    Nf2 = 0,
                    Ntraits = 2,
                    TraitsIndex = 2,
                    Rgen = matrix(c(1.00 , 0.48,
                                    0.48 , 1.00),
                                   nrow = 2),
                    Rcom = matrix(c(1.00 , 0.5,
                                    0.5 , 1.00),
                                   nrow = 2),
                    Rres = matrix(c(1.00 , 0.32,
                                    0.32 , 1.00),
                                   nrow = 2),
                    mean = c(50,500),
                    a_var = c(200,8000),
                    c_var = c(150,1000),
                    e_var = c(250,12000))

Mating <- randommating(gen = 0,
                      Nfam_FS = 120)

for(fam in 1: nrow(Mating)){
  ped <- offspringFSfam(gen = 1,
                       No = 100,
                       probmale = 0.5,
                       sire = Mating$Sire[fam],
```

```
dam = Mating$Dam[fam],
Ntraits = 2,
TraitsIndex = 2,
Rgen = matrix(c(1.00 , 0.48,
               0.48 , 1.00),
              nrow = 2),
Rcom = matrix(c(1.00 , 0.5,
               0.5 , 1.00),
              nrow = 2),
Rres = matrix(c(1.00 , 0.32,
               0.32 , 1.00),
              nrow = 2),
a_var = c(200,8000),
c_var = c(150,1000),
e_var = c(250,12000))
}

ped <- preselphen(gen = 1,
                 Nenv = 2,
                 Npresel = c(25,15),
                 trait = 1,
                 Ntraits = 2)

ped <- survive(gen = 1,
              presel = 2,
              surv = 0.8)
```

Index

avail_selection, 2
breeding_values, 4
cor_var, 7
deltaG_F, 8
founderpopfam, 9
founderpopgroup, 12
gen_param, 16
groupmating, 16
offspringFSfam, 18
offspringFSgroup, 21
preselphen, 23
preselrandom, 26
preselcand, 28
randommating, 30
select, 32
survive, 34