**ADAM**

**Multi**

**Program to simulate selective-breeding schemes**

**for animals & plants**

**Developer**

Thinh Tuan Chu

Just Jensen

**Developers of previous version**

Anders Christian Sørensen

Mark Henryon

Huiming Liu

Louise Dybdahl Pedersen

Saeid Ansari-Mahyari

Peer Berg

Aarhus University

Center for Quantitative Genetics and Genomics

Blichers Allé 20

P.O. Box 50

8830 Tjele

Denmark

Seges, Danish Pig Research Centre

Axeltorv 3

1609 Copenhagen

Denmark

**Contact information**

Thinh Tuan Chu

[chu.thinh@qgg.au.dk](mailto:chu.thinh@qgg.au.dk)

+45 7144 9746

Version 272

22/03/2024

**Index**

[1. Program description 6](#_Toc109904111)

[2. Directories 6](#_Toc109904112)

[3. Files 6](#_Toc109904113)

[4. Computer resource 8](#_Toc109904114)

[5. Script file 8](#_Toc109904115)

[6. Input file 9](#_Toc109904116)

[7. Which namelists do I need? 10](#_Toc109904117)

[8. Working process 12](#_Toc109904118)

[9. Genetic model and selection 13](#_Toc109904119)

[10. Realising traits 18](#_Toc109904120)

[11. Founder population and base with genomic information 18](#_Toc109904121)

[12. Sampling and ebv\_obs 19](#_Toc109904122)

[13. Genomic-relationship matrices 28](#_Toc109904123)

[14. Description of namelists 32](#_Toc109904124)

[&OUTPUTDIRECTORY 32](#_Toc109904125)

[&CONTROLPARAMETERS 33](#_Toc109904126)

[&CONTROL\_GENOME 35](#_Toc109904127)

[&LD 37](#_Toc109904128)

[&RANDOMNUMBERSEEDS 47](#_Toc109904129)

[&POPULATIONPARAMETERS 51](#_Toc109904130)

[&BASEPOPULATION 56](#_Toc109904131)

[&SELECTION 59](#_Toc109904132)

[&PHENOTHRESHOLDS 72](#_Toc109904133)

[&RULES 73](#_Toc109904134)

[&FAMILYPARAMETERS 78](#_Toc109904135)

[&EVA 79](#_Toc109904136)

[&EVAPARAMETERS 91](#_Toc109904137)

[&PHENOWEIGHTS 94](#_Toc109904138)

[&GROUPPARAMETERS 95](#_Toc109904139)

[&GENOMICBLUPPARAMETERS 97](#_Toc109904140)

[&IBDBLUPPARAMETERS 101](#_Toc109904141)

[&IBSBLUPPARAMETERS 102](#_Toc109904142)

[&EVAGENOMICRELATIONSHIPPARAMETERS 103](#_Toc109904143)

[&EVAIBDRELATIONSHIPPARAMETERS 104](#_Toc109904144)

[&EVAIBSRELATIONSHIPPARAMETERS 105](#_Toc109904145)

[&BLUPPARAMETERS 106](#_Toc109904146)

[&BAYESPPARAMETERS 108](#_Toc109904147)

[&MATINGPARAMETERS 109](#_Toc109904148)

[&MATRICES 110](#_Toc109904149)

[&DESIGN\_MATRICES 112](#_Toc109904150)

[&COMBINEDOBSERVATIONS 114](#_Toc109904151)

[&OBSERVATIONCONSTRAINTS 116](#_Toc109904152)

[&ECONOMICVALUES 117](#_Toc109904153)

[&OBSERVATIONS 120](#_Toc109904154)

[&CATEGORICALS 128](#_Toc109904155)

[&PSEUDOCATEGORICALS 129](#_Toc109904156)

[&REPORT 130](#_Toc109904157)

[Definition: Write true and estimated breeding values of all 132](#_Toc109904158)

[animals in the population to output file, 132](#_Toc109904159)

[*printBreedingValues* Definition: Write true and estimated breeding values of each 133](#_Toc109904160)

[*printCoVariances* Definition: Write (co)variances estimated by DMU at each 133](#_Toc109904161)

[specified time step and selection step to output 133](#_Toc109904162)

[files, 133](#_Toc109904163)

[‘covariancesRep<*rep*>Time<*time*>Stage<*stage*>.res’, 133](#_Toc109904164)

[15. Examples of input.prm files 140](#_Toc109904165)

[Example 1: Polygenic model; one trait in the breeding objective 140](#_Toc109904166)

[Example 2: Polygenic model; nine traits in the breeding objective 141](#_Toc109904167)

[Example 3: Polygenic model with pseudo-genomic selection; four traits in the breeding objective, ‘genotyped’ animals are truncation selected 141](#_Toc109904168)

[Example 4: Polygenic model with pseudo-genomic selection; four traits in the breeding objective, ‘genotyped’ animals are EVA selected 141](#_Toc109904169)

[16. Output files 142](#_Toc109904170)

[geneticTrendsMeans.res 143](#_Toc109904171)

[geneticTrendsRep<rep>.res (rep=start\_rep … start\_rep+nrep-1) 144](#_Toc109904172)

[geneticTrendsPlots.pdf 145](#_Toc109904173)

[input.log 147](#_Toc109904174)

[seeds.in 147](#_Toc109904175)

[error.txt 147](#_Toc109904176)

[memory.log 147](#_Toc109904177)

[adam\_log.txt 147](#_Toc109904178)

[adamlog.log 147](#_Toc109904179)

[biasAccuracyMeans.res 148](#_Toc109904180)

[biasAccuracyRep<rep>.res (rep=start\_rep … start\_rep+nrep-1) 148](#_Toc109904181)

[breedingValuesRep<rep>.res.bz2 (rep=start\_rep … start\_rep+nrep-1) 149](#_Toc109904182)

[dmuAdamRep<rep>Time<time>Stage<stage>.lst (rep=start\_rep … start\_rep+nrep-1, time=1 … ntime, stage=1 … selection\_groups where breeding values predicted using DMU) 150](#_Toc109904183)

[heterozygosity.dat 150](#_Toc109904184)

[ldSummaryOutputOverTime.dat 151](#_Toc109904185)

[LD\_bin.dat 151](#_Toc109904186)

[ldVersusDistanceOverTime.dat 152](#_Toc109904187)

[QTLdistribution.dat 152](#_Toc109904188)

[markerDistribution.dat 152](#_Toc109904189)

[ldPlots.pdf 153](#_Toc109904190)

[baseHaplotypes.dat 153](#_Toc109904191)

[geneticArchitecture.dat 153](#_Toc109904192)

[animal#.res 154](#_Toc109904193)

[mean\_obs#.res 154](#_Toc109904194)

[mean\_QTL#.res 154](#_Toc109904195)

[mean\_template#.res 154](#_Toc109904196)

[mean\_obs#.res 154](#_Toc109904197)

[mean\_QTL#.res 154](#_Toc109904198)

[mean\_template#.res 154](#_Toc109904199)

[mean\_timesteps\_replicate.res 154](#_Toc109904200)

[obs#.res 154](#_Toc109904201)

[qtl#.res 154](#_Toc109904202)

[template#.res 155](#_Toc109904203)

[template\_info#.res 155](#_Toc109904204)

[References 162](#_Toc109904205)

# Program description

ADAM is a computer program that models selective breeding schemes for animals and plants using stochastic simulation. The program simulates population(s) and traces the genetic changes in the population(s) under different selective-breeding scenarios. It caters to different population structures, genetic models, selection strategies, and mating designs. ADAM can be used to evaluate breeding schemes and generate genetic data to test statistical tools. The program is written in FORTRAN.

# Directories

There are three main directories associated with an ADAM simulation: submission, working, and output directories.

***Submission directory***

Directory from which a simulation is submitted.

***Working directory***

Temporary directory where a simulation is run. This directory is created by the Portable Batch System (PBS) on Foulum’s Linux system.

***Output directory***

Directory to which output files from a simulation are written.

The submission and output directories can be the same directory.

# Files

**Script file**

*adam.sh*, script file that submits a simulation. The directory in which *adam.sh* is saved and from which a simulation is submitted becomes the submission directory.

The following files are located in the submission directory. All files, with the exception of input file, *input.prm*, are optional.

**Input file**

*input.prm*, provides input parameters.

**Seed files**

Files with random seeds when seeds are provided manually. File names are provided in namelist &RANDOMNUMBERSEEDS.

*seedsFile*, input file with seeds when *randomSeeds* ‘file’ in namelist &RANDOMNUMBERSEEDS.

*founderSeedsFile*, input file with seeds when *randomSeeds* ‘manual’ and *founderSeeds* ‘file’.

*baseSeedsFile*, input file with seeds when *randomSeeds* ‘manual’ and *baseSeeds* ‘file’.

*selectionSeedsFile*, input file with seeds when *randomSeeds* ‘manual’ and *selectionSeeds* ‘file’.

These files contains two integer seeds per row (*seed1* and *seed2*), and one row per replicate. Each replicate is initiated with the seeds provided in the files.

**DMU files**

Files required when DMU is used to estimate breeding values. The files required are determined by the breeding values that are to be estimated. For details of the files themselves, refer to the User’s Manual for DMU.

(1) Polgenic breeding values (selection criterion ‘polyblup’ specified in namelists &SELECTION and &EVA of *input.prm*)

*dmu.polyblup.dir*, DMU-input parameters *(name of this file can be set by users)*

*dmu.polyblup.parm* (optional), (co)variance components

The files *dmudat* and *dmuped* (files with DMU-input data and polygenic-pedigree) must be specified in *dmu.polyblup.dir*. These files are created by ADAM.

The file *dmu.polyblup.parm* is optional. It can be used to provide (co)variance components, in which case, it must be specified in *dmu.polyblup.dir*. The alternative is to specify the (co)variance components in *dmu.polyblup.dir*.

(2) Genomic breeding values (selection criterion ‘genomicblup’)

*dmu.genomicblup.dir*, DMU-input parameters *(name of the file can be set by users)*

*dmu.genomicblup.parm* (optional), (co)variance components

The files *dmudat*, *dmuped*, *genotyped.dat*, and *gMatrix.prediction* (files with DMU-input data, polygenic-pedigree, genotyped animals, and genomic-relationships) must be specified in *dmu.genomicblup.dir*. These files are created by ADAM. Option for *genotyped.dat*, and *gMatrix.prediction* can be specified using input from users in namelist *Genomicblupparameter*.

The file *dmu.genomicblup.parm* is optional. It can be used to provide (co)variance components, in which case, it must be specified in *dmu.genomicblup.dir*. The alternative is to specify the (co)variance components in *dmu.genomicblup.dir*.

$COMMENT ADAM-DMU interface

$ANALYSE 12 2 0 0

$DATA ASCII (6,2,-9998.0) dmudat

$VARIABLE

id dam sex mn hys time trait wt

$MODEL

1

0

1 0 2 5 1

1 1

0

0

$VAR\_STR 1 PGMIX 1 ASCII dmuped genotyped.dat gmatrix.prediction 0.25 G-ADJUST

$PRIOR

1 1 1 1.0000001

2 1 1 4.0000000

Files used to estimate polygenic breeding values [*dmu.polyblup.dir* and *dmu.polyblup.parm* (optional)] are also required when polygenic breeding values are estimated instead of genomic breeding values.

(3) IBD breeding values (selection criterion ‘ibdblup’)

*dmu.ibdblup.dir*, DMU-input parameters *(name of this file can be set by users)*

*dmu.ibdblup.parm* (optional), (co)variance components

The files *dmudat*, *dmuped*, *genotyped.dat*, and *gMatrix.prediction* (files with DMU-input data, polygenic-pedigree, genotyped animals, and genomic-relationships) must be specified in *dmu.ibdblup.dir*. These files are created by ADAM.

The file *dmu.ibdblup.parm* is optional. It can be used to provide (co)variance components, in which case, it must be specified in *dmu. ibdblup.dir*. The alternative is to specify the (co)variance components in *dmu. ibdblup.dir*.

(4) EBV prediction program by users

Prediction of EBV for selection can be done by own program from users. Program, scripts and files to run the program must be provided by users.

The files *dmudat*, *dmuped*, *marker.RawData, map.RawData* (files with DMU-input data, polygenic-pedigree, marker data, and map file) are created by ADAM for prediction of EBV.

NB. DMU and EVA files generated by ADAM, DMU, and EVA are copied to the output directory when debugOutput ‘yes’ in namelist &CONTROLPARAMETERS. These files are dmu-script file(s), G-matrix script and parameter files, dmudat, dmuped, map.dat, marker.dat, gMatrix.dat, genotyped.dat, eva.dat, eva.log, eva-solution file, and dmu.lst. These files are copied after they have been created; newly-created files overwrite older files.

# Computer resource

ADAM must be run on 64-bit machine when genomic-breeding values are used as selection criterion in breeding schemes.

# Script file

The following is an example of a script file, adam.sh

#!/bin/bash

#SBATCH -p ghpc # Name of the queue

#SBATCH -J testAdam # job name

#SBATCH --mem=20000 # Memory in MiB(10 GiB = 10 \* 1024 MiB)

#SBATCH -t 100:00:00 # Job max time - Format = MM or MM:SS or HH:MM:SS or DD-HH or DD-HH:MM

#SBATCH -N 1 # number of nodes

#SBATCH -n 2 # number of cores

#SBATCH --output=slurm\_%A.out # STDOUT

#SBATCH --error=slurm\_%A.err # STDERR

#SBATCH --export=ALL

TMPDIR=/scratch/$USER/$SLURM\_JOBID

export TMPDIR

mkdir -p $TMPDIR

cd $TMPDIR

pwd

echo Job started at $(date '+%d\_%m\_%y\_%H\_%M\_%S')

echo $SLURM\_JOB\_NODELIST

echo Job submitted from $SLURM\_SUBMIT\_HOST

echo Submission directory $SLURM\_SUBMIT\_DIR

ulimit -s unlimited

echo number of cores assigned: $SLURM\_NTASKS

export OMP\_NUM\_THREADS=$SLURM\_NTASKS

cp -f /usr/home/qgg/chuthinh/adam/adam2720 adam

echo Running adam

./adam > $SLURM\_SUBMIT\_DIR/adamlog.log

echo adam completed

#=========================================================================#

# Cleanup: DO NOT REMOVE OR CHANGE #

#=========================================================================#

cd $SLURM\_SUBMIT\_DIR

rm -rf /scratch/$USER/$SLURM\_JOBID

echo Job completed at $(date '+%d\_%m\_%y\_%H\_%M\_%S')

# Input file

Complete list of input parameters (namelists) in input file, input.prm:

[&OUTPUTDIRECTORY **Fejl! Bogmærke er ikke defineret.**](#_Toc361213011)

[&CONTROLPARAMETERS **Fejl! Bogmærke er ikke defineret.**](#_Toc361213012)

[&CONTROL\_GENOME **Fejl! Bogmærke er ikke defineret.**](#_Toc361213027)

[&LD **Fejl! Bogmærke er ikke defineret.**](#_Toc361213028)

[&RANDOMNUMBERSEEDS **Fejl! Bogmærke er ikke defineret.**](#_Toc361213013)

[&POPULATIONPARAMETERS **Fejl! Bogmærke er ikke defineret.**](#_Toc361213014)

[&BASEPOPULATION **Fejl! Bogmærke er ikke defineret.**](#_Toc361213015)

[&SELECTION **Fejl! Bogmærke er ikke defineret.**](#_Toc361213016)

[&PHENOTHRESHOLDS **Fejl! Bogmærke er ikke defineret.**](#_Toc361213017)

[&RULES **Fejl! Bogmærke er ikke defineret.**](#_Toc361213019)

[&EVA **Fejl! Bogmærke er ikke defineret.**](#_Toc361213020)

[&EVAPARAMETERS **Fejl! Bogmærke er ikke defineret.**](#_Toc361213020)

&PHENOWEIGHTS

&GENOMICBLUPPARAMETERS

&IBDBLUPPARAMETERS

&IBSBLUPPARAMETERS

&EVAGENOMICRELATIONSHIPPARAMETERS

&EVAIBDRELATIONSHIPPPARAMETERS

&EVAIBSRELATIONSHIPPPARAMETERS

[&BLUPPARAMETERS **Fejl! Bogmærke er ikke defineret.**](#_Toc361213021)

[&BAYESPPARAMETERS **Fejl! Bogmærke er ikke defineret.**](#_Toc361213021)

[&MATINGPARAMETERS **Fejl! Bogmærke er ikke defineret.**](#_Toc361213022)

[&COMBINEDOBSERVATIONS **Fejl! Bogmærke er ikke defineret.**](#_Toc361213023)

[&OBSERVATIONCONSTRAINTS **Fejl! Bogmærke er ikke defineret.**](#_Toc361213023)

[&ECONOMICVALUES **Fejl! Bogmærke er ikke defineret.**](#_Toc361213023)

[&OBSERVATIONS **Fejl! Bogmærke er ikke defineret.**](#_Toc361213023)

[&CATEGORICALS **Fejl! Bogmærke er ikke defineret.**](#_Toc361213024)

[&MATRICES **Fejl! Bogmærke er ikke defineret.**](#_Toc361213025)

[&DESIGN\_MATRICES **Fejl! Bogmærke er ikke defineret.**](#_Toc361213026)

[&REPORT **Fejl! Bogmærke er ikke defineret.**](#_Toc361213040)

# Which namelists do I need?

All simulations require five namelists:

&OUTPUTDIRECTORY

&CONTROLPARAMETERS

&RANDOMNUMBERSEEDS

&ECONOMICVALUES

&REPORT

The remaining required namelists are determined by the variable *geneticModel* in namelist &CONTROLPARAMETERS. geneticModel can be ‘polygenic’, ‘genomic’, or ‘ldonly’.

**polygenic, genomic, qtl**

&POPULATIONPARAMETERS

&BASEPOPULATION

&MATINGPARAMETERS

&OBSERVATIONS

&CATEGORICALS

&MATRICES

&DESIGNMATRICES

&SELECTION

&PHENOTHRESHOLDS

&PHENOWEIGHTS

&RULES

*if phenotypic selection (thresholds)*

*if phenotypic selection (weights)*

*if selection rules*

*if categorical traits*

*if BLUP-estimation*

&EVA

&BLUPPARAMETERS

*if EVA selection*

*if BLUP-estimation*

*if genomic-EVA*

&EVAPARAMETERS

&EVAGENOMICRELATIONSHIPS



# Working process

Read input from input file, input.prm

Write input to input log file, input.log

Start replicate i (i=start\_rep, start\_rep+nrep)

Generate random seeds

Generate base population

Start timestep *j* (*j*=1-*gestation\_length*, *ntime*)

Realise observations

Control herd sizes

EBV estimation

Selection

Mating

Sample offspring

Culling

Report replicate

timestep=*ntime*

timestep<*ntime*

Final report

replicate=start\_rep+nrep

replicate<start\_rep+nrep

if phenotyping

Realise maternal observations

Update reproductive cycles

Update reproduction statuses

Generate founder population

Generate random seeds for founder population

Generate random seeds for base population

Generate random seeds for selection

# Genetic model and selection

Genetic model is the method used to generate breeding values. Selection involves the method used to select animals and the criterion by which animals are ranked. The selection criterion is often an estimated breeding value derived from a statistical analysis.

*Genetic models*. There are two genetic models available:

(a) Polygenic model

(b) Genomic model with linkage disequilibrium between QTL and markers

An additional option, ‘linkage disequilibrium only’ (ldonly), is available

to generate linkage disequilibrium with the genomic model. It is used to

study linkage disquilibrium; a breeding scheme is not simulated.

*Selection methods*. There are three selection methods:

(a) Threshold selection

(b) Truncation selection

(c) Optimum-contribution selection (EVA selection)

*Selection criteria*. There are eight selection criteria:

(a) True breeding value

(b) Random variable

(c) Null variable

(d) Phenotypic observation(s) with associated threshold(s)

(e) Phenotypic observation(s) with associated weight(s)

(f) Polygenic-BLUP breeding value

(g) Genomic-BLUP breeding value

(h) IBD-breeding value

(j) BayesP

*Genetic model and selection method*. All three selection methods are available with each genetic model.

*Genetic model and selection criterion*. The genetic model determines the selection criteria that are available.

(a) Polygenic: (i) True breeding value

(ii) Random variable

(iii) Null variable

(iv) Phenotypic observation(s) with associated threshold(s)

(v) Phenotypic observation(s) with associated weight(s)

(vi) Polygenic-BLUP breeding value

(b) Genomic: (i) True breeding value

(ii) Random variable

(iii) Null variable

(iv) Phenotypic observation(s) with associated threshold(s)

(v) Phenotypic observation(s) with associated weight(s)

(vi) Polygenic-BLUP breeding value

(vii) Genomic-BLUP breeding value

(viii) IBD-BLUP breeding value

(x) BayesP

*Selection method and selection criterion*. The selection method determines the selection criteria that are available.

(a) Threshold selection: Phenotypic observation(s) with associated threshold(s)

(b) Truncation selection: (i) True breeding value

(ii) Random variable

(iii) Phenotypic observation(s) with associated weight(s)

(iv) Polygenic-BLUP breeding value

(v) Genomic-BLUP breeding value

(vi) IBD-breeding value

(viii) BayesP

(c) EVA selection: (i) True breeding value

(ii) Null variable

(iii) Polygenic-BLUP breeding value

(iv) Genomic-BLUP breeding value

(v) IBD-breeding value

(vii) BayesP

*Selection criteria and phenotypic and genotypic observations.* Selection and the selection criterion used to rank animals can be influenced by the phenotypic and genotypic observations that are available.

(a) True breeding value, random variable, and null variable

Selection candidates are always selected

(b) Phenotypic observation(s) with associated threshold(s) or weight(s)

Selection candidates must have all specified phenotypes observed to be

selected

(c) Polygenic breeding value

Polygenic breeding values requires that animals have phenotype(s) observed.

When they are not, a random variable is used.

 (d) IBD breeding value

IBD breeding values requires that animals have phenotype(s) observed.

When they are not, a random variable is used. ~~All selection candidates are~~

~~presumed to be genotyped.~~

(d) Genomic breeding value

Genomic breeding values requires that animals have phenotypes and genotypes

observed. When they are not observed, polygenic breeding value or random

variable is used.



If relationship used to constrain rate of in breeding in EVA specified as ‘genomic’, but not animals are genotyped, ‘pedigree’ is used.

(e) BayesP

BayesP breeding value requires that animals have phenotypes observed (all

animals are presumed genotyped ~~at birth~~). When they are not observed,

random variable is used.

*geneticModel* must be ‘genomic’, *nebv*==1, and *genotyped\_at\_birth*=='yes'

SUBROUTINE blupSelectionCriteria to see which *selection\_criteria* are compatable in the same breeding scheme

# Realising traits

Phenotypic observations can be realised in three ways:

1) At the start of each time step; specified in namelist &OBSERVATIONS

2) At selection; specified in namelists &SELECTION, &EVA, and &OBSERVATIONS

3) As maternal traits; specified in namelist &OBSERVATIONS

There is a variable *realiseBaseObservations* in namelist &BASEPOPULATIONS that enables phenotyping of base animals that fulfil the criteria specified in namelist &OBSERVATIONS.

# Founder population and base with genomic information

With genomic information, *geneticModel* ‘genomic’

*Founder population*. Multiple populations with different allele frequency between populations can be used.

*Genomic-base population*. Genomic-base animals/population are/is generated by sampling from *nchrom* x *ploidy* (>=2) pools of chromosome haplotypes generated in the *LD\_npat* and *LD\_nmat* founder population. When we generate a founder population with *LD\_npat* and *LD\_nmat* founder males and females. This generates *LD\_npat* x *ploidy* (>=2) (paternal and maternal) and *LD\_nmat* x *ploidy* (>=2) haplotypes for male and female founder animals for the *nchrom* chromosomes. Sex of founder population is ignored. Users put equal number of males and females that have the same allele frequency.

We sample QTL effects so that the QTL/genomic (co)variances equal those specified by the user assuming an overall allele frequency from all founder populations. If this assumption were not desirable, users could provide qtl effects for each allele. We assume in each locus there are two alleles 1 and 2.

For each genomic-base animal, we sample paternal and maternal haplotypes/chromosome arms for each chromosome by sampling haplotypes/chromosome arms from the *nchrom* x *ploidy* pools, size *LD\_npat* x *ploidy* and *LD\_nmat* x *ploidy*.

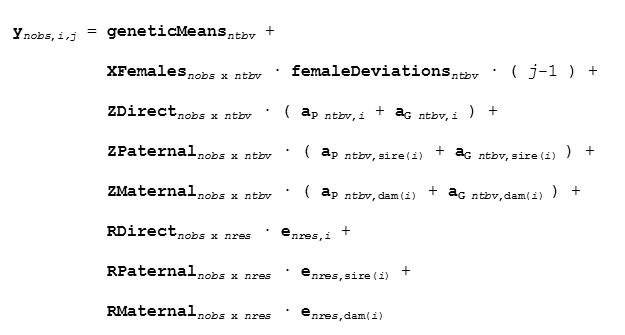
For base animal id …

# Sampling and ebv\_obs

(i) Observations

Symbol *a* refers to breeding values in additive models, or genetic values (a+d) in dominance model.

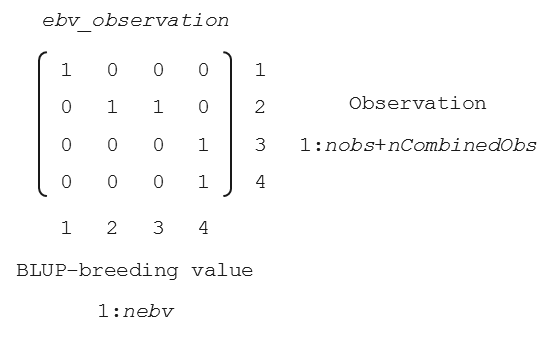
Vector of *nobs* observations, **y***nobs*,*i*,*j* , realised for animal *i* with sex *j* (*j* = 1 for males, 2 females), sire(*i*), and dam(*i*):



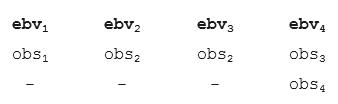
where

***(ii) ebv\_observation***

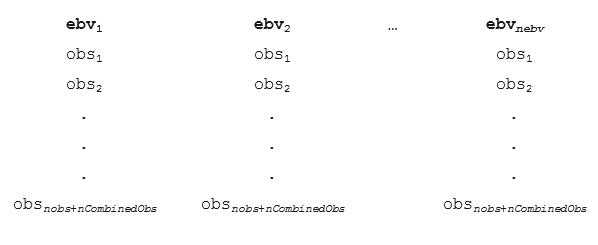
The *nobs*+*nCombinedObs* x *nebv* matrix *ebv\_observation*, specified in namelist DMUPARAMETERS, variable *ebv\_observation,* is used to create DMU-input data (*dmudat*) and retrieve BLUP-breeding values estimated by DMU. [Translation/Transition matrix between ADAM and DMU] Elements of *ebv\_observation* associate observations with BLUP-breeding values. Elements can be ‘0’ or ‘1’.



*DMU-input data*. Element *ebv\_observation*(*i,j*) (*i* = 1 … *nobs*+*nCombinedObs*, *j* = 1 … *nebv*) set to ‘1’ results in the *i*th observation for the *k*th animal being written to DMU-input data under ebv *j*~~, where a trait is a characteristic for which a BLUP-breeding value is estimated~~. The structure of the input data for the *k*th animal with *ebv\_observation* (above) is:



The general structure for DMU-input data is:



where the columns of the input data correspond to the columns of *ebv\_observation* and the observations written to each column correspond to rows within each column of *ebv\_observation* set to ‘1’.

Columns of *ebv\_observation* with ‘all-zeros’ (i.e., no elements set to ‘1’) result in the associated estimated breeding value not being written to the input data. Rows of *ebv\_observation* with ‘all-zeros’ result in the associated observations not being written.

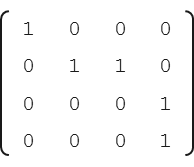
If the *i*th row has multiple elements set to ‘1’, the *i*th observation for the *k*th animal is written to DMU-input data under multiple traits; traits with columns in row *i* of *ebv\_observation*(*i,j*) set to ‘1’. If the *j*th column has multiple elements set to ‘1’, observations *n* for the *k*th animal are written under the *j*th trait, where the *n* observations are a subset of rows in column *j* with *ebv\_observation*(*i,j*) set to ‘1’.

*BLUP-breeding values*. Columns containing a ‘1’ requires the *j*th BLUP-breeding values to be estimated by DMU.

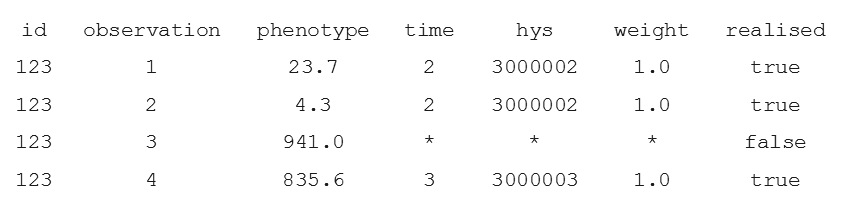
*ebv\_observation* and the model fitted to the DMU-input data by DMU must be aligned. The model fitted by DMU must be aligned with the DMU-input data and the *nebv* BLUP-breeding values expected to be estimated.

**Example 1: Typical DMU-input data and DMU model. Includes genetic-maternal effect.**

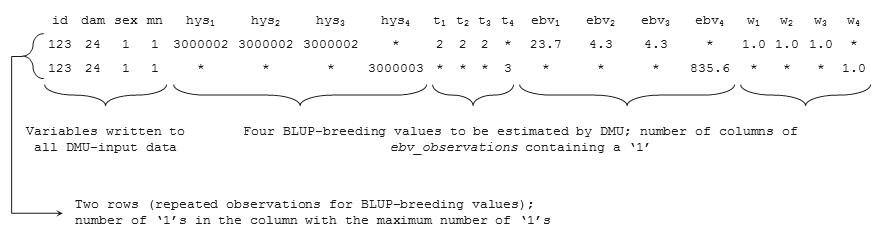
*ebv\_observation*



Phenotypic observations generated and realised for an animal ~~during a simulation~~



DMU-input data

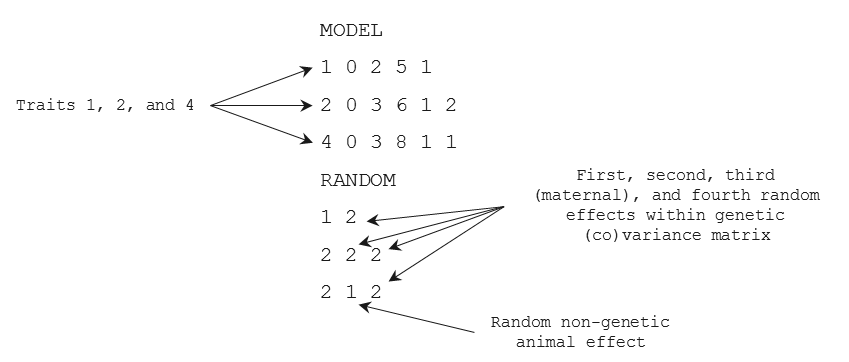


where trait*i* is the *i*th characteristic for which a BLUP-breeding value is estimated, hys*i* and t*i* are herd-year-season and time associated with the *i*th trait, and \* is the largest representable integer for hys*i* and t*i* and -9999.0 for obs*i* and w*i*.

Note that observation 2, associated with BLUP-breeding values 2 and 3, is written twice.

Observation must be realised and, if specified, realised within *previousObsTimeSteps*, otherwise observation set to -9999.0. Fixed variables set to huge(1).

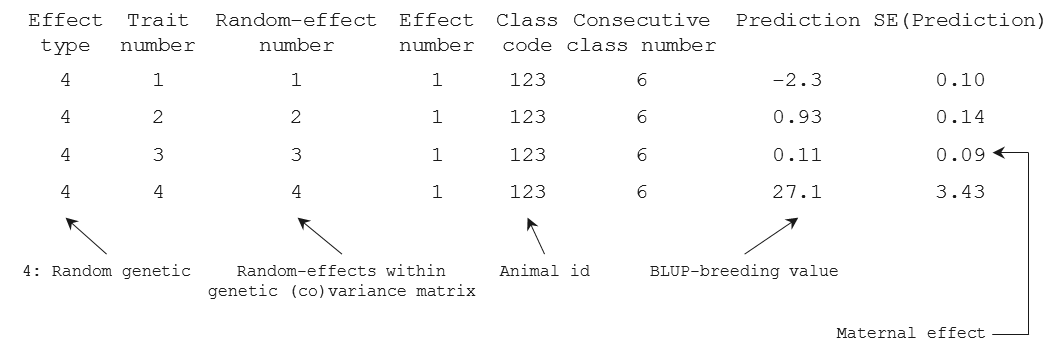
DMU model



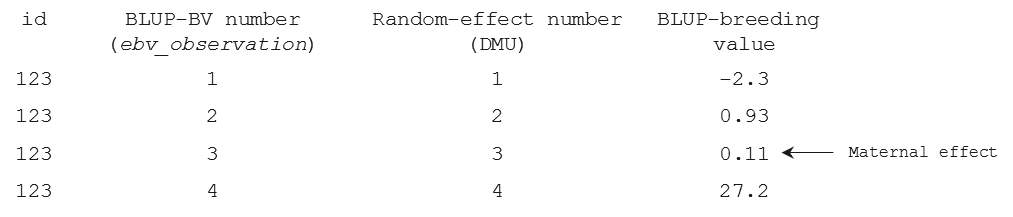
*The prior variances and covariances must be specified in random factor number sequence i.e. priors for random factor 1 must be specified before priors for random factor 2 and so on. Each line consists of 3 integers a real number (free format). The first integer is the random factor number followed by row-column (trait) combination and finally the prior (co)variance.*

Model must be specified with the order of the BLUP-breeding values to be estimated (random effect within genetic (co)variance matrix) 1:*nebv*.

Solution (SOL) file generated by DMU

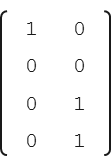


BLUP-breeding values transferred from solution file

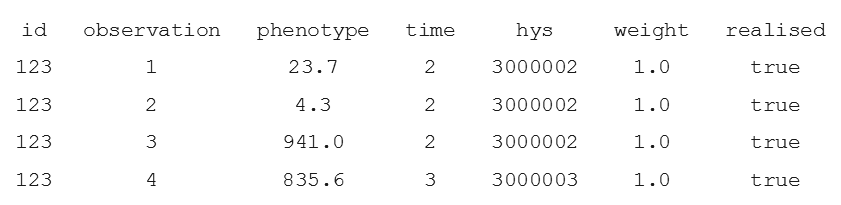


**Example 2: Phenotypic observation not written to DMU-input data**

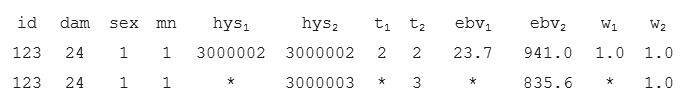
*ebv\_observation*



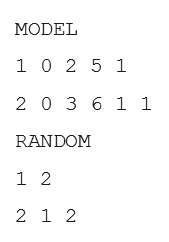
Phenotypic observations



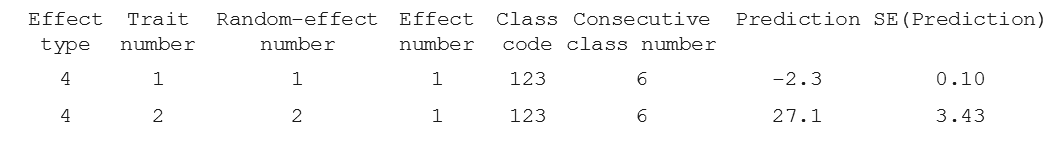
DMU-input data



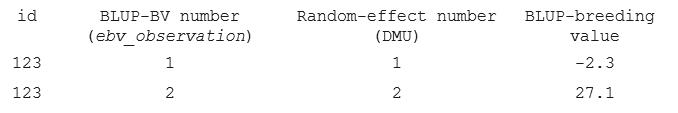
DMU model



Solution (SOL) file generated by DMU

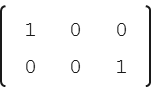


BLUP-breeding values transferred from solution file

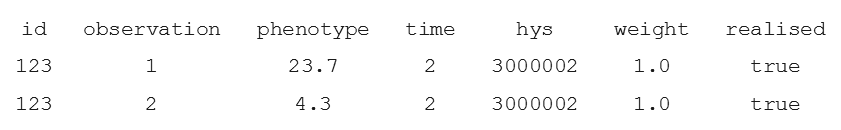


**Example 3: BLUP-breeding value not estimated by DMU**

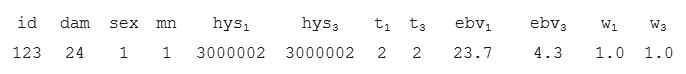
*ebv\_observation*



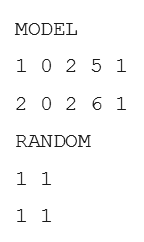
Phenotypic observations



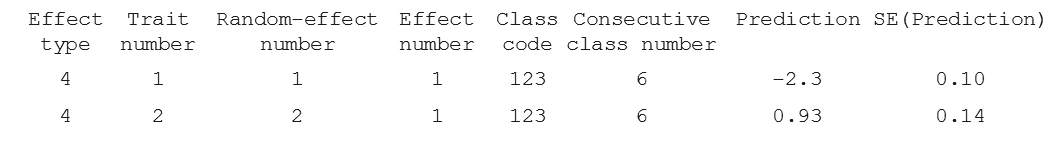
DMU-input data



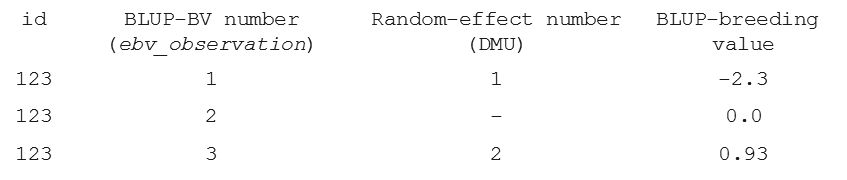
DMU model



Solution (SOL) file generated by DMU



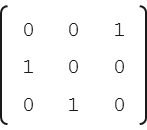
BLUP-breeding values transferred from solution file



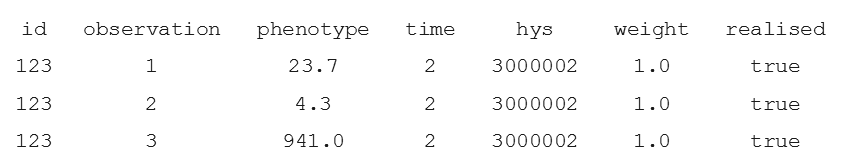
NB. BLUP-breeding value 2 was not estimated by DMU. It was set to 0.0.

**Example 4: Unusual order of *ebv\_observation***

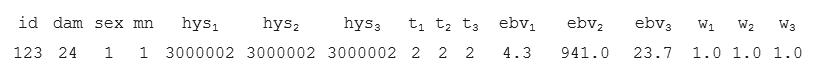
*ebv\_observation*



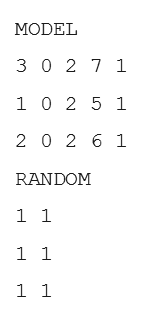
Phenotypic observations



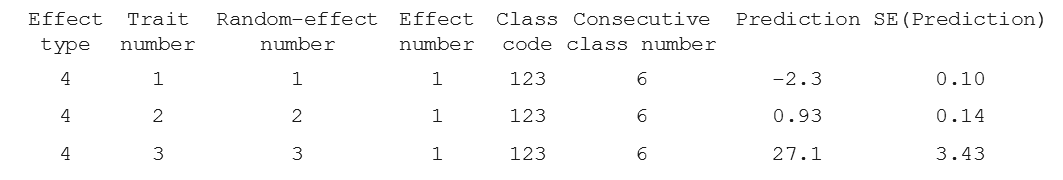
DMU-input data



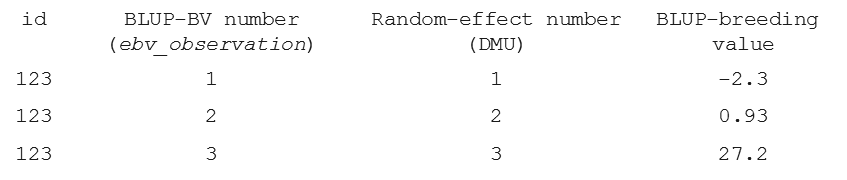
DMU model



Solution (SOL) file generated by DMU

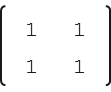


BLUP-breeding values transferred from solution file

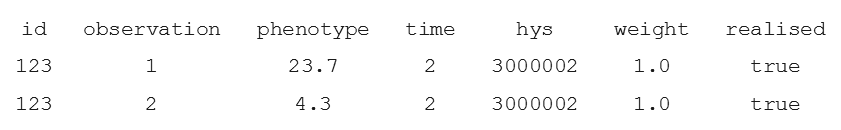


**Example 5: Excessive paramaterisation; impractical example**

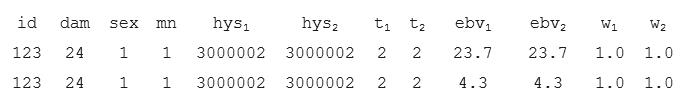
*ebv\_observation*



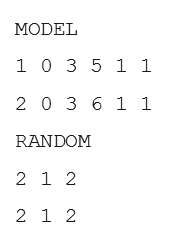
Phenotypic observations



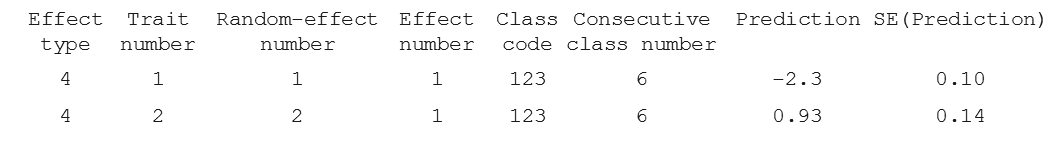
DMU-input data



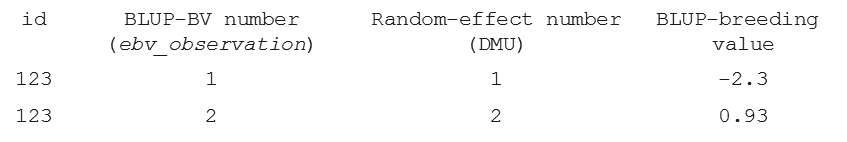
DMU model



Solution (SOL) file generated by DMU



BLUP-breeding values transferred from solution file



If own prediction program used:

Input file are specified as above. The output file must have the following format:

* no header
* col1: ebvNumber/trait number (1<=value<=nebv).
* Col2: ID.
* Col3: EBV (real)

Phenotypic observations can be realised in three ways:

Hvis du har en fænotype, som både har direkte og maternelle effekter, og de maternelle effekter både er genetiske og miljømæssige, så skal du bruge to tbv'er og to res'er pr. obs.

[10:31:44] Anders Christian Sørensen: Det samme hvis du har en fællse kuldeffekt, så vil det modelleres som en ekstra residual hos moderen.

[10:32:05] Anders Christian Sørensen: nres kan altså sagtens tænkes at være forskellig fra nobs

obs(:)=MATMUL(REAL(Zbv,sp),pop(id)%tbv\_poly) + MATMUL(REAL(Wres,sp),pop(id)%res) ! Add bv+residual

if (pop(id)%sex==2) obs(:) = obs(:) + MATMUL(REAL(xmean,sp),sexdif)

dam=pop(id)%dam

if (dam/=0) then

obs(:) = obs(:) + MATMUL(REAL(Zmatbv,sp),pop(dam)%tbv\_poly) + MATMUL(REAL(Wmres,sp),pop(dam)%res)

! add maternal genetic and environmental effects

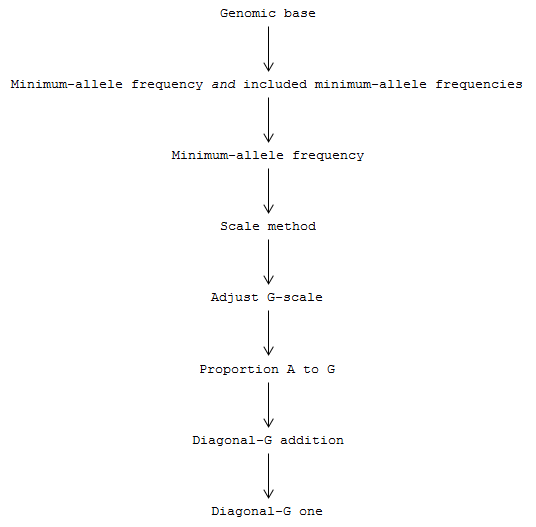
end if

# Genomic-relationship matrices

Genomic-relationship matrices are constructed as **G** = **ZZ'**/s, where **G** is a *n* x *n* matrix of genomic relationships between *n* animals, **Z** = **M**-**1**(2**p**)**'**, **M** is a *n* x *m* matrix of counts of the mutant allele for the *n* animals at the *m* marker loci with element M*ij* = 0, 1, or 2 for animal *i* at marker locus *j* (*i* = 1 … *n*, *j* = 1 … *m*), **1** is a *n* vector of ones, **p** = (p1, p2 … p*m*) is a *m* vector with p*j* the frequency of the mutant allele at marker locus *j* ~~in the base population~~, and s ~~= 2~~**~~p'~~**~~(~~**~~1~~**~~-~~**~~p~~**~~) transforms~~ scales **G** towards the same scale as a pedigree-relationship matrix (?)(adapted from VanRaden 2008).

ADAM allows different matrices to be created.

The following flow diagram outlines the sequence used to generate genomic-relationship matrices used to predict genomic-breeding values, control rates of inbreeding using genomic information and EVA, and monitor rates of inbreeding:



Let **X***m*,*n* be a *m* x *n* matrix of marker genotypes for *n* individuals at *m* loci. The marker genotype for individual *j* at locus *i* is denoted x*i*,*j* = 0, 1, or 2 when the genotype is 11, 12, or 22 and marker alleles ~~at locus~~ *~~i~~* are denoted 1 and 2.

**Alternative: X***n*,*m USE ALTERNATIVE, more conventional*

, where …

genomic base define the animals included in the genomic-base population that are

used to calculate marker-allele frequencies, p*i* and 1-p*i*, used to ’centre’ and ’scale’ genotypes at locus *i*,

all All genotyped animals in the population,

<included in dmudat>

base All animals in the base population

ignoring genotyping

genotypedbase All genotyped animals in the base

population

tracedbase All animals in the base population

<included in dmudat> ignoring genotyping

tracedgenotypedbase All genotyped animals in the base

population <included in dmudat>

*loci*, *maf*, and *mafInclude* defines the loci used to construct genomic matrices.

*loci* defines the loci used as genetic markers (QTL and markers, QTL, or markers),

*maf* Threshold for minimum-allele frequency

*mafInclude* Minimum-alle frequency of loci included.

*loci* all QTL and markers

qtl QTL

markers Markers

*maf* Definition: Threshold for minimum-allele frequency

*mafInclude* Definition: Minimum-alle frequency of loci included

above Include loci with minimum-allele frequency

greater than or equal to *maf*

below Include loci with minimum-allele frequency

less than or equal to *maf*

*Centring*

Marker genotypes are centred by substracting the mean of the marker genotypes:

x*ij* = x*ij* – 2p*i*, where p*i* is frequency of allele 2 at locus *i* (and 2p*i* is the mean of the marker genotype at locus *i*, which is the same as averaging marker genotypes at locus *i* across *n* individuals). In matrix notation, this is **X***m*,*n* = **X***m*,*n* – (2**p)1***n*T, where **p** is a column vector **p** = (p1, p2 … p*m*) and **1***n*T is a 1 x *n* ~~sum vector~~ transposed ~~column~~ sum vector having all of its elements equal to one. ~~[IS p FREQUENCY AT ALLELE 2 OR MAF? WHAT DOES SU’S PROGRAM DO?]~~

Alternative: **X***n*,*m* – **1***n*(2**p)**T

Equivalent is **X***n*,*m* be a *n* x *m* matrix of marker genotypes for *n* individuals at *m* loci. The marker genotype for individual *j* at locus *i* is denoted x*i*,*j* = -1, 0, or 1 when the genotype is 11, 12, or 22 and marker alleles ~~at locus~~ *~~i~~* are denoted 1 and 2. Marker genotypes are centred by substracting the mean of the marker genotypes:

x*ij* = x*ij* – (2p*i* – 1). In matrix notation, this is **X***n*,*m* = **X***n*,*m* – **1***n* (2**p** - 1**)**T.

Either way, end up with deviations from the respective means that that are the same. Centering sets mean values of the allele effects [genotypes?] to 0.

*locusWeight* is the weight applied to each locus. How? [Order!]

one Weight 1.0 applied to each locus; each locus has

equal weight

goddard Weight applied to each locus as a function of allele

frequency, 1.0/(p*i*(1-p*i*)), where p*i* is the minimum-

allele frequency at locus *i* (after Goddard 2009)

jannink Weight applied to each locus as a function of allele

frequency, (arcsin(1)-arcsin(sqrt(p*i*)))/(p*i* (1-p*i*))

(after Jannink 2010)

Default: *locusWeight* must be specified

*scaleMethod* Definition: Method to scale **G**-matrix

Type: Integer

Options:

1 Divide **G** by sum of 2p(1-p)

2 Divide each locus genotype by sqrt(2p*i*(1-p*i*)), where p*i*

is frequency of an allele at locus *i*

Default: *scaleMethod* must be specified

*Scaling (locusWeight* and *scaleMethod)*

Marker genotypes are standardised to a unit variance

scaleMethod 1: **X***n*,*m*∙**X***n*,*m*T s

x*ij* = x*ij*/s (*scaleMethod* 1), where s = ν*i*/, or x*ij* = x*ij*/ (*scaleMethod* 2. In matrix notation, this is **X***m*,*n* = **X***m*,*n*/s and **X***m*,*n* = **X***m*,*n*/. Note: s is a constant, there is no rescaling of markers relative to each other. Or is it: **X***m*,*n* = **X***m*,*n*/, where ν*i* is the weight applied to locus *i*.

Alternative: **X***n*,*m* = **X***n*,*m*/s and **X***n*,*m* = **X***n*,*m*/

G = **X***m*,*n*T**X***m*,*n*

Alternative: G = **X***n*,*m* **X***n*,*m*T

**If weighting** then

({0,1,2}-2p)\*SQRT(weight/(2p(1-p))

Then XXT

But

*scaleGToA* Definition: Adjust G to same scale as A

Type: Integer

Options:

Default: *scaleGToA* must be specified

*Adjusting marker-based relationship matrix*

Parameters α and β are estimated by fitting **G** to **A** and solving two equations:

 and ,  and  are means of all elements in the two matrices, and *d***G** and *d***A** are means of diagonal elements in the two matrices.

*propAToG* Definition: Proportion of A-matrix to modify G-matrix

Type: Real

Options: 0.0≤*propAToG*≤1.0

Default: *percentAToG* must be specified

*addDiagG*  Definition: Value added to diagonal elements of G-matrix *<to*

*make the matrix positive-definite>*

Type: Real

Options: *addDiagG*≥0.0 *[Set to 0.0 if A-matrix used]*

Default: *addDiagG* must be specified

*diagGOne* Definition: Scale mean of diagonal elements of G-matrix so

the mean of the diagonal equals *diagGOne*

Type: Real

Options: *diagGOne*≥0.0, where 0.0 does not bring about scaling

Default: *diagGOne* must be specified

Genomic-matrix **G**: The genomic-inbreeding coefficient for individual *i* is **G***ii* – 1. Genomic relationships between individuals *i* and *j*, which are analogous to the relationship coefficients of Wright (1922), are obtained by dividing elements **G***ij*/ … Wright’s formulae is A*ij* = 2ƒ*ij*/, which means that off-diagonal elements are equal to 2ƒ*ij*?

*Questions*

*Do we account for inbreeding coefficient of parents when using off-diagonal elements of* ***G*** *(parents) to calculate inbreeding coefficients of offspring or is the inbreeding coefficient of offspring simply equal to* ƒ*ij (i.e., pop(iid)%genomicFParent(i)=0.5\*r)?*

*Is it better to work with* ***G*** *accounting for inbreeding,* that is, A*ij* = 2ƒ*ij*/?

*What if several markers that have correlation 1?*

**IBD and IBS**

IBD and IBS between two individuals: , where  is the allele sharing status at locus *k*, which is equal to 1 if allele *i* form the first individual is identical to the allele *j* from the second individual, and 0 otherwise.

Use ½ to realise relationship (¼ realises coancestry)

# Description of namelists

## **&OUTPUTDIRECTORY**

OutDirectory=*OutDirectory* /

\*\*Task\*\*

Destination for output files

\*\*Properties of name\*\*

*OutDirectory* Definition: Directory for output

Type: Character

Options: Any existing and accessible directory

Default: Submission folder is assumed ($PBS\_O\_WORKDIR, $SLURM\_SUBMIT\_DIR, or $ADAM\_SUBMIT\_DIR)

\*\*Additional information\*\*

*OutDirectory* must be enclosed in quotation marks (“” or ‘’) and end with a forward

slash (/). For example, OutDirectory=’/bob/adam/test/’

## **&CONTROLPARAMETERS**

geneticModel=*geneticModel*

nrep=*nrep*

start\_rep=*start\_rep*

ntime=*ntime*

cull\_males=*cull\_males*

timeStepsSeason=*timeStepsSeason*

constant\_mendelian\_variance=*constant\_mendelian\_variance*

bisexualspecies=*bisexualspecies*

qtlEffectsProvided=*qtlEffectsProvided*

SimulateDominance=*SimulateDominance*

SimulatePairEpistasis=*SimulatePairEpistasis*

genomicEffectModel*=genomicEffectModel*

ModelOfParameters*=* *ModelOfParameters*

nTotal=*nTotal* /

\*\*Task\*\*

Input that controls the physical aspects of the simulation

\*\*Properties of names\*\*

*geneticModel* Definition: Genetic model used to generate breeding values

Type: Integer

Options:

polygenic Polygenic model

genomic Genomic model with linkage disequilibrium

between QTL and markers

ldonly Genomic LD used to generate linkage

disquilibrium; breeding scheme is not simulated

Default: polygenic

*nrep* Definition: Number of replicates simulated. Also applies to

the number of replicates in the LD-generating

routine when *geneticModel* ‘ldonly.

Type: Integer

Options: ≥1

Default: 1

*start\_rep* Definition: Starting replicate number

Type: Integer

Options: ≥1

Default: 1

*ntime* Definition: Number of time steps

Type: Integer

Options: ≥1

Default: 1

*cull\_males* Definition: Defines whether surplus males are stored in

memory

Type: Integer

Options:

0 Sampled males are stored in memory

1 Sampled males are only stored if the dam was

selected as a bull dam.

Default: 0

*timeStepsSeason* Definition: Number of time steps per season; used to

simulate HYS effects

Type: Real

Options: ≥0.0

Default: 1.0

*constant\_mendelian\_variance*

Definition: Determines whether inbreeding of parents

reduces the variance of the Mendelian

sampling term of the offspring

Type: Character

Options:

no Uses inbreeding of parents to determine the

variance of the Mendelian sampling term of the

offspring

yes Uses 0.5 for variance [half of the additive genetic (co)variance(s)] of the Mendelian sampling term of the offspring independent of the

inbreeding of the parents [Mendelian-sampling terms for offspring are independent of the inbreeding of the parents; Mendelian-sampling variance is half genetic variances, specified for **G**]

Default: no

*bisexualspecies* Definition: species is bisexual

Type: character

Options: yes: individuals are bisexual. Can be both sire & dam

no: individuals are either males or females (sire or dam)

Default: no

*genomicEffectModel*  Type: Integer

Options: 0, 11, 12

Default: 0 or 11

Definition:

! 0 (default) genomic model not simulated.

! 11 (default) Simulation models based on QTL effects (not subtitution effect) for each allele in the locus.

! Effects of each allele/ pairEpis sample from distributions

! count number of each allele (x)

! calculate covariate for effects:

! additive: xA = (x - ploidy/2) (2 / ploidy)

! Dominance: xD = x\*(ploidy - x) (2 / ploidy)^2

! Epistasis: xAA = xA1k \* xA1l + xA1k \* xA2l + xA2k \* xA1l + xA2k \* xA2l ! Note: nAlleleInLocus=2. If nAlleleInLocus>2, it will be pairwise effects of locus is the sum of effects \* covariate.

! 12 Simulation models based on biological subtitution effect of one allele over other allele.

! This model requires nAlleleInLocus=2

! Effects of each allele/ pairEpis sample from distributions

! count number of allele 1 (x)

! calculate covariate for effects:

! additive: xA = (x - ploidy/2) (2 / ploidy)

! Dominance: xD = x\*(ploidy - x) (2 / ploidy)^2

! Epistasis: xAA = xAk \* xAl

! effects of locus is the sum of effects \* covariate.

*qtlEffectsProvided*  Type: Integer

Options: 0, 3, 30

Default: 0

Definition:

QTL effects provided by users (in file geneticArchitecture(Rep(rep)).dat) or generated by ADAM to simulate tbv traits.

Options:

0 qtl effects simulated by defaults from ADAM

11 qtl effects provided for nAlleleInLocus alleles in each locus (nAlleleInLocus values for each tbv). nploidy can be greater than 2.

corresponding to genomicEffectModel=11

Example: nAlleleInLocus=3, two traits. Format of file geneticArchitecture(Rep(rep)).dat:

READ (UNIT=io\_dat,FMT=\*,IOSTAT=ier) seg\_locus(i)%al\_freq, seg\_locus(i)%link\_group, seg\_locus(i)%position, seg\_locus(i)%qtl, seg\_locus(i)%a\_qtlInput(1:ntbv,1), seg\_locus(i)%a\_qtlInput(1:ntbv,2), seg\_locus(i)%a\_qtlInput(1:ntbv,3), seg\_locus(i)%d\_qtlInput(1:ntbv,1), seg\_locus(i)%d\_qtlInput(1:ntbv,2), seg\_locus(i)%d\_qtlInput(1:ntbv,3)

Note that seg\_locus(i)%al\_freq has dimension of 3 (3 values). d\_qtlInput are optional depennding on SimulateDominance

Format of file episPairLociEffect(Rep(rep)).dat (optional depennding on SimulatePairEpistasis):

READ (UNIT=io\_dat,FMT=\*,IOSTAT=ier) nEpisPairLoci

READ (UNIT=io\_dat,FMT=\*,IOSTAT=ier) EpisPairLoci(ipair)%locusOne,EpisPairLoci(ipair)%locusTwo,

episEff\_AA(:,:,tbv1),episEff\_AA(:,:,tbv2),...episEff\_AA(:,:,ntbv) ! SimulatePairEpistasis(1) /=0

episEff\_AD(:,:,tbv1),episEff\_AD(:,:,tbv2),...episEff\_AD(:,:,ntbv) ! SimulatePairEpistasis(2) /=0

episEff\_DD(:,:,tbv1),episEff\_DD(:,:,tbv2),...episEff\_DD(:,:,ntbv) ! SimulatePairEpistasis(3) /=0

12 qtl effects provided for nAlleleInLocus=2

corresponding to genomicEffectModel=12

Example: two traits. Format of file geneticArchitecture(Rep(rep)).dat:

READ (UNIT=io\_dat,FMT=\*,IOSTAT=ier) seg\_locus(i)%al\_freq, seg\_locus(i)%link\_group, seg\_locus(i)%position, seg\_locus(i)%qtl, seg\_locus(i)%a\_qtlInput(1:ntbv,1),

seg\_locus(i)%d\_qtlInput(1:ntbv,1)

Format of file episPairLociEffect(Rep(rep)).dat (optional depennding on SimulatePairEpistasis):

READ (UNIT=io\_dat,FMT=\*,IOSTAT=ier) nEpisPairLoci

READ (UNIT=io\_dat,FMT=\*,IOSTAT=ier) EpisPairLoci(ipair)%locusOne,EpisPairLoci(ipair)%locusTwo,

episEff\_AA(1:ntbv), ! SimulatePairEpistasis(1) /=0

episEff\_AD(1:ntbv), ! SimulatePairEpistasis(2) /=0

episEff\_DD(1:ntbv) ! SimulatePairEpistasis(3) /=0

*SimulateDominance*  Type: Integer

Options: 1

Default: 0

Definition:

! 1 : dominance simulated. Only work with genomic models.

! 0 : (default) Dominance not simulated.

*SimulatePairEpistasis*  Type: Integer(3)

Dimension: 3

Options: 1, 1, 1

Default: 0 0 0

Definition:

1 1 1 : Epistasis simulated (AA, AD, DD)

0 0 0 : (default) Epistasis not simulated.

*nTotal*  Type: Integer

Options: >1000

Default: 0 (optional)

Definition:

! >1000 : a rough estimate of max total number of indivdiuals in pops. (unnecessary but for memory efficient purposes)

! 0 : (default) Don’t need to provide. Be careful if this number is used.

\*\*Relation to subsequent namelists\*\*

*geneticModel* is a control parameter; it largely determines the subsequent

namelists required

polygenic Required &POPULATIONPARAMETERS, &BASEPOPULATION, &SELECTION,

&MATINGPARAMETERS, &OBSERVATIONS, &MATRICES, and

&DESIGN\_MATRICES

Optional &RULES, &EVA, &BLUPPARAMETERS, &PHENOTHRESHOLDS,

&PHENOWEIGHTS, and &CATEGORICALS

genomic Required &POPULATIONPARAMETERS, &BASEPOPULATION, &SELECTION,

&MATINGPARAMETERS, &OBSERVATIONS, &MATRICES, and

&DESIGN\_MATRICES, [&CONTROL\_GENOME](http://adam.agrsci.dk/manual_website.htm#CONTROL_GENOME), and &LD

Optional &RULES, &EVA, &BLUPPARAMETERS, &PHENOTHRESHOLDS,

&PHENOWEIGHTS, and &CATEGORICALS

ldonly Required [&CONTROL\_GENOME](http://adam.agrsci.dk/manual_website.htm#CONTROL_GENOME) and &LD

\*\*Additional information\*\*

When *geneticModel* ‘ldonly’, *geneticModel*, *nrep*, and *start\_rep* are the only parameters

required in namelist &CONTROL\_GENOME.

## **&CONTROL\_GENOME**

ploidy=*ploidy*

recombCrossoverModel=*recombCrossoverModel*

nAlleleInLocus=*nAlleleInLocus*

nchrom=*nchrom*

ChromLengths=*ChromLengths*

nEpisPairLoci=*nEpisPairLoci*

nQtlEachQtlPairwithEpistasis=*nQtlEachQtlPairwithEpistasis*

genotyped\_at\_birth=*genotyped\_at\_birth*

templateDensity=*templateDensity*

nSingleGenes=*nSingleGenes* /

\*\*Task\*\*

Chromosome information and methods used to simulate QTL and genetic markers.

Specify first time step at which non-polygenic breeding values are used as

selection criterion.

Only read when *geneticModel* 'genomic' or 'ldonly'

\*\*Properties of names\*\*

*ploidy* Definition: the number of complete sets of chromosomes.

Even number: 2n (diploid),4n (tetraploid),6n,8n...

Type: Integer

Options: ≥2

Default: 2

*recombCrossoverModel* Definition: methods for recombinations and cross-over in meosis

use if ploidy >2.

Type: Integer

Options: 100: multi-somic, random bivalent pairing

102: disomic.

Default: 100

*nchrom* Definition: Number of chromosomes in the genome. A

chromosome is defined as a single segment of

DNA, whose inheritance is completely

independent of any other chromosomes.

Type: Integer

Options: ≥1

Default: 0

*ChromLengths* Definition: Length (cM) of each chromosome

Type: Integer

Dimension: *nchrom*\*1; a length required for each chromosome

Options: ≥0

Default: -1

*genotyped\_at\_birth* Definition: Animals genotyped at birth

Type: character

Options:

yes Animals are genotyped at birth

no Animals are not genotyped at birth

Default: no

*templateDensity* Definition: Density of markers on marker template; number of

markers per cM. *templateDensity*>0.0 generates

marker and QTL templates.

Marker template

is a dense marker map that provides unique genotypes for all base individuals.

It is sampled for all offspring according to Mendelian sampling terms allowing

for recombination. The markers are not available for selection purposes but

only serve to monitor inbreeding.

Type: Real

Options:

0.0 Marker template is not created

>0.0 Marker template is created with marker density

*templateDensity*

Default: 0.0

*nSingleGenes* Definition:

nAlleleInLocus

Definition: (max) number of alleles at a locus (simulate multi-allelic QTL)

Type: Integer

Options: nAlleleInLocus>=2

Default: 2

nQtlEachQtlPairwithEpistasis

Definition: (max) number of Qtl that each QTL interacts/pairs with when

simulating epistasis.

Type: Integer

Options: 0<nQtlEachQtlPairwithEpistasis<(nQtl-1).

Default: 1

No need to provide in input.prm if SimulatePairEpistasis=30.

nEpisPairLoci

Definition: number of nEpisPairLoci pairs. (can be defined by user. This means not all interactions pairs used. default (max used).)

Type: Integer

Options: 1 < nEpisPairLoci < nqtl\*nQtlEachQtlPairEpis/2.

Default: nqtl\*nQtlEachQtlPairEpis/2

No need to provide in input.prm if SimulatePairEpistasis=30.

*ModelOfParameters*  Type: Integer

Options: 0, 11, 12, 21, 22

Default: 0 or 11

! 0 (default) genomic model not simulated.

! 11 (default) Functional biological model. Variance by chromosome & individuals (Chu 2023).

! TTC: following methods are under development.

! 12 Functional biological model. Variance by loci (Chu 2023). (only works for nAlleleInLocus=2)

! 21 statistical model (only work with nAlleleInLocus=2 & nQtlEachQtlPairEpis=1). Variance by individuals (Chu 2023).

! Parameter inputs of variances are from a statistical model.

! This method: Statistical effects are sampled, then converted to functional effects by NOIA formula (Chu 2023).

! 22 statistical model (only work with nAlleleInLocus=2 & nQtlEachQtlPairEpis=1). Variance by loci (Chu 2023).

! Parameter inputs of variances are from a statistical model.

! This method: Statistical effects are sampled, then converted to functional effects by NOIA formula (Chu 2023).

\*\*Relation to subsequent namelists\*\*

Subsequent namelists required is largely determined by *geneticModel* in namelist

&CONTROLPARAMETERS. The relevant options for *geneticModel* are ‘genomic’ and ‘ldonly’.

\*\*Additional information\*\*

1) *templateDensity* can only be greater than 0.0 when *geneticModel* ‘genomic’.

2) *templateDensity*>0.0 generates marker and QTL templates.

*templateDensity*>0.0 is the trigger to generate marker and QTL templates. IBD-inbreeding

coefficients are reported in output file(s). Selection method ‘ibdBlup’ can be used for prediction and ‘ibd’ relationship matrix can be used to restrict average relationship with EVA.

If *templateDensity* 0.0, the templates are not generated and selection method ‘ibdBlup’ and ‘ibd’ relationship matrix are not valid.

\*\*Example\*\*

The following illustrates how the input for this namelist is provided. In the

example, there are two chromosomes. The lengths of the chromosomes are 110 and

97 cM. Only variables *nchrom* and *ChromLength* are illustrated.

&CONTROL\_GENOME

nchrom=2

ChromLength=

110

97

genotyped\_at\_birth=no /

\*\*Additional information\*\*

1) *genotyped\_at\_birth* is only used when *geneticModel* ‘genomic’.

When *geneticModel* ‘ldOnly’, *genotyped\_at\_birth* is set to ‘no’.

## **&LD**

useStoredHaplotypes=*useStoredHaplotypes*

npopBaseHaplotype=*npopBaseHaplotype*

useStoredHaplotypesDirectory=*useStoredHaplotypesDirectory*

StoredArchitectureOneAlleleFreq=*StoredArchitectureOneAlleleFreq*

userDefinedPairEpis=*userDefinedPairEpis*

historicalPopulation=*historicalPopulation*

LD\_ngen=*LD\_ngen*

nPotentialLoci=*nPotentialLoci*

nmarker\_per\_qtl=*nmarker\_per\_qtl*

mutation\_rate=*mutation\_rate*

CullingProportion=*CullingProportion*

ProportionPositiveMutations=*ProportionPositiveMutations*

populationHistory=*populationHistory*

nmale\_start=*nmale\_start*

nfemale\_start=*nfemale\_start*

nmale\_end=*nmale\_end*

nfemale\_end=*nfemale\_end*

nmale\_bottleneck=*nmale\_bottleneck*

nfemale\_bottleneck=*nfemale\_bottleneck*

gen\_bottle=*gen\_bottle*

nmale\_mean=*nmale\_mean*

nmale\_std=*nmale\_std*

nfemale\_mean=*nfemale\_mean*

nfemale\_std=*nfemale\_std*

ldInterval=*ldInterval*

maxNQtls=*maxNQtls*

markersMaf=*markersMaf*

maxNMarkers=*maxNMarkers*

nMarkersMaxMaf=*nMarkersMaxMaf*

qtlDistribution=*qtlDistribution*

storeHaplotypes=*storeHaplotypes*

storeHaplotypesDirectory=*storeHaplotypesDirectory* /

\*\*Task\*\*

Parameters to generate founder population with linkage disequilibrium between QTL and

marker alleles

\*\*Properties of names\*\*

*useStoredHaplotypes* Definition: Founder population(s) read from saved external-data

files; *geneticarchitecture.dat* and

*basehaplotypes.dat* or

*geneticarchitectureRep<rep>.dat* and

*basehaplotypesRep<rep>.dat*, where *rep* is replicate

number

Type: Character

Options:

yes Founder population(s) read from external-data files

no Founder population(s) sampled by generating linkage

disequilibrium

Default: no

*npopBaseHaplotype* Definition: Number of base populations with Haplotypes

provided. All pops must be provided in *basehaplotypes.dat*

Type: Integer

Options: ≥1

Default: 1

Haplotype of pop 1,2,..n will be used to generate base animals in pop 1,2,..n, respectively.

*useStoredHaplotypesDirectory*

Definition: Directory where external-data files with founder

population(s) are saved

Type: Character

Options:

Existing directory

Default: *OutDirectory*

*userDefinedPairEpis*

Definition: specify whether epistasis pairs are defined by users or not. If qtlEffectsProvided /=0, this option will not be read.

Type: integer

Default: 0 (default) defined by ADAM or qtlEffectsProvided /=0

Options: 1

! Format of file episPairLoci(Rep(rep)).dat ! READ (UNIT=io\_dat,FMT=\*,IOSTAT=ier) nEpisPairLoci

! READ (UNIT=io\_dat,FMT=\*,IOSTAT=ier) EpisPairLoci(ipair)%locusOne, EpisPairLoci(ipair)%locusTwo

*historicalPopulation* Definition: Defines founder population(s) used across

replicates

Type: Character

Options:

same Same founder population used across replicates.

Founder population read (*useStoredHaplotypes*

‘yes’) or generated (*useStoredHaplotypes* ‘no’) and

QTL and markers sampled from segregating loci

(*maxNQtls*, *markersMaf*, *maxNMarkers*, or

*nMarkersMaxMaf*) in first replicate; founder

population with sampled QTL and markers used as

founder population across replicates.

reshuffle Different founder populations used across

replicates. Same founder population read from

external-data files across replicates

(*useStoredHaplotypes* ‘yes’) or founder population

generated in first replicate and read from

external-data files in subsequent replicates

(*useStoredHaplotypes* ‘no’). In each replicate,

different QTL and markers sampled from segregating

loci (*maxNQtls*, *markersMaf*, *maxNMarkers*, or

*nMarkersMaxMaf*) to generate unique founder

populations across replicates.

unique Different founder populations used across

replicates. In each replicate, different founder

population read from external-data files

(*useStoredHaplotypes* ‘yes’) or new founder

population generated (*useStoredHaplotypes* ‘no’)

and different QTL and markers sampled from

segregating loci (*maxNQtls*, *markersMaf*,

*maxNMarkers*, or *nMarkersMaxMaf*).

Default: same

*LD\_ngen*  Definition: Number of generations simulated to generate LD

Type: Integer

Options: ≥1

Default: 1000

*nPotentialLoci* Definition: Number of QTL and marker loci on the genome

Type: Integer

Options: ≥1

Default: 1

*nmarker\_per\_qtl* Definition: Ratio of marker to QTL loci

Type: Integer

Options: 7, 15, 31

Default: 7

*mutation\_rate*  Definition: Mutation rate; probability of a mutation occurring

at each diploid locus in each individual. The rate

of mutation at each locus allele in each individual

is 0.5\**mutation\_rate*.

Type: Real

Options: 0.0≤*mutation\_rate*≤1.0

Default: 0.0

sample\_nmutation=random\_poisson(0.5\*mutation\_rate\*REAL(nPotentialLoci))

*CullingProportion* Definition: Desired proportion of sampled animals culled

before becoming founders [ancestor in founder

population]

Type: Real

Options: 0.0≤*CullingProportion*≤0.5

Default: 0.0

*ProportionPositiveMutations*

Definition: Proportion of mutations that generate a positive

effect at QTL loci

Type: Real

Options: 0.0<*ProportionPositiveMutations*<1.0

Default: 0.0

*populationHistory* Definition: Population size over generations

Type: Character

Options:

constant Constant population size

expanding Population size increases

decreasing Population size decreases

bottlenecked Population size goes through a bottleneck

fluctuating Random population size at each generation

Default: constant

*nmale\_start* Definition: Number of male parents; number in first generation

Type: Integer

Options: ≥1

Default: 100

*nfemale\_start* Definition: Number of female parents; number in first

generation

Type: Integer

Options: ≥1

Default: 100

*nmale\_end* Definition: Number of male parents at in the final

generation

Type: Integer

Options: ≥1

Default: 100

*nfemale\_end*  Definition: Number of female parents in the final generation

Type: Integer

Options: ≥1

Default: 100

*nmale\_bottleneck*  Definition: Number of male parents at peak of population

bottleneck; when population size is lowest

Type: Integer

Options: ≥1

Default: 100

*nfemale\_bottleneck* Definition: Number of female parents at peak of population

bottleneck

Type: Integer

Options: ≥1

Default: 100

*gen\_bottle*  Definition: Generation in which population bottleneck peaks

Type: Integer

Options: 1≤*gen\_bottle*≤*LD\_ngen*

Default: 500

*nmale\_mean*  Definition: Mean number of male parents in each generation

Type: Integer

Options: ≥1

Default: 100

*nmale\_std*  Definition: Standard deviation of number of male parents in

each generation

Type: Integer

Options: ≥1

Default: 10

*nfemale\_mean*  Definition: Mean number of female parents in each generation

Type: Integer

Options: ≥1

Default: 100

*nfemale\_std*  Definition: Standard deviation of number of female parents

in each generation

Type: Integer

Options: ≥1

Default: 10

*ldInterval*  Definition: Genomic-distance (cM) between loci used to

generate output files with LD versus genomic

distance

Type: Real

Options: 0.0125≤*ldInterval*≤sum(*ChromLengths*)/40

Default: 0.25 (0-10cM)

*maxNQtls* Definition: Maximum number of QTL sampled for each chromosome.

QTL are randomly sampled from segregating QTL-loci

generated by linkage disequilibrium. Sampled QTL

are used in subsequent simulations.

Type: Integer

Dimension: *nchrom,* where *nchrom* is the number of chromosomes in

the genome provided in namelist &CONTROL\_GENOME

Options: ≥0

Default: huge(1)

*maxNMarkers* Definition: Maximum number of markers sampled for each

chromosome. Markers are randomly sampled from

segregating marker-loci generated by linkage

disequilibrium. Sampled markers are used in

subsequent simulations.

Type: Integer

Dimension: *nchrom,* where *nchrom* is the number of chromosomes in

the genome provided in namelist &CONTROL\_GENOME

Options: ≥0 with sum(*maxNMarkers*)≥1

Default: huge(1)

*markersMaf* Definition: Threshold for marker minimum-allele frequency on

each chromosome; segregating markers generated by

linkage disequilibrium with minimum-allele

frequency larger than *markersMaf* sampled and used

in simulations, where minimum-allele frequency is

calculated using founder animals

Type: Real

Dimension: *nchrom,* where *nchrom* is the number of chromosomes in

the genome provided in namelist &CONTROL\_GENOME

Options: 0.0≤*markersMaf*≤0.5

Default: 0.0

*nMarkersMaxMaf* Definition: Maximum number of markers retained on each

chromosome, where the *nMarkersMaxMaf* segregating

markers generated by linkage disequilibrium with

the highest minimum-allele frequencies are retained

and minimum-allele frequency is calculated using

founder animals. Maximum number of markers sampled

for each chromosome. Markers are sampled from

segregating marker-loci generated by linkage

disequilibrium. Sampled markers are used in

subsequent simulations.

Type: Integer

Dimension: *nchrom,* where *nchrom* is the number of chromosomes in

the genome provided in namelist &CONTROL\_GENOME

Options: ≥0 with sum(*nMarkersMaxMaf*)≥1

Default: huge(1)

*qtlDistribution* Definition: Distribution of QTL-effects

Type: Character

Options:

exponential Exponential distribution (only for *ntbv*=1)

with scale parameter (1/rate) of the

exponential set to 0.01

normal Normal distribution with mean 0

Default: exponential

*storeHaplotypes* Definition: Founder population(s) saved in external-data

files; *geneticarchitecture.dat* and

*basehaplotypes.dat* or

*geneticarchitectureRep<rep>.dat* and

*basehaplotypesRep<rep>.dat*, where *rep* is replicate

number

Type: Character

Options:

yes Founder population(s) saved in external-data files

no Founder population(s) not saved

Default: no

*storeHaplotypesDirectory*

Definition: Directory to save founder population(s) in

external-data files haplotype files

Type: Character

Options:

Existing directory or directory that can be created

Default: *OutDirectory*

*StoredArchitectureOneAlleleFreq*

Definition: Genetic architecture format has one allele freq column only.

Type: Character

Options: ‘yes’, ‘no’

Default: ‘no’

! Genetic architecture format has one allele freq column only.

Default: 'no' (allele of p and p... & for nAlleleInLocus>2).

'yes' (allele of p only. this is to read old ADAM input where nAlleleInLocus=2).

\*\*Additional information\*\*

• Namelist &LD only read when *geneticModel* ‘genomic’ or ‘ldonly’

• Parameters required is determined by *geneticModel* and *populationHistory*:

(a) *geneticModel* ‘genomic’

*useStoredHaplotypes*

*useStoredHaplotypesDirectory*

*historicalPopulation*

*LD\_ngen*

*nPotentialLoci*

*nmarker\_per\_qtl*

*mutation\_rate*

*CullingProportion*

*ProportionPositiveMutations*

*populationHistory*

*populationHistory* ‘constant’, ‘expanding’, ‘decreasing’, ‘bottlenecked’

*nmale\_start*

*nfemale\_start*

*populationHistory* ‘expanding’, ‘decreasing’, ‘bottlenecked’

*nmale\_end*

*nfemale\_end*

*populationHistory* ‘bottlenecked’

*nmale\_bottleneck*

*nfemale\_bottleneck*

*gen\_bottle*

*populationHistory* ‘fluctuating’

*nmale\_mean*

*nmale\_std*

*nfemale\_mean*

*nfemale\_std*

*ldInterval*

*maxNQtls*

*markersMaf*

*maxNMarkers*

*nMarkersMaxMaf*

*qtlDistribution*

*storeHaplotypes*

*storeHaplotypesDirectory*

(b) *geneticModel* ‘ldonly’

*LD\_ngen*

*nPotentialLoci*

*nmarker\_per\_qtl*

*mutation\_rate*

*CullingProportion*

*ProportionPositiveMutations*

*populationHistory*

*populationHistory* ‘constant’, ‘expanding’, ‘decreasing’, ‘bottlenecked’

*nmale\_start*

*nfemale\_start*

*populationHistory* ‘expanding’, ‘decreasing’, ’bottlenecked’

*nmale\_end*

*nfemale\_end*

*populationHistory* ‘bottlenecked’

*nmale\_bottleneck*

*nfemale\_bottleneck*

*gen\_bottle*

*populationHistory* ‘fluctuating’

*nmale\_mean*

*nmale\_std*

*nfemale\_mean*

*nfemale\_std*

*ldInterval*

*storeHaplotypes*

*storeHaplotypesDirectory*

Parameters *useStoredHaplotypes*, *useStoredHaplotypesDirectory*, *historicalPopulation*,

*maxNQtls*, *markersMaf*, *maxNMarkers*, *nMarkersMaxMaf*, and *qtlDistribution* are only

used when *geneticModel* ‘genomic’.

• *useStoredHaplotypesDirectory* only used when *useStoredHaplotypes* ‘yes’.

*useStoredHaplotypesDirectory* must end with ‘/’ and be provided in inverted commas

(‘’ or “”) e.g., *useStoredHaplotypesDirectory*=‘/user/donald/duck/’.

• When *useStoredHaplotypes* ‘yes’ and *historicalPopulation* ‘same’ or ‘reshuffle’,

files *geneticarchitecture.dat* and *basehaplotypes.dat* must exist in directory

*useStoredHaplotypesDirectory.*

When *useStoredHaplotypes* ‘yes’ and *historicalPopulation* ‘unique’, files

*geneticarchitectureRep<rep>.dat* and *basehaplotypes Rep<rep>.dat* must exist in

directory *useStoredHaplotypesDirectory*, where *rep* is replicate number

(*rep*=*start\_rep*:*start\_rep*+*nrep*-1)*.*

• (a) When *geneticModel* ‘genomic’ and *useStoredHaplotypes* ‘no’, a founder

population(s) with QTL and markers in linkage disequilibrium is generated. The

pool of genotypes for these animals is sampled with replacement to generate

base populations used in subsequent breeding scheme.

(b) When *geneticModel* ‘ldonly’, LD is generated to study linkage disequilibrium. A

subsequent breeding scheme is not simulated and, therefore, the pool of

genotypes is not used to generate the base population. Founder populations can

be saved saved in external-data files.

Describe how genotypes of base animals are sampled from founder genotypes.

• When *geneticModel* ‘genomic’ and *useStoredHaplotypes* ‘no’ or *geneticModel* ‘ldonly’,

output from analyses of linkage disequilibrium is written to files in

*OutDirectory*.

Files heterozygosityRep<*rep*>.dat, QTLdistributionRep<*rep*>.dat,

markerDistributionRep<*rep*>.dat, LD\_binRep<*rep*>.dat, and ldPlotsRep<*rep*>.pdf, where

*rep* is replicate number, are written when *printLdHayesFiles* ‘yes’.

When *printLdMutations* ‘yes’, mutationsRep<*rep*>.dat is written to *OutDirectory*.

• *nmarker\_per\_qtl*<nBits

• *CullingProportion* … (Expected) number of offspring that are culled by ‘natural’

selection.

ACS: I’ve eplaced the variable SelectionCoefficient used in sampling genomic LD by the variables CullingProportion, which gives the desired proportion of animals culled in each generation of the founder population, and ExponentialScale, which gives the scale parameter for the exponential distribution for sampling QTL effects. Notice the parametrisation of the exponential distribution here. It is the scale parameter (1/lambda), which is the inverse of the rate parameter, lambda. ExponentialScale is ~~only~~ used to sample QTL effects when generating LD. If the exponential distribution is chosen for sampling QTL effects in breeding scheme, these effects are maintained. If normal distribution is used instead, normal deviates are sampled for segregating alleles.

• When *populationHistory* ‘constant’, only *nmale\_start* and *nfemale\_start* are used

When *populationHistory* ‘expanding’ or ‘decreasing’, *nmale\_start,* *nfemale\_start*,

*nmale\_end*, and *nfemale\_end* are used

When populationHistory ‘bottlenecked’, *nmale\_start*, *nfemale\_start*, *nmale\_end*,

*nfemale\_end*, *nmale\_bottleneck*, *nfemale\_bottleneck*, and *gen\_bottle* are used

When populationHistory ‘fluctuating’, *nmale\_mean*, *nmale\_std*, *nfemale\_mean*, and

*nfemale\_std* are used

• *ldInterval* only used when printLdHayesFiles ‘yes’ [mh: CHECK?]. Otherwise, file

with LD versus genomic distance are not generated and ldInterval not required.

• If *maxNQtls* is provided for one chromosome, it must be provided for all *nchrom*

chromosomes (*maxNQtls*<huge(1)). If *maxNQtls* is not provided, all segregating QTL

are sampled/used in subsequent simulations. All segregating QTL are used when the

number of segregating QTL-loci on each chrosomome is less than or equal to the

corresponding elements of *maxNQtls*.

If *maxNMarkers* is provided for one chromosome, it must be provided for all *nchrom*

Chromosomes (*maxNMarkers*<huge(1)). If *maxNMarkers* is not provided, no segregating

markers are removed from subsequent simulations due to *maxNMarkers*. No marker-loci

are removed due to *maxNMarkers* when the number of segregating marker-loci on each

chromosome is less than or equal to the corresponding elements of *maxNMarkers*.

If *markersMaf* is provided for one chromosome, it must be provided for all *nchrom*

Chromosomes (*markersMaf*>0.0). If *markersMaf* is not provided, no segregating markers

are removed from subsequent simulations due to *markersMaf*.

If *nMarkersMaxMaf* is provided for one chromosome, it must be provided for all

*nchrom* chromosomes *nMarkersMaxMaf* (*nMarkersMaxMaf*<huge(1)). If *nMarkersMaxMaf* is

not provided, no segregating markers are removed from subsequent simulations due to

*nMarkersMaxMaf*. No marker-loci are removed due to *nMarkersMaxMaf* when the number of

segregating marker-loci on each chromosome is less than or equal to the

corresponding elements of *nMarkersMaxMaf*.

• If neither *maxNMarkers*, *markersMaf*, or *nMarkersMaxMaf* are not provided, all

segregating markers are sampled/used in subsequent simulations.

• *maxNMarkers* and *markerMaf* can be used together; neither *maxNMarkers* or *markerMaf*

can be used with *nMarkersMaxMaf*.

• *markerMaf* has higher precedence over *maxNMarkers*. Markers are sampled for *markerMaf*

before *maxNMarkers*.

• If *historicalPopulation* 'reshuffle', *maxNQtl* or *maxNMarker* must be provided.

• When *geneticModel* ‘genomic’ and *useStoredHaplotypes* ‘no’ or *geneticModel* ‘ldonly’,

linkage disequilbrium is generated by simulating a single trait with QTL-effects

sampled from an exponential distribution.

When *geneticModel* ‘genomic’ and *qtlDistribution* ‘exponential’, QTL-effects sampled

from the exponential distribution while generating linkage disequilbrium are used

in subsequent simulations.

When *geneticModel* ‘genomic’ and *qtlDistribution* ‘normal’, QTL-effects sampled

from multivariate-normal distributions in subsequent simulations.

QTL-effects used in subsequent simulations are scaled so that the total QTL

(co)variance(s) for traits are equal to pre-defined (co)variances.

output from analyses of linkage disequilibrium is written to files in

*OutDirectory*.

*X* ~ exp(λ) with E[X] = 1/λ, where 1/λ=0.01 and λ=100 [exponential distribution: λ>0 is the parameter of the distribution, often called the *rate parameter*]. In ADAM, we provide ExponentialScale=E[X]=1/λ=0.01 in input.prm. Scale is a constant, not important, as exponetialfordelingen er en gammafordeling med shape parameter = 1, dvs. samme shape. Exponential distributions with different λ viser jo også den samme form, nemlig konstant halveringstid (hvor x-aksen er tid).

Store/Use haplotypes: Det er kun haplotyper og qtl indikator, men ikke effekter. Det eneste der skal matche med input er antal Chromosomer.

Så vidt jeg kan se/huske, gemmer du alle haplotyper og qtl indikatorer ved slutningen af DoGenomicLD. Hvis man vælger at bruge "stored haplotypes", springer man over DoGenomicLD. Derefter "vælger/fjerner" du qtl'er og markører samt tilføjer qtl-effekter.

Ja, men de kan sagtens bruges med en anden fordeling af effekter og en anden ntbv

When *geneticModel* ‘genomic’ and *useStoreHaplotypes* ‘yes’, *nchrom* and *ChromLengths* are used and must conform with the number of chromosomes and chromosome lengths read from saved external-data files.

Beware! The lengths of the chromosomes in *ChromLengths* need not be the same as the lengths saved in stored haplotypes [MH: CHECK. Do we change this?].

• QTL-effects sampled from an exponential distribution for the single trait when

generating founder populations with linkage disequilibrium between QTL and

marker alleles are always read from and stored in saved external-data files;

*geneticarchitecture.dat* and *basehaplotypes.dat* or *geneticarchitectureRep<rep>.dat*

and *basehaplotypesRep<rep>.dat*. That is, the original QTL-effects are always

available for subsequent simulations, even when reading the founder population(s)

that has been saved after a simulation with *geneticModel* ‘genomic’, *storeHaplotypes*

‘yes’, *qtlDistribution* ‘normal’, and *maxNQtl*, *maxNMarker*, *markerMaf*, or

*nMarkerMaxMaf* provided.

• *storeHaplotypesDirectory* only used when *useStoredHaplotypes* ‘yes’.

*storeHaplotypesDirectory* must end with ‘/’ and be provided in inverted commas

(‘’ or “”) e.g., *storeHaplotypesDirectory* =‘/user/anders/and/’.

• When *geneticModel* ‘genomic’ and *useStoredHaplotypes* 'yes', storeHaplotypes can only

be ‘yes’ when *maxNQtl*, *maxNMarker*, *markerMaf*, or *nMarkerMaxMaf* are provided.

• When *geneticModel* ‘genomic’ and both *useStoredHaplotypes* and *storeHaplotypes* are

‘yes’, *useStoredHaplotypesDirectory* and *storeHaplotypesDirectory* must be different

directories.

• When *storeHaplotypesDirectory* already exists, all old founder-population files in

*storeHaplotypesDirectory* are deleted overwritten.

• When *geneticModel* 'ldonly' *[or geneticModel 'genomic' and historicalPopulation*

*‘reshuffle’ or ‘unique’]* and *storeHaplotypes* 'yes', founder populations generated

are saved in external-data files *geneticarchitectureRep<rep>.dat* and

*basehaplotypesRep<rep>.dat* in directory *storeHaplotypesDirectory*, where *rep* is

replicate number.

When *geneticModel* 'genomic', *historicalPopulation* ‘same’, and *storeHaplotypes*

'yes', a single founder population is generated and saved in external-data files

*geneticarchitecture.dat* and *basehaplotypes.dat* in directory

*storeHaplotypesDirectory*.

When *geneticModel* 'genomic', *historicalPopulation* ‘reshuffle’ or ‘unique’, and

*storeHaplotypes* 'yes', founder populations are saved in external-data files

*geneticarchitectureRep<rep>.dat* and *basehaplotypesRep<rep>.dat* in directory

*storeHaplotypesDirectory*, where *rep* is replicate number

(*rep*=*start\_rep*:*start\_rep*+*nrep*-1).

## **&RANDOMNUMBERSEEDS**

randomSeeds=*randomSeeds*

seed1=*seed1*

seed2=*seed2*

seedsFile=*seedsFile*

founderSeeds=*founderSeeds*

founderSeed1=*founderSeed1*

founderSeed2=*founderSeed2*

founderSeedsFile=*founderSeedsFile*

baseSeeds=*baseSeeds*

baseSeed1=*baseSeed1*

baseSeed2=*baseSeed2*

baseSeedsFile=*baseSeedsFile*

selectionSeeds=*selectionSeeds*

selectionSeedsFile=*selectionSeedsFile* /

\*\*Task\*\*

Provide seeds to initialise Randlib’s random-number generator; *seed1* and *seed2*

\*\*Properties of names\*\*

*randomSeeds* Definition: Method used to generate seeds that initialise

Randlib’s random-number generator

Type: Character

Options:

computer Seeds randomly generated from /dev/urandom

user Seeds provided by user, *seed1* and *seed2*

file Seeds provided by user in file s*eedsFile*; *seed1*

and *seed2* for each replicate

manual Seeds provided by user to generate founder

populations, base population, and carry out

selection

Default: *randomSeeds* must be specified

*seed1* Definition: First seed of current random-number generator;

used when *randomSeeds* ‘user’

Type: Integer\*4

Options: 0≤*seed1*≤huge(1)

Default: 123456789

*seed2* Definition: Second seed of current random-number generator;

used when *randomSeeds* ‘user’

Type: Integer\*4

Options: 0≤*seed2*≤huge(1)

Default: 987654321

*seedsFile* Definition: Input file with seeds when *randomSeeds* ‘file’;

two integer seeds per row, *seed1* and *seed2*, one

row per replicate. Each replicate is initiated

with specified seeds.

Type: Character

Options: Valid filename with *start\_rep*+*nrep*-1 rows,

0≤*seed1*≤huge(1), and 0≤*seed2*≤huge(1)

Default: seeds.in provided in submission directory

*founderSeeds* Definition: Method used to provide seeds to generate

founder population(s) when *randomSeeds* ‘manual’

Type: Character

Options:

computer Seeds randomly generated from /dev/urandom

user Seeds provided by user; *founderSeed1* and

*founderSeed2*

file Seeds provided by user; *seed1* and *seed2 for each*

replicate specified in file *founderSeedsFile*

Default: computer

*founderSeed1* Definition: First seed of current random-number generator

when *randomSeeds* ‘manual’ and *founderSeeds*

‘user’; *seed1* is set to *founderSeed1*

Type: Integer\*4

Options: 0≤*founderSeed1*≤huge(1)

Default: 123456789

*founderSeed2* Definition: Second seed of current random-number generator

when *randomSeeds* ‘manual’ and *founderSeeds*

‘user’; *seed2* is set to *founderSeed2*

Type: Integer\*4

Options: 0≤*founderSeed2*≤huge(1)

Default: 987654321

*founderSeedsFile* Definition: Input file with seeds when *randomSeeds* ‘manual’

and *founderSeeds* ‘file’; two integer seeds per

row, *seed1* and *seed2*, one row per replicate.

Founder populations in each replicate are

generated with specified seeds.

Type: Character

Options: Valid filename with *start\_rep*+*nrep*-1 rows,

0≤*seed1*≤huge(1), and 0≤*seed2*≤huge(1) with

*geneticModel* ‘genomic’, *founderSeeds* ‘file’, and

*historicalPopulation* ‘reshuffle’ and ‘unique’.

With *historicalPopulation* ‘same’, only one row is

used.

Default: founderSeeds.in provided in submission directory

*baseSeeds* Definition: Method used to provide seeds to generate

base population(s) when *randomSeeds* manual

Type: Character

Options:

computer Seeds randomly generated from /dev/urandom

user Seeds provided by user; *baseSeed1* and *baseSeed2*

file Seeds provided by user; *seed1* and *seed2* for

each replicate specified in file, *baseSeedsFile*

Default: computer

*baseSeed1* Definition: First seed of current random-number generator

when *randomSeeds* ‘manual’ and *baseSeeds* ‘user’;

*seed1* is set to *baseSeed1*

Type: Integer\*4

Options: 0≤*baseSeed1*≤huge(1)

Default: 123456789

*baseSeed2* Definition: Second seed of current random-number generator

when *randomSeeds* ‘manual’ and *baseSeeds* ‘user’;

*seed2* is set to *baseSeed2*

Type: Integer\*4

Options: 0≤*baseSeed2*≤huge(1)

Default: 987654321

*baseSeedsFile*  Definition: Input file with seeds when *randomSeeds* manual

and *baseSeeds* file; two integer seeds per

row, *seed1* and *seed2*, one row per replicate.

Base populations in each replicate are

generated with specified seeds.

Type: Character

Options: Valid filename with *start\_rep*+*nrep*-1 rows,

0≤*seed1*≤huge(1), and 0≤*seed2*≤huge(1)

Default: baseSeeds.in provided in submission directory

*selectionSeeds*  Definition: Method used to provide seeds to carry out

selection when *randomSeeds* manual

Type: Character

Options:

computer Seeds randomly generated from /dev/urandom

file Seeds provided by user; *seed1* and *seed2* for

each replicate specified in file,

*selectionSeedsFile*

Default: computer

*selectionSeedsFile*  Definition: Input file with seeds when *randomSeeds* manual

and *selectionSeeds* file; two integer seeds per

row, *seed1* and *seed2*, one row per replicate.

Selection process in each replicate is

initiated with specified seeds.

Type: Character

Options: Valid filename with *start\_rep*+*nrep*-1 rows,

0≤*seed1*≤huge(1), and 0≤*seed2*≤huge(1)

Default: selectionSeeds.in provided in submission directory

\*\*Additional information\*\*

• *seed1* and *seed2* remain the seeds of Randlib’s random-number generator throughout

simulations. They are inititated and reinitialised using the information provided

in namelist **RANDOMNUMBERSEEDS**.

• (a) When *randomSeeds* ‘computer’, *seed1* and *seed2* used by RandLib are initiated in

the first replicate by linux's random-number generator; with *randomSeeds*

‘user’, seeds are provided by user. In subsequent replicates, the ‘state of the

current generator’ is reinitialised; seeds at this point are used by RandLib to

simulate each replicate.

When *randomSeeds* ‘file’, *seed1* and *seed2* for each rplicate are provided in

*seedsFile*.

(b) When *geneticModel* ‘genomic’, *randomSeeds* ‘manual’, and *founderSeeds* ‘computer’,

seeds used by RandLib to generate founder populations are initiated by linux's

random-number generator at the start of each replicate. When *founderSeeds*

‘user’, seeds used by RandLib in each replicate are *founderSeeds1* and

*founderSeeds2*. This means that unique founder populations are generated with

*founderSeeds* ‘computer’ in each replicate; with *founderSeeds* ‘user’, the

same founder population is generated in each replicate.

When *geneticModel* ‘genomic’, *randomSeeds* ‘manual’, and *founderSeeds* ‘file’,

seeds used by RandLib to generate founder population in each replicate are

provided in *founderSeedsFile*.

(c) When *randomSeeds* ‘manual’, and *baseSeeds* ‘computer’, seeds used by RandLib to

generate base populations are initiated by linux's random-number generator

before generating the base population in each replicate. When *baseSeeds* ‘user’,

seeds used by RandLib in each replicate are *baseSeeds1* and *baseSeeds2*. This

means that unique base populations are generated with *baseSeeds* ‘computer’ in

each replicate; with *baseSeeds* ‘user’, the same base population is generated in

each replicate with *geneticModel* ‘polygenic’ and *geneticModel* ‘genomic’,

provided the same founder population is used across replicates.

When *randomSeeds* ‘manual’, and *baseSeeds* ‘file’, seeds used by RandLib to

generate base population in each replicate are provided in *baseSeedsFile*.

(d) When *randomSeeds* ‘manual’, and *selectionSeeds* ‘computer’, seeds used by RandLib

to carry out selection are initiated by linux's random-number generator

before carrying out selection in each replicate.

When *randomSeeds* ‘manual’, and *selectionSeeds* ‘file’, seeds used by RandLib to

generate base population in each replicate are provided in *selectionSeedsFile*.

• When *geneticModel* ‘ldonly’, *randomSeeds* cannot be ‘manual’

• (a) *seed1 and seed2* are only used when *randomSeeds* ‘user’

(b) *seedsFile* is only used as input when when *randomSeeds* ‘file’

(c) *founderSeeds*, *baseSeeds*, and *selectionSeeds* are only used when randomSeeds

‘manual’

• *founderSeeds* can only be used when *geneticModel* ‘genomic’ and randomSeeds ‘manual’

• (a) *founderSeed1 and founderSeed2* are only used when *randomSeeds* ‘manual’ and

*founderSeeds* ‘user’

(b) *founderSeedsFile* is only used when *randomSeeds* ‘manual’ and *founderSeeds* ‘file’

• (a) *baseSeed1 and baseSeed2* are only used when *randomSeeds* ‘manual’ and *baseSeeds*

‘user’

(b) *baseSeedsFile* is only used when *randomSeeds* ‘manual’ and *baseSeeds* ‘file’

• *selectionSeedsFile* is only used when *randomSeeds* ‘manual’ and selectionSeeds

‘file’.

• *seedsFile*, *founderSeedsFile*, *baseSeedsFile*, and *selectionSeedsFile* must be provided

in the submission directory.

• The format of input files *seedsFile*, *founderSeedsFile*, *baseSeedsFile*,

*selectionSeedsFile* is:

<*seed1* for replicate 1> <*seed2* for replicate 1>

<*seed1* for replicate 2> <*seed2* for replicate 2>

.

.

.

<*seed1* for replicate *start\_rep*+*nrep*-1> <*seed2* for replicate *start\_rep*+*nrep*-1>

• When *randomSeeds* ‘computer’ or ‘user’, seeds are written to output file seeds.out

in *OutDirectory*.

When *geneticModel* ‘genomic’, *randomSeeds* ‘manual’, and *founderSeeds* ‘computer’,

founder seeds used by RandLib to generate founder populations are written to

output file founderSeeds.out in *OutDirectory*.

When *randomSeeds* ‘manual’, and *baseSeeds* ‘computer’, base seeds used by RandLib to

generate base populations are written to output file baseSeeds.out in *OutDirectory*.

When *randomSeeds* ‘manual’, and *selectionSeeds* ‘computer’, selection seeds used by

RandLib to generate selection are written to output file selectionSeeds.out in

*OutDirectory*.

• Output files seeds.out, founderSeeds.out, baseSeeds.out, and selectionSeeds.out

have *nrep* rows (*start\_rep*:*start\_rep*+*nrep*-1); two seeds per row, *seed1* and *seed2*,

one row per replicate.

The exception is *geneticModel* ‘genomic’, *randomSeeds* ‘manual’, and

*historicalPopulation* ‘same’, where only one founder population is read or generated

and founderSeeds.out has one row.

• Output files seeds.out, founderSeeds.out, baseSeeds.out, and selectionSeeds.out can

be renamed seeds.in, founderSeeds.in, baseSeeds.in, and selectionSeeds.in and used

as seeds in subsequent simulations.

• The format of output files seeds.out, founderSeeds.out, baseSeeds.out, and

selectionSeeds.out is:

<*seed1* for replicate *start\_rep*> <*seed2* for *start\_rep* >

<*seed1* for replicate *start\_rep*+1> <*seed2* for replicate *start\_rep*+1>

.

.

.

<*seed1* for replicate *start\_rep*+*nrep*-1> <*seed2* for replicate *start\_rep*+*nrep*-1>

## **&POPULATIONPARAMETERS**

npop=*npop*

nherd=*nherd*

ntbv=*ntbv*

nobs=*nobs*

nres=*nres*

nCombinedObs=*nCombinedObs*

nObsConstraints=*nObsConstraints*

nebv=*nebv*

gestation\_length=*gestation\_length*

daysOpen=*daysOpen*

femaleCycleLength=*femaleCycleLength*

proportionEmptyMales=*proportionEmptyMales*

proportionEmptyFemales=*proportionEmptyFemales*

OldAgeMales=*OldAgeMales*

OldAgeFemales=*OldAgeFemales*

preferentialTreatment=*preferentialTreatment*

/

NB! Number of lines provided for RandomCullingGroups must equal

nRandomCullingGroups

\*\*Task\*\*

Basic input to generate population(s)

\*\*Properties of names\*\*

*npop* Definition: Number of populations. If npopBaseHaplotype /=0,

npop = npopBaseHaplotype+ npopcrossbred.

Type: Integer

Options: ≥1

Default: 1

*npopcrossbred* Definition: crossbred populations. Base animals are not

simulated in crossbred pops. Popid is:

(npopBaseHaplotype+1) : npop

Type: Integer

Options: ≥0

Default: 0

*nherd* Definition: Max number of herds from populations (as each

population could have different no. of herds).

Type: Integer

Options: ≥1

Default: 1

*ntbv* Definition: Number of traits, where a trait is a characteristic

for which a true breeding value is sampled

Type: Integer

Options: ≥1

Default: 1

*nobs* Definition: Number of observations, where an observation is

the phenotype of a *ntbv* trait.

Type: Integer

Options: ≥1

Default: 1

*nres* Definition: Number of residuals, where a residual is an

environmental deviation sampled for a *nobs*

observation. Environmental deviation includes all

non-additive genetic effects.

Type: Integer

Options: ≥1

Default: 1

*nCombinedObs* Definition: Number of combined observations, where a combined

observation is a phenotype generated by combining

*nobs* observations.

Type: Integer

Options: ≥0

Default: 0

*nObsConstraints* Definition: Number of constraints applied to *nobs* and

*nCombinedObs* observations, where a constraint is a

restriction by which observations must comply

Type: Integer

Options: ≥0

Default: 0

*nebv* Definition: Number of selection criteria used to calculate

estimated aggregate-breeding value

Type: Integer

Options: ≥0

Default: 0

*gestation\_length* Definition: Gestation length; time steps from mating to birth

Type: Integer

Options: ≥0

Default: 0

*daysOpen* Definition: Days open; time steps from end of gestation

length to end of female reproductive cycle

Type: Integer

Options: ≥0

Default: 0

*femaleCycleLength* Definition: Number of time steps between ovulations

Type: Integer

Options: ≥1

Default: 1

*proportionEmptyMales* Definition: Proportion of reproductive males that do not

reproduce at a given time step

Type: Real

Options: 0.0≤*proportionEmptyMales*≤1.0

Default: 0.0

*proportionEmptyFemales*

Definition: Proportion of reproductive females that do not

become pregnant at ovulation

Type: Real

Options: 0.0≤*proportionEmptyFemales*≤1.0

Default: 0.0

*OldAgeMales* Definition: Age at which males are culled for old age

Type: Integer

Options: ≥1

Default: 1

*OldAgeFemales* Definition: Age at which females are culled for old age

Type: Integer

Options: ≥1

Default: 1

*preferentialTreatment* Definition: Criterion by which to apply preferential treatments

Type: Character

Options:

parentaverage Preferential treatment applied as a function

of an individual’s parent average

tbv Preferential treatment applied as a function

of an in individual’s true-breeding value

Default: parentaverage

Preferential treatments applied as a function of ‘parentaverage’ or ‘tbv’, where function is regression coefficient with intercept cohort mean.

Only applies when geneticResidualMatrix has a non-zero element(s).

\*\*Additional information\*\*

1) *nherd* must be greater than or equal to *herd\_id(1)* and *herd\_id(2)* provided in

namelist &SELECTION, variable selection\_scheme

2) Can only set *nCombinedObs*>0 when *geneticModel* ‘polygenic’ or 'genomic' in

&CONTROLPARAMETERS. Otherwise, *nCombinedObs* must be 0.

3) Offspring are sampled in the same time step as parents are mated; they are born

*gestation\_length* time steps after the parents are mated. When *gestation\_length*

0, offspring are born in the same time step as parents are mated.

4) Males and females must be reproductive to be selected for reproductive. To be

reproductive, females must be alive, they must be ovulating, they must not be

in gestation or days open, and they must be between the minimum and maximum

reproductive ages for females. Males must be alive and be between the minimum

and maximum reproductive ages for males. The minimum and maximum reproductive

ages for males and females are derived from input provided in namelists

&SELECTION and &EVA.

5) (a) Non-reproductive animals are not candidates for selection when *destiny\_sel*

includes reproduction. That is, non-reproductive males are not candidates

when *destiny\_sel* 1, 12, 14, 16, 126, 91, 912, 914, 916, 9126. Non-

reproductive females are not candidates when *destiny\_sel* 1, 12, 14, 17,

127, 147, 91, 912, 914, 917, 9127, or 9147.

(b) Non-reproductive animals are not candidates for EVA-selection when destiny

for selection in the subsequent EVA-selection stage includes reproduction.

That is, non-reproductive males are not candidates when *MaleDestinySel* in

namelist &EVA, variable EvaSelection 1, 12, 14, 16, 126, 91, 912, 914, 916,

9126. Non-reproductive females are not candidates when *FeMaleDestinySel* in

namelist &EVA, variable EvaSelection 1, 12, 14, 17, 127, 147, 91, 912, 914,

917, 9127, or 9147.

6) (a) *femaleCycleLength* represents the length of the female ovulation cycle in

time steps. *femaleCycleLength* 1 means that there is one time step between

ovulations and females ovulate at each time step; 2 means that females

ovulate at alternate time steps; 3 females ovulate at each third time step.

Only females that are ovulating are reproductive.

(b) Females start ovulating at the minimum reproductive age for females.

Ovulation stage is sampled for each female when they reach the minimum

reproductive age. Sampling is from a uniform distribution and females have

equal probability of starting their reproductive life at each of the

possible ovulation stages. For example, *femaleCycleLength* 1 means that all

females are ovulating when they reach the minimum age; 2 half of the

females are ovulating at the minimium age, the remainder are not; 3 a third

of the females are ovulating at the minimum age, the remainder are at the

two non-ovulating stages of the ovulation cycle. At each time step, females

move to the next stage of ovulation cycle.

(c) Females that complete reproduction start a new ovulation cycle. Ovulation

stage is sampled for each female. Sampling is from a uniform distribution

as described in (b).

7) Females are in gestation or days open for *gestation\_length*+*daysOpen* time steps

after mating. During the first *gestation\_length*+*daysOpen*-1 time steps, these

females cannot be culled, they can only be selected for a limited number of

selection destinies (see namelists &SELECTION and &EVA), and they cannot be

culled for not being selected. Females in gestation or days open are available

for culling and unrestricted selection at *gestation\_length*+*daysOpen* time steps

after mating. Females in gestation or days open are females that are mated and

produce offspring.

8) (a) *proportionEmptyMales* represents the proportion of reproductive males that

do not mate at the current time step. They can be mated at the next

timestep.

(b) *proportionEmptyFemales* represents the proportion of ovulating females that

do not mate at current time step. They can be mated at their next ovulation

cycle.

9) *OldAgeMales* and *OldAgeFemales* must be greater than the maximum reproduction

ages of living males and females. The maximum reproduction ages are specified

in namelist &SELECTION, variable selection\_scheme, via *sex\_code*, *age(2)*, and

destiny\_sel.

10) Animals that are randomly culled can have observations realised at culling.

The observations realised and the age at which culling needs to take place

for a trait(s) to be realised is specified in &OBSERVATIONS. The exception

is that animals randomly culled after conception and prior to birth (i.e.,

age 1-*gestation\_length* to -1) can never have observations realised. Animals

must be born to have observations realised.

11) RandomCullingGroups is only required when *nRandomCullingGroups*>0. That is,

there is no random culling when *nRandomCullingGroups* 0.

12) Number of lines provided for RandomCullingGroups must equal

nRandomCullingGroups

13) *sex*, *proportion*, and *cause* are the only variables that need to be considered

when culling randomly. *sex* must be 1 or 2, 0.0≤*proportion*≤1.0, *cause* 1 or 3.

All other variables associated with random culling, *firstAge*, *lastAge*,

firstHerd, lastHerd, timeSinceLastMating, and parity, do not need to be

considered. They can all be -9999 or -9.

14) *firstAge* and *lastAge* can range from 1-*gestation\_length* to OldAge-1, where

OldAge is *OldAgeMales* for males and *OldAgeFemales* for females. This implies

that males and females can be randomly culled after they have been

conceived and prior to birth. Not only can this make sense biologically, it

can be useful from a simulation view point. In some instances, it can

reduce the number of time steps required.

15) *firstAge* -9999, *lastAge* must be -9999. When *lastAge* -9999, *firstAge* must

be -9999.

16) *firstHerd* -9, *lastHerd* must be -9. When *lastHerd* -9, *firstHerd* must be -9.

17) The largest parity derived from the input provided in namelist selection,

variable *selection\_scheme* is the largest possible parity a female can

obtain. It does not necessarily mean that females obtain this parity. All

females may, for example, be culled before they obtain this or earlier

parities. In these cases, some obervations will not be realised.

\*\*Relation to subsequent namelists\*\*

&BLUPPARAMETERS

Namelist &BLUPPARAMETERS is required when *nebv*>0 and there is a selection stage(s)

using estimated aggregate-breeding value as selection criterion. These selection

stages are stages of namelist &SELECTION, variable *selection\_groups* with (i) *sex\_code*

0:6 and *selection\_criterion* ‘polyblup’, ‘genomicblup’, ‘ibdblup’, ~~or ‘gas’~~, and (ii)

*sex\_code* 7 and *MaleSelCrit* or *FemaleSelCrit* ‘polyblup’, ‘genomicblup’, ‘ibdblup’, ~~or~~

~~‘gas’~~ in corresponding EVA-selection stage of namelist &EVA, variable *EvaSelection*.

&COMBINEDOBSERVATIONS

Namelist &COMBINEDOBSERVATIONS is required when *nCombinedObs*>0

&OBSERVATIONCONSTRAINTS

Namelist &OBSERVATIONCONSTRAINTS is required when *nObsConstraints*>0

&ECONOMICVALUES

Namelist &ECONOMICVALUES, variable *economicValueEbv* is required when *nebv*>0

## **&BASEPOPULATION**

age\_class\_base=*age\_class\_base*

max\_age\_base\_males=*max\_age\_base\_males*

max\_age\_base\_females=*max\_age\_base\_females*

age\_class\_male=*age\_class\_male*

age\_class\_female=*age\_class\_female*

sampleFounderAsBase=*sampleFounderAsBase*

realiseBaseObservations=*realiseBaseObservations* /

\*\*Task\*\*

Input to generate the base population.

\*\*Properties of names\*\*

*age\_class\_base* Definition: Method used to generate the age structure of

the base population

Type: Character

Options:

no\_structure All base animals generated have

birth time 0. The numbers of

base males and females generated in

each herd is determined by the size

of the herds. Herd sizes are

generated using information provided

in namelist &SELECTION, variable

selection\_scheme.

create\_structure Age structure generated using

information provided in namelist

&SELECTION, variable

*selection\_scheme*. Birth times for

the base animals range from

1-AgeMales-*gestation\_length* to 0 for

males and

1-AgeFemales-*gestation\_length* to 0

for females, where AgeMales and

AgeFemales are the maximum

reproductive ages for male and

females (see Additional

Information). This method ensures

enough base males and females to

cover reproduction in each time

step.

input Age structure of base population is

user-provided. Input is provided in

*max\_age\_base\_males*,

*max\_age\_base\_females*, *age\_class\_male*,

and *age\_class\_female*.

Default: no\_structure

*max\_age\_base\_males* Definition: Oldest age at which males in the base

population are reproductive; measured in time

steps. Used when *age\_class\_base* ‘input’.

Type: Integer

Options: ≥1

Default: 1

*max\_age\_base\_females* Definition: Oldest age at which females in the base

population are reproductive; measured in time

steps. Used when *age\_class\_base* ‘input’.

Type: Integer

Options: ≥1

Default: 1

*age\_class\_male* Definition: Matrix with numbers of base males in each

npop x (age-class x herd) combination at time step

1-*gestation\_length*. Used when *age\_class\_base*

‘input’.

Dimension:(gestation\_length+max\_age\_base\_females x nherd)xnpop

Bounds: (1-*gestation\_length*):*max\_age\_base\_females*, 1:*nherd,*

1:*npop*

Type: Integer

*age\_class\_female* Definition: Matrix with numbers of base females in each

npop x (age-class x herd) combination at time step

1-*gestation\_length*. Used when *age\_class\_base*

‘input’.

Dimension:

(gestation\_length+max\_age\_base\_females x nherd) x npop

Bounds: (1-*gestation\_length*):*max\_age\_base\_females*, 1:*nherd,*

1:*npop*

Type: Integer

*realiseBaseObservations*

Definition: Observations realised for base animals that

fulfil the criteria specified in namelist

&OBSERVATIONS

Type: Character

Options:

yes Observations are realised

no Observations are not realised

Default: yes

*sampleFounderAsBase*

Definition: Sample founder indivdiuals as base individuals.

Type: Character

Options:

yes sample founder indivdiuals as base indivdiuals.

no (default) Sample haplotype to make base individuals.

Default: no

\*\*Additional information\*\*

1) Information provided in namelists &POPULATIONPARAMETERS, &SELECTION, and &EVA

is used to generate the base population when *age\_class\_base* ‘no\_structure’ and

‘create\_structure’

2) *age\_class\_base* ‘no\_structure’

(a) Variables max\_age\_base\_males, max\_age\_base\_females, age\_class\_male and

*age\_class\_female* are not used

(b) All base males and females have the same age. They are all born at time

0. The consequence of this is that, when *gestation\_length* and/or the

minimum reproductive age of the base animals is greater than 0, there

will be time steps with no selection candidates available for

reproduction. This does not cause the program to stop.

(d) When *nherd*>1, base animals are only allocated to those herds where animals

are selected for reproduction; irrespective of the age of the base animals.

[MH: Check if this applies to ‘no\_structure’!]

(d) When germ plasm is used for reproduction,

(i) Germ plasm is stored for all base males if male germ plasm is used

for reproduction. The number of germ plasm stored is the sum of

all germ plasm stored for males during a time step. That is, the

sum of germ plasm stored at each stage of &NAMELIST selection,

variable *selection\_scheme* with *sex\_code* 1, 3, or 5 and *destiny\_sel*

2, 12, 26, or 126 and &NAMELIST EVA, variable *EvaSelection* with

*MaleDestinySel* 2, 12, 26, or 126.

(ii) Germ plasm is stored for all base females if female germ plasm is

used for reproduction. The number of germ plasm stored is the sum

of all germ plasm stored for females during a time step. That is,

the sum of germ plasm stored at each stage of &NAMELIST selection,

variable *selection\_scheme* with with *sex\_code* 2, 4, or 6 and

*destiny\_sel* 2, 12, 127, or 147 and &NAMELIST EVA, variable

*EvaSelection* with *FemaleDestinySel* 2, 12, 127, or 147.

3) *age\_class\_base* ‘create\_structure’

(a) Variables *max\_age\_base\_males*, *max\_age\_base\_females*, *age\_class\_male*, and

*age\_class\_female* are not used

(b) The age structure of base animals in each herd ranges from

1-*gestation\_length* to AgeMales for males and 1-*gestation\_length* to

AgeFemales for females, where AgeMales and AgeFemales are the maximum

reproductive ages for male and females in the herd. This converts to

birth times that range from 1-*gestation\_length*-AgeMales to 0 for males

and 1-*gestation\_length*-AgeFemales to 0 for females. The consequence of

this is that there should be sufficient selection candidates available

for reproduction in each time step.

(c) The first offspring are born in time step 1

(d) When *nherd*>1, base animals are only allocated to those herds where animals

are selected for reproduction; irrespective of the age of the base animals.

(e) When germ plasm is used for reproduction,

(i) Germ plasm is stored for base males if male germ plasm is used

for reproduction. All males that are younger than or at the same

age as the maximum age at which male germ plasm is used have germ

plasm stored. The number of germ plasm stored is the sum of all

germ plasm stored for males during a time step, as described for

*age\_class\_base* ‘no\_structure’.

(ii) When the maximum age at which males are used for reproduction is

greater than *OldAgeMales*-1, it is assumed that base males older than

*OldAgeMales*-1 are dead and died from old age at time BirthMale*i* +

*OldAgeMales*, where BirthMale*i* is the birth time of the *i*th male.

(iii) 2(b) (i) and (ii) applies to females.

4) *age\_class\_base* ‘input’

(a) Variables *max\_age\_base\_males*, *max\_age\_base\_females*, *age\_class\_male*, and

*age\_class\_female* are used

(b) It is useful to set *max\_age\_base\_males* and *max\_age\_base\_females* to

the maximum reproductive ages for males and females specified in namelist

&SELECTION, variable *selection\_scheme*.

(c) *age\_class\_male* and *age\_class\_female* provide the age structure of base males

and females at time step 1-*gestation\_length*

(d) Bounds of *age\_class\_male* and *age\_class\_female* are

(1-*gestation\_length*):*max\_age\_base\_males* x 1:*nherd* and

(1-*gestation\_length*):*max\_age\_base\_females* x 1:*nherd*. The first input datum

provided for *age\_class\_male* is element (1-*gestation\_length*, 1); the number

of males with age 1-*gestation\_length* in herd 1 at time step

1-*gestation\_length*. Birth times of the base animals range from

1-*gestation\_length*-*max\_age\_base\_males* to 0 for males and

1-*gestation\_length*-*max\_age\_base\_females* to 0 for females.

(e) The program stops when insufficient or surplus data is provided

(f) The consequence of the age structure being user specified is that, if

insufficient base animals are specified, there will be time steps with

insufficient selection candidates available for reproduction. This does not

cause the program to stop.

(g) When germ plasm is used for reproduction, germ plasm is stored for base

males, as described for *age\_class\_base* ‘create\_structure’.

5) When *realiseBaseObservations* ‘yes’, *nherd*>1, and *age\_class\_base* ‘no\_structure’

or ‘create\_structure’, some base animals may not fulfil the criteria, *FirstHerd*

and *LastHerd*, specified in namelist &OBSERVATIONS. These animals will not be

observed for those traits for which these criteria are specified. Using

*age\_class\_base* ‘input’ with correct input will enable base animals to be

observed for traits specified in namelist &OBSERVATIONS when *nherd*>1.

## **&****SELECTION**

selection\_groups=*selection\_groups*

selection\_scheme=

*FirstTime LastTime sex\_code live\_status population\_id(1) population\_id(2)*

*age(1) age(2) herd\_id(1) herd\_id(2) identity(1) identity(2) parity(1) parity (2)*

*selection\_method selection\_criterion MendelianIndexWeight RunBlup numbers*

*related\_stage othertags tag destiny\_sel matingGroup TestPopulation TestHerd*

*destiny\_unsel repro\_capacity litterSizeAllocation litter\_size sexAllocation*

*sex\_ratio germ\_plasm SelectionRule* *otherSelectionLayer* /

NB! Number of lines in selection\_scheme must equal *selection\_groups*

When *sex\_code* 7 (EVA selection), all remaining variables in selection\_scheme

are read, but not used. See 1) of ‘Additional information’.

\*\*Task\*\*

Input for selection

\*\*Properties of names\*\*

*selection\_groups* Definition: Number of selection stages

Type: Integer

Options: ≥1

Default: 1

*FirstTime*  Definition: First time step where the selection group is used

Type: Integer

Options:

1-gestation\_length≤FirstTime≤LastTime

Default: *FirstTime* must be specified

*LastTime*  Definition: Last time step where the selection group is used

Type: Integer

Options: *FirstTime*≤*LastTime*≤*ntime*

Default: *LastTime* must be specified

*sex\_code* Definition: Type of selection

Type: Integer

Options:

0 Truncation or select-all selection ignoring sex

of candidates

1 Truncation or select-all selection of males

2 Truncation or select-all selection of females

3 Truncation or select-all selection of males born

out of bull-dams

4 Truncation or select-all selection among bull

dams previously selected in current time step

5 Truncation or select-all selection among males

previously tagged in current time step; tag

specified by variable tag

6 Truncation or select-all selection among females

previously tagged in current time step; tag

specified by variable tag

7 EVA-selection among males and/or females tagged for

EVA-selection

8 Group selection; truncation or select-all selection of

groups of animals

9 Family selection; truncation or select-all selection of

full-sib families

Default: *sex\_code* must be specified

*live\_status* Definition: Live status of selection candidates

Type: Integer

Options:

0 Animal is candidate for selection if it is alive

or it is not alive and has germplasm stored

1 Only live animals are candidates for selection

2 Only animals not alive and with germplasm stored are

candidates for selection

3 Live and dead animals are candidates for selection

~~4 Only dead animals are candidates for selection~~

Default: *live\_status* must be specified

Note: *live\_status=3* may increase computation challenges. And some options may not work due to logical reasons (dead animals cant reproduce…). It doesnot work with OCS.

*population\_id(1)* Definition: First pop within which population is selected.

Type: Integer

Options: 1≤*population\_id(1)*≤ *population\_id(2)*

Default: *population\_id(1)* must be specified

*population\_id(2)* Definition: Last pop within which population is selected.

Type: Integer

Options: *population\_id(1)*≤*population\_id(2)*≤npop

Default: *population\_id(2)* must be specified

*age(1)* Definition: Youngest age at which animals are considered

for selection

Type: Integer

Options:

*age(1)*≥1 when *gestation\_length* 0

*age(1)*≥0 when *gestation\_length*>0

Default: *age(1)* must be specified

*age(2)* Definition: Oldest age at which animals are considered

for selection

Type: Integer

Options: *age(2)*≥*age(1)*

Default: *age(2)* must be specified

*herd\_id(1)* Definition: First herd within which animals are considered

for selection

Type: Integer

Options: 1≤*herd\_id(1)*≤*nherd*

Default: *herd\_id(1)* must be specified

*herd\_id(2)* Definition: Last herd within which animals are considered

for selection

Type: Integer

Options: *herd\_id(1)*≤*herd\_id(2)*≤*nherd*

Default: *herd\_id(2)* must be specified

*identity(1)* Definition: Generation/identity in a breeding cycle (plant).

This refers to P0, F1, F2...Fn. First identity

is considered for selection.

Type: Integer

Options: -9 identity ignored

1≤ *identity(1)*≤ *identity(2)*

Default: *identity(1)* must be specified

*identity(2)* Definition: Generation/identity in a breeding cycle (plant).

This refers to P0, F1, F2...Fn. Last identity

is considered for selection.

Type: Integer

Options: -9 identity ignored

*identity(1)*≤ *identity(2)*

Default: *identity(2)* must be specified

*parity(1)* Definition: Parity of animals.

First parity is considered for selection.

Type: Integer

Options: -9 *parity* ignored

1≤ *parity (1)*≤ *parity (2)*

Default: *parity (1)* must be specified

*parity(2)* Definition: Parity of animals.

Last parity is considered for selection.

Type: Integer

Options: -9 *parity* ignored

*parity (1)*≤ *parity (2)*

Default: *parity (2)* must be specified

*selection\_method* Definition: Method of selection

Type: Character

Options:

select\_all Select all animals

threshold Threshold selection

truncation Truncation selection

Default: *selection\_method* must be specified

*selection\_criterion* Definition: Criterion on which animals are selected

Type: Character

Options:

tbv True breeding value

random Random variable

phenothreshold Phenotypic observation(s) with associated

threshold(s)

phenoweight Phenotypic selection with observations

weighted

polyblup Polygenic-BLUP breeding value

genomicblup Genomic-BLUP breeding value

ibdblup IBD-BLUP breeding value

bayesp BayesP-breeding values

Default: *selection\_criterion* must be specified

When bayesP, *geneticModel* must be ‘genomic’, nebv==1, and genotyped\_at\_birth)/='yes'

*MendelianIndexWeight* Definition: Mendelian-index weight used as modify selection

criterion. The modified criterion for individual

*i* is c*i*=*b*\*(*ai*-*PAi*)+(1-*b*)\**PAi*, where *b* is

*MendelianIndexWeight*, *ai* is its breeding value,

and PA*i* is its parental average (efter Grundy *et*

al. 1999, Wu & Schaeffer 2000).

Type: Real

Options: 0.0<*MendelianIndexWeight*<1.0

Default: *MendelianIndexWeight* must be specified

*RunBlup* Definition: Estimate BLUP-breeding values

Type: Integer

Options:

0 Breeding values are not estimated

1 Breeding values are estimated using DMU

2 Breeding values are estimated using own program from users

Default: *RunBlup* must be specified

*numbers* Definition: Total number of animals to be selected. This must

be provided unless selection method is select\_all.

Type: Integer

Options: ≥1

Default: *numbers* must be specified

*related\_stage* Definition: Animals selected in the current stage of

selection are automatically selected in

selection stage *related\_stage* of the current

time step. The total number of selected animals

in selection stage *related\_stage* includes the

number of animals selected in the current

selection stage.

Type: Integer

Options: 0 No related stages

2≤*related\_stage*≤selection\_groups

Default: *related\_stage* must be specified

*otherTags*  Definition: other type of tags

Type: Integer

Options:

0 Ignore tags

1..9 need more descriptions. Add here later

Default: *CheckHerdsize* must be specified

! 0: Ignore tags

! 1,2,3,4,5,..9: selected individuals in this selection stage are tagged with a tag number as 1,2,..9 (apart from destination of selection, animals could be also tagged with a number from 1...9).

! 11,12,13...19: only individuals with tag number of 1,2,...9 are consider as candidate of selections.

! 21,22,23...29: only individuals with tag number of 1,2,...9 are consider as candidate of selections. If animals are selected, tag of the individual removed.

! 31,32,33...39: Tag number of 1,2,...9 for only individuals with tag number of 1,2,...9, respectively are removed for the selected indivdiuals.

! 41,42,43...49: tag number of 1,2,...9 are revomed for the individuals that pass conditions of all other selection criteria (herd, pop, age...).

! 50: all tag numbers are revomed for the individuals that pass conditions of all other selection criteria (herd, pop, age...).

! 101,102...109: Par1 of selected individuals in this selection stage are tagged with a tag number as 1,2,..9 (apart from destination of selection, animals could be also tagged with a number from 1...9).

! 111,112,113...119: only individuals that are offspring of par1 with tag number of 1,2,...9 are consider as candidate of selections.

! 121,122,123...129: only individuals that are offspring of par1 with tag number of 1,2,...9 are consider as candidate of selections. If the individuals are selected, tag of the par1 removed.

! 131,132,133...139: Tag number of 1,2,...9 for par1 of individuals with tag number of 1,2,...9, respectively, are removed for par1 of the selected indivdiuals.

! 141,142,143...149: tag number of 1,2,...9 are revomed for par1 of the individuals that pass conditions of all other selection criteria (herd, pop, age...).

! 150: all tag numbers are revomed for par1 of the individuals that pass conditions of all other selection criteria (herd, pop, age...).

! 201,202...209: Par2 of selected individuals in this selection stage are tagged with a tag number as 1,2,..9 (apart from destination of selection, animals could be also tagged with a number from 1...9).

! 211,212,213...219: only individuals that are offspring of par2 with tag number of 1,2,...9 are consider as candidate of selections.

! 221,222,223...229: only individuals that are offspring of par2 with tag number of 1,2,...9 are consider as candidate of selections. If the individuals are selected, tag of the par2 removed.

! 231,232,233...239: Tag number of 1,2,...9 for par2 of individuals with tag number of 1,2,...9, respectively, are removed for par2 of the selected indivdiuals.

! 241,242,243...249: tag number of 1,2,...9 are revomed for par2 of the individuals that pass conditions of all other selection criteria (herd, pop, age...).

! 250: all tag numbers are revomed for par2 of the individuals that pass conditions of all other selection criteria (herd, pop, age...).

! 301,302,303,..309: Not yet implemented. Offspring as results of matings from selected individuals in this selection stage are tagged with a tag number as 1,2,..9.

*tag* Definition: Animals with specified tag are selection

candidates when *sex\_code* 5 or 6

Type: Integer

Options:

90 Selection among candidates tagged to be left

alive

91 Selection among candidates tagged to reproduce

(mate)

92 Selection among candidates tagged to have germ

plasm stored

93 Selection among candidates tagged for genotyping

94 Selection among candidates tagged for transfer

to another (test) herd

955 Selection among candidates tagged to be allocated to

groups

96 Selection among candidates tagged to have DYD

sampled (only available for males)

98 Selection among candidates tagged to have phenotypic

observations realised at current selection stage

99 Selection among candidates tagged to be culled

voluntarily

912 Selection among candidates tagged to reproduce

and have germ plasm stored

914 Selection among candidates tagged to reproduce

after transfer to another (test) herd

916 Selection among candidates tagged to reproduce

and have DYD sampled (only available for males)

917 Selection among candidates tagged to reproduce

as bull-dams (only available for females)

926 Selection among candidates tagged to have germ

plasm stored and DYD sampled (only available for

males)

9126 Selection among candidates tagged to reproduce,

have germ plasm stored, and DYD sampled (only

available for males)

9127 Selection among candidates tagged to reproduce

as bull-dams and have germ plasm stored (only

available for females)

9147 Selection among candidates tagged to reproduce

as bull-dams after transfer to another (test)

herd (only available for females)

Default: *tag* must be specified

*destiny\_sel* Definition: Defines the destiny of selected animals

Type: Integer

Options:

0 Selected candidates are left alive

1 Selected candidates are reproduced (mated)

2 Selected candidates have germ plasm stored

3 Selected candidates are genotyped

4 Selected candidates are transferred to another

(test) herd; the herd is specified by *TestHerd*

5 Selected candidates are allocated to groups

6 Selected candidates have DYD sampled (only

available for males)

8 Selected candidates have phenotypic observation(s)

realised at current selection stage

9 Selected candidates are culled voluntarily (only for

*selection\_method* ‘select\_all’)

12 Selected candidates are reproduced and have germ

plasm stored

14 Selected candidates are reproduced after

transfer to another (test) herd; the herd is

specified by *TestHerd*

16 Selected candidates are reproduced and have DYD

sampled (only available for males)

17 Selected candidates are reproduced as bull-dams

(only available for females)

26 Selected candidates have germ plasm stored and

DYD sampled (only available for males)

48 Selected candidates are transferred to another

(test) herd and have phenotypic observation(s)

realised at current selection stage

126 Selected candidates are reproduced, have germ

plasm stored, and DYD sampled (only available

for males)

127 Selected candidates are reproduced as bull-dams

and have germ plasm stored (only available for

females)

147 Selected candidates are reproduced as bull-dams

after transfer to another (test) herd; the herd

is specified by *TestHerd* (only available for

females)

90 Selected candidates are tagged as candidates to

be left alive

91 Selected candidates are tagged as candidates to

reproduce (mate)

92 Selected candidates are tagged as candidates to

have germ plasm stored

93 Selected candidates are tagged as candidates for

genotyping

94 Selected candidates are tagged as candidates for

transfer to another (test) herd

95 Selected candidates are tagged as candidates for

EVA selection

~~955 Selected candidates are tagged as candidates to be~~

~~allocated to groups~~

96 Selected candidates are tagged as candidates to

have DYD sampled (only available for males)

98 Selected candidates are tagged as candidates to have

a phenotypic observation(s) realised at a subsequent

selection stage

99 Selected candidates are tagged as candidates to be

culled voluntarily

912 Selected candidates are tagged as candidates to

reproduce and have germ plasm stored

914 Selected candidates are tagged as candidates to

reproduce after transfer to another (test) herd

916 Selected candidates are tagged as candidates to

reproduce and have DYD sampled (only available

for males)

917 Selected candidates are tagged as candidates to

reproduce as bull-dams (only available for

females)

926 Selected candidates are tagged as candidates to

have germ plasm stored and DYD sampled (only

available for males)

9126 Selected candidates are tagged as candidates to

reproduce, have germ plasm stored, and DYD

sampled (only available for males)

9127 Selected candidates are tagged as candidates to

reproduce as bull-dams and have germ plasm

stored (only available for females)

9147 Selected candidates are tagged as candidates to

reproduce as bull-dams after transfer to another

(test) herd (only available for females)

Default: *destiny\_sel* must be specified

*matingGroup* Definition: Mating group for animals selected for reproduction

Type: Integer

Options: 0 No mating group; animals not selected for

reproduction

1≤*matingGroup*≤*nMatingGroups*, where *nMatingGroups* is

the number of mating groups

Default: *matingGroup* must be specified

*TestPopulation* Definition: Test pop to which selected animals are transferred

Type: Integer

Options: 1<=*TestPopulation*<=npop

Default: *TestPopulation* must be specified

*TestHerd* Definition: Test herd to which selected animals are transferred

Type: Integer

Options: 1≤*TestHerd*≤*nherd*

Default: *TestHerd* must be specified

*destiny\_unsel* Definition: Destiny of candidates that are not selected

Type: Integer

Options:

0 Animal is culled

1 Animal is left alive

Default: *destiny\_unsel* must be specified

*repro\_capacity* Definition: Reproductive capacity (number of matings)

of selected individuals (biologically

constrained or otherwise)

Type: Integer

Options: ≥1

Default: *repro\_capacity* must be specified.

*litterSizeAllocation*  Definition: Method used to assign number of offspring in each

litter

Type: Character

Options:

fix Fixed number of offspring in each litter; litter

size specified by *litter\_size*

phenotype Number of offspring is phenotype of dam; sampled

for observation number provided in *litter\_size*

Default: *litterSizeAllocation* must be specified

*litter\_size* Definition: Number of offspring per mating or observation

number of trait used to sample number of

offspring

Type: Integer

Options: ≥1

Default: *litter\_size* must be specified

*sexAllocation* Definition: Method to assign sex to offspring within litters

Type: Character

Options:

fix Fixed number of males and females in each litter

sample Sex of each offspring is sampled

Default: *sexAllocation* must be specified

*sex\_ratio* Definition: Probability of an offspring being male

Type: Real

Options: 0.0≤*sex\_ratio*≤1.0

Default: *sex\_ratio* must be specified

*germ\_plasm* Definition: Number of units of germplasm stored per

selected animal

Type: Integer

Options: *germ\_plasm*≥0

Default: *germ\_plasm* must be specified

*SelectionRule* Definition: Selection rule number; only valid for

truncation selection

Type: Integer

Options:

0 No selection rule

≥1 Selection rule. Rule number must correspond to a

rule number specified in namelist rules, variable

SelectionRules.

Default: *SelectionRule* must be specified

*otherSelectionLayer* Definition: apply selection rules across, and/or within unit

at selection stage.

Type: Integer

Options:

0: No otherSelectionLayer applied at selection stage

1: 2nd layer applied between and within unit.

With this option, Namelist &SELECTIONTOOLEXTRA must

be provided

Default: *otherSelectionLayer* must be specified

\*\*Relation to subsequent namelists\*\*

&PHENOTHRESHOLDS

Namelist &PHENOTHRESHOLDS is required when *selection\_criterion* ‘phenothreshold’

in any selection stage of namelist &SELECTION, variable selection\_scheme

&RULES

Namelist &RULES is required when *sex\_code* 0:6, *selection\_method* ‘truncation’, and

*rule*>0, in any selection stage of namelist &SELECTION, variable selection\_scheme

&EVA

Namelist &EVA is required when *sex\_code* 7 in any selection stage of namelist

&SELECTION, variable selection\_scheme. The number of EVA-selection stages in

namelist &EVA, variable EvaSelection must equal the number of selection stages in

namelist &SELECTION, variable selection\_scheme with EVA-selection (i.e., *sex\_code*

7)

&PHENOWEIGHTS

Namelist &PHENOWEIGHTS is required when *selection\_criterion* ‘phenoweight’

in any selection stage of namelist &SELECTION, variable selection\_scheme

&GROUPPARAMETERS

Namelist &GROUPPARAMETERS is required when *destiny\_sel* 55 at any selection stage of

namelist &SELECTION, variable selection\_scheme

&BLUPPARAMETERS

Namelist &BLUPPARAMETERS is required when *selection\_criterion* ‘polyblup’,

‘genomicblup’, ‘ibdblup’, ~~or ‘gas’~~ at any selection stage of namelist &SELECTION,

variable selection\_scheme

**&GENOMICBLUPPARAMETERS**

&IBDBLUPPARAMETERS

&BAYESPPARAMETERS

\*\*Additional information\*\*

1) All variables in namelist &SELECTION, variable selection\_scheme are read

and, therefore, must be specified. However, there are cases where some

variables are not used. The most striking case is when *sex\_code* 7. This

specifies that EVA selection is to be carried out. The information required to

carry out EVA selection is not provided in selection\_scheme, but in namelist

&EVA. Another case is when selected animals are not destined to be mated. In

this case, variables, such as *repro\_capacity*, *litter\_size, and sex\_ratio*, are

read, but not used. For variables that are not used, the input provided is

irrelevant. However, for clarity, we recommend providing 0 for integer

variables, 0.0 for *sex\_ratio* (the only real), and ‘ ‘ for character variables.

2) sex\_code

(a) *sex\_code* must be 0:7

(b) The number of selection stages with *sex\_code* 7 (selection stages with EVA-

selection) must equal the number of EVA-selection stages provided in

namelist &EVA

3) (a) *live\_status* must be 0, 1, or 2 when *sex\_code* 0:6

(b) *live\_status* can only be 0 and 2 when

(i) *sex\_code* 1:4 and *destiny\_sel* 1

(ii) *sex\_code* 5 or 6, *tag* 91, and *destiny\_sel* 1

(iii) *sex\_code* 0:4 and *destiny\_sel* 91 or 95

(iv) *selection\_unit* not ’family’

(c) When *live\_status* 0 or 2, *destiny\_sel* 1, and a deceased animal with stored

germ plasm is selected for reproduction, the animal’s reproductive capacity

is defined as follows. If the number of stored germ plasm is greater or

equal to *repro\_capacity*, the animal’s reproductive capacity is

*repro\_capacity*. If, however, the number of stored germ plasm is less

than *repro\_capacity*, the animal’s reproductive capacity is the number

of stored germ plasm. The consequence of this is that, insufficient

stored germ plasm can lead to less matings being carried out than

planned.

5) *population\_id* must be 1 when *sex\_code* 0:6

7) (a) *age(1)* and *age(2)* must be provided when *sex\_code* 0:6

(b) age(2)<max(OldAgeMales,OldAgeFemales) when sex\_code 0 and live\_status 1

(c) age(2)<OldAgeMales when sex\_code 1, 3, or 5 and live\_status 1

(d) age(2)<OldAgeFemales when sex\_code 2, 4, or 6 and live\_status 1

(e) There is no limit on *age(1)* and *age(2)* when *live\_status* 0 or 2

(f) *age(2)*≥*age(1)*≥0 implies that animals cannot be selected before they are

born

8) *herd\_id(1)* and *herd\_id(2)* must be provided when *sex\_code* 0:6 and

s*election\_unit* ‘across\_herd’, ‘within\_herd’, or ‘family’. Otherwise,

*herd\_id(1)* and *herd\_id(2)* are not used.

9) (a) *selection\_method* must be ‘select\_all’, ‘threshold’, or ‘truncation’ when

*sex\_code* 0:6

(b) When selection\_method ‘select\_all’, *selection\_criterion*,

*MendelianIndexWeight*, *RunBlup*, *numbers*, *related\_stage*, *CheckHerdsize*,

*destiny\_unsel*, *repro\_capacity*, *litter\_size*, *sex\_ratio*, and *SelectionRule*

are not used. They are set to 0; 0.5 for *MendelianIndexWeight*.

(c) When selection\_method ‘threshold’, *MendelianIndexWeight*, *RunBlup*, *numbers*,

*related\_stage*, *CheckHerdsize*, *repro\_capacity*, *litter\_size*, *sex\_ratio*, and

*SelectionRule* are not used. They are set to 0; 0.5 for

MendelianIndexWeight.

(d) When *selection\_method* ‘select\_all’ or ‘threshold’, the destiny of the

selected candidates cannot include reproduction. That is, *destiny\_sel*

cannot be 1, 12, 14, 16, 17, 126, 127, or 147.

(e) When *selection\_method* ‘select\_all’, *selection\_unit* cannot be ‘family’

10) (a) *selection\_criterion* must be ‘tbv’, ‘random’, ‘phenoweights’, ‘polyblup’,

‘genomicblup’, ‘ibdblup’, ~~or ‘gas’~~ when *sex\_code* 0:6 and *selection\_method*

‘truncation’

(b) *selection\_criterion* must be ‘phenothreshold’ when *sex\_code* 0:6 and

*selection\_method* ‘threshold’

(c) When *geneticModel* ‘polygenic’, *selection\_criterion* cannot be

‘genomicblup’, ‘ibdblup’, ~~or ‘gas’~~ at any selection stage

(e) When *geneticModel* ‘genomic’, both ‘genomicblup’ ~~and ‘gas’~~ cannot be used as

selection criterion in a simulated breeding scheme. That is, both

‘genomicblup’ ~~and ‘gas’~~ cannot be specified as *selection\_criterion* in

namelist &selection, variable selection\_scheme, and/or as *MaleSelCrit* or

*FemaleSelCrit* in any EVA-selection stage of namelist &EVA, variable

EvaSelection.

(f) When *selection\_criterion* ‘tbv’, economic weight(s) provided in

*economic\_weight* in namelist &MATRICES are used to weight the true breeding

values for each trait

When *selection\_criterion* ‘polyblup’, ‘genomicblup’, ‘ibdblup’, ~~or ‘gas’,~~

economic weight(s) provided in *economic\_weight* in namelist &MATRICES

are used to weight the estimated breeding values for each trait

When *selection\_criterion* ‘phenoweights’, phenotypic weight(s) provided in

namelist &PHENOWEIGHTS are used to weight the phenotypes for each trait

When bayesP, *geneticModel* must be ‘genomic’, nebv==1, and genotyped\_at\_birth)/='yes'

11) (a) *MendelianIndexWeight* available when *selection\_method* ‘truncation’ and

*selection\_criterion* ‘random’, ‘polyblup’, ‘genomicblup’, ‘ibdblup’, ~~or ‘gas’~~

(b) Most-common values for *MendeliaIndexWeight*:

0.0 Between-family selection

0.5 Selection for breeding value

1.0 Selection for Mendelian-sampling term

See Grundy et al. (1998) JABG 115:39-51 and Wu & Schaeffer (2000) JABG 117:361-374 for details

12) (a) *RunBlup* must be 0:1 when *sex\_code* 0:6, *selection\_method* ‘truncation’, and

*selection\_criterion* ‘polyblup’, ‘genomicblup’, ‘ibdblup’, ~~or ‘gas’~~

(b) *RunBlup* is not used when *sex\_code* 7, *selection\_method* ‘select\_all’ or

‘threshold’, or *selection\_criterion* ‘tbv’, ‘random’, ‘phenothreshold’, or

‘phenoweights’. It is set to 0.

(d) When *RunBlup* 1, *sex\_code* 0:6, *selection\_method* ‘truncation’, and

*selection\_criterion* ‘polyblup’, ‘ibdblup’, ~~or ‘gas’,~~ ~~and~~

~~selection stage not the first selection stage where~~ *~~selection\_criterion~~*

~~‘polyblup’, ‘genomicblup’, or ‘gas’,~~ BLUP-breeding values are only

estimated if a phenotypic observation(s) has been realised. If

BLUP-breeding values have been estimated previously, they are only

re-estimated when phenotypic observations have been realised since the

previous estimation.

(e) When *RunBlup* 1, *sex\_code* 0:6, *selection\_method* ‘truncation’, and

*selection\_criterion* ‘genomicblup’, BLUP-breeding values are only

estimated if a phenotypic observation(s) has been realised and animals have

been genotyped. If genomic-BLUP breeding values have been estimated

previously, they are only re-estimated when phenotypic observations have been

realised or additional animals have been genotyped since the previous

estimation.

13) (a) *numbers*>0 when *sex\_code* 0:6 and *selection\_method* ‘truncation’

(b) When *selection\_unit* ‘across\_pop’, ‘population’, or ‘across\_herd’,

*numbers* refers to the total number of animals to be selected. When

*selection\_unit* ‘within\_herd’ or ‘family’, *numbers* refers to the number

of animals to be selected from each herd or full-sib family.

14) (a) *related\_stage*≥0 and *related\_stage*≤*selection\_groups* when *sex\_code* 1:6

and *selection\_method* ‘truncation’

(b) *related\_stage* cannot be used when *sex\_code* 0 or *selection\_method*

‘select\_all’ or ‘threshold’ in either the selection stage or related

selection stage. In both cases, related\_stage must be set to 0.

(c) *related\_stage* must be greater than selection stage. This implies that

the minimum related selection stage is 2 in selection stage 1.

(d) *related\_stage* is not used at selection stage *selection\_groups*, the last

selection stage. It is set to 0.

(e) Related selection stage cannot have *selection\_criterion* ‘family’

(f) *related\_stage* cannot be used in combination with EVA selection

(g) (i) When *sex\_code* 1 or 5 in selection stage, the related selection stage

must have *sex\_code* 1 or 5

(ii) When *sex\_code* 2 or 6 in selection stage, the related selection stage

must have *sex\_code* 2 or 6

(iii) When *sex\_code* 3 in selection stage, the related selection stage

must have *sex\_code* 1, 3, or 5

(iv) When *sex\_code* 4 in selection stage, the related selection stage

must have *sex\_code* 2, 4, or 6

(h) Several selection stages can have the same *related\_stage*. All animals

selected in each of these stages are included in the number of animals

selected at selection stage *related\_stage*.

16) (a) *tag* must be specified when *sex\_code* 5:6

(b) *tag* must be 91 and *destiny\_sel* 1 when *sex\_code* 5:6 and *live\_status* 0 or 2

(c) *tag* must correspond to *destiny\_sel*. That is, when *tag* 90, *destiny\_sel*

must be 0, *tag* 91 *destiny\_sel* 1, *tag* 92 *destiny\_sel* 2, *tag* 93

*destiny\_sel* 3, and so on

(d) Tagged animals cannot be re-tagged. That is, *destiny\_sel* cannot be

90:9147 when *sex\_code* 5:6.

(e) Tagged animals must also fulfil the criteria specified by *live\_status*,

population\_id, selection\_unit, age(1), age(2), herd\_id(1), and herd\_id(2)

to be selection candidates

17) (a) *destiny\_sel* must be specified when *sex\_code* 0:6

(b) *destiny\_sel* can only be 1, 91, or 95 when *live\_status* 0 or 2

(c) *destiny\_sel* cannot include reproduction when *selection\_unit* ‘family’ or

selection\_method ‘select\_all’ or ‘threshold’. That is, *destiny\_sel* cannot

be 1, 12, 14, 16, 17, 126, 127, or 147.

Note, however, that animals selected by within-family selection, or by

selection methods ‘select\_all’ and ‘threshold’, can be used for

reproduction by using tags. For example, animals are selected by within-

family selection and tagged. The tagged animals are then selected for

reproduction. In this way, there is control of the number of selection

candidates.

(d) *destiny\_sel* cannot include reproduction or storage of germ plasm when

*sex\_code* 0. That is, *destiny\_sel* cannot be 1, 2, 12, 14, 16, 17, 26,

126, 127, or 147.

(f) Non-reproductive animals are not candidates for selection when *destiny\_sel*

includes reproduction. The criteria to be reproductive are defined in

namelist &POPULATIONPARAMETERS (‘Additional information’). Non-reproductive

males are not candidates when *destiny\_sel* 1, 12, 14, 16, 126, 91, 912, 914,

916, 9126. Non-reproductive females are not candidates when *destiny\_sel* 1,

12, 14, 17, 127, 147, 91, 912, 914, 917, 9127, or 9147.

(g) Non-reproductive animals are not candidates for EVA-selection when destiny

for selection in the subsequent EVA-selection stage includes reproduction.

That is, non-reproductive males are not candidates when *MaleDestinySel* in

namelist &EVA, variable EvaSelection 1, 12, 14, 16, 126, 91, 912, 914, 916,

9126. Non-reproductive females are not candidates when *FeMaleDestinySel* in

namelist &EVA, variable EvaSelection 1, 12, 14, 17, 127, 147, 91, 912, 914,

917, 9127, or 9147.

(h) When *destiny\_sel* 9 (animal culled voluntarity), *selection\_method* must be

‘select\_all’

(i) Females during the first *gestation\_length*+*daysOpen*-1 time steps of

gestation or days opencan only be candidates for selection when:

(i) *destiny\_sel* 0, 3, 8, 90, 93, or 98, and

(ii) *destiny\_sel* 95 and the destiny for EVA-selected females

(*FemaleDestinySel* in namelist &EVA, variable *EvaSelection*) in the

subsequent EVA-selection stage is 0, 3, 8, 90, 93, or 98

(j) *destiny\_sel* can only be 4, 14, 147, 94, 914, or 9147 [transfer to another

(test) herd] when *nherd*>1

(k) Females cannot have DYD sampled. That is, *destiny\_sel* cannot be 6, 16,

26, 126, 96, 916, 926, or 9126 when *sex\_code* 0, 2, 4, or 6.

(l) Males cannot be bull-dams. That is, *destiny\_sel* cannot be 17, 127, 147,

917, 9127, or 9147 when *sex\_code* 0, 1, 3, or 5.

(m) There is only a limited number of valid destinies for selected bull

dams. They are *destiny\_sel* 0, 1, 2, 3, 8, 12, or 95 when *sex\_code* 4.

(n) *destiny\_sel* cannot be 95 (tagged for EVA-selection) when *sex\_code* 0, 5,

or 6

(o) Animals can only be allocated to one group. Candidates already allocated a

group with *destiny\_sel* 55 or *MaleDestinySel* or *FemaleDestinySel* 55 in any

eva-selection stage of namelist &EVA, variable *EvaSelection* are not allocated

to a new group if selected to be allocated to a group in subsequent

selection stages or time steps.

(p) Tagging can be carried out over several selection stages. Tagging is

carried out with two intentions:

(i) Animals are tagged with a view to selection with *sex\_code* 5 or 6

at a subsequent selection stage during the same time step. This

form for tagging is specified with *sex\_code* 0:4 or 7 and *destiny\_sel*

90, 91, 92, 93, 94, 96, 98, 99, 912, 914, 916, 917, 926, 9126, 9127,

or 9147. Animals remain tagged until they are candidates for selection

in a selection stage with *sex\_code* 5 or 6. That is, they fulfil the

criteria specified in a selection stage with *sex\_code* 5 or 6.

(ii) Animals are tagged for EVA selection at a subsequent selection

stage during the same time step with *sex\_code* 1:4 and *destiny\_sel*

95. The one restriction is that, when tagging females, both normal

females or bull dams cannot be tagged together. That is, females

selected for EVA selection cannot be both normal females (*sex\_code* 2)

and bull dams (*sex\_code* 4).

(iii) Non-reproductive animals are not candidates for EVA-selection when

destiny for selection in the subsequent EVA-selection stage includes

reproduction.

• (a) *matingGroup* only used when *selection\_method* 'threshold' and *destiny\_sel*

includes reproduction; that is *destiny\_sel* 1, 12, 14, 16, 17, 126, 127, or 147.

When *matingGroup* is not used, it is set to 0.

(b) Numbering of mating groups must go from 1 … *nMatingGroups*, where *nMatingGroups*

is the number of mating groups.

(c) Animals can also be allocated to *matingGroups* as *MaleSelCrit* or *FemaleSelCrit*

in any EVA-selection stage of namelist &EVA, variable EvaSelection.

(d) Mating groups are allocated within time steps. Mating groups with the same

mating-group number across times steps are different mating groups.

18) Removing tags

(a) Following a selection stage with *sex\_code* 5:6, all tagged candidates,

whether selected or unselected, lose the tag specified by *tag*. Other tags

are not removed. Tagged candidates are animals with the specified *tag* and

fulfil the criteria specified by *live\_status*, *population\_id*,

selection\_unit, age(1), age(2), herd\_id(1), and herd\_id(2). Animals with

the specified *tag* that do not fulfil the specified criteria do not lose

their tag.

(b) Following a selection stage with *sex\_code* 7 (EVA-selection), all animals

tagged for EVA selection (*tag* 95) lose their tag. Therefore, animals need

to be tagged again before EVA-selection can be carried out again.

19) *TestHerd* must be specified when *destiny\_sel* includes transfer to another

(test) herd. That is, *destiny\_sel* 4, 14, or 147

20) *destiny\_unsel* must be 0:1 when *sex\_code* 0:6 and *selection\_method* ‘truncation’

21) *repro\_capacity*>0 when *sex\_code* 1:6, *selection\_method* ‘truncation’,

and *destiny\_sel* 1, 12, 14, 17, 127, or 147. Otherwise, it is set to 0.

22) *litter\_size*>0 when *sex\_code* 2, 4, or 6, *selection\_method* ‘truncation’,

and *destiny\_sel* 1, 12, 14, 17, 127, or 147. Otherwise, it is set to 0.

23) 0.0≤*sex\_ratio*≤1.0 when *sex\_code* 2, 4, or 6, *selection\_method* ‘truncation’,

and *destiny\_sel* 1, 12, 14, 17, 127, or 147. Otherwise, it is set to 0.0.

24) (a) *germ\_plasm*>0 when *sex\_code* 1:6 and *destiny\_sel* 2, 12, 26, 126, or 127

(b) If an animal is reselected to have germ plasm stored within a time step

or during a subsequent time step, the new number of germ plasm is added

to the number currently stored for the animal. The exception is males

and females in the base population.

(c) When *live\_status* 0 or 2, *destiny\_sel* 1, and deceased animals with stored

germ plasm are selected for reproduction, the number of stored germ plasm

for each selected animal is reduced by the number of germ plasm used for

reproduction. The number of germ plasm used for reproduction is

*repro\_capacity* when the number of stored germ plasm is greater than or

equal to *repro\_capacity*. Otherwise, it is the number of stored germ plasm.

25) *SelectionRule* can be used when *sex\_code* 0:6 and *selection\_method* ‘truncation’

26) When the options of only using some selection groups in any particular time

step, the user is extra responsible that all selection groups fits, as there

are no internal checks to control irregularities across clusters of selection

groups.

## **&PHENOTHRESHOLDS**

nPhenotypicThresholds=*nPhenotypicThresholds*

phenotypicThresholds=*stage observation minThreshold maxThreshold* /

NB! *nPhenotypicThresholds* must equal the number of selection stages with selection

criterion ‘phenothreshold’ in namelist &SELECTION, variable selection\_scheme

The number of stages specified in *phenotypicThresholds* must equal

*nPhenotypicThresholds*

\*\*Task\*\*

Specify thresholds for phenotypic selection with associated thresholds

\*\*Properties of names\*\*

*nPhenotypicThresholds* Definition: Number of selection stages where phenotypic

selection with associated thresholds is applied

Type: Integer

Options: 1≤*nPhenotypicThresholds*≤*selection\_groups*

Default: 0

*stage*  Definition: Selection stage

Type: Integer

Options: 1≤*stage*≤*selection\_groups*

Default: *stage* must be specified

*observation*  Definition: Observation number; observation to which phenotypic

thresholds are applied

Type: Integer

Options: 1≤*observation*≤*nobs*+*nCombinedObs*

Default: *observation* must be specified

*minThreshold* Definition: Minimum threshold; animals with an phenotypic value

smaller than *minThreshold* are culled

Type: Real

Options: ≥-999.0

-9999.0 if minimum threshold not applied

Default: *minThreshold* must be specified

*maxThreshold* Definition: Maximum threshold; animals with an phenotypic value

greater than *maxThreshold* are culled

Type: Real

Options: ≥-999.0

-9999.0 if maximum threshold not applied

Default: *maxThreshold* must be specified

\*\*Additional information\*\*

1) *selection\_criterion* in namelist &SELECTION, variable *selection\_scheme* must be

‘phenothreshold’ at each *stage* specified in variable phenotypicThresholds

2) Each *stage* can only be represented once in phenotypicThresholds

3) Both *minThreshold* and *maxThreshold* cannot be -999.0 at the same *stage*

## **&RULES**

SelectionRules=

*rule SireTimeTotal DamTimeTotal FamilyTimeTotal SireMatingsTimeTotal*

*DamMatingsTimeTotal SireTotal DamTotal FamilyTotal SireMatingsTotal*

*DamMatingsTotal reselect*

SireTimeTotals=*SireTimeTotals*

DamTimeTotals=*DamTimeTotals*

FamilyTimeTotals=*FamilyTimeTotals*

SireMatingsTimeTotals=*SireMatingsTimeTotals*

DamMatingsTimeTotals=*DamMatingsTimeTotals*

SireTotals=*SireTotals*

DamTotals=*DamTotals*

FamilyTotals=*FamilyTotals*

SireMatingsTotals=*SireMatingsTotals*

DamMatingsTotals=*DamMatingsTotals* /

NB! Number of selection rules provided in SelectionRules must equal the number of

different rules specified by *SelectionRule* in namelist &SELECTION, variable

selection\_scheme

\*\*Task\*\*

Specify selection rule(s) for truncation selection.

\*\*Properties of names\*\*

The following variables are single integers

*rule* Definition: Selection-rule number; must correspond to

a selection rule in namelist &SELECTION,

variable *SelectionRule*

Type: Integer

Options: ≥1

Default: *rule* must be specified

*SireTimeTotal* Definition: Specifies which element of *SireTimeTotals*

contains the maximum number of offspring

that can be selected from each sire during

each time step

Type: Integer

Options:

≥1 Element of *SireTimeTotals* that contains the

maximum number of offspring that can be selected

from each sire during each time step

-9 Maximum number of offspring not applicable

Default: *SireTimeTotal* must be specified

*DamTimeTotal* Definition: Specifies which element of *DamTimeTotals*

contains the maximum number of offspring

that can be selected from each dam during

each time step

Type: Integer

Options:

≥1 Element of *DamTimeTotals* that contains the

maximum number of offspring that can be selected

from each dam during each time step

-9 Maximum number of offspring not applicable

Default: *DamTimeTotal* must be specified

*FamilyTimeTotal* Definition: Specifies which element of *FamilyTimeTotals*

contains the maximum number of offspring

that can be selected from each full-sib

family during each time step

Type: Integer

Options:

≥1 Element of *FamilyTimeTotals* that contains the

maximum number of offspring that can be selected

from each full-sib family during each time step

-9 Maximum number of offspring not applicable

Default: *FamilyTimeTotal* must be specified

*SireMatingsTimeTotal* Definition: Specifies which element of

*SireMatingsTimeTotals* contains the maximum

number of matings for each sire during

each time step

Type: Integer

Options:

≥1 Element of *SireMatingsTimeTotals* that contains

the maximum number of matings for each sire

during each time step

-9 Maximum number of matings not applicable

Default: *SireMatingsTimeTotals* must be specified

*DamMatingsTimeTotal* Definition: Specifies which element of

*DamMatingsTimeTotals* contains the maximum

number of matings for each dam during each

time step

Type: Integer

Options:

≥1 Element of *DamMatingsTimeTotals* that contains

the maximum number of matings for each dam during

each time step

-9 Maximum number of matings not applicable

Default: *DamMatingsTimeTotals* must be specified

*SireTotal* Definition: Specifies which element of *SireTotals*

contains the maximum number of offspring

that can be selected from each sire across

time steps

Type: Integer

Options:

≥1 Element of *SireTotals* that contains the

maximum number of offspring that can be selected

from each sire

-9 Maximum number of offspring not applicable

Default: *SireTotal* must be specified

*DamTotal* Definition: Specifies which element of *DamTotals*

contains the maximum number of offspring

that can be selected from each dam across

time steps

Type: Integer

Options:

≥1 Element of *DamTotals* that contains the

maximum number of offspring that can be selected

from each dam

-9 Maximum number of offspring not applicable

Default: *DamTotal* must be specified

*FamilyTotal* Definition: Specifies which element of *FamilyTotals*

contains the maximum number of offspring

that can be selected from each full-sib

family across time steps

Type: Integer

Options:

≥1 Element of *FamilyTotals* that contains the

maximum number of offspring that can be selected

from each full-sib family

-9 Maximum number of offspring not applicable

Default: *FamilyTotal* must be specified

*SireMatingsTotal* Definition: Specifies which element of *SireMatingsTotals*

contains the maximum number of matings for

each sire across time steps

Type: Integer

Options:

≥1 Element of *SireMatingsTotals* that contains

the maximum number of matings for each sire

-9 Maximum number of matings not applicable

Default: *SireMatingsTotals* must be specified

*DamMatingsTotal* Definition: Specifies which element of *DamMatingsTotals*

contains the maximum number of matings for

each dam across time steps

Type: Integer

Options:

≥1 Element of *DamMatingsTotals* that contains

the maximum number of matings for each dam

-9 Maximum number of matings not applicable

Default: *DamMatingsTotals* must be specified

*reselect* Definition: Specifies whether the highest-ranking

candidates that do not fulfil the rule(s)

are selected when there are insufficient

selection candidates to fulfil the selection

rule(s)

Type: Integer

Options:

0 Highest-ranking candidates are not selected

1 Highest-ranking candidates are selected

Default: *reselect* must be specified

The following variables are arrays of integers

*SireTimeTotals* Definition: Array that contains the maximum number(s) of

offspring that can be selected from each

sire during each time step

Type: Integer

Dimension: 1 x maximum number specified for

*SireTimeTotal* across rules

Options:

≥1 For each element specified in *SireTimeTotal*

0 For each element not specified in *SireTimeTotal*

Default: *SireTimeTotals* must be specified when

*SireTimeTotal*≥1 for any rule

*DamTimeTotals* Definition: Array that contains the maximum number(s) of

offspring that can be selected from each

dam during each time step

Type: Integer

Dimension: 1 x maximum number specified for

*DamTimeTotal* across rules

Options:

≥1 For each element specified in *DamTimeTotal*

0 For each element not specified in *DamTimeTotal*

Default: *DamTimeTotals* must be specified when

*DamTimeTotal*≥1 for any rule

*FamilyTimeTotals* Definition: Array that contains the maximum number(s) of

offspring that can be selected from each

full-sib family during each time step

Type: Integer

Dimension: 1 x maximum number specified for

*FamilyTimeTotal* across rules

Options:

≥1 For each element specified in *FamilyTimeTotal*

0 For each element not specified in *FamilyTimeTotal*

Default: *FamilyTimeTotals* must be specified when

*FamilyTimeTotal*≥1 for any rule

*SireMatingsTimeTotals* Definition: Array that contains the maximum number(s) of

matings for each sire during each time step

Type: Integer

Dimension: 1 x maximum number specified for

*SireMatingsTimeTotal* across rules

Options:

≥1 For each element specified in *SireMatingsTimeTotal*

0 For each element not specified in

*SireMatingsTimeTotal*

Default: *SireMatingsTimeTotals* must be specified when

*SireMatingsTimeTotal*≥1 for any rule

*DamMatingsTimeTotals* Definition: Array that contains the maximum number(s) of

matings for each dam during each time step

Type: Integer

Dimension: 1 x maximum number specified for

*DamMatingsTimeTotal* across rules

Options:

≥1 For each element specified in *DamMatingsTimeTotal*

0 For each element not specified in *DamMatingsTimeTotal*

Default: *DamMatingsTimeTotals* must be specified when

*DamMatingsTimeTotal*≥1 for any rule

*SireTotals* Definition: Array that contains the maximum number(s) of

offspring that can be selected from each sire

across time steps

Type: Integer

Dimension: 1 x maximum number specified for

*SireTotal* across rules

Options:

≥1 For each element specified in *SireTotal*

0 For each element not specified in *SireTotal*

Default: *SireTotals* must be specified when *SireTotal*≥1

for any rule

*DamTotals* Definition: Array that contains the maximum number(s) of

offspring that can be selected from each

dam across time steps

Type: Integer

Dimension: 1 x maximum number specified for

*DamTotal* across rules

Options:

≥1 For each element specified in *DamTotal*

0 For each element not specified in *DamTotal*

Default: *DamTotals* must be specified when

*DamTotal*≥1 for any rule

*FamilyTotals* Definition: Array that contains the maximum number(s) of

offspring that can be selected from each

full-sib family across time steps

Type: Integer

Dimension: 1 x maximum number specified for

*FamilyTotal* across rules

Options:

≥1 For each element specified in *FamilyTotal*

0 For each element not specified in *FamilyTotal*

Default: *FamilyTotals* must be specified when

*FamilyTotal*≥1 for any rule

*SireMatingsTotals* Definition: Array that contains the maximum number(s) of

matings for each sire across time steps

Type: Integer

Dimension: 1 x maximum number specified for

*SireMatingsTotal* across rules

Options:

≥1 For each element specified in *SireMatingsTotal*

0 For each element not specified in

*SireMatingsTotal*

Default: *SireMatingsTotals* must be specified when

*SireMatingsTotal*≥1 for any rule

*DamMatingsTotals* Definition: Array that contains the maximum number(s) of

matings for each dam across time steps

Type: Integer

Dimension: 1 x maximum number specified for

*DamMatingsTotal* across rules

Options:

≥1 For each element specified in *DamMatingsTotal*

0 For each element not specified in *DamMatingsTotal*

Default: *DamMatingsTotals* must be specified when

*DamMatingsTotal*≥1 for any rule

\*\*Additional information\*\*

1) Selection rules do not apply to animals in the base population

2) reselect

(a) When *reselect* 0 and an insufficient number of selection candidates

fulfil the selection rule(s), the breeding program continues with fewer

selected candidates from the corresponding selection stage

(b) When *reselect* 1 and an insufficient number of selection candidates

fulfil the selection rule(s), the highest-ranking candidates that do not

fulfil the rule(s) are selected until the required number of animals are

selected. If there is still and insufficient number of selection

candidates, the breeding program continues with fewer selected candidates

for the corresponding selection stage.

## **&FAMILYPARAMETERS**

parameters=

*stage criterion nSibs nMales nFemales* /

NB! Number of lines in parameters must equal the number of selection stages with

family selection; number of selection stages with *sex\_code* 9 in namelist &SELECTION,

variable selection\_scheme.

\*\*Task\*\*

Parameters used to carry out selection within selected full-sib families. Namelist

is read when *sex\_code* 9 in any selection stage of namelist &SELECTION, variable

selection\_scheme.

\*\*Properties of names\*\*

*stage* Definition: Stage of selection. Must correspond with a

selection stage with *sex\_code* 9 in namelist

&SELECTION, variable selection\_scheme.

Type: Integer

Options: 1≤*stage*≤*selection\_groups*

Default: *stage* must be specified

*criterion* Definition: Criterion used to carry out selection within

selected full-sib families

Type: Character

Options:

tbv True breeding value

random Selection on a random number, i.e. random

selection

phenoweight Phenotypic selection with observations weighted

polyblup Selection on BLUP-EBV based on polygenic-

relationship matrix

genomicblup Selection on BLUP-EBV based on genomic-

relationship matrix

ibdblup Selection on BLUP-EBV based on IBD-relationship

matrix

ibsblup Selection on BLUP-EBV based on IBS-relationship

matrix

gas Selection on QTL-effect and BLUP-EBV based on

polygenic-relationship matrix

bayesp Selection on BLUP-EBV based on BayesP

Default: *criterion* must be specified

*nSibs* Definition: Number of sibs selected from each full-sib family

Type: Integer

Options:

-9 *nSibs* ignored; no sibs in each full-sib family selected

-1 All sibs in each full-sib family selected

≥1 Number of sibs in each full-sib family selected

Default: *nSibs* must be specified

*nMales* Definition: Number of males selected from each full-sib family

Type: Integer

Options:

-9 *nMales* ignored; all males in each full-sib family

selected

-1 All males in each full-sib family selected

≥1 Number of males in each full-sib family selected

Default: *nMales* must be specified

*nFemales* Definition: Number of females selected from each full-sib

family

Type: Integer

Options:

-9 *nFemales* ignored; all females in each full-sib family

selected

-1 All females in each full-sib family selected

≥1 Number of females in each full-sib family selected

Default: *nFemales* must be specified

## **&EVA**

EvaSelection=

*stage MalesSelected FemalesSelected MaleSelCrit MaleRunBlup MaleDestinySel*

*maleMatingGroup MaleTestHerd MaleDestinyUnsel MaleMaxMatings MaleGermplasm*

*FemaleSelCrit FemaleRunBlup FemaleDestinySel femaleMatingGroup FemaleTestHerd*

*FemaleDestinyUnsel FemaleMaxMatings FemaleGermplasm relationshipMatrix*

*nMalePreSel nFemalePreSel*  *malePreSelUnit femalePreSelUnit firstHerd lastHerd*

*birthGroups nMatings litterSizeAllocation LitterSize* *sexAllocation SexRatio* /

\*\*Task\*\*

Input for EVA-selection. Namelist is read when *sex\_code* 7 in any selection

stage of namelist &SELECTION, variable selection\_scheme.

\*\*Properties of names\*\*

*stage* Definition: Stage of selection; must correspond to

a selection stage in namelist &SELECTION,

variable selection\_scheme with *sex\_code* 7

Type: Integer

Options: 2≤*stage*≤*selection\_groups*

Default: *stage* must be specified

*MalesSelected* Definition: Males EVA-selected

Type: Character

Options:

yes Males are selected

no Males are not selected

Default: *MalesSelected* must be specified

*FemalesSelected*  Definition: Females EVA-selected

Type: Character

Options:

yes Females are selected

no Females are not selected

Default: *FemalesSelected* must be specified.

*MaleSelCrit* Definition: Criterion used to select males

Type: Character

Options:

tbv True breeding value

null No selection criterion

phenoweight Phenotypic selection with observations

weighted

polyblup Polygenic-BLUP breeding value

genomicblup Genomic-BLUP breeding value

ibdblup IBD-BLUP breeding value

~~gas GAS-breeding value~~

bayesp BayesP-breeding values

Default: *MaleSelCrit* must be specified

When bayesP, *geneticModel* must be ‘genomic’, nebv==1, and genotyped\_at\_birth)/='yes'

*MaleRunBlup* Definition: Estimate BLUP breeding values

Type: Integer

Options:

0 Breeding values are not estimated

1 Breeding values are estimated

Default: *MaleRunBlup* must be specified

*MaleDestinySel* Definition: Destiny of selected males

Type: Integer

Options:

0 Selected males are left alive

1 Selected males are reproduced (mated)

2 Selected males have germ plasm stored

3 Selected males are genotyped

4 Selected males are transferred to another

(test) herd; the herd is specified by

*MaleTestHerd*

5 Selected males are allocated to groups

6 Selected males have DYD sampled

8 Selected males have phenotypic observation(s)

realised at current selection stage

12 Selected males are reproduced and have germ

plasm stored

14 Selected males are reproduced after

transfer to another (test) herd; the herd is

specified by *MaleTestHerd*

16 Selected males are reproduced and have DYD

sampled

26 Selected males have germ plasm stored and

DYD sampled

126 Selected males are reproduced, have germ

plasm stored, and DYD sampled

90 Selected males are tagged as candidates to

be left alive

91 Selected males are tagged as candidates to

reproduce (mate)

92 Selected males are tagged as candidates to

have germ plasm stored

93 Selected males are tagged as candidates for

genotyping

94 Selected males are tagged as candidates for

transfer to another (test) herd

955 Selected males are tagged as candidates to be

allocated to groups

96 Selected males are tagged as candidates to

have DYD sampled

98 Selected males are tagged as candidates to have

a phenotypic observation(s) realised at a subsequent

selection stage

99 Selected males are tagged as candidates to be

culled voluntarily

912 Selected males are tagged as candidates to

reproduce and have germ plasm stored

914 Selected males are tagged as candidates to

reproduce after transfer to another (test) herd

916 Selected males are tagged as candidates to

reproduce and have DYD sampled

926 Selected males are tagged as candidates to

have germ plasm stored and DYD sampled

9126 Selected males are tagged as candidates to

reproduce, have germ plasm stored, and DYD

sampled

Default: *MaleDestinySel* must be specified

*maleMatingGroup* Definition: Mating group for males selected for reproduction

Type: Integer

Options: 0 No mating group; males not selected for reproduction

1≤ *maleMatingGroup* ≤*nMatingGroups*, where *nMatingGroups*

is the number of mating groups

Default: *maleMatingGroup* must be specified

*MaleTestHerd* Definition: Test herd to which selected males are

transferred

Type: Integer

Options: 1≤*MaleTestHerd*≤*nherd*

Default: *MaleTestHerd* must be specified

*MaleDestinyUnsel* Definition: Destiny of unselected males

Type: Integer

Options:

0 Males are culled

1 Males are left alive

Default: *MaleDestinyUnsel* must be specified

*MaleMaxMatings* Definition: Maximum number of matings allocated to

selected males; male reproductive capacity

Type: Integer

Options: ≥1

Default: *MaleMaxMatings* must be specified

*MaleGermplasm* Definition: Number of units of germplasm stored for

selected males

Type: Integer

Options:

≥0 Number of units stored for each selected male

-1 Number of units stored for each selected male is set

to the number of matings allocated to each male

Default: *MaleGermplasm* must be specified

*FemaleSelCrit* Definition: Criterion used to select females

Type: Character

Options:

tbv True breeding value

null No selection criterion

phenoweight Phenotypic selection with observations

weighted

polyblup Polygenic-BLUP breeding value

genomicblup Genomic-BLUP breeding value

ibdblup IBD-BLUP breeding value

~~gas GAS-breeding value~~

bayesp BayesP-breeding values

Default: *FemaleSelCrit* must be specified

When bayesP, *geneticModel* must be ‘genomic’, nebv==1, and genotyped\_at\_birth)/='yes'

*FemaleRunBlup* Definition: Estimate BLUP breeding values

Type: Integer

Options:

0 Breeding values are not estimated

1 Breeding values are estimated

Default: *FemaleRunBlup* must be specified

*FemaleDestinySel* Definition: Destiny of selected females

Type: Integer

Options:

0 Selected females are left alive

1 Selected females are reproduced (mated)

2 Selected females have germ plasm stored

3 Selected females are genotyped

4 Selected females are transferred to another

(test) herd; the herd is specified by

*FemaleTestHerd*

5 Selected females are allocated to groups

8 Selected females have phenotypic observation(s)

realised at current selection stage

12 Selected females are reproduced and have germ

plasm stored

14 Selected females are reproduced after

transfer to another (test) herd; the herd is

specified by *FemaleTestHerd*

17 Selected females are reproduced as bull-dams

127 Selected candidates are reproduced as bull-dams

and have germ plasm stored

147 Selected females are reproduced as bull-dams

after transfer to another (test) herd; the herd

is specified by *FemaleTestHerd*

90 Selected females are tagged as candidates to

be left alive

91 Selected females are tagged as candidates to

reproduce (mate)

92 Selected females are tagged as candidates to

have germ plasm stored

93 Selected females are tagged as candidates for

genotyping

94 Selected females are tagged as candidates for

transfer to another (test) herd

955 Selected females are tagged as candidates to be

allocated to groups

98 Selected females are tagged as candidates to have

a phenotypic observation(s) realised at a subsequent

selection stage

99 Selected females are tagged as candidates to be

culled voluntarily

912 Selected females are tagged as candidates to

reproduce and have germ plasm stored

914 Selected females are tagged as candidates to

reproduce after transfer to another (test) herd

917 Selected females are tagged as candidates to

reproduce as bull-dams

9127 Selected females are tagged as candidates to

reproduce as bull-dams and have germ plasm

stored

9147 Selected females are tagged as candidates to

reproduce as bull-dams after transfer to another

(test) herd

Default: *FemaleDestinySel* must be specified

*maleMatingGroup* Definition: Mating group for females selected for reproduction

Type: Integer

Options: 0 No mating group; females not selected for

reproduction

1≤*femaleMatingGroup* ≤*nMatingGroups*, where

*nMatingGroups* is the number of mating groups

Default: *femaleMatingGroup* must be specified

*FemaleTestHerd* Definition: Test herd to which selected females are

transferred

Type: Integer

Options: 1≤*FemaleTestHerd*≤*nherd*

Default: *FemaleTestHerd* must be specified

*FemaleDestinyUnsel* Definition: Destiny of unselected females

Type: Integer

Options:

0 Females are culled

1 Females are left alive

Default: *FemaleDestinyUnsel* must be specified

*FemaleMaxMatings* Definition: Maximum number of matings allocated to

selected females; female reproductive

capacity

Type: Integer

Options: ≥1

Default: *FemaleMaxMatings* must be specified

*FemaleGermplasm* Definition: Number of units of germplasm stored for

selected females

Type: Integer

Options:

≥0 Number of units stored for each selected female

-1 Number of units stored for each selected female is set

to the number of matings allocated to each female

Default: *FemaleGermplasm* must be specified

*relationshipMatrix* Definition: Method used to construct relationship matrix

Type: Character

Options:

pedigree Matrix constructed with pedigree

genomic Matrix constructed with markers and pedigree

ibs Matric constructed with IBS-markers

ibd Matrix constructed with IBD-markers

Default: *RelationshipMatrix* must be specified

The following six parameters modify EVA-input datasets

*nMalePreSel* Definition: Number of male candidates for EVA-selection that

can be allocated matings. Males that cannot be

allocated matings have *MaleMaxMatings* set to zero

in EVA-input datasets.

Type: Integer

Options:

-9 Reduction occurs across herds

≥1 Reduction occurs within herds

Default: *nMalePreSel* must be specified

*nFemalePreSel* Definition: Number of female candidates for EVA-selection that

can be allocated matings. Feales that cannot be

allocated matings have *FemaleMaxMatings* set to zero

in EVA-input datasets.

Type: Integer

Options:

-9 Reduction occurs across herds

≥1 Reduction occurs within herds

Default: *nFemalePreSel* must be specified

*malePreSelUnit* Definition: Selection unit used to reduce the number of male

candidates for EVA-selection that can be allocated

matings. Males that cannot be allocated matings

have *MaleMaxMatings* set to zero in EVA-input

datasets.

Type: Character

Options:

acrossherd Reduction occurs across herds

withinherd Reduction occurs within herds

Default: *malePreSelUnit* must be specified

*femalePreSelUnit* Definition: Selection unit used to reduce the number of female

candidates for EVA-selection that can be allocated

matings. Females that cannot be allocated matings

have *FemaleMaxMatings* set to zero in EVA-input

datasets.

Type: Character

Options:

acrossherd Reduction occurs across herds

withinherd Reduction occurs within herds

Default: *femalePreSelUnit* must be specified

*firstHerd*  Definition: First herd within which animals are included in

input-data set used by EVA

Type: Integer

Options:

*firstHerd*≥1

-9 Herd not considered

-1 All animals from all herds included

Default: *firstHerd* must be specified

*lastHerd* Definition: Last herd within which animals are included in

input-data set used by EVA

Type: Integer

Options:

*firstHerd*≤*lastHerd*≤*nherd*

-9 Herd not considered

-1 All animals from all herds included

Default: *LastHerd* must be specified

*firstHerd* and *lastHerd*: Include individuals - in addition to individuals tagged for EVA-selection - to pedigree file; primarily intended for young unreproductive animals

*birthGroups* Definition: Parameter used to allocate animals into birth

groups when generating EVA-input data. Animals

are allocated to birth groups as

abs(minval(pop(:)%birth))+

floor(real(pop(id)%birth)/

(*birthGroups*+*gestation\_length*))

Type: Integer

Options: 1≤*birthGroups*≤rAgemin, where rAgemin is the minimum

reproductive age of EVA-selection candidates

Default: *birthGroups* must be specified

*nMatings* Definition: Total number of matings

Type: Integer

Options: ≥1

Default: *nMating* must be specified

*litterSizeAllocation*  Definition: Method used to assign number of offspring in each

litter

Type: Character

Options:

fix Fixed number of offspring in each litter; litter

size specified by *LitterSize*

phenotype Number of offspring is phenotype of dam for

observation number provided in *LitterSize*

Default: *litterSizeAllocation* must be specified

*LitterSize* Definition: Number of offspring per mating or observation

number of trait used to sample number of offspring

Type: Integer

Options: ≥1

Default: *LitterSize* must be specified

*sexAllocation* Definition: Method to assign sex to offspring within litters

Type: Character

Options:

fix Fixed number of males and females in each litter

sample Sex of each offspring is sampled

Default: *sexAllocation* must be specified

*SexRatio* Definition: Probability of an offspring being a male

Type: Real

Options: 0.0≤ *SexRatio* ≤1.0

Default: *SexRatio* must be specified

\*\*Relation to subsequent namelists\*\*

&EVAPARAMETERS

Namelist &EVAPARAMETERS is required when EVA-selection. The number of

stages in namelist &EVAPARAMETERS, variable parameters must equal the

number of stages in namelist &EVA, variable *EvaSelection*.

&EVAPHENOWEIGHT

Namelist &EVAPHENOWEIGHTS is required when *MaleSelCrit* and/or *FemaleSelCrit*

‘phenoweight’ in any eva-selection stage of namelist &EVA, variable *EvaSelection*

&GROUPPARAMETERS

Namelist &GROUPPARAMETERS is required when *MaleDestinySel* or *FemaleDestinySel* 55 in

any eva-selection stage of namelist &EVA, variable *EvaSelection*

&IBDBLUPPARAMETERS

&EVAGENOMICRELATIONSHIP

Namelist &EVAGENOMICRELATIONSHIPS is required when *relationshipMatrix* ‘genomic’

in any eva-selection stage of namelist &EVA, variable *EvaSelection*. The number of

stages in namelist &EVAGENOMICRELATIONSHIPS, variable genomicRelationships must

equal the number of stages in namelist &EVA, variable *relationshipMatrix*

‘genomic’.

&EVAIBDRELATIONSHIPPARAMETERS

&BLUPPARAMETERS

Namelist &BLUPPARAMETERS is required when *selection\_criterion* ‘polyblup’,

‘genomicblup’, ‘ibdblup’, ~~or ‘gas’~~ in any eva-selection stage of namelist &EVA,

variable EvaSelection

~~&GAS\_PARAMETERS~~

~~Namelist &GAS\_PARAMETERS is required when~~ *~~geneticModel~~* ~~‘qtl’ in namelist~~

~~&CONTROLPARAMETERS and~~ *~~MaleSelCrit~~* ~~and/or~~ *~~FemaleSelCrit~~* ~~‘gas’ in any EVA-selection~~

~~stage of namelist &EVA, variable EvaSelection~~

&BAYESPPARAMETERS

\*\*Additional information\*\*

1) Only animals tagged for EVA-selection are selection candidates. That is,

animals tagged in selection stages prior to the EVA-selection stage and, if

there is more than one EVA-selection stage, after the previous

EVA-selection stage. Animals are tagged for EVA selection in namelist

selection, variable selection\_scheme.

2) EVA-selection tags are removed from all animals following a stage of EVA

selection

3) All variables in namelist eva, variable EvaSelection are read

and, therefore, must be specified. However, there are cases where some

variables are not used. The most striking case is when *MalesSelected* or

*FemalesSelected* set to ‘no’. Other cases are when selected animals are not

destined to be mated. In these cases, variables, such as *Littersize* and

*SexRatio* are not used. For variables that are not used, the input provided

is irrelevant. However, for clarity, we recommend providing 0 for integer

variables and 0.0 for *SexRatio* (the only real).

4) Variables that are always used are *stage*, *MalesSelected*, F*emalesSelected*,

*FutMaleBreeders*, *FutFemaleBreeders*, ~~nTracedGen,~~ *nMatings*, and *EvaParameterFile*.

The use of all other variables depends on the options specified for these and

the other variables in namelist eva.

5) *stage* must correspond to a selection stage in namelist &SELECTION, variable

selection\_scheme with *sex\_code* 7

6) (a) Either *MalesSelected* or *FemalesSelected*, or both, must be ‘yes’. That is,

EVA selection is not carried out when both *MalesSelected* and

*FemalesSelected* are ‘no’. In this case, the program stops.

(b) (i) When *MalesSelected* ‘yes’, *MaleSelCrit*, *MaleDestinySel*,

*MaleDestinyUnsel,* and *MaleMaxMatings* are used and must be specified

with valid options

(ii) When FemalesSelected ‘yes’, *FemaleSelCrit*, *FemaleDestinySel*,

*FemaleDestinyUnsel,* and *FemaleMaxMatings* are used and must be

specified with valid options

(c) (i) When *MalesSelected* ‘no’, *MaleSelCrit*, *MaleDestinySel*,

*MaleDestinyUnsel*, *MaleMaxMatings*, *MaleRunBlup*, *MaleTestHerd*, and

*MaleGermplasm* are not used. *MaleSelCrit* is set to ‘ ‘ and

*MaleDestinySel*, *MaleDestinyUnsel*, *MaleMaxMatings*, *MaleRunBlup*,

*MaleTestHerd*, and *MaleGermplasm*,are set to 0.

(ii) When *FemalesSelected* ‘no’, *FemaleSelCrit*, *FemaleDestinySel*,

*FemaleDestinyUnsel*, *FemaleMaxMatings*, *FemaleRunBlup*, *FemaleTestHerd*,

*FemaleGermplasm*, *Littersize*, and *SexRatio* are not used. *FemaleSelCrit*

is set to ‘ ‘, *FemaleDestinySel*, *FemaleDestinyUnsel*, *FemaleMaxMatings*,

*FemaleRunBlup*, *FemaleTestHerd*, *FemaleGermplasm*, and *Littersize* are

set to 0, and *SexRatio* is set to 0.0.

7) (a) (i) When *MalesSelected* ‘no’, males are not EVA-selected. Females are EVA-

selected while ignoring the genetic relationships of males.

(ii) When *FemalesSelected* ‘no’, females are not EVA-selected. Males are

EVA-selected while ignoring the genetic relationships of females.

(b) Setting *MalesSelected* or *FemalesSelected* to ‘no’ is different from setting

*MalesSelected* and/or *FemalesSelected* to ‘yes’ and *MaleSelCrit* and/or

*FemaleSelCrit* to ‘null’ (see information point 19)

Information points 8) to 16) assume that *MalesSelected* ‘yes’ and/or

FemalesSelected ‘yes’

phenoweights

8) (a) *MaleSelCrit* and *FemaleSelCrit* must be ‘tbv’, ‘null’, ‘polyblup’,

‘genomicblup’, ‘ibdblup’~~, or ‘gas’~~

(b) When *MaleSelCrit* and/or *FemaleSelCrit* ‘tbv’, economic weight(s) provided in

namelist &MATRICES, variable *economic\_weight* in are used to weight the true

breeding values for each trait

When *MaleSelCrit* and/or *FemaleSelCrit* ‘polyblup’, ‘genomicblup’, ‘ibdblup’,

~~or ‘gas’,~~ economic weight(s) provided in namelist &MATRICES, variabl*e*

*economic\_weight* are used to weight the estimated breeding values for each

trait

(c) When *geneticModel* ‘polygenic’, *selection\_criterion* cannot be

‘genomicblup’, ‘ibdblup’~~, or ‘gas’~~ at any selection stage

(e) When *geneticModel* ‘genomic’, ~~both~~ ‘genomicblup’ ~~and ‘gas’~~ cannot be used as

selection criterion in the same simulated breeding scheme. That is, ~~both~~

‘genomicblup’ ~~and ‘gas’~~ cannot be specified as *selection\_criterion* in

namelist &selection, variable selection\_scheme, and/or as *MaleSelCrit* or

*FemaleSelCrit* in any EVA-selection stage of namelist &EVA, variable

EvaSelection.

When bayesP, *geneticModel* must be ‘genomic’, nebv==1, and genotyped\_at\_birth)/='yes'

9) (a) *MaleRunBlup* must be 0:1 when *MaleSelCrit* ‘polyblup’, ‘genomicblup’, ‘ibdblup’,

~~or ‘gas’~~; *FemaleRunBlup* must be 0:1 when *FemaleSelCrit* ‘polyblup’,

‘genomicblup’, ‘ibdblup’~~, or ‘gas’~~

(b) *MaleRunBlup* is not used when *MaleSelCrit* ‘tbv’ or‘null’; *FemaleRunBlup* is

not used when *FemaleSelCrit* ‘tbv’ or ‘null’. They are set to 0.

~~(c)~~ *~~MaleRunBlup~~* ~~and/or~~ *~~FemaleRunBlup~~* ~~must be 1 if the EVA-selection stage~~

~~is the first selection stage where~~ *~~selection\_criterion~~* ~~‘polyblup’,~~

~~‘genomicblup’, or ‘gas’. The first selection stage(s) can be a~~

~~truncation-selection stage(s).~~

(d) When *MaleRunBlup* and/or *FemaleRunBlup* 1 and *selection\_criterion* ‘polyblup’,

‘ibdblup’, ~~or ‘gas’, and selection stage not the first selection~~

~~stage where~~ *~~selection\_criterion~~* ~~‘polyblup’, ‘genomicblup’, or ‘gas’~~,

BLUP-breeding values are only estimated if a phenotypic observation(s) has

been realised. If BLUP-breeding values have been estimated previously, they

are only re-estimated when phenotypic observations have been realised since

the previous estimation.

(e) When *MaleRunBlup* and/or *FemaleRunBlup* 1 and *selection\_criterion* ‘genomicblup’,

BLUP-breeding values are only estimated if a phenotypic observation(s) has

been realised and animals have been genotyped. If genomic-BLUP breeding values

have been estimated previously, they are only re-estimated if phenotypic

observations have been realised or additional animals have been genotyped

since the previous estimation.

• (a) When males that are not alive can be amongst those males tagged for EVA

selection (i.e., *live\_status* 0 or 2 in namelist &SELECTION, variable

selection\_scheme),

(i) *MaleDestinySel* must be 1 or 91,

(ii) When *MaleDestinySel* 1, and a deceased animal with stored germ plasm is

selected for reproduction, the animal’s reproductive capacity

is defined as follows. If the number of stored germ plasm is

greater or equal to *MaleMaxMatings*, the animal’s reproductive

capacity is *MaleMaxMatings*. If, however, the number of stored germ

plasm is less than *MaleMaxMatings*, the animal’s reproductive

capacity is the number of stored germ plasm. The consequence of

this is that the total number of male matings available may be less

than *nMatings*.

(b) 10)(a) applies to females

• Females tagged for EVA selection can be either normal females or bull dams.

That is, both normal females (*sex\_code* 2 or 6 in namelist &SELECTION,

variable *selection\_scheme*) and bull dams (*sex\_code* 4) cannot be selected

for EVA selection.

• (a) *~~MaleDestinySel~~* ~~or~~ *~~FemaleDestinySel~~* ~~cannot be 3 (genotyping) at any EVA-~~

~~selection stage when~~ *~~MaleSelCrit~~* ~~or~~ *~~FemaleSelCrit~~* ~~‘gas’ at any EVA-~~

~~selection stage or~~ *~~selection\_criterion~~* ~~‘gas’ at any selection stage of~~

~~namelist &SELECTION, variable selection\_scheme.~~

(b) *MaleDestinySel* can only be 4, 14, 94, and 914 [transfer to another (test)

herd] when *nherd*>1; *FemaleDestinySel* can only be 4, 14, 147, 94, 914, and

9147

(c) Animals can be tagged with a view to selection with in *sex\_code* 5 or 6

(namelist &SELECTION, variable selection\_scheme) at a subsequent selection

stage during the same time step. That is, *MaleDestinySel* can be 90, 91, 92,

93, 94, 96, 98, 99, 912, 914, 916, 926, and 9126; *FemaleDestinySel* can be

91, 92, 93, 94, 98, 99, 912, 914, 917, 9127, and 9147.

(d) Animals cannot be tagged for EVA selection. That is, *MaleDestinySel* and

*FemaleDestinySel* cannot be 95.

(e) Males cannot be bull-dams. That is, *destiny\_sel* cannot be 17, 127, 147,

917, 9127, or 9147 when *sex\_code* 0, 1, 3, or 5.

(f) Females cannot have DYD sampled. That is, *FemaleDestinySel* cannot be 6, 16,

26, 126, 96, 916, 926, or 9126.

(j) There is only a limited number of valid destinies for selected bull

dams. They are *FemaleDestinySel* 0, 1, 2, 3, 8, and 12.

(o) Animals can only be allocated to one group. Candidates already allocated a

group with *MaleDestinySel* or *FemaleDestinySel* 55 or *destiny\_sel* 55 in in any

selection stage of namelist &SELECTION, variable selection\_scheme are not

allocated to a new group if selected to be allocated to a group in subsequent

selection stages or time steps.

(k) Non-reproductive males cannot be among the candidates for selection when

*MaleDestinySel* includes reproduction. That is, non-reproductive males are

not candidates when *MaleDestinySel* 1, 12, 14, 16, 126, 91, 912, 914, 916,

9126. Non-reproductive females cannot be among the candidates for selection

when *FemaleDestinySel* includes reproduction. They are not candidates when

*FemaleDestinySel* 1, 12, 14, 17, 127, 147, 91, 912, 914, 917, 9127, or 9147.

(f) Non-reproductive animals are not candidates for selection when destiny for

selection includes reproduction. That is, non-reproductive males are not

candidates when *MaleDestinySel* 1, 12, 14, 16, 126, 91, 912, 914, 916, 9126.

Non-reproductive females are not candidates when *FemaleDestinySel* 1, 12, 14,

17, 127, 147, 91, 912, 914, 917, 9127, or 9147.

(g) Non-reproductive animals are not candidates for EVA-selection when

destiny for selection in the subsequent EVA-selection stage includes

reproduction. That is, non-reproductive males are not candidates when

*MaleDestinySel* in namelist &EVA, variable EvaSelection 1, 12, 14, 16,

126, 91, 912, 914, 916, 9126. Non-reproductive females are not

candidates when FeMaleDestinySel in namelist &EVA, variable

EvaSelection 1, 12, 14, 17, 127, 147, 91, 912, 914, 917, 9127, or 9147.

• (a) *maleMatingGroup* and/or *femaleMatingGroup* only used when *MaleDestinySel* and/or

*FemaleDestinySel* includes reproduction; that is *MaleDestinySel* 1, 12, 14, 16,

126 and/or *FemaleDestinySel* 1, 12, 14, 17, 127, or 147. When *maleMatingGroup*

and *femaleMatingGroup* are not used, they are set to 0.

(b) Numbering of mating groups must go from 1 … *nMatingGroups*, where *nMatingGroups*

is the number of mating groups.

(c) Animals can also be allocated to mating groups in namelist &SELECTION, variable

*selection\_scheme*.

(d) Mating groups are allocated within time steps. Mating groups with the same

mating-group number across times steps are different mating groups.

• (a) *MaleTestHerd* must be specified when *MaleDestinySel* includes transfer to

another (test) herd. That is, *MaleDestinySel* 4 or 14.

(b) *FemaleTestHerd* must be specified when *FemaleDestinySel* includes transfer

to another (test) herd. That is, *FemaleDestinySel* 4, 14, or 147.

• (a) MaleMaxMatings>0

(b) FemaleMaxMatings>0

• (a) *MaleGermplasm*>0 or -1 when *MaleDestinySel* 2, 12, 26, or 126

(b) *FemaleGermplasm*>0 or -1 when *FemaleDestinySel* 2, 12, or 127

(c) If an animal is reselected to have germ plasm stored within a time step

or during a subsequent time step, the new number of germ plasm is added

to the number currently stored for the animal. The exception is males

and females in the base population.

(d) When animals that are not alive can be amongst those tagged for EVA

selection (i.e., *live\_status* 0 or 2 in namelist &SELECTION, variable

selection\_scheme), *destiny\_sel* 1, and deceased animals with stored

germ plasm are selected for reproduction, the number of stored germ plasm

for each selected animal is reduced by the number of germ plasm used for

reproduction. The number of of germ plasm used for reproduction is

the number of matings allocated to each male or female.

• *LitterSize>0* and 0.0≤*SexRatio*≤1.0 when the destiny for selected females

includes reproduction. That is, *FemaleDestinySel* 1, 2, 14, 17, 127, or 147.

• (a) *relationshipMatrix* can be ‘pedigree’, ‘genomic’, ‘ibs’, or ‘ibd’ when

*geneticModel* ‘genomic’ in namelist &CONTROLPARAMETERS. It can only be

‘pedigree’ when *geneticModel* ‘polygenic’. Genotyping is considered with

*relationshipMatrix* ‘genomic’; it is ignored with ‘ibs’ and ‘ibd’. It is ignored

with ‘ibd’ because this is a theoretical situation; it is ignored with ‘ibs’

because we have not implemented a way to build relationship matrices for a

mixture of genotyped and non-genotyped animals.

(b) If *relationshipMatrix* ‘genomic’ and no animals have been genotyped,

*relationshipMatrix* ‘pedigree’ is used. *Any other problems will be detected when*

*constructing genomic-relationship matrices.*

• *FutMaleBreeders* and *FutFemaleBreeders* must be ‘none’, ‘herd’, or ‘population’

19) *~~nTracedGen~~* ~~is applied to increase the speed of EVA-calculations. It reduces the~~

~~size of the relationship matrix, the main restricter of speed.~~

• (a) *birthGroups* can determine the speed of EVA-calculations. The speed is

quickest when *birthGroups* is equal to the minimum reproductive age of

the candidates for EVA-selection.

(b) If *birthGroups* is greater than the minimum reproductive age, no input error

or warning is given, but the program will crash during EVA-calculations.

The reason being that some offpring will be allocated to the same birth

group as their parents.

(c) *birthGroups* can be smaller than the minimum reproductive age of the

candidates for EVA-selection. The only drawback is that this may slow the

speed of the EVA-calculations.

• Setting *MalesSelected* or *FemalesSelected* to ‘no’ is different from setting

*MalesSelected* and/or *FemalesSelected* to ‘yes’ and *MaleSelCrit* and/or

*FemaleSelCrit* to ‘null’

(a) (i) When *MalesSelected* ‘no’, males are not selected. Females are EVA-

selected without considering the genetic relationships of males.

(ii) When *FemalesSelected* ‘no’, females are not selected. Males are EVA-

selected without considering the genetic relationships of females.

(b) (i) When *MalesSelected* ‘yes’ and *MaleSelCrit* ‘null’, males are EVA-

selected. The selection criterion of the males is set to 0.0 and the

genetic relationships of the males is considered in the selection of

the males and, possibly, the females.

(ii) When *FemalesSelected* ‘yes’ and *FemaleSelCrit* ‘null’, females are EVA-

selected. The selection criterion of the females is set to 0.0 and the

genetic relationships of the females is considered in the selection of

the females and, possibly, the males.

A description of the program EVA is provided in Appendix I

## **&EVAPARAMETERS**

parameters=

*stage* *optimise wMerit* *wRelationship dFconstraint limitMaleMatings*

*nSelectedMales wMales nGenerations*

*nGenNoImprovement* *popSize nOffspring restartInterval exchangeAlgorithm*

*mutateProb crossoverProb directedMutateProb* *seed* /

\*\*Task\*\*

Input to run EVA, including parameters for the evolutionary algorithm

\*\*Properties of names\*\*

*stage* Definition: Stage of selection. Must correspond to a

selection stage in *select\_scheme* with *sex\_code* 7

Type: Integer

Options: 2≤*stage*≤*selection\_groups*

Default: *stage* must be specified

*optimise* Definition:

Type: Character

Options:

penalty Penalty applied to average relationship

constraint Rate of inbreeding constrained to specified rate

merit Selection on merit

Default: *optimise* must be specified

‘penalty’ uses *wMerit* and *wRelationship*. ‘constraint’ uses *dFconstraint. ‘merit’ uses neither wMerit*, *wRelationship*, or *dFconstraint*; it sets *wMerit* 1 and *wRelationship* 0.

*wMerit* Definition: Weight applied to genetic merit; estimated breeding

value or index

Type: Real

Options: Any real; negative values impose a penalty on

genetic merit

Default: *wMerit* must be specified

Applied when *optimise* ‘penalty’

*wRelationship* Definition: Weight applied to the average additive-genetic

relationship of the current generation, which

includes the new cohort

Type: Real

Options: Any real; negative values impose a penalty on

additive-genetic relationship

Default: *wRelationship* must be specified

Applied when *optimise* ‘penalty’

*dFconstraint*  Definition: Pre-defined rate of inbreeding

Type: Real

Options: 0.0≤*dFconstraint*≤1.0

Default: *dFconstraint* must be specified

Applied when *optimise* ‘constraint’

*limitMaleMatings* Definition: Multiple by which matings are allocated to male

selection candidates. For example, when

*limitMaleMatings* 5, matings are allocated in

multiples of 5 and males can only be allocated

0, 5, 10, 15, … *MaleMaxMatingsi* matings, where

*MaleMaxMatingsi* is the maximum number of matings

that can be allocated to the *i*th male.

Type: Integer

Options: *limitMaleMatings*≥1

Default: *limitMaleMatings* must be specified

*nSelectedMales* Definition: Number of males selected (allocated matings)

Type: Integer

Options: *nSelectedMales*≥0

0 Optimal number of males selected

≥0 Number of males selected restricted to *nSelectedMales*

Default: *nSelectedMales* must be specified

*wMales* Definition: Weight applied to selecting males; negative value

specifies a penalty applied to selecting more males

Type: Real

Options: Any real

Default: *wMales* must be specified

The following variables control the evolutionary algorithm

*nGenerations* Definition: Maximim number of generations run by the

evolutionary algorithm

Type: Integer

Options: *nGenerations*≥1

Default: nGenerations must be specified

*nGenNoImprovement* Definition: Number of generations of the evolutionary algorithm

that are run without generating an improved

solution before EVA stops

Type: Integer

Options: *nGenNoImprovement*≥1

Default: *nGenNoImprovement* must be specified

*popSize* Definition: Size of population of solutions maintained by

the evolutionary algorithm

Type: Integer

Options: *popSize*≥1

Default: *popSize* must be specified

*nOffspring* Definition: Number of new solutions generated in each

generation of the evolutionary algorithm

Type: Integer

Options: *nOffspring*≥1

Default: *nOffspring* must be specified

*restartInterval* Definition: Number of generations after which more variation

is generated by the evolutionary algorithm. More

variation is generated by increasing the

mutation variance in a single generation.

Increasing the variance is only applied when a

better solution is not been found for

*restartInterval* generations. Better solutions

are those that increase the optimisation

criterion, which is a function of genetic merit,

average additive-genetic relationship, *wMerit*,

and *wRelationship*.

Type: Integer

Options: *restartInterval*≥1

Default: *restartInterval* must be specified

*exchangeAlgorithm* Definition: Number of generations after which an exchange

algorithm is used to iteratively optimise

solutions generated by the evolutionary

algorithm

Type: Integer

Options: *exchangeAlgorithm*≥1

Default: *exchangeAlgorithm* must be specified

*mutateProb*  Definition: Probability of mutating an (each)individual in a

solution generated by the evolutionary

algorithm by randomly exchanging the

individual for another individual (in the

solution or resampling?). Suggested value

1/(4·Nmatings).

Type: Real

Options: 0.0≤*mutateProb*≤1.0

Default: *mutateProb* must be specified

*crossoverProb*  Definition: Probability of crossovers when the evolutionary

algorithm generates new solutions from two

parental solutions. Suggested value between

1/nMatings: 1/(2·nMatings).

Type: Real

Options: 0.0≤*crossoverProb*≤1.0

Default: *crossoverProb* must be specified

*directedMutateProb*  Definition: Probability of mutating an individual in a

solution generated by the evolutionary algorithm

by exchanging the individual for an individual that increases the evaluation of the solution. Suggested value 1/(2\*nMatings).

Type: Real

Options: 0.0≤*directedMutateProb*≤1.0

Default: *directedMutateProb* must be specified

*seed* Definition: Seed used by random-number generator in EVA

Type: Integer

Options: 0 Seed sampled by computer clock

>0 Specified seed

Default: *seed* must be specified

\*\*Additional information\*\*

A description of the program EVA is provided in the Appendix

## **&SELECTIONTOOLEXTRA**

SelectUnitRuleParm=

*stage selection\_unit nUnitSelect CriterionBetweenUnit minSizeselected*

*maxSizeselected CriterionWithinUnit*

TimeSeqBy=*TimeSeqBy*

TimeSkipBy=*TimeSkipBy*

ApplyQtlSelection=*ApplyQtlSelection*

selectQtlRules=*selectQtlRules*

/

\*\*Task\*\*

Extra tools and rules applied for selection\_scheme. The tools added here are usually optional inputs for selection\_scheme. These tools are not directly added to selection\_scheme because this could advoid changes in input formats for different versions of adam.

SelectUnitRuleParm: Selection is applied within or between selection units when truncation selection has *select\_scheme*%otherSelectionLayer=1. All animals must pass conditions set by the corresponding *select\_scheme* first, before going to this selection layer.

TimeSeqBy & TimeSkipBy: specifies which selection stages are used or skiped at each time step.

\*\*Properties of names\*\*

*stage* Definition: Stage of selection. Must correspond to a

selection stage in *select\_scheme*

Type: Integer

Options: 1≤*stage*≤*selection\_groups*

Default: *stage* must be specified

*selection\_unit* Definition: Unit of selection

Type: Character

Default: *selection\_unit* must be specified

Options: "population" : Population of animals

"herd" : herd

"family"

'fullfamily'

'familyorigins'

'grandparent'

'group'

*nUnitSelect* Definition: Number of units to be selected

Type: Integer

Options: >0

Default: *nUnitSelect* must be specified

*CriterionBetweenUnit* Definition: selection Criterion used for between units

Type: Character

Default: *CriterionBetweenUnit* must be specified

Options: "asselectscheme": same as the Criterion provided in

The corresponding selection scheme

"random" : random selection.

*minSizeselected* Definition: minimum number of individuals within each unit are

selected

Type: Integer

Options: >0

Default: *minSizeselected* must be specified

*maxSizeselected* Definition: maximum number of individuals within each unit are

selected

Type: Integer

Options: >0

Default: *maxSizeselected* must be specified

*CriterionWithinUnit* Definition: selection Criterion used for between units

Type: Character

Default: *CriterionWithinUnit* must be specified

Options: "asselectscheme": same as the Criterion provided in

the corresponding selection scheme

"random" : random selection.

*TimeSeqBy* Definition: By default (By=1) the time step sequence from First time step to last time step for selection\_scheme at stage are used: Eg. 1,2,3...ntime. By=n the time step sequence from First time step to last time step for selection\_scheme at stage are used i,i+1n,i+2n,i+3n,...ntime.

Type: Integer

Options: <2 (if n<2, n=1)

>1 (if n>1, n=n)

dimension: selection\_groups.

Default: *optional input.*

*TimeSkipBy* Definition: By default (no skip < 2) the time step sequence from First time step to last time step for selection\_scheme at stage are used: Eg. 1,2,3...ntime. By=n the time step sequence skipped from First time step to last time step for selection\_scheme at stage are i+1n,i+2n,i+3n,...ntime.

Type: Integer

Options: <2 (if n<2, n=0)

>1 (if n>1, n=n)

dimension: selection\_groups.

Default: *optional input.*

*ApplyQtlSelection* Definition: apply qtl selection at stages using rules for qtlith.

Type: Integer

Options: 0…nRules

dimension: selection\_groups.

Default: *optional input.*

*selectQtlRules* Definition: apply qtl selection for qtlith using ith rule.

! Parameters for selection of SNP rules. SNP will be randomly selected.

! If select for QTL, QTL with the highest effect for a trait is selected.

SnpSelith: (interger) snp selected th

! >=1 : higher rank, defined first. higher rank, higher effects.

! once SNP defined, it will not be redefined. Different rules could have same SnpSelith.

*Qtl:*  (character): options: qtl ,or marker

*AlleleFreqRange(1):*  (real): options: 1>= *AlleleFreqRange(1)*>=0

! First element : minimum frequency of minor allele considered(>=0)

*AlleleFreqRange(2):*  (real): options: 1>= *AlleleFreqRange(2)*>= *AlleleFreqRange(1)*

! Second element: maximum frequency of minor allele considered(<=1.0)

nAlleleCopy(1): (interger) ! 0,1,2..ploidy

! First element : first number of allele copy considered (>=0)

nAlleleCopy(2): (interger) ! 0,1,2..ploidy

! Second element: Last number of allele copy considered (>=0)

selectfortrait: (interger) ! 0 <= selectfortrait <= ntbv. If SNP is marker, selectfortrait=0.

signfortrait ! -1,0,1. If SNP is marker, signfortrait=0.

## **&PHENOWEIGHTS**

nPhenotypicWeights=*nPhenotypicWeights*

phenotypicWeights=*stage sex observation weight* /

NB! The number of lines specified in phenotypicWeights must equal *nPhenotypicWeights*

\*\*Task\*\*

Specify weights applied to each observation when aggregate phenotype used as

selection criterion

\*\*Properties of names\*\*

*nPhenotypicWeights* Definition: Number of stage-observation combinations specified

in phenotypicWeights

Type: Integer

Options: 1≤*nPhenotypicWeights*≤(nP+nm+nf)\*(*nobs*+*nCombinedObs*),

where nP is the number of selection stages with

*sex\_code* 0:6, *selection\_method* ‘truncation’, and

*selection\_criterion* ’phenoweight’; nm is the number of

EVA-selection stages with *MaleSelCrit* ’phenoweight’;

and nf is the number of EVA-selection stages with

*FemaleSelCrit* ’phenoweight’.

Default: 0

*stage* Definition: Stage of selection at which phenotypic

selection is applied. Must correspond to a

selection stage in *select\_scheme*.

Type: Integer

Options: 1≤*stage*≤*selection\_groups*

Default: *stage* must be specified

*sex* Definition: Sex of selection candidates to which phenotypic

selection is applied

Type: Integer

Options:

0 Sex ignored, weights applied to both sexes

1 Males

2 Females

Default: *sex* must be specified

*observation*  Definition: Observation number

Type: Integer

Options: 1≤*observation*≤*obs*

Default: *observation* must be specified

*weight* Definition: Weight applied to observation

Type: Real

Options: Any real value

Default: *weight* must be specified

\*\*Additional information\*\*

1) Namelist &PHENOWEIGHTS must be provided when (i) *selection\_criterion* ’phenoweight’

at any selection stage of namelist &SELECTION, variable *selection\_scheme* with

*sex\_code* 0:6, or (ii) *MaleSelCrit* ’phenoweight’ or *FemaleSelCrit* ’phenoweight’

at any EVA-selection stage of namelist &EVA, variableEvaSelection

2) At least one *weight* must be provided for each selection stage with phenotypic

selection

3) Each combination of *stage*, *sex*, and *observation* can only be provided once in

phenotypicWeights

4) *sex* must be 0 for all *stages* corresponding to selection stages of namelist

&SELECTION, variable *selection\_scheme* with *sex\_code* 0:6 and *selection\_criterion*

’phenoweight’

5) *weight* is set to 0.0 for all observations not specified in phenotypicWeights

6) Observations with *weight* 0.0 do not contribute to the aggregate phenotype used as

the selection criterion

7) Animals must have realised all observations allocated a *weight* to be candidates

for phenotypic selection at selection stage *stage*. That is, if an animal has not

had an observation realised, and the observation has *weight*≠0.0 at selection stage

*stage*, the animal is not considered a candidate for phenotypic selection.

## **&GROUPPARAMETERS**

parameters=*stage* *minGroupSize* *maxGroupSize* *interval* *criterion*

*nSibsPerSubfamily* /

NB! Number of lines in parameters must equal the number of selection stages, where

selected candidates are allocated to groups

\*\*Task\*\*

Input to allocate animals to groups

\*\*Properties of names\*\*

*stage* Definition: Stage of selection. Must correspond with selection

stage where (i) *destiny\_sel* 5 in namelist

&SELECTION, variable *selection\_scheme*, or (ii)

*MaleDestinySel* and/or *FemaleDestinySel* 5 in

namelist &EVA, variable EvaSelection.

Type: Integer

Options: 1≤*stage*≤*selection\_groups*

Default: *stage* must be specified

*minGroupSize* Definition: Minimum group size

Type: Integer

Options: *minGroupSize*≥2

Default: *minGroupSize* must be specified

*maxGroupSize* Definition: Maximum group size

Type: Integer

Options: *maxGroupSize*≥*minGroupSize*

Default: *maxGroupSize* must be specified

*interval* Definition: Interval between *minGroupSize* and *maxGroupSize*;

used to define group sizes: *minGroupSize*,

*minGroupSize*+i\**interval* ... *maxGroupSize*; *i*=1,2...

Type: Integer

Options: *interval*>1

Default: *interval* must be specified

*criterion* Definition: Criterion used to allocate candidates to groups

Type: Character

Options:

random Candidates allocated randomly

pseudorandom As for random except that all animals in each

group cannot come from the same full-sib family

family Animals allocated to groups as sub-families

with *nSibsPerSubfamily* full-sibs in each sub-

family

pseudofamily As for family except that only one sub-family

from each full-sib family can be allocated to a

group

Default: *criterion* must be specified

*nSibsPerSubfamily* Definition: Number of full-sibs in each sub-family

Type: Integer

Options:

-9 *nSibsPerSubfamily* not considered

≥1 Number of full-fibs in each family

Default: *nSibsPerSubfamily* must be specified

\*\*Additional information\*\*

• When *minGroupSize*=*maxGroupSize*, the number of animals selected to be grouped must

be a multiple of *maxGroupSize*. This number excludes selected animals that have

already been allocated a group.

• When *minGroupSize*≠*maxGroupSize*, group sizes are sampled from a random-uniform

distribution on the interval [*minGroupSize*, *maxGroupSize*]. If the sum of groups

sizes is not equal (greater than) to the number of animals selected to be grouped,

groups sizes are resampled. This number of animals selected to be grouped excludes

selected animals that have already been allocated a group.

• Difference between *maxGroupSize* and *minGroupSize* must be a multiple of *interval*.

• When *criterion* ‘random’ or ‘pseudorandom’, *nSibsPerSubfamily* must be -9. When

*‘*family’ or ‘pseudofamily’, *nSibsPerSubfamily* must be ≥1.

• When *criterion* *‘*family’ or ‘pseudofamily’, all group sizes must be a multiple of

*nSibsPerSubfamily*.

## **&GENOMICBLUPPARAMETERS**

parameters=*stage centre scale weight firstChrom lastChrom*  *loci scaleGToA*

*propAToG addDiagG diagGOne unrelateZero includeGenotyped incGenoPop*

*incGenoTS(1) incGenoTS(2) previousGenoTS parentsgenokept dmuGrel*

centre=*stage genomicBase firstPop lastPop firstHerd lastHerd*

*firstGeneration lastGeneration maf mafInclude*

scale= *stage genomicBase firstPop lastPop firstHerd lastHerd*

*firstGeneration lastGeneration maf mafInclude*

weight= *stage genomicBase firstPop lastPop firstHerd lastHerd*

*firstGeneration lastGeneration maf mafInclude*

scaleGToA= *stage genomicBase firstPop lastPop firstHerd lastHerd*

*firstGeneration lastGeneration*

singleGeneParameters= *stage genomicBase firstPop lastPop firstHerd lastHerd*

*firstGeneration lastGeneration maf mafInclude*

IncludePopHerdGenoData= *stage* (npop\*nherd)

checkDetGmatrix=*checkDetGmatrix*

/

NB! Number of lines in parameters must equal the number of selection stages using

genomic-breeding value as selection criterion. *stage* in parameters must have a

corresponding stage in centre, scale, and weight when *centre* ‘yes’, *scale* not

‘no’, and *weight* not ‘one’.

\*\*Task\*\*

Input to construct genomic-**G** matrices using Guosheng’s program, invhmatrix. Genomic-**G**

matrices are used by DMU to predict genomic-breeding values.

\*\*Properties of names\*\*

*stage* Definition: Stage of selection. Must correspond with a

selection stage that uses (i) *selection\_criterion*

‘genomicblup’, and *RunBlup* 1 in namelist

&SELECTION, variable *selection\_scheme*, or (ii)

*MaleSelCrit* and/or *FemaleSelCrit* ‘genomicblup’ and

*MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist

&EVA, variable EvaSelection.

Type: Integer

Options: 1≤*stage*≤*selection\_groups*

Default: *stage* must be specified

*centre* Definition: Loci centred

Type: Character

Options:

no Loci centred with **p**=0.5; equivalent to SNP genotypes

-1 0 1 without centering

yes Loci centred; **M**-2**p**

Default: *centre* must be specified

*scale* Definition: Method to scale G-matrix

Type: Character

Options:

no G-matrix divided by sum(2pq) with p=q=0.5

sum2pq G-matrix divided by sum(2pq); 2pq summed over

loci

sqrt2pq Genotypes at each locus divided by sqrt(2pq)

Default: *scale* must be specified

**ADAM**: The genomic-relationship matrix was constructed as **G** = **ZDZ’**, where **D** is a diagonal matrix with diagonal elements **D***ii* =  (*scale* ‘sum2pq’ in variable *parameters*, namelist **&GenomicBlupParameters**) or **D***ii* = √2p*i*(1-p*i*) (*scale* ‘sqrt2pq’) **D** = 

*weight* Definition: Weight applied to each locus

Type: Character

Options:

one Weight 1.0 applied to each locus; each locus has

equal weight

goddard Weight applied to each locus as a function of allele

frequency, 1.0/(p*i*(1-p*i*)), where p*i* is the minimum-

allele frequency at locus *i* (after Goddard 2009)

jannink Weight applied to each locus as a function of allele

frequency, (arcsin(1)-arcsin(sqrt(p*i*)))/(p*i* (1-p*i*))

(after Jannink 2010)

Default: *locusWeight* must be specified

*firstChrom* Definition: First chromosome used to construct genomic-

relationship matrices

Type: Integer

Options: 1≤*firstChrom*≤*nchrom*

Default: *firstChrom* must be specified

*lastChrom* Definition: Last chromosome with loci used to construct

genomic-relationship matrices

Type: Integer

Options: *firstChrom*≤*lastChrom*≤*nchrom*

Default: *lastChrom* must be specified

Loci on chromosomes *firstChrom*:*lastChrom* are used to construct matrices

*loci* Definition: Loci used as genetic markers

Type: Character

Options:

all QTL and markers

qtl QTL

markers Markers

Default: *loci* must be specified

*scaleGToA* Definition: Adjust G to same scale as A

Type: Character

Options:

no G-matrix not scaled to A-matrix

all G-matrix scaled with α and β estimated using

all animals in the population

~~genotyped G-matrix scaled with α and β estimated using~~

~~all genotyped animals in the population~~

base G-matrix scaled with α and β estimated using

all base animals in the population

~~genotypedbase G-matrix scaled with α and β estimated using~~

~~genotyped animals in the base population~~

Default: *scaleGToA* must be specified

all and base with ‘prediction’, ‘eva’, and ‘eva relationships’; all, genotyped, base, genotypedbase with ‘genomic-inbreeding’

genotyped and genotypedbase are redundant with ‘prediction’, ‘eva’, and ‘eva relationships’ because only animals included in *marker file* can be animals used to estimate α and β. Animals included in the *marker file* are traced and genotyped.

‘traced’ options in *genomicBase* are not available for *scaleGToA* as they are also redundant. With ‘prediction’, ‘eva’, and ‘eva relationships’, animals included in *marker file* are already traced. With ‘genomic-inbreeding’, all animals are by definition traced (i.e., included in the pedigree) because we calculate genomic-inbreeding coefficients for all animals in the population.

*propAToG* Definition: Weighting applied to A-matrix (relative to G-

matrix) to construct H-matrix

Type: Real

Options: 0.0≤*propAToG*≤1.0

Default: *percentAToG* must be specified

*addDiagG*  Definition: Value added to diagonal elements of G-matrix *<to*

*make the matrix positive-definite>*

Type: Real

Options: *addDiagG*≥0.0 *[Set to 0.0 if A-matrix used]*

Default: *addDiagG* must be specified

*diagGOne* Definition: Scale mean of diagonal elements of G-matrix so

the mean of the diagonal equals *diagGOne*

Type: Real

Options: *diagGOne*≥0.0, where 0.0 does not bring about scaling

Default: *diagGOne* must be specified

*unrelateZero* Definition: Off-diagonal elements in G-matrix are set to zero

for individuals that are unrelated according to pedigree

Type: Character

Options:

no

yes

Default: *unrelateZero* must be specified

*includeGenotyped* Definition: All genotyped animals – including those that are

not phenotyped or not ancestors of phenotyped

animals – are included in genomic-relationship matrix.

Type: Character

Options:

no

yes

Default: *includeGenotyped* must be specified

*incGenoPop* Definition: whether all genotyped individauls or parts

(in certain herds and populations) are included

in BLUP prediction for EBV.

Type: Character

Options:

all : all genotyped animals

part : parts of genotyped animals. Specified in

IncludePopHerdGenoData

Default: *incGenoPop* must be specified

*incGenoTS(1)* Definition: whether all genotyped individauls or parts

(from certain timesteps) are included in BLUP prediction for EBV.

Type: integer

Options: -9 : includes genotyped animals from all TS

*0=< incGenoTS(1) =< incGenoTS(2)<= ntime*

Default: *incGenoPop* must be specified

*incGenoTS(2)* Definition: whether all genotyped individauls or parts

(to certain timesteps) are included in BLUP prediction for EBV.

Type: integer

Options: -9 : includes genotyped animals from all TS

*0=< incGenoTS(1) =< incGenoTS(2)<= ntime*

Default: *incGenoPop* must be specified

*previousGenoTS* Definition: includes genotype data from (current\_time -previousGenoTS)

to current\_time for prediction of EBV.

Type: integer

Options: -9 : includes genotyped animals from all TS

0< previousGenoTS <= ntime

Default: *previousGenoTS* must be specified

*parentsgenokept* Definition:if previousGenoTS>=1,genotyped parents are usedto calculate

Gmatrix even when they are born prior to current\_time-previousGenoTS.

Type: integer

Options: -9 : not applicable

1 kept (if previousGenoTS>=1).

0 not kept (if previousGenoTS>=1).

Default: *previousGenoTS* must be specified

*dmuGrel* Definition: Matrix to be calculated by GS program for DMU input.

Example of using: GBLUP only, not HBLUP;

enabling to use DMU4 11 30 0 0 used to handle dense matrix

Type: Character

Options:

gmat : Only Gmatrix is created

ginv : Only inversed Gmatrix is created

gboth : Both matrix created

Default: *dmuGrel* must be specified

*genomicBase* Definition: Define animals in genomic-base population that are

used to calculate marker-allele frequencies, p and

1-p. Allele frequencies are used to ’centre’ and

’scale’ genotypes at loci used as genetic markers

in prediction.

Type: Character

Options:

all All animals in the population

traced All traced animals in the population

genotyped All genotyped animals in the population

base All base animals in the population

tracedgenotyped Traced and genotyped animals in the

population

tracedbase Traced animals in the base population

genotypedbase Genotyped animals in the base population

tracedgenotypedbase Traced and genotyped animals in the base

population

Default: *genomicBase* must be specified

all, traced, base, and tracedbase ignore genotyping; hypothetical choose of genomic base

all, genotyped, base, and genotypedbase ignore pedigree tracing; animals not in the pedigree can be part of the genomic base

genomicbase belongs to *firstPop:lastPop firstHerd:lastHerd*

*firstGeneration:lastGeneration*

if integers of firstPop,lastPop,firstHerd,lastHerd,firstGeneration,lastGeneration =-9, conditions are ignored. Otherwise, can be specifed.

*maf* Definition: Threshold for minimum-allele frequency

Type: Real

Options: 0.0≤*maf*≤0.50

Default: *maf* must be specified

*mafInclude* Definition: Minimum-allele frequency of loci included. Loci

with minimum-allele frequency 0.0, that is, are

fixed and do not segregate, are excluded.

Type: Character

Options:

above Include loci with minimum-allele frequency

greater than or equal to *maf*

below Include loci with minimum-allele frequency

less than or equal to *maf*

Default: *mafInclude* must be specified

*IncludePopHerdGenoData*= stage (npop\*nherd)

Description: only genotype data from some populations and herds will be included to

estimate EBV if *incGenoPop=’part’*.

(npop\*nherd): matrix with dimension (npop\*nherd). 1 indicate included,

0 indicate not included/used

*checkDetGmatrix=checkDeterminantGmatrix*

Type: integer

Dimension: prmstage (nrows of parameter)

Options: 1: checked; 0: no check

Description: Check log determinant of genomic-relationship matrix calculated by Gousheng program. Other programs do not work. This optional input is placed here to reduce changes of parameter input files.

\*\*Additional information\*\*

1) An overview of the program invhmatrix is provided in section 13. The manual is

provided in the Appendix.

2) Namelist &**GENOMICBLUPPARAMETERS** is only read when there is a selection stage(s)

using genomic-breeding value as the selection criterion.

3) The character variables, *genomicBase*, *loci*, *mafInclude*, and *locusWeight*, must be

provided in inverted commas (‘’).

4) Fixed loci, loci with minimum-allele frequency 0.0, are not used to construct

genomic matrices.

5)

When constructing base file for AGSCALE 2:

do iid=1,maxid

if (.not. genomicBasePopulation(iid)) cycle

select case (genomicBase)

case ('base','genotypedbase','tracedbase')

select case (gMatrixTask)

case ('prediction')

if (.not. lpedGenoBlup(iid) .or. .not. pop(iid)%genotyped) cycle

case ('eva')

if (.not. lpedEva(iid) .or. .not. pop(iid)%genotyped) cycle

end select

end select

6) Selection stages using genomic-breeding value as selection criterion are stages of

namelist &SELECTION, variable *selection\_groups* with (i) *sex\_code* 0:6,

*selection\_criterion* ‘genomicblup’, and *RunBlup* 1, and (ii) *sex\_code* 7, *MaleSelCrit*

or *FemaleSelCrit* ‘genomicblup’, and *MaleRunBlup* and/or *FemaleRunBlup* 1 in

corresponding EVA-selection stage of namelist &EVA, variableEvaSelection.

x) Stages can be provided in any order; all stages must be provided

x) Selection stages using genomic-breeding value as selection criterion must be

represented once in variable *economicValueEbv*.

## **&IBDBLUPPARAMETERS**

parameters=*stage firstChrom lastChrom loci* /

NB! Number of lines in parameters must equal the number of selection stages using

IBD-breeding value as selection criterion

\*\*Task\*\*

Input to generate IBD-relationship matrices for IDB-BLUP.

\*\*Properties of names\*\*

*stage* Definition: Stage of selection. Must correspond with a

selection stage that uses (i) *selection\_criterion*

‘ibdblup’, and *RunBlup* 1 in namelist

&SELECTION, variable *selection\_scheme*, or (ii)

*MaleSelCrit* and/or *FemaleSelCrit* ‘ibdblup’ and

*MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist

&EVA, variable EvaSelection.

Type: Integer

Options: 1≤*stage*≤*selection\_groups*

Default: *stage* must be specified

*firstChrom* Definition: First chromosome used to construct IBD-

relationship matrices

Type: Integer

Options: 1≤*firstChrom*≤*nchrom*

Default: *firstChrom* must be specified

*lastChrom* Definition: Last chromosome with loci used to construct

IBD-relationship matrices

Type: Integer

Options: *firstChrom*≤*lastChrom*≤*nchrom*

Default: *lastChrom* must be specified

Loci on chromosomes *firstChrom*:*lastChrom* are used to construct matrices

*loci* Definition: Loci used as genetic markers

Type: Character

Options:

markers Markers

qtl QTL

all QTL and markers

Default: *loci* must be specified

\*\*Additional information\*\*

1) Namelist &**IBDBLUPPARAMETERS** is only read when there is a selection stage(s)

using ibd-breeding value as the selection criterion.

*geneticModel* must be ‘genomic’. ~~Should we allow ‘ibdblup’ with~~ *~~geneticModel~~* ~~‘qtl’?~~

‘ibdblup’ does not rely on genotypings. Animals do not need to be genotyped to be part of ibd-relationship matrix.

2) The character variable, *loci*, must be provided in inverted commas (‘’).

3) Selection stages using ibd-breeding value as selection criterion are stages of

namelist &SELECTION, variable *selection\_groups* with (i) *sex\_code* 0:6,

*selection\_criterion* ‘ibdblup’, and *RunBlup* 1, and (ii) *sex\_code* 7, *MaleSelCrit*

or *FemaleSelCrit* ‘ibdblup’, and *MaleRunBlup* and/or *FemaleRunBlup* 1 in

corresponding EVA-selection stage of namelist &EVA, variableEvaSelection.

x) Stages can be provided in any order; all stages must be provided

x) Selection stages using ibd-breeding value as selection criterion must be

represented once in variable *economicValueEbv*.

## **&IBSBLUPPARAMETERS**

parameters=*stage firstChrom lastChrom loci* /

NB! Number of lines in parameters must equal the number of selection stages using

IBS-breeding value as selection criterion

\*\*Task\*\*

Input to generate IBS-relationship matrices for IDB-BLUP.

\*\*Properties of names\*\*

*stage* Definition: Stage of selection. Must correspond with a

selection stage that uses (i) *selection\_criterion*

‘ibsblup’, and *RunBlup* 1 in namelist

&SELECTION, variable *selection\_scheme*, or (ii)

*MaleSelCrit* and/or *FemaleSelCrit* ‘ibsblup’ and

*MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist

&EVA, variable EvaSelection.

Type: Integer

Options: 1≤*stage*≤*selection\_groups*

Default: *stage* must be specified

*firstChrom* Definition: First chromosome used to construct IBS-

relationship matrices

Type: Integer

Options: 1≤*firstChrom*≤*nchrom*

Default: *firstChrom* must be specified

*lastChrom* Definition: Last chromosome with loci used to construct

IBS-relationship matrices

Type: Integer

Options: *firstChrom*≤*lastChrom*≤*nchrom*

Default: *lastChrom* must be specified

Loci on chromosomes *firstChrom*:*lastChrom* are used to construct matrices

*loci* Definition: Loci used as genetic markers

Type: Character

Options:

markers Markers

qtl QTL

all QTL and markers

Default: *loci* must be specified

\*\*Additional information\*\*

1) Namelist &**IBSBLUPPARAMETERS** is only read when there is a selection stage(s)

using ibs-breeding value as the selection criterion.

*geneticModel* must be ‘genomic’. ~~Should we allow ‘ibsblup’ with~~ *~~geneticModel~~* ~~‘qtl’?~~

‘ibsblup’ does not rely on genotypings. Animals do not need to be genotyped to be part of ibs-relationship matrix.

2) The character variable, *loci*, must be provided in inverted commas (‘’).

3) Selection stages using ibs-breeding value as selection criterion are stages of

namelist &SELECTION, variable *selection\_groups* with (i) *sex\_code* 0:6,

*selection\_criterion* ‘ibsblup’, and *RunBlup* 1, and (ii) *sex\_code* 7, *MaleSelCrit*

or *FemaleSelCrit* ‘ibsblup’, and *MaleRunBlup* and/or *FemaleRunBlup* 1 in

corresponding EVA-selection stage of namelist &EVA, variableEvaSelection.

x) Stages can be provided in any order; all stages must be provided

x) Selection stages using ibs-breeding value as selection criterion must be

represented once in variable *economicValueEbv*.

## **&EVAGENOMICRELATIONSHIPPARAMETERS**

parameters=*stage centre scale weight firstChrom lastChrom*  *loci scaleGToA*

*propAToG addDiagG diagGOne unrelateZero*

centre=*stage genomicBase firstPop lastPop firstHerd lastHerd*

*firstGeneration lastGeneration maf mafInclude*

scale= *stage genomicBase firstPop lastPop firstHerd lastHerd*

*firstGeneration lastGeneration maf mafInclude*

weight= *stage genomicBase firstPop lastPop firstHerd lastHerd*

*firstGeneration lastGeneration maf mafInclude*

scaleGToA= *stage genomicBase firstPop lastPop firstHerd lastHerd*

*firstGeneration lastGeneration*

singleGeneParameters= *stage genomicBase firstPop lastPop firstHerd lastHerd*

*firstGeneration lastGeneration maf mafInclude*

/

NB! Number of lines in parameters must equal the number of selection stages using

genomic-relationship matrix to constrain average relationship in EVA-selection.

\*\*Task\*\*

Input to construct genomic-**H** matrices using the program, invhmatrix. Genomic-**H**

matrices are used by EVA to constrain average relationship. The namelist is read

when *relationshipMatrix* ‘genomic’ in any EVA-selection stage of namelist &EVA,

variable *EvaSelection*.

\*\*Properties of names\*\*

Description of variables in parameters is as described for namelist

&GENOMICBLUPPARAMETERS with the exception that:

(i) *genomicBase* ‘all’ includes all genotyped animals that are candidates for EVA-

selection (tagged for EVA-selection) or too young to be selection candidates and

all genotyped individuals traced back from these animals, ‘tracedbase’ are base

animals traced back from candidates for EVA-selection (tagged for EVA-selection)

or too young to be selection candidates, and ‘tracedgenotypedbase’ are are

genotyped base animals traced back from candidates for EVA-selection (tagged for

EVA-selection) or too young to be selection candidates.

## **&EVAIBDRELATIONSHIPPARAMETERS**

parameters=*stage firstChrom lastChrom loci* /

NB! Number of lines in parameters must equal the number of selection stages using

ibd-relationship matrix to constrain average relationship in EVA-selection.

\*\*Task\*\*

Loci used to generate IBD-relationship matrices that constrain average relationship

in EVA-selection.

\*\*Properties of names\*\*

*stage* Definition: Stage of selection. Must correspond with a

selection stage that uses (i) *selection\_criterion*

‘ibdblup’, and *RunBlup* 1 in namelist

&SELECTION, variable *selection\_scheme*, or (ii)

*MaleSelCrit* and/or *FemaleSelCrit* ‘ibdblup’ and

*MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist

&EVA, variable EvaSelection.

Type: Integer

Options: 1≤*stage*≤*selection\_groups*

Default: *stage* must be specified

*firstChrom* Definition: First chromosome used to construct IBD-

relationship matrices

Type: Integer

Options: 1≤*firstChrom*≤*nchrom*

Default: *firstChrom* must be specified

*lastChrom* Definition: Last chromosome with loci used to construct

IBD-relationship matrices

Type: Integer

Options: *firstChrom*≤*lastChrom*≤*nchrom*

Default: *lastChrom* must be specified

Loci on chromosomes *firstChrom*:*lastChrom* are used to construct matrices

*loci* Definition: Loci used as genetic markers

Type: Character

Options:

markers Markers

qtl QTL

all QTL and markers

Default: *loci* must be specified

\*\*Additional information\*\*

1) Namelist &**IBDBLUPPARAMETERS** is only read when there is a selection stage(s)

using ibd-breeding value as the selection criterion.

*geneticModel* must be ‘genomic’. ~~Should we allow ‘ibd’ with~~ *~~geneticModel~~* ~~‘qtl’?~~

‘ibdblup’ does not rely on genotypings. Animals do not need to be genotyped to be part of ibd-relationship matrix.

2) The character variable, *loci*, must be provided in inverted commas (‘’).

3) Selection stages using ibd-breeding value as selection criterion are stages of

namelist &SELECTION, variable *selection\_groups* with (i) *sex\_code* 0:6,

*selection\_criterion* ‘ibdblup’, and *RunBlup* 1, and (ii) *sex\_code* 7, *MaleSelCrit*

or *FemaleSelCrit* ‘ibdblup’, and *MaleRunBlup* and/or *FemaleRunBlup* 1 in

corresponding EVA-selection stage of namelist &EVA, variableEvaSelection.

x) Stages can be provided in any order; all stages must be provided

x) Selection stages using ibd-breeding value as selection criterion must be

represented once in variable *economicValueEbv*.

## **&EVAIBSRELATIONSHIPPARAMETERS**

parameters=*stage firstChrom lastChrom loci* /

NB! Number of lines in parameters must equal the number of selection stages using

ibs-relationship matrix to constrain average relationship in EVA-selection.

\*\*Task\*\*

Loci used to generate IBS-relationship matrices that constrain average relationship

in EVA-selection.

\*\*Properties of names\*\*

*stage* Definition: Stage of selection. Must correspond with a

selection stage that uses (i) *selection\_criterion*

‘ibsblup’, and *RunBlup* 1 in namelist

&SELECTION, variable *selection\_scheme*, or (ii)

*MaleSelCrit* and/or *FemaleSelCrit* ‘ibsblup’ and

*MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist

&EVA, variable EvaSelection.

Type: Integer

Options: 1≤*stage*≤*selection\_groups*

Default: *stage* must be specified

*firstChrom* Definition: First chromosome used to construct IBS-

relationship matrices

Type: Integer

Options: 1≤*firstChrom*≤*nchrom*

Default: *firstChrom* must be specified

*lastChrom* Definition: Last chromosome with loci used to construct

IBS-relationship matrices

Type: Integer

Options: *firstChrom*≤*lastChrom*≤*nchrom*

Default: *lastChrom* must be specified

Loci on chromosomes *firstChrom*:*lastChrom* are used to construct matrices

*loci* Definition: Loci used as genetic markers

Type: Character

Options:

markers Markers

qtl QTL

all QTL and markers

Default: *loci* must be specified

\*\*Additional information\*\*

1) Namelist &**IBSBLUPPARAMETERS** is only read when there is a selection stage(s)

using ibs-breeding value as the selection criterion.

*geneticModel* must be ‘genomic’. ~~Should we allow ‘ibs’ with~~ *~~geneticModel~~* ~~‘qtl’?~~

‘ibsblup’ does not rely on genotypings. Animals do not need to be genotyped to be part of ibs-relationship matrix.

2) The character variable, *loci*, must be provided in inverted commas (‘’).

3) Selection stages using ibs-breeding value as selection criterion are stages of

namelist &SELECTION, variable *selection\_groups* with (i) *sex\_code* 0:6,

*selection\_criterion* ‘ibsblup’, and *RunBlup* 1, and (ii) *sex\_code* 7, *MaleSelCrit*

or *FemaleSelCrit* ‘ibsblup’, and *MaleRunBlup* and/or *FemaleRunBlup* 1 in

corresponding EVA-selection stage of namelist &EVA, variableEvaSelection.

x) Stages can be provided in any order; all stages must be provided

x) Selection stages using ibs-breeding value as selection criterion must be

represented once in variable *economicValueEbv*.

## **&BLUPPARAMETERS**

ebv\_observation=*ebv\_observation*

Dmublupparm= *stage dmuVersion runDMUbyUserScript dirFileName parFileName nfiles*

*lstFileNeeded dmuScriptFile incPhenoPop incObsFirtTS incObsLastTS*

*previousObsTS incPedFirstTS incPedLastTS previousPedTS*

manualblupparm= *stage nfiles lstFileNeeded cmdrun logfile solfile dataHeader*

*genoData loci GenoDataFormat GenoDataDelimiter GenoDataNSKIP*

IncludePopHerdPhenoData=*stage* (npop\*nherd)

litterObservations=*litterObs*

printDmuLstFiles=*printDmuLstFiles*

idinfo=*idinfo*

familyOriginInfo=*familyOriginInfo*

addFactors=*addFactors* /

\*\*Task\*\*

Input for estimation of breeding values using DMU or own program from user. Dmublupparm is required input when runBlup=1. manualblupparm is required input when runBlup=2.

*ebv\_observation* Definition: Matrix relating observations to BLUP-breeding

values. Used to create DMU-input data and read

breeding values predicted by DMU.

Dimension: (*nobs*+*nProgenyObs*+*nCombinedObs*)\**nebv*

Options: Matrix elements 0 or 1

Type: Integer

Default: *ebv\_observation* must be specified

\*\*Properties of names for Dmublupparm\*\*

*stage* Definition: Stage of selection. Must correspond with a

selection stage that uses *selection\_criterion*

and *RunBlup* 1.

Type: Integer

Options: 1≤*stage*≤*selection\_groups*

Default: *stage* must be specified

*dmuVersion* Definition: DMU version used to estimate breeding values

at the stage. Different stages can use different version

Type: Character

Options:

dmuai DMUAI used

dmu4 DMU4 used

dmu5 DMU5 used

Default: *dmuVersion* must be specified

*runDMUbyUserScript* Definition: specify script to run DMU generated by user or ADAM

Type: Integer

Options:

0 generated by ADAM

1 user provided script for running DMU.

Default: *runDMUbyUserScript* must be specified

*dirFileName* Definition: a directive file name used for running DMU.

Type: Character

Options:

’’ if runDMUbyUserScript=1

‘filename’ if runDMUbyUserScript=0

Default: *dirFileName* must be specified

*parFileName* Definition: a parout file name used for parameter input of

the directive file. if dmuVersion='dmuai', this will be the file name for

Parout file. If not given, parFileName=ParInput[Stage].txt dmuai used.

Type: Character

Options:

’’ if runDMUbyUserScript=1, this file is not read.

‘filename’ if runDMUbyUserScript=0, this file is read, but optional

Default: *parFileName* must be specified

n*files* Definition: number of files to be copied and used for running

user scripts if runDMUbyUserScript=1.

Type: Integer

Options:

0 if runDMUbyUserScript=0

>=1 if runDMUbyUserScript=1

Default: *nfiles* must be specified

*lstFileNeeded* Definition: lists of all files needed for running script in the

current working folder if runDMUbyUserScript=1.

Type: Character

Options:

’’ if runDMUbyUserScript=0

‘filename1 filename2’ if runDMUbyUserScript=1 & nfiles=2.

list of files are seprerate with space.

Default: *lstFileNeeded* must be specified

*dmuScriptFile* Definition: file name of the script to run user program on unix.

this file must be in the list of lstFileNeeded if runDMUbyUserScript=1.

Type: Character

Options:

’’ if runDMUbyUserScript=0

‘filename’ if runDMUbyUserScript=1

Default: *dmuScriptFile* must be specified

Note: if runDMUbyUserScript=1, the end result must be:

* Solution file must be named SOL,
* Same formats as DMU sol file: first row indicate descriptor (integer and real value); column 3 of integer part is ebvNumber/trait number (1<=value<=nebv); column 5 of integer part is ID; and column 1 of real part is ebv.

Following input allows flexible inclusion of phenotype data in prediction. Need to be careful using this option. Only recommended for highly experienced user. Most of the time, put all and -9 for all following parameters

*incPhenoPop* Definition: whether all phenotyped individauls or parts

(in certain herds and populations) are included

in BLUP prediction for EBV.

Type: Character

Options:

all : all phenotyped animals

genotyped : phenotyped data of all genotyped individuals

part : parts of phenotyped animals. Specified in

IncludePopHerdPhenoData

Default: *incPhenoPop* must be specified

*incObsFirtTS* Definition: whether all phenotyped individauls or parts

(from certain timesteps) are included in BLUP prediction for EBV.

Type: integer

Options: -9 : includes phenotyped animals from all TS

*0=< incObsFirtTS =< incObsLastTS <= ntime*

Default: *incObsFirtTS* must be specified

*incObsLastTS* Definition: whether all phenotyped individauls or parts

(to certain timesteps) are included in BLUP prediction for EBV.

Type: integer

Options: -9 : includes phenotyped animals from all TS

*0=< incObsFirtTS =< incObsLastTS <= ntime*

Default: *incObsLastTS* must be specified

*previousObsTS* Definition: includes phenotype data from (current\_time-previousGenoTS)

to current\_time for prediction of EBV.

Type: integer

Options: -9 : includes phenotyped animals from all TS

0< *previousObsTS* <= ntime

Default: *previousObsTS* must be specified

*! warning: extremely careful when using following options. these options may lead to some ID donot have ebv, or need adding negative values for base animals (auto in DMU).*

*It must be also in agreement with options in genomicRelBlupParm if relevant. In short, users must know what they are doing.*

*incPedFirtTS* Definition: whether all individauls or parts in pedigree

(from certain timesteps) are included in BLUP prediction for EBV.

Type: integer

Options: -9 : includes all animals from all TS in ped

*0=< incPedFirtTS =< incPedLastTS <= ntime*

Default: *incPedFirtTS* must be specified

*incPedLastTS* Definition: whether all genotyped individauls or parts

(to certain timesteps) are included in BLUP prediction for EBV.

Type: integer

Options: -9 : includes phenotyped animals from all TS

*0=< incPedFirtTS =< incPedLastTS <= ntime*

Default: *incPedLastTS* must be specified

*previousPedTS* Definition: includes phenotype data from (current\_time-previousGenoTS)

to current\_time for prediction of EBV.

Type: integer

Options: -9 : includes phenotyped animals from all TS

0< *previousPedTS* <= ntime

Default: *previousPedTS* must be specified

*IncludePopHerdGenoData*= stage (npop\*nherd)

Description: only genotype data from some populations and herds will be included to

estimate EBV if *incGenoPop=’part’*.

(npop\*nherd): matrix with dimension (npop\*nherd). 1 indicate included,

0 indicate not included/used

\*\*Properties of names for manualblupparm \*\*

Experienced user + good skills with unix.

*stage* Definition: Stage of selection. Must correspond with a

selection stage that uses *selection\_criterion*

and *RunBlup* 2.

Type: Integer

Options: 1≤*stage*≤*selection\_groups*

Default: *stage* must be specified

n*files* Definition: number of files to be copied and used for running

user scripts.

Type: Integer

Options:

>=1

Default: *nfiles* must be specified

*lstFileNeeded* Definition: lists of all files needed for running script in the

current working folder.

Type: Character

Options:

‘filename1 filename2’ if nfiles=2.

list of files are seprerate with space.

Default: *lstFileNeeded* must be specified

*Cmdrun* Definition: command to run the scripts on unix.

Type: Character

Options:

Default: *cmdrun* must be specified

*logfile* Definition: name of log/lst file to be reported to user.

Type: Character

Options: ‘filename1’

Default: ‘*blup.lst’*

*solfile* Definition: solution file to be read by ADAM.

Type: Character

Options: ‘filename’

Default: ‘*SOL’*

! format of solfile (no header) must be:

* col1: ebvNumber/trait number (1<=value<=nebv).
* Col2: ID.
* Col3: EBV (real.

*dataHeader* Definition: ADAM will provide phenotype data (dmudat) with or

without header.

Type: Integer

Options:

0 no header format

1 format with header in the first line

Default: *dataHeader* must be specified

*genoData* Definition: genotype data provided or not.

Type: Integer

Options:

0 no genotype data needed

1 yes, provide data, named marker.RawData

Default: *genoData* must be specified

Other files provided with genotype data: map.RawData, genotyped.dat

*loci* Definition: Loci used as markers for BLUP prediction

Type: Character

Options:

'all' QTL and markers

'qtl' QTL

'markers' Markers

Default: *loci* must be specified

*GenoDataFormat* Definition: Format of the genotype data.

Type: Integer

Options:

-9 no genotype data needed

1 marker data with allele code (integer)

Eg: A1A1 A1A2 A2A2: 1 1 1 2 2 2

2 marker data with genotype code (integer)

Eg: A1A1 A1A2 A2A2: 0 1 2

Default: *GenoDataFormat* must be specified

*GenoDataDelimiter* Definition: Format of the genotype data.

Type: Integer

Options:

-9 no genotype data needed

1 no seperation between marker code

2 seperation with space

Default: *GenoDataDelimiter* must be specified

*GenoDataNSKIP* Definition: Number of variables between ID and marker,

which might not be read. All number for the variables are 9.

Type: Integer

Options:

-9 no genotype data needed

-1: no id needed or added to genotype file (for LMT)

>=0 Number of variables between ID and marker

Default: *GenoDataNSKIP* must be specified

*printDmuLstFiles* Definition: Write DMU-lst or log files to subdirectory,

dmuLstFiles. Subdirectory DmuLstFiles

is created in the output directory specified

in namelist &OUTPUTDIRECTORY, variable

*OutDirectory.*

Type: Character

Options:

no DMU-lst or log files are not written to subdirectory

yes DMU-lst or log files are written

Default: no

*addFactors*  Definition: add fixed factors of *SireTSidTS SireTSdamTS*

*SireTSdamTSidTS* to dmudat file to challenge BLUP model. These fators correct source of genetic level differences between selection candidates with overlapping generations. The factors are often included in stastical model in real genetic evaluation.

Type: Character

Options:

no not add the factors

yes add the factors

Default: no

*Idinfo*  Definition: generate a dataset with individuals' information

(birthpop, currentpop, birth herds...).

Type: Character

Options:

no the file not generated.

yes the file generated.

Default: no

*familyOriginInfo* Definition: generate variable family origins to phenotype file

(sireFamOrigin damFamOrigin famOrigin identity...)

Type: Character

Options:

no variables not added.

yes variables added.

Default: no

\*\*Additional information\*\*

1) *dmuPolyBlup* is used when polygenic breeding values are used as selection

criterion at any selection stage, *dmuGenomicBlup* is used when genomic breeding

values are used~~, and~~ *~~dmuGasBlup~~* ~~when GAS breeding values are used~~.

2) (a) *dmuPolyBlup* is also used when genomic breeding values are specified as the

selection criterion but polygenic breeding values are estimated instead of

genomic breeding values. This occurs when there is a lack of genotypic

and/or phenotypic observations (see section *Genetic model and selection;*

ADAM’s motor)~~, or when~~ *~~startGenomicSelectionTime~~*~~>1 (namelist~~

~~&CONTROL\_GENOME)~~.

~~(b)~~ *~~dmuPolyBlup~~* ~~is also used when gas breeding values are specified as the~~

~~selection criterion but polygenic breeding values are estimated instead of~~

~~gas breeding values. This occurs when there is a lack of phenotypic~~

~~observations (see section~~ *~~Genetic model and selection~~*~~), or~~

~~when~~ *~~startGasTime~~*~~>1 (namelist &CONTROL\_GENOME)~~.

3) When *printDmuLstFiles* ‘yes’:

(a) DMU-lst files are written to directory dmuLstFiles at selection stages with

(i) truncation or EVA selection, and (ii) selection criterion polygenic,

genomic, ~~or GAS~~ breeding values

(b) Directory DmuLstFiles is a subdirectory of the output directory

*OutDirectory* (namelist &OUTPUTDIRECTORY, variable *OutDirectory*)

(c) DMU-lst files are written as dmuAdamRep<*rep*>Time<*time*>Stage<*stage*>.lst,

where <*rep*> is replicate number, <*time*> is time step, and <*stage*> is

selection stage

(d) If the subdirectory dmuLstFiles does not exist, it is created

(e) If the subdirectory dmuLstFiles exists and it contains old DMU-lst files

(files from an earlier simulation), the old DMU-lst files are renamed

dmuAdamRep<*rep*>Time<*time*>Stage<*stage*>.prev

## **&BAYESPPARAMETERS**

There is now a version of ADAM that can use BayesP instead of GBLUP for genomic prediction. It is restricted to fewer cases than GBLUP, e.g. only a single trait and all animals genotyped.

BayesP can be used when specifying ‘bayesp’ in &SELECTION rather than ‘polyblup’ or ‘genomicblup’ as SelectionCriterion.

The attached input.prm works with the new version of ADAM. Here are some comments to the additional parameters involved.

There is a new namelist with up to seven parameters:

&BAYESPPARAMETERS

nIterations=1000

priorProbability=0.05

priorResidualVariance=1.0

priorGeneticVariance=1.0

bayesPLoci='markers'

bayespMafInclude='above'

mafBayesP=0.0

/

The values given for the parameters in the example above are the default values. That means that if these values suit you, you don’t need to put them in the namelist.

nIterations Is the number of rounds run in the BayesP program. This is a parameter that Theo knows a lot more about than me.

priorProbability Is the mixing proportion in the program, i.e. the proportion of loci with big effect.

priorResidualVariance Is the residual variance. This should almost always fit the parameter given in r\_matrix.

priorGeneticVariance is the genetic variance. This should almost always fit the parameter given in *genomicMatrix*.

bayesPLoci controls which loci to use whether ‘all’, ‘markers’, or ‘qtl’.

bayespMafInclude specifies whether to use loci with MAF ‘above’ or ‘below’ the MAF-threshold.

mafBayesP is the MAF-threshold.

## **&MATINGPARAMETERS**

mating\_scheme= *matingGroupNumber mating\_design offspringDestinyPop offspringDestinyHerd*

*offspringDestinyIdentity MatingUnit CrossChosen nCrosses nTSgestation*

*nTSmatingkept redefineParmMatingkept McacGenerations*

KeptMatingRedefine= *matingGroupNumber matingkeptNumber offspringDestinyPop*

*offspringDestinyHerd offspringDestinyIdentity litterSizeAllocation*

*litter\_size sexAllocation sex\_ratio*

plantMatingPlot= *plantMatingPlot*

nSelfingAfterBorn*=nSelfingAfterBorn*

familyOriginOfParentAsOffsprings*=familyOriginOfParentAsOffsprings*

/

\*\*Task\*\*

Each mating group could have different mating designs.

\*\*Properties of names mating\_scheme\*\*

*matingGroupNumber* Definition: Mating group number

Type: Integer

Options: 1≤ *matingGroupNumber* ≤*nMatingGroups*,

where *nMatingGroups* is the max number of mating groups

Default: *matingGroupNumber* must be specified

*mating\_design* Definition: Method to mate selected animals; animals are mated

within mating groups

Type: Character

Default: *mating\_design* must be specified

Options:

‘random\_mating’ :

Parents mated randomly. For plant: It could be sex\_code 1 (in selection\_scheme) mated with sex\_code 2. sex\_code 1 cant mate with individual from sex\_code 1, or sex\_code 0.

‘random\_unique\_mating’ :

same as random\_mating, but unique pairs. Mating pairs are not repeated. In factorial mating, 1 male mates with several females, and 1 female mates with several males. So, there are cases where the same pair of male and female happens more than 1 in random\_mating. This mating design advoids same pairing (thus call unique mating).

‘mc\_mating’ :

Mating using Minimum-coancestry mating. sex could be actual sex of individuals (in animals). Or for plant: it could also be sex\_code 1 mated with sex\_code 2. sex\_code 1 cant mate with individual from sex\_code 1.

‘mcac\_mating’ :

Mating minimising covariance of ancestral genetic contributions as described by Henryon et al. (2010). sex could be actual sex of individuals (in animals). Or for plant: it could also be sex\_code 1 mated with sex\_code 2. sex\_code 1 cant mate with individual from sex\_code 1.

‘assortative\_mating’ :

Best male parents mated to the best females, where best is the criterion used to select males and females. sex could be actual sex of individuals (in animals). Or for plant: it could also be sex\_code 1 mated with sex\_code 2. sex\_code 1 cant mate with individual from sex\_code 1.

‘desortative\_mating’ :

Best male parents mated to the worse females, where best is the criterion used to select males and females. sex could be actual sex of individuals (in animals). Or for plant: it could also be sex\_code 1 mated with sex\_code 2. sex\_code 1 cant mate with individual from sex\_code 1.

‘pseudorandom\_mating’ :

same as random mating but indiviudals can't mate with itselt. Or no self-polinated.

‘pseudorandom\_unique\_mating’ :

same as pseudorandom mating but unique pairs. Mating pairs are not repeated.

‘selfing’ :

Self-polination. Sexcode contribute to this mating group must be 0.

‘clone’ :

individuals are cloned. Sexcode contribute to this mating group must be 0.

‘dhs’ :

doubling haploid. Sexcode contribute to this mating group must be 0.

‘random\_crossing’ :

individual can mate with every other individual. But only maximum repro\_capacity number of mating result in offsprings. Sexcode contribute to this mating group must be 0.

‘pseudorandom\_crossing’ :

same as random\_crossing, except that no self-polinated.

‘backcross\_par1’ :

backcrossing between animal and its parent 1 (sire). The backcross animals (selection destination of 1) has sex code of 1 in selection\_scheme. The backcross animal lists could have many more than needed. The animals to be backcrossed (selection destination of 1) has sex code of 2 in selection\_scheme. Reproduction capacity of sex code 2 must be ==1. Total number of matings will be equal to number of individuals selected in stage with sex code =2.

‘backcross\_par2’ :

backcrossing between animal and its parent 2 (dam). Other chracteristics same as above.

‘backcross\_par1fsib’ :

backcrossing between animal and its parent 1 (sire)'s fullsib. Other chracteristics same as above.

‘backcross\_par2fsib’ :

backcrossing between animal and its parent 2 (dam)'s fullsib. Other chracteristics same as above.

‘backcross\_par1or2’ :

backcrossing between animal and its parent 1 or 2 (sire or dam). Other chracteristics same as above.

‘backcross\_par1or2fsib’ :

backcrossing between animal and its parent 1 or 2 (sire or dam) 's fullsib. Other chracteristics same as above.

Note: if sex=0 individuals will be added to both sire and dam mating\_lists

*offspringDestinyPop* Definition: Offspring destinied to population number.

Type: Integer

Options:

1<=offspringDestinyPop<=npop : offspring are destined to the population number.

This is used for crossbreeding.

offspringDestinyPop=-2,0 : offspring are born at the population of its dam.

offspringDestinyPop=-1 : offspring are born at the population of its sire.

Default: *mating\_design* must be specified

*offspringDestinyHerd* Definition: Offspring destinied to herd number.

Type: Integer

Options:

1<=offspringDestinyHerd<=npop : offspring are destined to the herd number.

-2,0 : offspring are born at the herd of its dam.

-1 : offspring are born at the herd of its sire.

Default: *offspringDestinyHerd* must be specified

*offspringDestinyIdentity* Definition: Offspring destinied to identity number.

Type: Integer

Options:

>= : offspring are destined to Identity number.

<0 : offspring have Identity of its dam +1.

Default: *offspringDestinyIdentity* must be specified

Note: in most of case in animal breeding scheme: offspringDestinyIdentity=0. This option is used for produce P0, F1..Fn.

*MatingUnit*  Definition: mating happens between individuals

within mating units

Type: character

Default: *mating\_design* must be specified

Options:

‘defaults’ : mating unit is not defined. Mating is regardless of population

id, herd id, family id… of the 2 individuals

‘within\_herd’ : within herd (within population) mating.

Mating couple is from same herd (of the same population).

‘within\_fullsibfamily’ : within full-sibs family mating.

Mating couple is from the same full-sib family

‘within\_familyorigins’ : within FamilyOrigins mating.

Mating couple is from the same FamilyOrigins

‘within\_grandparent’ : within grandparent mating. Mating couple is from the same

grandparent (in case of selfing for two generations)

*CrossChosen*  Definition: in crossing of plants, there are many possiblity of

crossing. Matings can only happen for some crosses.

This process of crossing is random.

Type: character

Options:

'all’ : all crosses (mating) are chosen.

In most cases of animal breeding, this is the choice.

‘random’ : crosses are chosen randomly to produce offspring

Default: *CrossChosen* must be specified

*nTSgestation*  Definition: This option allows more flexibility for

gestation\_length for each mating group.

Type: Integer

Options:

0 : offspring born at the end of current timestep.

1 <= *nTSgestation* <= *ntime-currentTS* : offspring born at the end of current

*timestep+nTSgestation*

Default: *nTSgestation* must be specified

*nTSmatingkept*  Definition: same mating could have offspring in several TS.

Type: Integer

Options: =1 these matings only used in this current timestep

>1 same mating is kept and use to produce offspring in

nTSmatingkept TS.

Default: *nTSmatingkept* must be specified

*redefineParmMatingkept*  Definition: redefine parameters for the mating.

If nTSmatingkept >1, parameters (offspringDestinyPop,

offspringDestinyHerd, offspringDestinyIdentity…) can be

redefined for consecutive TS.

Type: Integer

Options: 0 same parameters are used

1 these matings only used in this current timestep.

parameters (offspringDestinyPop, offspringDestinyHerd, offspringDestinyIdentity…) has to be redefined for consecutive TS.

Default: *redefineParmMatingkept*  must be specified

*mcacGenerations*  Definition: Number of generations ancestors of selected

animals are traced back when using minimum-

coancestry-covariance mating (MCAC mating).

Type: Integer

Options: ≥0

Default: *mcacGenerations* must be specified

*mcacGenerations*  Definition: Number of generations ancestors of selected

animals are traced back when using minimum-

coancestry-covariance mating (MCAC mating).

Type: Integer

Options: ≥0

Default: *mating\_design* must be specified

*mcacGenerations*  Definition: Number of generations ancestors of selected

animals are traced back when using minimum-

coancestry-covariance mating (MCAC mating).

Type: Integer

Options: ≥0

Default: *mating\_design* must be specified

*mcacGenerations*  Definition: Number of generations ancestors of selected

animals are traced back when using minimum-

coancestry-covariance mating (MCAC mating).

Type: Integer

Options: ≥0

Default: 5

\*\*Properties of names KeptMatingRedefine parameters\*\* (optional parameters).

Each line specifies parameters for 1 TS of 1 mating group.

*matingGroupNumber :* see description above

*matingkeptNumber :* integer. mating kept number (from 2..nTSmatingkept)

*offspringDestinyPop offspringDestinyHerd* : see description above

*offspringDestinyIdentity*

*litterSizeAllocation litter\_size* : see description in selection\_scheme

*sexAllocation sex\_ratio*

\*\*Properties of names plantMatingPlot parameters\*\* (optional parameters).

plantMatingPlot is an average biological genetic effects of different genotypes from the mating (eg. 2 parents/ full-sibs):

Each number specifies parameters for 1 mating group in the same order as for

*matingGroupNumber*

plantMatingPlot Definition: composition of plot from same mating (fullsibs).

Type: interger

Default: optional 0

Dimension: *nMatingGroups*

Options: 0,1 … Eg. With *nMatingGroups =4: 1 0 0 300*

0: individual (default), or normal obs

1: plot obs (infinite genotypes within plot)

>1: plot obs (n of genotypes within plot)

nSelfingAfterBorn

Definition: nSelfingEfterBorn is number of (automatically) selfing after the indiviudal is born. This option is used for skipping selfing step.

Type: interger

Default: optional 0 … 0

Dimension: *nMatingGroups*

Options:

0: no selfing after born

>0: number of selfing imidiately after born.

Eg. With *nMatingGroups =4: 1 0 0 4*

familyOriginOfParentAsOffsprings

Definition: Keep family Origin from parent(s) as family Origin of offspring.

Type: interger

Default: optional 0 … 0

Dimension: *nMatingGroups*

Options:

0: (default) not keeping, default tracing

1: family Origin of par1

2: family Origin of par2

Eg. With *nMatingGroups =4: 1 0 0 2*

\*\*Additional information\*\*

1) Animals are allocated to mating groups in namelist &SELECTION, variable

*selection\_scheme*.

3) When sum of sire matings does not equal sum of dam matings within a mating group,

excess matings are removed from randomly selected sires or dams.

5) *mcacGenerations* is only used and required when *mating\_design* ‘mcac\_mating’.

## **&MATRICES**

polygenicMatrix=*polygenicMatrix*

genomicMatrix=*genomicMatrix*

dominanceDegreeMeanVector=*dominanceDegreeMeanVect*

dominanceDegreeVarMatrix=*dominanceDegreeVarMat*

dominanceMatrix=*dominanceMat*

EpistasisAxAMatrix=*EpistasisAxAMatrix*

EpistasisAxDMatrix=*EpistasisAxDMatrix*

EpistasisDxDMatrix=*EpistasisDxDMatrix*

familyMatrix=*familyMatrix*

residualMatrix=*residualMatrix*

groupMatrix=*groupMatrix*

socialResidualMatrix1=*socialResidualMatrix1*

socialResidualMatrix2=*socialResidualMatrix2*

socialResidualMatrix3=*socialResidualMatrix3*

socialResidualMatrix4=*socialResidualMatrix4*

socialResidualMatrix5=*socialResidualMatrix5*

socialResidualMatrix6=*socialResidualMatrix6*

socialResidualMatrix7=*socialResidualMatrix7*

socialResidualMatrix8=*socialResidualMatrix8*

socialResidualMatrix9=*socialResidualMatrix9*

geneticResidualMatrix=*geneticResidualMatrix*

idMatrix=idMatrix

sireMatrix=sireMatrix

damMatrix=damMatrix

mateMatrix=mateMatrix

geneticMeanVector=*geneticMeanVector*

femaleDeviationVector=*femaleDeviationVector* /

\*\*Task\*\*

Provide genetic and residual (co)variance matrices and vectors of genetic means and

female deviations (fixed-sex effects)

\*\*Properties of names\*\*

*polygenicMatrix*  Definition: Polygenic (co)variance matrix. Elements are

read in the following order: (row 1,column 1),

(1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *ntbv*\**ntbv*

Type: Real

Default: Matrix of 0.0

*genomicMatrix* Definition: Genomic (co)variance matrix. Elements are

read in the following order: (row 1,column 1),

(1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *ntbv*\**ntbv*

Type: Real

Default: Matrix of 0.0

Must be 0.0 when geneticModel not ‘genomic’

*dominanceDegreeMeanVect* Definition: dominance Degree Mean Vector.

Dimension: *ntbv*

Type: Real

Default: Vector of 0.0 if SimulateDominance=0

Vector of 0.193 if SimulateDominance/=0

*dominanceDegreeVarMat* Definition: dominance Degree covariance matrix. Elements are

read in the following order: (row 1,column 1),

(1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *ntbv*\**ntbv*

Type: Real

Default: Matrix of 0.0 if SimulateDominance=0

Matrix of 0.0973 in diagonal, 0.0 off-diagonal if SimulateDominance/=0

*dominanceMatrix*  Definition: dominance covariance matrix. Elements are

read in the following order: (row 1,column 1),

(1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *ntbv*\**ntbv*

Type: Real

Default: Matrix of 0.0

Must be 0.0 when geneticModel not ‘genomic’ or SimulateDominance=0

*EpistasisAxAMatrix* Definition: 2-way epistatic interaction additive x additive covariance matrix. Elements are read in the following order: (row 1,column 1), (1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *ntbv*\**ntbv*

Type: Real

Default: Matrix of 0.0

Must be 0.0 when geneticModel not ‘genomic’ or SimulatePairEpistasis(1)=0

*EpistasisAxDMatrix* Definition: 2-way epistatic interaction additive x dominance covariance matrix. Elements are read in the following order: (row 1,column 1), (1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *ntbv*\**ntbv*

Type: Real

Default: Matrix of 0.0

Must be 0.0 when geneticModel not ‘genomic’ or SimulatePairEpistasis(2)=0

*EpistasisDxDMatrix* Definition: 2-way epistatic interaction dominance x dominance covariance matrix. Elements are read in the following order: (row 1,column 1), (1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *ntbv*\**ntbv*

Type: Real

Default: Matrix of 0.0

Must be 0.0 when geneticModel not ‘genomic’ or SimulatePairEpistasis(3)=0

*familyMatrix* Definition: Family (co)variance matrix. Elements are

read in the following order: (row 1,column 1),

(1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *nobs*\**nobs*

Type: Real

Default: Matrix of 0.0

*residualMatrix* Definition: Residual (co)variance matrix. Elements are

read in the following order: (row 1, column 1),

(1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *nres*\**nres*

Type: Real

Default: Matrix of 0.0

*groupMatrix* Definition: Group (co)variance matrix. Elements are

read in the following order: (row 1,column 1),

(1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *nobs*\**nobs*

Type: Real

Default: Matrix of 0.0

*socialResidualMatrix1* Definition: Social-residual (co)variance matrix. Elements are

read in the following order: (row 1,column 1),

(1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *maxGroupSize*\**maxGroupSize*, maxGroupSize=maxval(groupParm%maxGroupSize)

Type: Real

Default: Matrix of 0.0

*geneticResidualMatrix* Definition: Genetic-residual covariance matrix. Elements are

read in the following order: (row 1, column 1),

(1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *ntbv*\**nres*

Type: Real

Default: Matrix of 0.0

*preferentialTreatment* only applies when a non-zero element(s) in *geneticResidualMatrix*.

Converted to regression coefficients: *geneticResidualMatrix*(*i*,*j*)/(*polygenicMatrix*(*i*,*i*)+*genomicMatrix*(*i*,*i*))

Preferential treatments applied as a function of ‘parentaverage’ or ‘tbv’, where function is regression coefficient with intercept cohort mean.

*idMat*  Definition: Permanent id (co)variance matrix. Elements are

read in the following order: (row 1,column 1),

(1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *nobs*\**nobs*

Type: Real

Default: Matrix of 0.0

*sireMat*  Definition: Permanent sire (co)variance matrix. Elements are

read in the following order: (row 1,column 1),

(1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *nobs*\**nobs*

Type: Real

Default: Matrix of 0.0

*damMat*  Definition: Permanent dam (co)variance matrix. Elements are

read in the following order: (row 1,column 1),

(1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *nobs*\**nobs*

Type: Real

Default: Matrix of 0.0

*mateMat*  Definition: Permanent mate (co)variance matrix. Elements are

read in the following order: (row 1,column 1),

(1,2), ..., (1,column n), (2,1), (2,2) ...

Dimension: *nobs*\**nobs*

Type: Real

Default: Matrix of 0.0

*geneticMeanVector* Definition: Vector of genetic means

Dimension: *ntbv*

Type: Real

Default: Vector of 0.0

*femaleDeviationVector* Definition: Vector of female deviations

Dimension: *ntbv*

Type: Real

Default: Vector of 0.0

\*\*Additional information\*\*

1) Matrix of total-genetic (co)variances is the sum of *polygenicMatrix* and

*genomicMatrix*

2) *genomicMatrix* is only used when *geneticModel* ‘genomic’ in namelist

&CONTROLPARAMETERS

***&COVARIATEPARAMETERS***

parameters=

*obs covariateAnimal covariateObs regr* /

NB! Number of lines in parameters must equal *nCovariateParameters* in namelist

&POPULATIONPARAMETERS.

\*\*Task\*\*

Parameters used to add covariates to observations

\*\*Properties of names\*\*

*obs* Definition: Observation number; observation affected by *regr*

Type: Integer

Options: 1≤*obs*≤*nobs*

Default: *obs* must be specified

*covariateAnimal*  Definition: Animal covariate observed on; relative to animal

observed for *obs*

Type: Character

Options: 1≤*obs*≤*nobs*

Default: *covariateAnimal* must be specified

*covariateObs* Definition: Covariate observation; observed on *covariateAnimal*

Type: Integer

Options:

id Covariate an observation of animal

sire Covariate an observation of sire of animal

dam Covariate an observation of dam of animal

Default: *obs* must be specified

*regr* Definition: Regression

Type: Real

Default: *regr* must be specified

\*\*Additional information\*\*

1)

.not. covariateParm%obs==covariateParm%covariateObs

## **&DESIGN\_MATRICES**

ZDirectGenetic=*ZDirectGenetic*

WDirectError=*WDirectError*

ZPaternalGenetic=*ZPaternalGenetic*

WPaternalError=*WPaternalError*

ZMaternalGenetic=*ZMaternalGenetic*

WMaternalError=*WMaternalError*

ZSocialGenetic=*ZSocialGenetic*

WSocialError=*WSocialError*

ZMateGenetic=*ZMateGenetic*

WMateError=*WMateError*

XFemales=*XFemales* /

\*\*Task\*\*

Provide design matrices

\*\*Properties of names\*\*

*ZDirectGenetic* Definition: Design matrix associating true direct-genetic

effects to observations

Dimension: *nobs*\**ntbv*

Type: Integer

Options: 0, 1

Default: must be specified

*WDirectError* Definition: Design matrix associating direct-residual effects

to observations

Dimension: *nobs*\**nres*

Type: Integer

Options: 0, 1

Default: must be specified

*ZPaternalGenetic* Definition: Design matrix associating paternal-genetic effects

to observations

Dimension: *nobs*\**ntbv*

Type: Integer

Options: 0, 1

Default: Matrix of 0

*WPaternalError* Definition: Design matrix associating paternal-residual effects

to observations

Dimension: *nobs*\**nres*

Type: Integer

Options: 0, 1

Default: Matrix of 0

*ZMaternalGenetic* Definition: Design matrix associating maternal-genetic effects

to observations

Dimension: *nobs*\**ntbv*

Type: Integer

Options: 0, 1

Default: Matrix of 0

*WMaternalError* Definition: Design matrix associating maternal-residual effects

to observations

Dimension: *nobs*\**nres*

Type: Integer

Options: 0, 1

Default: Matrix of 0

*ZSocialGenetic* Definition: Design matrix associating social-genetic effects

to observations

Dimension: *nobs*\**ntbv*

Type: Integer

Options: 0, 1

Default: Matrix of 0

*WSocialError* Definition: Design matrix associating social-residual effects

to observations

Dimension: *nobs*\**nres*

Type: Integer

Options: 0, 1

Default: Matrix of 0

if (any(count(WSocialError==1,dim=1)>1)) then ‘error’

if (abs(rho-0.0)>epsilon(0.0)) then

if (all(WSocial==0)) ‘error’

end if

*ZMateGenetic* Definition: Design matrix associating mate-genetic effects

to observations

Dimension: *nobs*\**ntbv*

Type: Integer

Options: 0, 1

Default: Matrix of 0

*WMateError* Definition: Design matrix associating mate-residual effects

to observations

Dimension: *nobs*\**nres*

Type: Integer

Options: 0, 1

Default: Matrix of 0

Mate effects can only be specified for maternal observations; rows of *ZMateGenetic* and *WMateError* that are associated with maternal observations. Maternal observations are observations in *FemaleObservations* with *parity*≠-9.

Mate effects, where elements of *ZMateGenetic* and *WMateError* are set to 1, can only used when the reproductive capacity of females selected for reproduction is one. Reproductive capacities for selected females is provided in namelist selection, variable *selection\_scheme* and namelist eva, variable *EvaSelection*.

*XFemales* Definition: Design matrix associating female deviations to

observations

Dimension: *nobs*\**ntbv*

Type: Integer

Options: 0, 1

Default: Matrix of 0

\*\*Additional information\*\*

1) *ZMateGenetic* and *WMateError* apply to maternal traits; id of sire mated to dam prior to phenotyping of maternal traits.

## **&COMBINEDOBSERVATIONS**

combinations=*obs a b c d e* /

NB! Number of lines in combinations must equal *nCombinedObs*

\*\*Task\*\*

Input to generate combined traits. The *i*th combined trait (*i*=*nobs*+1 … *nobs*+*nCombinedObs*) for the *j*th animal, *yij*, is generated as *yij=*(*aj*+*bj*-*cj*)\**dj*/*ej*, where *aj*, *bj*, *cj*, *dj*, and *ej* are phenotypic observations generated for animal *j* and the observation numbers allocated to *a*, *b*, *c*, *d*, and *e* can be 1 … *nobs*.

\*\*Properties of names\*\*

*obs*  Definition: Observation number

Type: Integer

Options: *nobs*+1≤*obs*≤*nobs+nCombinedobs*

Default: *obs* must be specified

*a* Definition: Observation allocated to *a*

Type: Integer

Options:

-9 No observation is allocated to *a*

1:*nobs* Observation allocated to *a*

Default: *a* must be specified

*b* Definition: Observation allocated to *b*

Type: Integer

Options:

-9 No observation is allocated to *b*

1:*nobs* Observation allocated to *b*

Default: *b* must be specified

*c* Definition: Observation allocated to *c*

Type: Integer

Options:

-9 No observation is allocated to *c*

1:*nobs* Observation allocated to *c*

Default: *c* must be specified

*d* Definition: Observation allocated to *d*

Type: Integer

Options:

-9 No observation is allocated to *d*

1:*nobs* Observation allocated to *d*

Default: *d* must be specified

*e* Definition: Observation allocated to *e*

Type: Integer

Options:

-9 No observation is allocated to *e*

1:*nobs* Observation allocated to *e*

Default: *e* must be specified

\*\*Additional information\*\*

1) Namelist &COMBINEDTRAITS is only read and used when *geneticModel* ‘polygenic’

or ‘genomic’ in namelist &CONTROLPARAMETERS and *nCombinedTraits*>0 in namelist

&POPULATIONPARAMETERS

2) An observation number can only be represented once in variable combinations, *obs*.

This implies that all observation numbers in the sequence *nobs*+1 …

*nobs*+*nCombinedObs* must be represented in combinations, *obs*.

3) At least one of *a*, *b*, *c*, *d*, and *e* must be allocated an observation number

(1:*nobs*)for each combined observation. That is, all of *a*, *b*, *c*, *d*,

and *e* cannot be -9.

4) (i) If *a* -9, *b* -9, and *c* -9, *a* for each animal is set to 1.0, *b* to 0.0, and *c*

to 0.0.

(ii) If *d* -9, *d* for each animal is set to 1.0.

(iii) If *e* -9, *e* for each animal is set to 1.0.

5) ~~All component observations (1 …~~ *~~nobs~~*~~) that make up a combined observation must be~~

~~realised by an animal for the combined observation to be realised by the animal.~~ CHECK!!!!!!!!!

6) DYD???

## **&OBSERVATIONCONSTRAINTS**

constraints=*a b c d e* *sign* *y* /

NB! Number of lines in constraints must equal *nObsConstraints*

\*\*Task\*\*

Input for imposing constraints on sampled observations, where a constraint is a restriction by which observations must comply. Each constraint is imposed on the *j*th animal as (*aj*+*bj*-*cj*)\**dj*/*ej* *sign* *y*, where *aj*, *bj*, *cj*, *dj*, and *ej* are phenotypic observations generated for animal *j*, the observation numbers allocated to *a*, *b*, *c*, *d*, and *e* can be 1 … *nobs*+*nCombinedObs*, and *sign* is > or <. When an observation(s) for animal *j* does not fulfil all constraints, all observations for that animal are resampled.

\*\*Properties of names\*\*

*a* Definition: Observation allocated to *a*

Type: Integer

Options:

-9 No observation is allocated to *a*

1:*nobs*+*nCombinedObs* Observation allocated to *a*

Default: *a* must be specified

*b* Definition: Observation allocated to *b*

Type: Integer

Options:

-9 No observation is allocated to *b*

1:*nobs*+*nCombinedObs* Observation allocated to *b*

Default: *b* must be specified

*c* Definition: Observation allocated to *c*

Type: Integer

Options:

-9 No observation is allocated to *c*

1:*nobs*+*nCombinedObs* Observation allocated to *c*

Default: *c* must be specified

*d* Definition: Observation allocated to *d*

Type: Integer

Options:

-9 No observation is allocated to *d*

1:*nobs*+*nCombinedObs* Observation allocated to *d*

Default: *d* must be specified

*e* Definition: Observation allocated to *e*

Type: Integer

Options:

-9 No observation is allocated to *e*

1:*nobs*+*nCombinedObs* Observation allocated to *e*

Default: *e* must be specified

*sign* Definition: Equation sign

Type: Character

Options:

< Less than

> Greater than

Default: *sign* must be specified

*y* Definition: Equation sign

Type: Real

Options: Any real number

Default: *y* must be specified

\*\*Additional information\*\*

1) Namelist &OBSERVATIONCONSTRAINTS is only read and used when *nObsConstraints*>0 in

namelist &POPULATIONPARAMETERS

2) At least one of *a*, *b*, *c*, *d*, and *e* must be allocated an observation number

(1:*nobs*)for each combined observation. That is, all of *a*, *b*, *c*, *d*, and *e* cannot

be -9.

3) (i) If *a* -9, *b* -9, and *c* -9, *a* for each animal is set to 1.0, *b* to 0.0, and *c*

to 0.0.

(ii) If *d* -9, *d* for each animal is set to 1.0.

(iii) If *e* -9, *e* for each animal is set to 1.0.

4) When an observation(s) for an animaldoes not fulfil all constraints, all

observations for the animal are resampled by resampling the residual effects for

all observations.

5) If the observation(s) does not fulfil all constraints after 100 rounds of

resampling, the program stops. This implies that the constraints are too strict.

## **&ECONOMICVALUES**

economicValueTbv=*economicValueTbv*

nEconomicValuePbv*=nEconomicValuePbv*

economicValuePbv=*stage* *sex* *values*

nEconomicValueEbv=*nEconomicValueEbv*

economicValueEbv=*stage* *sex* *values* /

NB! Number of lines in *economicValuePbv* and *economicValueEbv* must equal *nEconomicValuePbv* and *nEconomicValueEbv*

\*\*Task\*\*

Provide economic values

\*\*Properties of names\*\*

*economicValueTbv* Definition: Vector of *ntbv* economic values used to calculate

true aggregate-breeding value. The aggregate-

breeding value is calculated by weighting *ntbv*

true-breeding values by *economicValueTbv*.

Dimension: *ntbv*

Type: Real

Options: Any reals

Default: *economicValueTbv* must be specified

*nEconomicValuePbv* Definition: Number of selection stages using pseudo-true

aggregate-breeding values for which economic

values are provided in *economicValuePbv*

Type: Integer

Options:

-9 Economic values provided in *economicValueTbv* applied to

all selection stages using pseudo-true aggregate-

breeding values. No information is provided in

*economicValuePbv*.

1 Provide one row of input in *economicValuePbv*. These

economic values are used in all selection stages where

estimated pseudo-true aggregate-breeding values are used

as selection criterion. The row of e*conomicValuePbv* must

be must have *stage* -9 and *sex* 0.

1≤*nEconomicValuePbv*≤*selection\_groups*

Provide economic values for all stages and sexes of

*economicValuePbv* where estimated pseudo-true aggregate-

breeding values are used as selection criterion. The

number of rows provided must equal *nEconomicValuePbv*.

Default: -9

economicValuePbv=

*stage* Definition: Stage of selection. Must correspond with a

selection stage using pseudo-true aggregate-

breeding value (‘tbv’) as selection criterion

Type: Integer

Options:

-9 Economic values applied to all selection stages; only

applies when *nEconomicValuePbv* 1

1≤*stage*≤*selection\_groups*

Default: *stage* must be specified when economicValuePbv

provided

*sex* Definition: Sex of selection candidates

Type: Integer

Options:

0 Sex ignored, economic values applied to both sexes

1 Males

2 Females

Default: *sex* must be specified when economicValuePbv

provided

*values* Definition: Vector of *ntbv* economic values used to calculate

pseudo-true aggregate-breeding value. The pseudo-

true aggregate-breeding value is calculated by

weighting *ntbv* true-breeding values by *values*.

Dimension: *ntbv*

Type: Real

Options: Any reals

Default: *values* must be specified when economicValuePbv

provided

*nEconomicValueEbv* Definition: Number of selection stages using estimated

aggregate-breeding values for which economic

values are provided in *economicValueEbv*

Type: Integer

Options:

-9 Economic values provided in *economicValueTbv* applied to

all selection stages using estimated aggregate-breeding

values. *nebv* must equal *ntbv* and no information is

provided in *economicValueEbv*.

1 Provide one row of input in *economicValueEbv*. These

economic values are used in all selection stages where

estimated aggregate-breeding values are used as

selection criterion. The row of e*conomicValueEbv* must be

must have *stage* -9 and *sex* 0.

1≤*nEconomicValueEbv*≤*selection\_groups*

Provide economic values for all stages and sexes of

*economicValueEbv* where estimated aggregate-breeding

values are used as selection criterion. The number of

rows provided must equal *nEconomicValueEbv*.

Default: -9

economicValueEbv=

*stage* Definition: Stage of selection. Must correspond with a

selection stage using estimated aggregate-breeding

value as selection criterion

Type: Integer

Options:

-9 Economic values applied to all selection stages; only

applies when *nEconomicValueEbv* 1

1≤*stage*≤*selection\_groups*

Default: *stage* must be specified when *nEconomicValueEbv*

provided

*sex* Definition: Sex of selection candidates

Type: Integer

Options:

0 Sex ignored, economic values applied to both sexes

1 Males

2 Females

Default: *sex* must be specified when *nEconomicValueEbv*

provided

*values* Definition: Vector of *nebv* economic values used to calculate

estimated aggregate-breeding value. The estimated

aggregate-breeding value is calculated by

weighting *nebv* selection criteria by *values*.

Dimension: *nebv*

Type: Real

Options: Any reals

Default: *values* must be specified when *nEconomicValueEbv*

provided

\*\*Additional information\*\*

1) Namelist &ECONOMICVALUES is always read and variable *economicValueTbv* is always

used. Variables *economicValuePbv* and *economicValueEbv* are only used when there

are selection stage(s) using pseudo-true and estimated aggregate-breeding values

as selection criterion.

2) Economic values can be allocated to *economicValuePbv* and *economicValueEbv* in

three ways:

(i) No input provided for *economicValuePbv* and *economicValueEbv*. The economic

values provided in *economicValueTbv* are allocated to all selection stages of

*economicValuePbv* and *economicValueEbv* where pseudo-true and estimated

aggregate-breeding values are used as selection criterion. *nEconomicValuePbv*

and *nEconomicValueEbv* must be -9, the default value. Using this option for

*economicValueEbv* requires *ntbv* equal to *nebv*.

(ii) Provide one row of input that is allocated to all selection stages of

*economicValuePbv* and *economicValueEbv*, where pseudo-true and estimated

aggregate-breeding values are used as selection criterion. *nEconomicValuePbv*

and *nEconomicValueEbv* must be 1, and single row in *economicValuePbv* and

*economicValueEbv* must have *stage* -9 and *sex* 0.

(iii) Provide economic values for all stages and sexes of *economicValuePbv* and

*economicValueEbv* where pseudo-true and estimated aggregate-breeding values

are used as selection criterion. The number of rows provided must equal

*nEconomicValuePbv* and *nEconomicValueEbv*. When economic values for all stages

and sexes are provided in *economicValuePbv* and *economicValueEbv*, *sex* can be:

0, 1, or 2 for *sex\_code* 0, 7, or 9 (family selection), where *sex\_code* is

provided in namelist &SELECTION, variable *selection\_groups* When *sex* 0, no

other rows of *economicValuePbv* or *economicValueEbv* can have this *stage*. When

sex 1, there must be another row with same *stage* and *sex* 2. When *sex* 2, there

must be another row with same *stage* and *sex* 1.

1 for *sex\_code* 1, 3, or 5. No other rows of *economicValuePbv* or

*economicValueEbv* can have this *stage*.

2 for *sex\_code* 2, 4, or 6. No other rows of *economicValuePbv* or

*economicValueEbv* can have this *stage*.

3) Selection stages using pseudo-true aggregate-breeding value as selection

criterion are stages of namelist &SELECTION, variable *selection\_groups* with (i)

*sex\_code* 0:6 and 9 and *selection\_criterion* ‘tbv’, and (ii) *sex\_code* 7 and

*MaleSelCrit* or *FemaleSelCrit* ‘tbv’ in corresponding EVA-selection stage of

namelist &EVA, variable *EvaSelection*. Selection stages using estimated aggregate-

breeding value as selection criterion are stages with *selection\_criterion*,

*MaleSelCrit*, or *FemaleSelCrit* ‘polyblup’, ‘genomicblup’, ‘ibdblup’, ‘gas’, or

‘bayesp’.

## **&OBSERVATIONS**

nMaleObs=*nMaleObs*

nFemaleObs=*nFemaleObs*

nGroupObs=*nGroupObs*

nDydObs=*nDydObs*

nCategoricalObs=*nCategoricalObs*

nPseudoCategoricalObs=*nPseudoCategoricalObs*

MaleObservations=

*obs RealisedSelectionStage FirstPop LastPop FirstHerd LastHerd alive age selection*

*CullingUnselected CullingOldAge CullingInvoluntary*

FemaleObservations=

*obs RealisedSelectionStage FirstPop LastPop FirstHerd LastHerd alive age selection CullingUnselected CullingOldAge CullingInvoluntary parity parityTime*

progenyObservations=

*obs realisedSelectionStage sex FirstPop LastPop firstHerd lastHerd alive age*

*progenyObs progenySex progenyFirstPop progenyLastPop progenyFirstHerd progenyLastHerd progenyAlive progenyAgeAtObs progenyObsStatus a b obsType mateLitter*

*[“obs realisedSelectionStage sex firstHerd lastHerd alive age” refers to sire or dam for which progeny obs is realised; progenyObs progenySex … realised obsType refers to progeny of sire or dam]*

groupObservations=

*obs realisedSelectionStage FirstPop LastPop firstHerd lastHerd alive age obsType*

DydObservations=

*obs RealisedSelectionStage nOffspring TimeLag AgeMates*

categoricalObservations=*categoricalObservations*

pseudoCategoricalObservations=*pseudoCategoricalObservations* /

NB! Number of observations for MaleObservations, FemaleObservations,

groupObservations, DydObservations, categoricalObservations, and

pseudoCategoricalObservations must equal *nMaleObs*, *nFemaleObs*, *nGroupObs*,

*nDydObs*, *nCategoricalObs*, and *nPseudoCategoricalObs*

When *nMaleObs*, *nFemaleObs*, *nGroupObs*, *nDydObs*, *nCategoricalObs*, and/or

*nPseudoCategoricalObs* are 0, MaleObservations, FemaleObservations,

groupObservations, DydObservations, categoricalObservations, and/or

pseudoCategoricalObservations are not required

if bisexualspecies=yes: observations need to specified in both MaleObservations & FemaleObservations

\*\*Task\*\*

Input for realising observations

\*\*Properties of names\*\*

*nMaleObs* Definition: Number of observations recorded for males

Type: Integer

Options: *nMaleObs*≥0

Default: 0

*nFemaleObs* Definition: Number of observations recorded for females

Type: Integer

Options: *nFemaleObs*≥0

Default: 0

*nGroupObs* Definition: Number of observations recorded on groups

Type: Integer

Options: *nGroupObs*≥0

Default: 0

*nDydObs* Definition: Number of DYD observations for males

Type: Integer

Options: *nDydObs*≥0

Default: 0

*nCategoricalObs* Definition: Number of categorical observations

Type: Integer

Options: 0≤*nCategoricalObs*≤*nobs*

Default: 0

*nPseudoCategoricalObs* Definition: Number of pseudo-categorical observations

Type: Integer

Options: 0≤*nPseudoCategoricalObs*≤*nobs*

Default: 0

*obs* Definition: Observation number

Type: Integer

Options: 1≤*obs*≤*nobs*+*nCombinedObs*

Default: *obs* must be specified

*RealisedSelectionStage* Definition: Specifiy when an observation is realised

Type: Integer

Options:

1≤RealisedSelectionStage≤selection\_groups

Selection stage at which an observation is realised

-9 Observation is realised at the start of each time step

or as a maternal trait. This is not an option for DYD.

Default: *RealisedSelectionStage* must be specified

Maternal traits can only be realised as maternal traits; they cannot be realised at the start of each time step and via selection. Maternal traits are defined by specifying *parity* and *parityTime* (i.e., not equal to -9).

At moment, can only handle one record of the *i*th maternal observation per female per parity. This suits livestock, where female mated to one male. Not so with fish, where each female could be mated with several males and, thereby, have several repeated records of the same maternal observation within parities.

*FirstPop* Definition: First pop within which animals are

candidates to have observation *obs* realised

Type: Integer

Options:

1≤*First*pop ≤*n*pop First pop number

-9 pop not considered

Default: *FirstHerd* must be specified

*LastPop* Definition: Last pop within which animals are

candidates to have observation *obs* realised

Type: Integer

Options:

1≤ *LastPop* ≤*n*pop Last pop number

-9 pop not considered

Default: *LastPop* must be specified

*FirstHerd* Definition: First herd within which animals are

candidates to have observation *obs* realised

Type: Integer

Options:

1≤*FirstHerd*≤*nherd* First herd number

-9 Herd not considered

Default: *FirstHerd* must be specified

*LastHerd* Definition: Last herd within which animals are

candidates to have observation *obs* realised

Type: Integer

Options:

1≤*LastHerd*≤*nherd* Last herd number

-9 Herd not considered

Default: *LastHerd* must be specified

*alive* Definition: Live status of candidates to have

observation *obs* realised

Type: Integer

Options:

1 Animal needs to be alive

-9 Live status not considered

Default: *alive* must be specified

*age* Definition: Age which animals are candidates to have

observation *obs* realised

Type: Integer

Options:

Males

*age*≥1 when *gestation\_length* 0, *alive* -9

*age*≥0 when *gestation\_length*>0, *alive* -9

1≤*age*<*OldAgeMales* when *gestation\_length* 0, *alive* 1

0≤*age*<*OldAgeMales* when *gestation\_length*>0, *alive* 1

Females

*age*≥1 when *gestation\_length* 0, *alive* -9

*age*≥0 when *gestation\_length*>0, *alive* -9

1≤*age*<*OldAgeFemales* when *gestation\_length* 0, *alive* 1

0≤*age*<*OldAgeFemales* when *gestation\_length*>0, *alive* 1

GROUPOBS

-9 Age not considered

Default: *age* must be specified

*selection* Definition: Age at which an animal needs to be selected

to reproduce for observation *obs* to be

realised ~~at age~~ *~~age~~*. Animals must produce

offspring for observation *obs* to be realised.

Type: Integer

Options:

Males

*selection*≥1 when *gestation\_length* 0, *alive* -9

*selection*≥0 when *gestation\_length*>0, *alive* -9

1≤*selection*<*OldAgeMales* when *gestation\_length* 0, *alive* 1

0≤*selection*<*OldAgeMales* when *gestation\_length*>0, *alive* 1

Females

*selection*≥1 when *gestation\_length* 0, *alive* -9

*selection*≥0 when *gestation\_length*>0, *alive* -9

1≤*selection*<*OldAgeFemales* when *gestation\_length* 0

0≤*selection*<*OldAgeFemales* when *gestation\_length*>0

-1 Observation *obs* available to be realised after selected

to reproduce for first time regardless of selection age

-9 Selection not considered

Default: *selection* must be specified

*obs* is realised in the time step following selection. For example, selected at time step *t*, *obs* realised at time *t*+1, after culling, but before updateReproductiveCycles and selection. A direct consequence of this is that when *selection* used with *age* (*age*≠-9) and *selection*≥0, *selection* must be less than *age*; *age*>*selection*. This is because the animals are one time step older than *selection* when observation is realised.

With -1, “available to be” and “for first time” as depends on *age*; *selection* -1 can be used with *age*.

*selection* can be used with *FirstHerd, LastHerd, alive, and age*

*CullingUnselected* Definition: Age at which an animal needs to be culled

because it was not selected to have observation

*obs* realised

Type: Integer

Options:

Males

1≤*CullingUnselected*<*OldAgeMales* when *gestation\_length* 0

0≤*CullingUnselected*<*OldAgeMales* when *gestation\_length*>0

Females

1≤*CullingUnselected*<*OldAgeFemales* when *gestation\_length* 0

0≤*CullingUnselected*<*OldAgeFemales* when *gestation\_length*>0

-1 Observation *obs* realised after culling regardless of culling

age

-9 Culling age not considered

Default: *CullingUnselected* must be specified

If *CullingUnselected*≠-9, *alive* and *age* must be -9

*CullingUnselected*<*OldAgeMales* and *OldAgeFemales* because animals must be alive to be culled and declared dead

*obs* is realised in the time step following selection. For example, animal culled for not being selected at time step *t*; age at time *t* becomes the animal’s culling age. *obs* for animal realised at time *t*+1, after culling, but before updateReproductiveCycles and selection, when *CullingUnselected* aligned with age at time *t*.

*CullingUnselected* can be used with *FirstHerd and LastHerd*

*CullingOldAge* Definition: Animal has observation *obs* realised when culled

for old age

Type: Integer

Options:

-1 Observation realised after culled for old age regardless

of culling age

-9 Culling not considered

Default: *CullingOldAge* must be specified

If *CullingOldAge*≠-9, *alive* and *age* must be -9

With *CullingOldAge* -1, “regardless of culling age” required because females in gestation or daysOpen cannot be culled for old age; they are culled for old age at a later age.

*obs* is realised in the same time step as culled; realised following culling for old age in time step *t*.

*CullingOldAge* can be used with *FirstHerd and LastHerd*

*CullingInvoluntary* Definition: Age at which an animal needs to be culled

involuntarily to have observation *obs* realised

Type: Integer

Options:

Males

1≤*CullingInvoluntary*<*OldAgeMales* when *gestation\_length* 0

0≤*CullingInvoluntary*<*OldAgeMales* when *gestation\_length*>0

Females

1≤*CullingInvoluntary*<OldAgeFemales when *gestation\_length* 0

0≤*CullingInvoluntary*<OldAgeFemales when *gestation\_length*>0

-1 Observation *obs* realised after culled involuntarily

regardless of culling age

-9 Culling not considered

Default: *CullingInvoluntary* must be specified

If *CullingInvoluntary*≠-9, *alive* and *age* must be -9

*obs* is realised in the same time step as culled; realised following culling involuntarily in time step *t*

*CullingInvoluntary* <*OldAgeMales* and *OldAgeFemales* because animals must be alive to be culled and declared dead

*CullingInvoluntary* can be used with *FirstHerd and LastHerd*

*parity* Definition: Parity during which an observation is realised

for females that have been selected to reproduce

Type: Integer

Options:

1≤*parity*≤1+MaxAge-MinAge when *gestation\_length* 0, where

MaxAge and MinAge are the maximum and minimum

reproductive ages for living females, derived from input

provided in namelist selection, variable

*selection\_scheme*

1≤*parity*≤1+floor((*MaxAge*-*MinAge*)/(*gestation\_length*+*daysOpen*))

when *gestation\_length*>0

-9 Parity not considered

Default: *parity* must be specified

Females must be alive; *alive* must be 1. Maternal traits are not released for females that are culled after selection for mating and before maternal traits are realised.

Repeated observation numbers must have the same *parity*

Observations with *parity*≠-9, cannot be provided in *MaleObservations* and *DydObservations*

*parity* can be used with *FirstHerd*, *LastHerd*, *age*, and *parityTime*

Mate effects can only be specified for maternal observations; rows of *ZMateGenetic* and *WMateError* that are associated with maternal observations.

*parityTime* Definition: Time of parity at which an observation is

realised for females that have been selected to

reproduce. Parity time is the period from mating

that includes gestation and days open.

Type: Integer

Options:

0 when *gestation\_length*+*daysOpen*=0

0≤*parityTime*<*gestation\_length*+*daysOpen* when

*gestation\_length*+*daysOpen*>0

-9 *ParityTime* not considered

Default: *parityTime* must be specified

*parityTime* can be used with *parity*; *parityTime*≠-9 when parity -9

*obsType* Definition: Observation type; individual or group observations

Type: Integer

Options:

1 Individual observations; animals in selected groups have

individual observations realised

2 Group observations; selected groups have group

observations realised

Default: *obsType* must be specified

*nOffspring* Definition: Number of offspring in progeny group

Type: Integer

Options: ≥1

Default: *noffspring* must be specified

*TimeLag* Definition: Time lag (in time steps) from selection

until the DYD is realised

Type: Integer

Options: 0≤*TimeLag*≤*ntime*

Default: *TimeLag* must be specified

*AgeMates* Definition: Age of mates when offspring are born.

Type: Integer

Options: 1≤*AgeMates*≤*ntime*, where 1 means that mates

are one time step older than offspring.

Default: *AgeMates* must be specified

*~~ObservationTypes~~* ~~Definition: Age of mates when offspring are born.~~

~~Type: Character~~

~~Options:~~

~~normal Normally-distributed trait~~

~~categorical Categorical trait~~

~~Default: normal~~

*categoricalObservations* Definition: List of categorical observations

Type: Integer

Options: 1≤*categoricalObservations*≤*nobs*

Default: categoricalObservations must be specified

*pseudoCategoricalObservations*

Definition: List of pseudo-categorical observations

Type: Integer

Options: 1≤*pseudoCategoricalObservations*≤*nobs*

Default: *pseudoCategoricalObservations* must be specified

obs in categoricalObservations cannot be in pseudoCategoricalObservations

\*\*Relation to subsequent namelists\*\*

&CATEGORICALS

Namelist &CATEGORICALS is required when *nCategoricalObs*>0 in namelist

&OBSERVATIONS

\*\*Additional information\*\*

1) At least one observation must be realised. That is, at least one observation in

*nMaleObs*, *nFemaleObs*, *nGroupObs*, and/or *nDydObs* must be greater than zero. Not all

observations need to be realised.

2) Male observations in MaleObservations can be realised in two ways. First, at

the start of each time step (*RealisedSelectionStage* -9 in namelist

&OBSERVATIONS, variable *MaleObservations*). Second, via selection (*destiny\_sel* 8

in namelist SELECTION, variable *selection\_scheme* or *MaleDestinySel* 8 in

namelist EVA, variable *EvaSelection*).

3) Female observations in FemaleObservations can be realised in three ways. First,

at the start of each time step (*RealisedSelectionStage* -9 in namelist

&OBSERVATIONS, variable *FemaleObservations*). Second, via selection (*destiny\_sel*

8 in namelist SELECTION, variable *selection\_scheme* or *FemaleDestinySel* 8 in

namelist EVA, variable *EvaSelection*). Third, maternal traits for females in

gestation or days open. Maternal traits are realised after offspring are

sampled/born. Females in gestation or days open that fulfil the criteria specified

in *FemaleObservations*, including *parity* and *parityTime*, have traits realised.

Note: Maternal traits can only be realised as maternal traits; they cannot be realised at the start of each time step and via selection. Maternal traits are defined as having having *parity* and *parityTime* specified (i.e., not equal to -9).

4) When RealisedSelectionStage≥1, FirstHerd, LastHerd, alive, age, selection,

CullingUnselected, CullingOldAge, CullingInvoluntary, parity, and parityTime

Must be -9.

5) When *FirstHerd* -9, *LastHerd* must be -9. When *LastHerd* -9, *FirstHerd* must be -9.

6) When *alive* 1 in MaleObservations, *age* must be less than *OldAgeMales*. When *live*

1 in FemaleObservations, *age* must be less than *OldAgeFemales*.

7) When *selection*≥0, *age* must be greater than *selection*. *alive* can be 1 or -9.

8) (a) When *CullingUnselected*≠-9, *age* must be -9. When *age*≠-9, *CullingUnselected*

must be -9.

(b) When *CullingUnselected*≠-9, *alive* must be -9. When *alive*≠-9,

*CullingUnselected* must be -9.

9) (a) When *CullingOldAge*≠-9, *age* must be -9. When *age*≠-9, *CullingOldAge* must be

–9.

(b) When *CullingOldAge*≠-9, *alive* must be -9. When *alive*≠-9, *CullingOldAge*

must be -9.

10) (a) When *CullingInvoluntary*≠-9, *age* must be -9. When *age*≠-9, *CullingInvoluntary*

must be -9.

(b) When *CullingInvoluntary*≠-9, *alive* must be -9. When *alive*≠-9,

*CullingInvoluntary* must be -9.

11) (a) The largest parity derived from the input provided in namelist selection,

variable *selection\_scheme* is the largest possible parity a female can

obtain. It does not necessarily mean that females obtain this parity. All

females may, for example, be culled before they obtain this or earlier

parities. In these cases, some obervations will not be realised.

(b) When *parity*≠-9, *alive* must be 1. When alive -9, *parity* must be -9.

(c) When *parity*≠-9, *age*≠-9, *FirstHerd*≠-9, and *LastHerd*≠-9 are valid options

12) (a) When *parity*≠-9, *parityTime* ~~must not~~ can be -9. When *parity* -9, *parityTime* must

be -9.

(b) When *gestation\_length*+*daysOpen*=0, maternal traits can only be realised in

the same time step in which females were mated

(c) When *gestation\_length*+*daysOpen*>0, maternal traits can be realised in the

same time step in which females were mated and in the next

*gestation\_length*+*daysOpen*-1 time steps. This implies that when

*gestation\_length*+*daysOpen*=1, maternal traits can only be realised in the

same time step in which the females were mated. Maternal taits are realised

after mating and have no influence in the current time step, and are first

used in the next time step.

13) (a) Only one of the following options can be set to a value other than -9 for

each observation in MaleObservations: *selection*, *CullingUnselected*,

CullingOldAge, and CullingInvoluntary.

(b) Only one of the following options can be set to a value other than -9 for

each observation in FemaleObservations: *selection*, *CullingUnselected*,

CullingOldAge, CullingInvoluntary, and parity.

14) DYDs cannot be sampled for traits influenced by maternal effects, whether

genetic or environmental

!gm DYD must be realised via selection (*destiny\_sel* 8 in namelist SELECTION, variable *selection\_scheme* or *FemaleDestinySel* 8 in namelist EVA, variable *EvaSelection*). They cannot be realised at the start of each time step (*RealisedSelectionStage* -9 in namelist &OBSERVATIONS, variable *FemaleObservations*).

) *Obs* cannot for combined traits in DydObservations

15) Observations not specified in MaleObservations, FemaleObservations, or

DydObservations are not realised

16) The same combination of *obs* and *RealisedSelectionStage* can only be represented

once in MaleObservations and FemaleObservations

17) Females in gestation or days open can have any traits realised

18) (a) (i) When *sex\_code* 0 and *destiny\_sel* 8 [observation(s) realised] at

selection stage *stage* in namelist &SELECTION, variable

selection\_scheme, *RealisedSelectionStage* must be *stage* for at least

one observation in MaleObservations and/or FemaleObservations

(ii) When *sex\_code* 0 and *destiny\_sel* not 8 at selection stage *stage*,

*RealisedSelectionStage* must not be *stage* for any observation in

MaleObservations and FemaleObservations

(b) (i) When *sex\_code* 1, 3, or 5 and *destiny\_sel* 8 at selection stage *stage*,

*RealisedSelectionStage* must be *stage* for at least one observation in

MaleObservations

(ii) When *sex\_code* 1, 3, or 5 and *destiny\_sel* not 8 at selection stage

*stage*, *RealisedSelectionStage* must not be *stage* for any observation in

MaleObservations

(c) (i) When *sex\_code* 2, 4, or 5 and *destiny\_sel* 8 at selection stage *stage*,

*RealisedSelectionStage* must be *stage* for at least one observation in

FemaleObservations

(ii) When *sex\_code* 2, 4, or 5 and *destiny\_sel* not 8 at selection stage

*stage*, *RealisedSelectionStage* must not be *stage* for any observation in

FemaleObservations

(d) (i) When *sex\_code* 7 at selection stage *stage* and *MaleDestinySel* 8 in

corresponding EVA-selection stage, *RealisedSelectionStage* must be

*stage* for at least one observation in MaleObservations

(ii) When *sex\_code* 7 at selection stage *stage* and *FemaleDestinySel* 8 in

corresponding EVA-selection stage, *RealisedSelectionStage* must be

*stage* for at least one observation in FemaleObservations

(iii) When *sex\_code* 7 at selection stage *stage* and *MaleDestinySel* not 8 in

corresponding EVA-selection stage, *RealisedSelectionStage* must not be

*stage* for any observation in MaleObservations

(iv) When *sex\_code* 7 at selection stage *stage* and *FemaleDestinySel* not 8 in

corresponding EVA-selection stage, *RealisedSelectionStage* must not be

*stage* for any observation in FemaleObservations

19) Observation only recorded once. The first observation. If already recorded, cannot be overwritten when animal falls into phenotyping category again.

Ja, R-matricen er uændret og gælder for en enkelt observation, både i input.prm og i .parm. Vægten bliver beregnet i ADAM og skrevet til dmudat og .dir fortæller DMU, at der skal bruges vægte for forskellige observationer. Hvis alle dyr med den pågældende fænotype havde DYD'er, så kunne man også gøre det i R-matricen, men hvis der både er DYD'er og enkelte fænotyper, så er man nødt til at bruge vægten i DMU.

DYD can only be realised via destiny\_sel 6 in &selection and &eva.

## **&CATEGORICALS**

nCategories=*nCategories*

proportions=*proportions*

firstCategories=*firstCategories* /

\*\*Task\*\*

Information specific to categorical traits

\*\*Properties of names\*\*

*nCategories* Definition: Number of categories for each categorical trait

Type: Integer

Dimension: *nCategoricalObs*

Options: ≥2

Default: *nCategories* must be specified

*proportions* Definition: Proportion of observations that fall within each

category interval

Type: Real

Dimension: *nCategoricalObs\*max(nCategories)*

Options:

0.0<*proportions*<1.0 Proportion for each category

-9.0 Otherwise

Default: *proportions* must be specified

*firstCategories* Definition: Array with value of first category for each

categorical trait

Type: Integer

Dimension: *nCategoricalObs*

Options: Any integer

Default: *firstCategories* must be specified

\*\*Additional information\*\*

1) Proportions provided in *proportions* must sum to 1.0 for each categorical trait

2) Unused elements in *proportions* must be -9.0

## **&PSEUDOCATEGORICALS**

minPseudoCategories=*minPseudoCategories*

maxPseudoCategories=*maxPseudoCategories* /

\*\*Task\*\*

Information specific to pseudo-categorical traits

\*\*Properties of names\*\*

*minPseudoCategories* Definition: Array with minimum category for each pseudo-

categorical trait

Type: Integer

Dimension: *nPseudoCategoricalObs*

Options: Any integer

Default: *minPseudoCategories* must be specified

*maxPseudoCategories* Definition: Array with maximum category for each pseudo-

categorical trait

Type: Integer

Dimension: *nPseudoCategoricalObs*

Options: Any integer

Default: *maxPseudoCategories* must be specified

\*\*Additional information\*\*

* Element *i* of *maxPseudoCategories* must be greater than the *i*th element of

*minPseudoCategories*

## **&REPORT**

nReportGroups=*nReportGroups*

reportGroupPopHerd=*reportGroupPopHerd*

reportSelectionTimes=*reportSelectionTimes*

reportSelectionStages=*reportSelectionStages*

reportHomozygosityTimes=*reportHomozygosityTimes*

nGenomicInbreeding=*nGenomicInbreeding*

genomicInbreedingParameters=*stage* *centre* *scale* *weight* *firstChrom* *lastChrom*

*loci* *scaleGToA* *propAToG* *addDiagG* *diagGOne*

genomicInbreedingCentre=*stage* *genomicBase* *maf* *mafInclude*

genomicInbreedingScale=*stage* *genomicBase* *maf* *mafInclude*

genomicInbreedingWeight=*stage* *genomicBase* *maf* *mafInclude*

printGenomicHomozygosity=*printGenomicHomozygosity*

printCategoricalProportions=*printCategoricalProportions*

printPopulationBreedingValues=*printPopulationBreedingValues*

printSimulatedData=*printSimulatedData*

printGenotypeTimes=*printGenotypeTimes*

printBiasAccuracy=*printBiasAccuracy*

printBreedingValues=*printBreedingValues*

printCoVariances=*printCoVariances*

printEvaRelationships=*printEvaRelationships*

genomicRelationshipParameters=*stage centre scale weight firstChrom lastChrom*

*loci scaleGToA propAToG addDiagG diagGOne*

genomicRelationshipCentre=*stage genomicBase* *maf mafInclude*

genomicRelationshipScale=*stage genomicBase* *maf mafInclude*

genomicRelationshipWeight=*stage genomicBase* *maf mafInclude*

ibsRelationshipParameters*=stage firstChrom lastChrom loci*

ibdRelationshipParameters*=stage firstChrom lastChrom loci*

printPedigreeRelationships=*printPedigreeRelationships*

printGenomicRelationship=*printGenomicRelationships*

printLdHayesFiles=*printLdHayesFiles*

printLdMutations=*printLdMutations*

debugOutput=*debugOutput* /

\*\*Task\*\*

Controls generation of output files. Output files are written to the output

directory specified in namelist &OUTPUTDIRECTORY, variable *OutDirectory*.

\*\*Properties of names\*\*

*nReportGroups* Definition: Number of report groups; applies to genetic-

trend and genomic-homozygosity files. Report groups

defined by herds with first and last herds in each

group provided in *firstReportHerds* and

*lastReportHerds.*

Type: Integer

Options: *nReportGroups*≥1

Default: 1

*reportGroupPopHerd* Definition: Array specify pop and herd, identity

in each report group

Type: Integer

Dimension: 6\**nReportGroups*

Options: ≥1

Default: 1 npop 1 nherd 0 10

*reportSelectionTimes* Definition: Array with time steps; applies to files generated

during selection

Type: Integer

Dimension: 1≤*nTimes*≤*ntime,* where *nTimes* is the number of time

steps for which output variables are written

Options: 1≤*reportSelectionTimesi*≤*ntime* (*i*=1,…,*nTimes*), where

*reportSelectionTimesi* is the *i*th integer of

*reportSelectionTimes* and

*reportSelectionTimesi*≠*reportSelectionTimesj* (i≠j)

Default: 1 … *ntime*

*reportSelectionStages* Definition: Array with selection stages; applies to files

generated during selection

Type: Integer

Dimension: 1≤*nStages*≤*selection\_groups,* where *nStages* is the

number of selection stages for which output

variables are to be written

Options: 1≤*reportSelectionStagesi*≤*selection\_groups*

(*i*=1,…,*nStages*), where *reportSelectionStagesi* is the

*i*th integer of *reportSelectionStages* and

*reportSelectionStagesi*≠*reportSelectionStagesj* (i≠j)

Default: 1 … *selection\_groups*

*reportHomozygosityTimes*

Definition: Array with time steps; applies to genomic-

homozygosity files

Type: Integer

Dimension: 1≤*nTimes*≤*ntime*+1*,* where *nTimes* is the number of time

steps for which output variables are written

Options: 0≤*reportHomozygosityTimesi*≤*ntime* (*i*=1,…,*nTimes*+1),

where *reportHomozygosityTimesi* is the *i*th integer of

*reportHomozygosityTimes* and

*reportHomozygosityTimesi*≠*reportHomozygosityTimesj*

(i≠j)

Default: 0 … *ntime*

*nGenomicInbreeding* Definition: Number of genomic-inbreeding coefficients to

calculate

Type: Integer

Options: *nGenomicF*≥0

Default: 0

Parameters of variables genomicInbreedingParameters, genomicInbreedingCentre,

genomicInbreedingScale, and genomicInbreedingWeight are as defined in namelist

**&GENOMICBLUPPARAMETERS:** *stage*, *centre*, *scale*, *weight*, *firstChrom*, *lastChrom*, *loci*,

*scaleGToA*, *propAToG*, *addDiagG*, *diagGOne*, *genomicBase*, *maf*, and *mafInclude*.

Description of variables as described for &GENOMICBLUPPARAMETERS with the exceptions

that:

*stage* in genomicInbreedingParameters must be aligned with *stages* in

genomicInbreedingCentre, genomicInbreedingScale, and genomicInbreedingWeight

Number of lines of parameters in genomicInbreedingParameters must equal number of

lines in genomicInbreedingCentre, genomicInbreedingScale, and genomicInbreedingWeight

*scaleGToA* can be ‘no’, ‘all’, or ‘base’

*scaleGToA* ‘traced’ options in *genomicBase* are not available as they are

also redundant. With ‘genomic-inbreeding’, all animals are by definition traced

(i.e., included in the pedigree) because we calculate genomic-inbreeding coefficients

for all animals in the population.

Input to construct genomic-**H** matrices using the program, invgmatrix. Genomic-**H**

matrices are used by EVA to constraint average relationship. The namelist is read

when *relationshipMatrix* ‘genomic’ in any EVA-selection stage of namelist &EVA,

variable *EvaSelection*.

*printGenomicHomozygosity*

Definition: Write genomic homozygosity at QTL, marker,

template-QTL, and template-marker loci to

genomic-homozygosity files

Type: Character

Options:

no Genomic homozygosity is not written to output files

yes Genomic homozygosity is written to output files

Default: no

*printCategoricalProportions*

Definition: Write categorical proportions to categorical files

Type: Character

Options:

no Categorical proportions is not written to output files

yes Categorical proportions is written to output files

Default: no

printPopulationBreedingValues

Definition: Write true and estimated breeding values of all

animals in the population to output file,

‘populationBreedingValuesRep<*rep*>.res’, where <*rep*>

is replicate number. Files are compressed by the

linux command,

‘bzip2 –f populationBreedingValuesRep*<rep>*.res’.

Type: Character

Options:

no Breeding values are not written to output files

yes Breeding values are written to output files

Default: no

*printSimulatedData* Definition: Write simulated datasets from each replicate to

output files in *OutDirectory*

Type: Integer

Dimension: 3

Options:

*printSimulatedData*(1)

0 True-breeding values and residuals of simulated

animals are not written to output file

1 True-breeding values and residuals of simulated

animals are written to output file

*printSimulatedData*(2)

0

1

*printSimulatedData*(3)

0 Genotypes of simulated animals are not written to output

file

1 Genotypes of simulated animals are written to output

file; genotypes written as 0, 1, or 2 at each locus, where

0 and 2 are homozygose for alleles 1 and 2, and 1 is

heterozygose

2 Genotypes of simulated animals are written to output

file; paternal allele written as 1 or 2 at each locus

followed by maternal alleles

Default: *printSimulatedData*=0 0 0

*printGenotypeTimes*

Definition: Array with time steps at which genotype of

individuals to be printed out.

Type: Integer

Dimension: 1≤*nTimes*≤*ntime*+1*,* where *nTimes* is the number of time

steps for which output variables are written

Options: 0≤ *printGenotypeTimes* ≤*ntime* (*i*=1,…,*nTimes*+1),

where *printGenotypeTimesi* is the *i*th integer of

*printGenotypeTimes* and

*printGenotypeTimesi*≠*printGenotypeTimesj*

(i≠j)

Default: huge(1)

*printBiasAccuracy* Definition: Write bias and accuracies to output files

Type: Character

Options:

no Biases and accuracies are not written to output files

yes Biases and accuracies are written to output files

Default: no

*printBreedingValues* Definition: Write true and estimated breeding values of each

selection candidate at each specified time

step and selection stage to output files,

‘breedingValuesRep<*rep*>.res’, where <*rep*> is

replicate number. Files are compressed by the

linux command,

‘bzip2 –f breedingValuesRep*<rep>*.res’.

Type: Character

Options:

no Breeding values are not written to output files

yes Breeding values are written to output files

Default: no

*printCoVariances* Definition: Write (co)variances estimated by DMU at each

specified time step and selection step to output

files,

‘covariancesRep<*rep*>Time<*time*>Stage<*stage*>.res’,

where <*rep*> is replicate number, <*time*> is time

step, and <stage> is selection stage

Type: Character

Options:

no Estimates are not written to output files

yes Estimates are written to output files

Default: no

\* Only used when *dmuPolyBlup*, *dmuGenomicBlup*, or *dmuIbdBlup* ‘dmuai’

\* *reportSelectionTimes* and *reportSelectionStages* applies to *printCoVariances*

*printEvaRelationships* Definition: Write genetic contributions, average relationships

of selected animals, and summaries of relationship

matrices for candidates and selected animals to

output files ‘averageRelationshipsReps.res’ and

averageRelationshipsMeans.res’; EVA-selection

stages only

Type: Character

Options:

no Contributions, relationships, and summaries are not

written to output files

yes Contributions, relationships, and summaries are written

to output files

Default: no

When *printEvaRelationships* ‘yes’, parameters of variables

genomicRelationshipParameters, genomicRelationshipCentre, genomicRelationshipScale,

genomicRelationshipWeight, ibsRelationshipParameters, and ibdRelationshipParameters

are as defined in namelist **&GENOMICBLUPPARAMETERS, &IBDBLUPPARAMETERS, and**

**&IBSBLUPPARAMETERS:** *stage*, *centre*, *scale*, *weight*, *firstChrom*, *lastChrom*, *loci*,

*scaleGToA*, *propAToG*, *addDiagG*, *diagGOne*, *genomicBase*, *maf*, *mafInclude*, *firstChrom*,

and *lastChrom*.

Description of variables as described for &GENOMICBLUPPARAMETERS**, &IBDBLUPPARAMETERS,**

**and &IBSBLUPPARAMETERS** with the exceptions that:

*stage* can be any integer value; it needs to be provided, although not used by

program/redundant. It is automatically set to 1 by program.

*scaleGToA* can only be ‘all’ and ‘base’

*scaleGToA* ‘genotyped’ and ‘genotypedbase’ are redundant because only animals included

in *marker file* can be animals used to estimate α and β. Animals included in the

*marker file* are traced and genotyped.

*scaleGToA* ‘traced’ options in *genomicBase* are not available as they are also

redundant. Animals included in *marker file* are already traced.

This is optional input. It is used to construct genomic-relationship matrix after

stages of EVA selection when *printEvaRelationships* ‘yes’, *geneticModel* ‘genomic’, and

*relationshipMatrix* in namelist &EVA, variable *EvaSelection* is not ‘genomic’. When

*relationshipMatrix* is ‘genomic’, the genomic-relationship matrix used to constrain

rates of inbreeding with EVA-selection is used. These relationship matrices are used

to calculate average relationships of selected animals and summaries of relationship

matrices for candidates and selected animals. At times

*firstReportTime*:*lastReportTime*. When all constraints are ‘genomic’ at times

*firstReportTime*:*lastReportTime*, *genomicRelationshipParameters* is not required.

Animals need to be genotyped to have their genotypes included in genomic-relationship

matrix. So, genotyping strategy is critical to *printEvaRelationships* when not using

genomic information to constrain rates of inbreeding in EVA.

Genotyping is ignored when constructing ‘ibs’ and ‘ibd’ relationship matrices; as for

matrices to constrain rates of inbreeding with EVA-selection.

If *relationshipMatrix* is ‘genomic’ in EVA-selection, but ‘pedigree’ used because no

animals are genotyped at times *firstReportTime*:*lastReportTime*, then output associated

with genomic-relationship matrices in *genomicRelationshipParameters* is not generated.

It is assumed that if genomic-relationship matrix cannot be created during EVA-

selection, it cannot be created during report.

Input to construct genomic-relationship matrices using the program, invgmatrix.

All *genomicBase* options available; genotyping is considered.

When *relationshipMatrix* is not ‘genomic’ in EVA-selection, the genomic-relationship

matrix generated with *printEvaRelationships* ‘yes’ is based on animals traced back

from the candidates for EVA-selection:

do i=1,nCandidates

call tracePedigreeGRM(contributions(i)%id)

end do

This option considers all stages of EVA-selection; selected animals may or may not be

destined for reproduction. Number of matings allocated to selected animals is used to

calculate genetic contributions; litter size is ignored. This is not a problem at

this stage as litter size is constant within selection stages.

ibdRelationshipParameters

Parameters used to construct IBD-relationship matrix after stages of EVA-selection

when printEvaRelationships 'yes', geneticModel 'genomic', and relationshipMatrix in

namelist &EVA, variable EvaSelection is not 'ibd'; EVA-selection stages only

ibsRelationshipParameters

Parameters used to construct IBS-relationship matrix after stages of EVA-selection

when printEvaRelationships 'yes', geneticModel 'genomic', and relationshipMatrix in

namelist &EVA, variable EvaSelection is not 'ibs';

EVA-selection stages only; required when printEvaRelationships 'yes', geneticModel

'genomic', EVA-selection in breeding scheme, and not all EVA-selection stages are

‘ibd’/’ibs’. If all EVA-selection stages, for example, are ‘ibd’, then do not need to

specify ibdRelationshipParameters, but will require ibsRelationshipParameters and

*genomicRelationshipParameters.*

ibdRelationshipParameters and ibsRelationshipParameters have same structure as

**&IBDBLUPPARAMETERS, variable parameters … [and the three other ibd and ibs namelists]**

*printPedigreeRelationships*

Definition: Write pedigree-relationship matrices to

output files in *OutDirectory* for selection stages

with (i) *selection\_criterion* ‘polyblup’ and *RunBlup*

1 in namelist &SELECTION, variable

selection\_scheme, or (ii) *MaleSelCrit*

‘polyblup’ and *MaleRunBlup* 1 and/or *FemaleSelCrit*

‘polyblup’ and *FemaleRunBlup* 1 in namelist &EVA,

variable EvaSelection;

aMatrix<*rep*>Time<*time*>Stage<*stage*>.res, where *rep*

is replicate, *time* is time step, and *stage* is

selection stage.

Type: Character

Options:

no Output is not written to output files

yes Output is written to output files

Default: no

Printed when *reportSelectionTimes* is current\_time and *reportSelectionStages* is selection stage

*printGenomicRelationships*

Definition: Write genomic-relationship matrices to

output files in *OutDirectory* when

*selection\_criterion* ‘genomicblup’ in any selection

stage of namelist &SELECTION, variable

selection\_scheme and/or *MaleSelCrit* or

*FemaleSelCrit* ‘genomicblup’in any EVA-selection

stage of namelist &EVA, variable EvaSelection;

gMatrix<*rep*>Time<*time*>Stage<*stage*>.res, where *rep*

is replicate, *time* is time step, and *stage* is

selection stage.

Type: Character

Options:

no Output is not written to output files

yes Output is written to output files

Default: no

*printLdHayesFiles* Definition: Write output from analysis of linkage

disequilibrium to output files in *OutDirectory*;

heterozygosityRep<*rep*>.dat,

QTLdistributionRep<*rep*>.dat,

markerDistributionRep<*rep*>.dat, LD\_binRep<*rep*>.dat,

and ldPlotsRep<*rep*>.pdf, where *rep* is replicate

number

Type: Character

Options:

no Output is not written to output files

yes Output is written to output files

Default: no

*printLdMutations* Definition: Write sampled positions of mutations from

analysis of linkage disequilibrium to output

file in *OutDirectory*;, mutationsRep<*rep*>.dat

Type: Character

Options:

no Output is not written to output file

yes Output is written to output file

Default: no

*debugOutput* Definition: Working-data and script files are copied to the

output directory. These files are listed in

‘Output files’. They are copied after they have

been created; newly-created files overwrite

older files. They will be the last files

created, should the program crash.

Type: Character

Options:

no Files are not copied to output directory

yes Files are copied to output directory

Default: no

\*\*Additional information\*\*

• Namelist &REPORT must be provided, even if it does not contain input

• When *geneticModel* ‘polygenic’ or ‘genomic’, genetic-trend files are written to

output directory *OutDirectory* by default:

geneticTrendsHerds<*first*>To<*last*>Rep<*rep*>.res

geneticTrendsHerds<*first*>To<*last*>Means.res

geneticTrendsHerds<*first*>To<*last*>Plots.res

phenotypicTrendsHerds<*first*>To<*last*>Rep<*rep*>.res

phenotypicTrendsHerds<*first*>To<*last*>Means.res

phenotypicTrendsHerds<*first*>To<*last*>Plots.res

where <*first*> and <*last*> are the first and last herds in a report group and <*rep*>

is replicate number.

• Genomic-inbreeding files are written to output directory *OutDirectory* when

*geneticModel* ‘genomic’ and *nGenomicInbreeding*>0:

genomicInbreedingHerds<*first*>To<*last*>Rep<*rep*>.res

genomicInbreedingHerds<*first*>To<*last*>Means.res

genomicInbreedingHerds<*first*>To<*last*>Plots.res

• Genomic-inbreeding parameters are used when *geneticModel* ‘genomic’ and

*nGenomicInbreeding*>0:

genomicInbreedingParameters

genomicInbreedingCentre

genomicInbreedingScale

genomicInbreedingWeight

• Genomic-homozygosity files are written to output directory *OutDirectory* when

*geneticModel* ‘genomic’ and *printGenomicHomozygosity* ‘yes’:

markerIbsHerds<*first*>To<*last*>Rep<*rep*>.res

qtlIbsHerds<*first*>To<*last*>Rep<*rep*>.res

templateMarkerIbdHerds<*first*>To<*last*>Rep<*rep*>.res

templateQtlIbdHerds<*first*>To<*last*>Rep<*rep*>.res

• Categorical files are written to output directory *OutDirectory* when

*printCategoricalProportions* ‘yes’ and *nCategoricalObs*>0:

categoricalProportionsHerds<first>To<last>Rep<rep>.res

• Animal and genotype data files are written to output directory *OutDirectory* when

*geneticModel* ‘genomic’ and *printSimulatedData* x x x:

xxxxxx

xxxxxx

xxxxxx

• Bias and accuracy files are written to output directory *OutDirectory* when

*printBiasAccuracy* ‘yes’:

biasAccuracyRep<*rep*>.res

biasAccuracyMeans.res

• Breeding value files are written to output directory *OutDirectory* when

*printBreedingValues* ‘yes’:

breedingValuesRep<*rep*>.res.bz2

Files breedingValuesRep*<rep>*.res’ are compressed with the linux command,

‘bzip2 –f breedingValuesRep*<rep>*.res’. bzip2 compresses files using the

Burrows-Wheeler block-sorting text-compression algorithm and Huffman coding.

They can be decompressed with the command, ‘bunzip2

breedingValuesRep*<rep>*.res.bz2’.

• Eva relationships files are written to output directory *OutDirectory* when

EVA-selection is carried out *printEvaRelationships* ‘yes’:

<files>

• Eva-relationship parameters are used when EVA-selection is carried out and

*printEvaRelationships* ‘yes’:

genomicRelationshipParameters

genomicRelationshipCentre

genomicRelationshipScale

genomicRelationshipWeight

ibsRelationshipParameters

ibdRelationshipParameters

• When *geneticModel* ‘ldonly’, *printLdHayesFiles*, *printLdMutations*, and

*debugOutput* are the only parameters used

• Parameters *nReportGroups*, *firstReportHerds*, and *lastReportHerds* apply to genetic-

trend, genomic-homozygosity, and categorical files:

geneticTrendsHerds<*first*>To<*last*>Rep<*rep*>.res

geneticTrendsHerds<*first*>To<*last*>Means.res

geneticTrendsHerds<*first*>To<*last*>Plots.res

phenotypicTrendsHerds<*first*>To<*last*>Rep<*rep*>.res

phenotypicTrendsHerds<*first*>To<*last*>Means.res

phenotypicTrendsHerds<*first*>To<*last*>Plots.res

genomicInbreedingHerds<*first*>To<*last*>Rep<*rep*>.res

genomicInbreedingHerds<*first*>To<*last*>Means.res

genomicInbreedingHerds<*first*>To<*last*>Plots.res

markerIbsHerds<*first*>To<*last*>Rep<*rep*>.res

qtlIbsHerds<*first*>To<*last*>Rep<*rep*>.res

templateMarkerIbdHerds<*first*>To<*last*>Rep<*rep*>.res

templateQtlIbdHerds<*first*>To<*last*>Rep<*rep*>.res

categoricalProportionsHerds<first>To<last>Rep<rep>.res

• Parameter *reportHomozygosityTimes* only applies to genomic-homozygosity files:

markerIbsHerds<*first*>To<*last*>Rep<*rep*>.res

qtlIbsHerds<*first*>To<*last*>Rep<*rep*>.res

templateMarkerIbdHerds<*first*>To<*last*>Rep<*rep*>.res

templateQtlIbdHerds<*first*>To<*last*>Rep<*rep*>.res

• Parameters *reportSelectionTimes* and *reportSelectionStages* apply to files generated

during selection – bias and accuracy, breeding value, and animal and genotype:

biasAccuracyRep<*rep*>.res

biasAccuracyMeans.res

breedingValuesRep<*rep*>.res.bz2

evaRelationship files

• All base animals born at and before time step 0 are used to calculate genetic

trend and genomic-inbreeding files for time 0

• All animals born at each time step are used to calculate output in each file

*(b) Selection stages specified in reportSelectionStages will generate output result when SelectionMethod=='truncation' or 'eva'. When SelectionMethod 'truncation', SelectionCriterion can be ’tbv', 'phenoweight', 'polyblup', 'genomicblup', ’ibdblup’, ~~'gas',~~ or ‘bayesp’. When SelectionMethod 'eva', MaleSelCrit can be 'tbv', 'polyblup', 'genomicblup', ’ibdblup’, ~~'gas',~~ or ‘bayesp’. FemaleSelCrit can be 'tbv', 'polyblup', 'genomicblup', ’ibdblup’, ~~'gas',~~ or ‘bayesp’. In all other cases, output is not written to the files.*

*(c) Selection stages specified in reportSelectionStages need not be ordered by*

*Selection stage*

*(d) Selection stages specified in reportSelectionStages cannot be repeated*

# Examples of input.prm files

## **Example 1: Polygenic model; one trait in the breeding objective**

## **Example 2: Polygenic model; nine traits in the breeding objective**

## **Example 3: Polygenic model with pseudo-genomic selection; four traits in the breeding objective, ‘genotyped’ animals are truncation selected**

## **Example 4: Polygenic model with pseudo-genomic selection; four traits in the breeding objective, ‘genotyped’ animals are EVA selected**

# Output files

Output files that are generated and written to output directory specified in namelist &OUTPUTDIRECTORY, variable *OutDirectory*:

geneticTrendsMeans.res

geneticTrendsRep<*rep*>.res **(*rep*=*start\_rep* … *start\_rep*+nrep-1)**

geneticTrendsPlots.pdf

input.log

seeds.in

error.txt

memory.log

adam\_log.txt

adamlog.log

biasAccuracyMeans.res

biasAccuracyRep<*rep*>.res

breedingValuesRep<*rep*>.res.bz2

dmuAdamRep<*rep*>Time<*time*>Stage<*stage*>.lst

animal#.res (optional)

marker#.res (optional)

mean\_timesteps.res

mean\_timesteps\_replicate.res

number\_genotyped.res (optional)

obs#.res (optional)

qtl#.res (optional)

template#.res

template\_info#.res (optional)

mean\_template#.res (optional)

mean\_founder#.res (optional)

## **geneticTrendsMeans.res**

Description: Means of genetic trends as a function of time across replicates

Column variables:

timeStep generation nReps BV sdBV polyBV sdPolyBV qtlBV sdQtlBV varBV sdVarBV varPolyBV sdVarPolyBV varQtlBV sdVarQtlBV varMend sdVarMend F sdF GI sdGI

When *ntbv*>1, variables specific to each trait are also presented:

timeStep generation nReps [necessary?] BV1 sdBV1 polyBV1 sdPolyBV1 qtlBV1 sdQtlBV1 varBV1 sdVarBV1 varPolyBV1 sdVarPolyBV1 varQtlBV1 sdVarQtlBV1 varMend1 sdVarMend1 … BV*ntbv* sdBV*ntbv* polyBV*ntbv* sdPolyBV*ntbv* qtlBV*ntbv* sdQtlBV*ntbv* varBV*ntbv* sdVarBV*ntbv* varPolyBV*ntbv* sdVarPolyBV*ntbv* varQtlBV*ntbv* sdVarQtlBV*ntbv* varMend*ntbv* sdVarMend*ntbv*

timeStep Birth time (0 … *ntime*)

generation Mean generation number across *nrep* replicates of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*. Generation number of animal *i* is calculated as ½(gNSire*i*+gNDam*i*)+1.0, where gNSire*i* and gNDam*i* are the generation numbers of the sire and dam of animal *i*. Generation numbers of base animals is 0.0.

nReps Number of replicates (*nrep*)

BV Mean aggregate-breeding value of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*. BV is the mean of *nrep* replicate means.

sdBV Standard deviation of BV (variation for replicate means)

polyBV Mean aggregate polygenic-breeding value across *nrep* replicates of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

sdPolyBV Standard deviation of polyBV

qtlBV Mean aggregate QTL-breeding value across *nrep* replicates of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

sdQtlBV Standard deviation of qtlBV

varBV Mean variance of aggregate-breeding values across *nrep* replicates for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

sdVarBV Standard deviation of varBV

varPolyBV Mean variance of aggregate polygenic-breeding values across *nrep* replicates for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

sdVarPolyBV Standard deviation of varPolyBV

varQtlBV Mean variance of aggregate QTL-breeding values across *nrep* replicates for animals born at time *timeStep* in herds F*irstReportHerd*≤herd≤*LastReportHerd*

sdVarQtlBV Standard deviation of varQtlBV

varMend Mean variance of aggregate-Mendelian deviations across *nrep* replicates for animals born at time *timeStep* in herds F*irstReportHerd*≤herd≤*LastReportHerd*

sdVarMend Standard deviation of varMend

F Mean inbreeding coefficient based on pedigree across *nrep* replicates for animals born at time *timeStep* in herds F*irstReportHerd*≤herd≤*LastReportHerd*

sdF Standard deviation of F

GI Mean generation interval across *nrep* replicates for animals born at time *timeStep* in herds F*irstReportHerd*≤herd≤*LastReportHerd*. GI for the *i*th animal is calculated as ½(2\*b*i*-s*i*-d*i*), where b*i* is the birth time of the *i*th animal born at time *timeStep*, and s*i* and d*i* are the birth times of the sire and dam of animal *i*.

sdGI Standard deviation of GI

BV*j* Mean breeding value for trait *j* (*j* = 1 … *ntbv*) across *nrep* replicates of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

sdBV*j* Standard deviation of BV*j* (variation for replicate means)

polyBV*j* Mean polygenic-breeding value for trait *j* across *nrep* replicates of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

sdPolyBV*j* Standard deviation of polyBV*j*

qtlBV*j* Mean QTL-breeding value for trait *j* across *nrep* replicates of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

sdQtlBV*j* Standard deviation of qtlBV*j*

varBV*j* Mean variance of breeding values for trait *j* across *nrep* replicates for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

sdVarBV*j* Standard deviation of varBV*j*

varPolyBV*j* Mean variance of polygenic-breeding values for trait *j* across *nrep* replicates for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

sdVarPolyBV*j* Standard deviation of varPolyBV*j*

varQtlBV*j* Mean variance of QTL-breeding values for trait *j* across *nrep* replicates for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

sdVarQtlBV*j* Standard deviation of varQtlBV*j*

varMend*j* Mean variance of aggregate-Mendelian deviations for trait *j* across *nrep* replicates for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

sdVarMend*j* Standard deviation of varMend*j*

Criteria for output: File always generated and written to *OutDirectory*

Additional information: Means are generated using the replicates run during the current simulation

## **geneticTrendsRep<rep>.res (rep=start\_rep … start\_rep+nrep-1)**

Description: Means of genetic trends as a function of time within replicates

Column variables:

timeStep generation nAnimals BV polyBV qtlBV varBV varPolyBV varQtlBV varMend F GI

When *ntbv*>1, variables specific to each trait are also presented:

timeStep generation BV1 polyBV1 qtlBV1 varBV1 varPolyBV1 varQtlBV1 varMend1 … polyBV*ntbv* qtlBV*ntbv* varBV*ntbv* varPolyBV*ntbv* varQtlBV*ntbv* varMend*ntbv*

timeStep Birth time (0 … *ntime*)

generation Mean generation number of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*. Generation number of animal *i* is calculated as ½(gNSire*i*+gNDam*i*)+1.0, where gNSire*i* and gNDam*i* are the generation numbers of the sire and dam of animal *i*. Generation numbers of base animals is 0.0.

nAnimals Number of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

BV Mean aggregate-breeding value of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

polyBV Mean aggregate polygenic-breeding value of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

qtlBV Mean aggregate QTL-breeding value of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

varBV Mean variance of aggregate-breeding values of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

varPolyBV Mean variance of aggregate polygenic-breeding values of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

varQtlBV Mean variance of aggregate QTL-breeding values of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

varMend Mean variance of aggregate-Mendelian deviations for animals born at time *timeStep* in herds F*irstReportHerd*≤herd≤*LastReportHerd*

F Mean inbreeding coefficient based on pedigree for animals born at time *timeStep* in herds F*irstReportHerd*≤herd≤*LastReportHerd*

GI Generation interval for animals born at time *timeStep* in herds F*irstReportHerd*≤herd≤*LastReportHerd*. GI is calculated as ½(2\*birth*i*-sireBirth*i*-damBirth*i*), where birth*i* is the birth time of the *i*th animal born at time *timeStep*, and sireBirth*i* and damBirth*i* are the birth times of the sire and dam of animal *i*.

BV*j* Mean breeding value for trait *j* (*j* = 1 … *ntbv*) of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

polyBV*j* Mean polygenic-breeding value for trait *j* of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

qtlBV*j* Mean QTL-breeding value for trait *j* of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

varBV*j* Mean variance of breeding values for trait *j* for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

varPolyBV*j* Mean variance of polygenic-breeding values for trait *j* for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

varQtlBV*j* Mean variance of QTL-breeding values for trait *j* for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

varMend*j* Mean variance of Mendelian deviations for trait *j* for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

Criteria for output: Files are always generated and written to *OutDirectory*

Additional information: Means are generated using the replicates run during the current simulation

## **geneticTrendsPlots.pdf**

Description: Plots of genetic gain, inbreeding, variances, and generation interval as functions of time

Plots:

Total genetic gain Mean aggregate-breeding value (± 95% confidence interval) as a function of time for animals born in herds *FirstReportHerd*≤herd≤*LastReportHerd*. BV is the mean of *nrep* replicate means.

Total polygenic gain Mean aggregate polygenic-breeding value (± 95% confidence interval) as a function of time across *nrep* replicates of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

Total QTL gain Mean aggregate QTL-breeding value (± 95% confidence interval) as a function of time across *nrep* replicates of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

Total genetic variance Mean variance of aggregate-breeding values (± 95% confidence interval) as a function of time across *nrep* replicates for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

Polygenic variance Mean variance of aggregate polygenic-breeding values (± 95% confidence interval) as a function of time across *nrep* replicates for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

QTL variance Mean variance of aggregate QTL-breeding values (± 95% confidence interval) as a function of time across *nrep* replicates for animals born at time *timeStep* in herds F*irstReportHerd*≤herd≤*LastReportHerd*

Mendelian variance Mean variance of aggregate-Mendelian deviations (± 95% confidence interval) as a function of time across *nrep* replicates for animals born at time *timeStep* in herds F*irstReportHerd*≤herd≤*LastReportHerd*

Inbreeding coefficient Mean inbreeding coefficient based on pedigree (± 95% confidence interval) as a function of time across *nrep* replicates for animals born at time *timeStep* in herds F*irstReportHerd*≤herd≤*LastReportHerd*

Generation interval Mean generation interval (± 95% confidence interval) as a function of time across *nrep* replicates for animals born at time *timeStep* in herds F*irstReportHerd*≤herd≤*LastReportHerd*. GI for the *i*th animal is calculated as ½(2\*b*i*-s*i*-d*i*), where b*i* is the birth time of the *i*th animal born at time *timeStep*, and s*i* and d*i* are the birth times of the sire and dam of animal *i*.

When *ntbv*>1, the following plots are also generated:

Genetic gain for trait *j* Mean breeding value for trait *j* (*j* = 1 … *ntbv*) (± 95% confidence interval) as a function of time across *nrep* replicates of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

Polygenic gain for trait *j* Mean polygenic-breeding value for trait *j* (± 95% confidence interval) as a function of time across *nrep* replicates of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

QTL gain for trait *j* Mean QTL-breeding value for trait *j* (± 95% confidence interval) as a function of time across *nrep* replicates of animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

Genetic variance for trait *j* Mean variance of breeding values for trait *j* (± 95% confidence interval) as a function of time across *nrep* replicates for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

Polygenic variance for trait *j* Mean variance of polygenic-breeding values for trait *j* (± 95% confidence interval) as a function of time across *nrep* replicates for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

QTL variance for trait *j* Mean variance of QTL-breeding values for trait *j* (± 95% confidence interval) as a function of time across *nrep* replicates for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

Mendelian varaiance for trait *j* Mean variance of Mendelian deviations for trait *j* (± 95% confidence interval) as a function of time across *nrep* replicates for animals born at time *timeStep* in herds *FirstReportHerd*≤herd≤*LastReportHerd*

## **input.log**

Description: Reconstruction of input file, input.prm

Criterion for output: Always generated

## **seeds.in**

Description:

## **error.txt**

Description: Error message

Criterion for output: Program stops because of an error in input file(s) or program code

## **memory.log**

Description: Memory usage

Criterion for output: Always generated

## **adam\_log.txt**

Description: Log file with a description of simulation at each time step of each replicate

Criterion for output: Always generated

## **adamlog.log**

Description:

## **biasAccuracyMeans.res**

Description: Mean bias and accuracy for each sex of selection candidates at each time and selection stage

Column variables:

time stage trait sex nReps mnNAnimals mnBias mnAccuracy sdNAnimals sdBias sdAccuracy

time Time stage (time=*firstReportTime* … *lastReportTime*)

stage Selection stage (stage=*reportSelectionStages*)

trait Trait number (trait=0 … *ntbv*, where 0 is aggregate-breeding value and 1 … *ntbv* is trait number)

sex Sex (sex=1 male, 2 female)

nReps Number of replicates

mnNAnimals Mean number of selection candidates across *nrep* replicates

mnBias Mean bias for selection candidates across *nrep* replicates, where bias is the regression of true-breeding value on estimated breeding value

mnAccuracy Mean accuracy for selection candidates across *nrep* replicates, where accuracy is the correlation of true-breeding value and estimated breeding value

sdNAnimals Standard deviation of mnNAnimals

sdBias Standard deviation of mnBias

sdAccuracy Standard deviation of mnAccuracy

Criteria for output: *printBiasAccuracy* ‘yes’ in namelist &REPORT

Additional information: No output written for times, stages, traits, and sex with no animals. No output is written at selection stages with no selection candidates or n<2. No output written when … (selection criterion) [CHECK]

(c) Selection scheme includes GAS breeding values

TotalAccuracy: AllCand GenoCand NonGenoCand, correlation between estimated

and true breeding values for all, genotyped, and non-

genotyped selection candidates

PolygenicAccuracy: AllCand GenoCand NonGenoCand, correlation between

estimated and true polygenic breeding values for

all, genotyped, and non-genotyped selection candidates

Derived from rep-files. So, excludes missing rep means and rep means with n<2.

Truncation: 'tbv','phenoweight','polyblup','genomicblup','gas', ‘bayesp’

EVA: 'tbv','polyblup','genomicblup','gas', ‘bayesp’

All combinations of time, stage, trait, sex presented?

## **biasAccuracyRep<rep>.res (rep=start\_rep … start\_rep+nrep-1)**

Description: Mean bias and accuracy for each sex of selection candidates at each time and selection stage

Column variables:

time stage trait sex nReps mnNAnimals mnBias mnAccuracy sdNAnimals sdBias sdAccuracy

time Time stage (time=*firstReportTime* … *lastReportTime*)

stage Selection stage (stage=*reportSelectionStages*)

trait Trait number (trait=0 … *ntbv*, where 0 is aggregate-breeding value and 1 … *ntbv* is trait number)

sex Sex (sex=1 male, 2 female)

nAnimals Mean number of selection candidates across *nrep* replicates

bias Mean bias for selection candidates across *nrep* replicates, where bias is the regression of true-breeding value on estimated breeding value

accuracy Mean accuracy for selection candidates across *nrep* replicates, where accuracy is the correlation of true-breeding value and estimated breeding value

Criteria for output: *printBiasAccuracy* ‘yes’ in namelist &REPORT

nAnimals>1 for output line is written

Truncation: 'tbv','phenoweight','polyblup','genomicblup','gas', ‘bayesp’

EVA: 'tbv','polyblup','genomicblup','gas', ‘bayesp’

## **breedingValuesRep<rep>.res.bz2 (rep=start\_rep … start\_rep+nrep-1)**

Description: Estimated and true breeding values of selection candidates at each time, selection stage, sex

Column variables:

time stage id sire dam sex age herd trait genotyped phenotyped selected tbv ebv

time Time step (time=*firstReportTime* … *lastReportTime*)

stage Selection stage (stage=*reportSelectionStages*)

id Animal id

sire Sire of animal

dam Dam of animal

sex Sex of animal (sex=1 male, 2 female)

age Age of animal

herd Herd of animal

trait Trait number (trait=0 … *ntbv*, where 0 is aggregate-breeding value and 1 … *ntbv* is trait number)

genotyped Animal genotyped (genotyped=0 not genotyped, 1 genotyped)

phenotyped Animal phenotyped for specific trait (phenotyped=0 not phenotyped, 1 phenotyped, -1 trait is 0 and number of traits>1). When

selected Animal selected in time and stage (selected=0 not selected, 1 selected)

tbv True-breeding value

ebv Estimated breeding value

Criteria for output: *printBreedingValues* ‘yes’ in namelist &REPORT

Additional information: Files are compressed by the linux command, ‘bzip2 –f breedingValuesRep*<rep>*.res’. They can be decompressed with the command,

‘bunzip2 breedingValuesRep*<rep>*.res.bz2’.

Breeding values are written to an output file at each selection stage with (i)

truncation or EVA selection, and (ii) selection criterion weighted phenotypes,

polygenic, genomic, or GAS breeding values

No output is written at selection stages with no selection candidates

**breedingValuesRep<*rep*>.res** (optional)

Estimated and true breeding values of selection candidates at each selection

stage

Breeding values are written to an output file at each selection stage with (i)

truncation or EVA selection, and (ii) selection criterion weighted phenotypes,

polygenic, genomic, or GAS breeding values

No output is written at selection stages with no selection candidates

Only non-missing rows and values included. If specified and not included, it is because not valid step or n<2, where n is the number of candidates in the calculation of b and r.

## **dmuAdamRep<rep>Time<time>Stage<stage>.lst (rep=start\_rep … start\_rep+nrep-1, time=1 … ntime, stage=1 … selection\_groups where breeding values predicted using DMU)**

Description: DMU-lst file generated by DMU when predicting breeding values

Criteria for output: *printDmuLstFiles* ‘yes’ in namelist &BLUPPARAMETERS

Additional information: A unique file is created at each selection stage involving DMU calculations:

<*rep*>, replicate number

<*time*>, time step

<*stage*>, selection stage

Files are written to directory dmuLstFiles, a subdirectory of the output directory *OutDirectory* (namelist &OUTPUTDIRECTORY, variable *OutDirectory*)

DMU-lst file is written to an output file at each selection stage with DMU Calculations. These selection stages have (i) truncation or EVA selection, and (ii) selection criterion polygenic, genomic, or GAS breeding values.

## **heterozygosity.dat**

Description: LD-summary parameters as a function time; exponential distribution; summary of each replicate

Column variables:

igen ld\_irep npar (count\_seg-nqtl) nqtl count\_mut count\_cul nneg heterozygosity(igen,ld\_irep,:) MeanFitness total\_var genic\_var ld\_quant

igen generation number of founder population

ld\_irep

npar

nmarkers number of segregating markers in generation igen

nqtl number of segregating QTL in generation igen

count\_mut number of new mutations in generation igen

count\_cul number of culled offspring (not selected) in generation igen

nneg

heterozygosity(igen,ld\_irep,:)

MeanFitness

total\_var

genic\_var

ld\_quant

Criteria for output: *printLdHayesFiles* ‘yes’ in namelist &REPORT

~~Additional information: ld\_irep is always 1 when~~ *~~geneticModel~~* ~~‘genomic’ in namelist &CONTROLPARAMETERS,~~ *~~LD\_method~~* ~~‘genomicld’ in namelist &LD. It ranges from 1:~~*~~nrep~~* ~~when~~ *~~geneticModel~~* ~~‘ld\_Only’.~~

## **ldSummaryOutputOverTime.dat**

Description: LD-summary parameters as a function time; exponential distribution; means of heterozygosity.dat across replicates

Column variables:

gen nReps nPar nMarkers nQTL nMutations nCulled nNegatives hetAll hetQtl meanFitness varA varG ld

gen

nReps

nPar

nMarkers

nQTL

nMutations

nCulled

nNegatives

hetAll

hetQtl

meanFitness

varA

varG

ld

Criteria for output: *printLdHayesFiles* ‘yes’ in namelist &REPORT

## **LD\_bin.dat**

Description: LD versus geneomic distrance markers as a function time; summary of each replicate

Column variables:

igen ld\_irep bin1 bin2 … bin40

igen Final generation number of LD-analysis

ld\_irep Replicate number

bin1 r2 at specified distance between markers

bin2

bin40

Criteria for output: *printLdHayesFiles* ‘yes’ in namelist &REPORT

~~Additional information: ld\_irep is always 1 when~~ *~~geneticModel~~* ~~‘genomic’ in namelist &CONTROLPARAMETERS,~~ *~~LD\_method~~* ~~‘genomicld’ in namelist &LD. It ranges from 1:~~*~~nrep~~* ~~when~~ *~~geneticModel~~* ~~‘ld\_Only’.~~

## **ldVersusDistanceOverTime.dat**

Description: LD versus genomic distance as a function time; exponential distribution; mean of LD\_bin.dat

Column variables:

gen nReps bin1 bin2 … bin40

gen Generation n umber of LD-analysis

nReps Number of replicates

bin1

bin2

bin40

Criteria for output: *printLdHayesFiles* ‘yes’ in namelist &REPORT

## **QTLdistribution.dat**

Description: Allele frequency and effect for each QTL locus at the end of the LD-analysis; summary of each replicate

Column variables:

igen ld\_irep freqency effect

igen Final generation number of LD-analysis

ld\_irep Replicate number

frequency Allele frequency at a QTL locus

effect QTL effect at a QTL locus

Criteria for output: *printLdHayesFiles* ‘yes’ in namelist &REPORT

~~Additional information: ld\_irep is always 1 when~~ *~~geneticModel~~* ~~‘genomic’ in namelist &CONTROLPARAMETERS,~~ *~~LD\_method~~* ~~‘genomicld’ in namelist &LD. It ranges from 1:~~*~~nrep~~* ~~when~~ *~~geneticModel~~* ~~‘ld\_Only’.~~

## **markerDistribution.dat**

Description: Allele frequency for each marker locus at the end of the LD-analysis; summary of each replicate

Column variables:

igen ld\_irep frequency

igen Final generation number of LD-analysis

ld\_irep Replicate number

frequency Allele frequency at a marker locus

Criteria for output: *printLdHayesFiles* ‘yes’ in namelist &REPORT

~~Additional information: ld\_irep is always 1 when~~ *~~geneticModel~~* ~~‘genomic’ in namelist &CONTROLPARAMETERS,~~ *~~LD\_method~~* ~~‘genomicld’ in namelist &LD. It ranges from 1:~~*~~nrep~~* ~~when~~ *~~geneticModel~~* ~~‘ld\_Only’.~~

## **ldPlots.pdf**

Description: Plots genetic gain, inbreeding, variances, and generation interval as functions of time; plots of **ldSummaryOutputOverTime.dat, ldVersusDistanceOverTime.dat, QTLdistribution.dat, and markerDistribution.dat**

Plots:

Number of segregating markers Number of segregating-marker loci as a function of founder generation

Number of segregating QTL Number of segregating-qtl loci as a function of founder generation

Number of mutations Number of new mutations as a function of founder generation

Number of culled animals Number of culled animals as a function of founder generation

Average heterozygosity of markers and QTL

Average heterozygosity of markers and QTL as a function of founder generation

Average heterozygosity of QTL Average heterozygosity of QTL as a function of founder generation

LD between segregating markers vs distance at generation *LD\_ngen*

LD between segregating markers *<0.125>* cM apart

Distribution of segregating−marker frequencies at generation *LD\_ngen*

Only frequencies from replicate 1 presented. If *geneticModel* ‘ld\_Only’ and *nrep*>1 in namelist &CONTROLPARAMETERS in namelist &LD, replicates 2:*nrep* not plotted.

Distribution of segregating−QTL frequencies at generation *LD\_ngen*

Distribution of segregating−QTL effects at generation *LD\_ngen*

Distribution of segregating−QTL variances at generation *LD\_ngen*

Effect versus frequency of segregating−QTL at generation *LD\_ngen*

## **baseHaplotypes.dat**

Description: Sampled haplotypes generated by LD

Criteria for output: *geneticModel* ‘genomic’ in namelist &CONTROLPARAMETERS~~,~~ *~~LD\_method~~* ~~‘genomicld’ in namelist &LD,~~ and *storeHaplotypes* ‘yes’ in namelist &LD

## **geneticArchitecture.dat**

Description: Sampled genetic architecture generated by LD

Criteria for output: *geneticModel* ‘genomic’ in namelist &CONTROLPARAMETERS~~,~~ *~~LD\_method~~* ~~‘genomicld’ in namelist &LD~~, and *storeHaplotypes* ‘yes’ in namelist &LD

## **animal#.res**

(Optional: Only printed if *printDataSets* in namelist &CONTROLPARAMETERS

has value yes)

File includes information on each individual animal including sire, dam,

sex, birth, death, herd, true and estimated breeding values etc.

**founder\_representation#.res**

(Optional: Only printed if *founder\_representation*=2 in namelist

TEMPLATE\_PARAMETERS)

For all QTL and a specified number of template loci the representation of

each founder (base animal) is given for each time step in replicate #. The

representation is given as the number of each founder allele and the number

of homozygotes for these alleles that the founder has contributed to the

population in the given time step.

**marker#.res**

(Optional: Only printed if *printDataSets* in namelist &CONTROLPARAMETERS

has value yes)

File includes marker genotypes for all individuals.

## **mean\_obs#.res**

## **mean\_QTL#.res**

## **mean\_template#.res**

## **mean\_obs#.res**

## **mean\_QTL#.res**

## **mean\_template#.res**

-all of the above output files are to be replaced by the following two files:

mean\_timesteps.res

## **mean\_timesteps\_replicate.res**

## **obs#.res**

(Optional: Only printed if *printDataSets* in namelist &REPORT

has value yes)

File includes realised phenotype, weight (ratio of residual variance to

variance of phenotype given breeding value), time of recording and herd-

year-season class associated with the recording for each observation for

each animal.

## **qtl#.res**

(Optional: Only printed if *printDataSets* in namelist &REPORT

has value yes)

File includes QTL genotypes for all individuals.

## **template#.res**

(Optional: Only printed if use\_marker\_template=2 in Namelist &CONTROL\_GENOME)

File includes average number of animals that are IBD in each of the marker

template loci for each time step in replicate #.

## **template\_info#.res**

(Optional: Only printed if use\_marker\_template=2 in Namelist &CONTROL\_GENOME)

File includes IBD information based on marker template for all individuals

in each time step for replicate #. If print\_genotypes=1 in Namelist

&TEMPLATE\_PARAMETERS marker template genotypes for all individuals are

printed as well.

When *debugOuput* ‘yes’ in namelist &REPORT, the following output files are generated and written to output directory specified in namelist &OUTPUTDIRECTORY, variable *OutDirectory*. Many of these files are regenerated during simulations. New files replace older files.

[Initialise]

***dmu4.polyblup.script***

Description:Script used to execute DMU4 when predicting polygenic-breeding values

Criteria for output: (i) *selection\_criterion* ‘polyblup’ in any selection stage of namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* or *FemaleSelCrit* ‘polyblup’ in any EVA-selection stage of namelist &EVA, variable EvaSelection, and (iii)variable *dmuPolyBlup* ‘dmu4’ in namelist &BLUPPARAMETERS

***dmu5.polyblup.script***

Description:Script used to execute DMU5 when predicting polygenic-breeding values

Criteria for output: (i) *selection\_criterion* ‘polyblup’ in any selection stage of namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* or *FemaleSelCrit* ‘polyblup’ in any EVA-selection stage of namelist &EVA, variable EvaSelection, and (iii)variable *dmuPolyBlup* ‘dmu5’ in namelist &BLUPPARAMETERS

***dmu4.genomicblup.script***

Description:Script used to execute DMU4 when predicting genomic-breeding values

Criteria for output: (i) *selection\_criterion* ‘genomicblup’ in any selection stage of namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* or *FemaleSelCrit* ‘genomicblup’ in any EVA-selection stage of namelist &EVA, variable EvaSelection, and (iii)variable *dmuGenomicBlup* ‘dmu4’ in namelist &BLUPPARAMETERS

***dmu5.genomicblup.script***

Description:Script used to execute DMU5 when predicting genomic-breeding values

Criteria for output: (i) *selection\_criterion* ‘genomicblup’ in any selection stage of namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* or *FemaleSelCrit* ‘genomicblup’ in any EVA-selection stage of namelist &EVA, variable EvaSelection, and (iii)variable *dmuGenomicBlup* ‘dmu5’ in namelist &BLUPPARAMETERS

***dmu4.gas.script***

Description:Script used to execute DMU4 when predicting GAS-breeding values

Criteria for output: (i) *selection\_criterion* ‘gas’ in any selection stage of namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* or *FemaleSelCrit* ‘gas’ in any EVA-selection stage of namelist &EVA, variable EvaSelection, and (iii)variable *dmuGasBlup* ‘dmu4’ in namelist &BLUPPARAMETERS

***dmu5.gas.script***

Description:Script used to execute DMU4 when predicting GAS-breeding values

Criteria for output: (i) *selection\_criterion* ‘gas’ in any selection stage of namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* or *FemaleSelCrit* ‘gas’ in any EVA-selection stage of namelist &EVA, variable EvaSelection, and (iii)variable *dmuGasBlup* ‘dmu5’ in namelist &BLUPPARAMETERSS

[Selection]

***eva.dat***

Description: Input-data file used by EVA to carry out EVA-selection

Criteria for output: *sex\_code* 7 in namelist &SELECTION, variable selection\_scheme. New data files replace older files written to output directory.

***eva.prm***

Description: Parameter file used by EVA to carry out EVA-selection

Criteria for output: *sex\_code* 7 in namelist &SELECTION, variable selection\_scheme. New data files replace older files written to output directory.

***map.hmatrix***

Description: Input-data file with marker loci. The file is used as input for Guosheng's program, invgmatrix, to construct genomic-relationship matrix with markers and pedigree

Criteria for output: *sex\_code* 7 in namelist &SELECTION, variable selection\_scheme and *relationshipMatrix* ‘genomic’, *current\_time*≥*startGenRelTime*, and *startGenRelTime≠-9* in namelist &EVA, variable EvaSelection. New data files replace older files written to output directory.

***marker.hmatrix***

Description: Input-data file with genotypes for each genotyped animal. The file is used as input to Guosheng's program, invgmatrix, to construct genomic-relationship matrix with markers and pedigree

Criteria for output: *sex\_code* 7 in namelist &SELECTION, variable selection\_scheme and *relationshipMatrix* ‘genomic’, current\_time≥*startGenRelTime*, and *startGenRelTime≠-9* in namelist &EVA, variable EvaSelection. New data files replace older files written to output directory.

***ped.hmatrix***

Description: Input-data file with pedigree of selection candidates. The file is used as input to Guosheng's program, invgmatrix, to construct genomic-relationship matrix with markers and pedigree

Criteria for output: *sex\_code* 7 in namelist &SELECTION, variable selection\_scheme and *relationshipMatrix* ‘genomic’, current\_time≥*startGenRelTime*, and *startGenRelTime≠-9* in namelist &EVA, variable EvaSelection. New data files replace older files written to output directory.

***par.hmatrix***

Description: Parameter file used by Guosheng's program, invgmatrix, to construct genomic-relationship matrix with markers and pedigree

Criteria for output: *sex\_code* 7 in namelist &SELECTION, variable selection\_scheme and *relationshipMatrix* ‘genomic’, current\_time≥*startGenRelTime*, and *startGenRelTime≠-9* in namelist &EVA, variable EvaSelection. New data files replace older files written to output directory.

***hMatrix.dat***

Description: H-matrix; genomic-relationship matrix contructed with markers and pedigree. The matrix is output from Guosheng's program, invgmatrix. It is used as input by DMU to generate genomic breeding values.

Criteria for output: Criteria for output: *sex\_code* 7 in namelist &SELECTION, variable selection\_scheme and *relationshipMatrix* ‘genomic’, current\_time≥*startGenRelTime*, and *startGenRelTime≠-9* in namelist &EVA, variable EvaSelection. New data files replace older files written to output directory.

***hMatrix.lst***

Description: Log-file from Guosheng's program, invgmatrix, when constructing genomic-relationship matrix with markers and pedigree.

Criteria for output: Criteria for output: *sex\_code* 7 in namelist &SELECTION, variable selection\_scheme and *relationshipMatrix* ‘genomic’, current\_time≥*startGenRelTime*, and *startGenRelTime≠-9* in namelist &EVA, variable EvaSelection. New data files replace older files written to output directory.

***eva.log***

Description: Log-file generated by EVA when carrying out EVA-selection

Criteria for output: *sex\_code* 7 in namelist &SELECTION, variable selection\_scheme. New data files replace older files written to output directory.

***eva\_MatingList.txt***

Description: Solution file with list of matings allocated by EVA when carrying out EVA-selection

Criteria for output: *sex\_code* 7 in namelist &SELECTION, variable selection\_scheme. New data files replace older files written to output directory.

[EBV]

***dmudat***

Description: Input-data file used by DMU to predict breeding values

Criteria for output: (i) *selection\_criterion* ‘polyblup, ‘genomicblup’, or ‘gas’ and *RunBlup* 1 in namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* and/or *FemaleSelCrit* ‘polyblup, ‘genomicblup’, or ‘gas’ and *MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist &EVA, variable EvaSelection, and (iii) and phenotypes and/or genotypes realised since breeding values last predicted

***dmuped***

Description: Pedigree-file used by DMU to predict breeding values

Criteria for output: (i) *selection\_criterion* ‘polyblup, ‘genomicblup’, or ‘gas’ and *RunBlup* 1 in namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* and/or *FemaleSelCrit* ‘polyblup, ‘genomicblup’, or ‘gas’ and *MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist &EVA, variable EvaSelection, and (iii) and phenotypes and/or genotypes realised since breeding values last predicted

***map.gmatrix***

Description: Input-data file with marker loci. The file is used as input for Guosheng's program, gmatrix, to generate genomic-relationship matrix.

Criteria for output: (i) *selection\_criterion* ‘genomicblup’ and *RunBlup* 1 in namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* or *FemaleSelCrit* ‘genomicblup’ and *MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist &EVA, variable EvaSelection, and (iii) and phenotypes and/or genotypes realised since breeding values last predicted

***marker.gmatrix***

Description: Input-data file with genotypes for each genotyped animal. The file is used as input to Guosheng's program, gmatrix, to generate genomic-relationship matrix.

Criteria for output: (i) *selection\_criterion* ‘genomicblup’ and *RunBlup* 1 in namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* or *FemaleSelCrit* ‘genomicblup’ and *MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist &EVA, variable EvaSelection, and (iii) and phenotypes and/or genotypes realised since breeding values last predicted

***par.gmatrix***

Description: Parameter file used by Guosheng's program, gmatrix, to generate genomic-relationship matrix.

Criteria for output: (i) *selection\_criterion* ‘genomicblup’ and *RunBlup* 1 in namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* or *FemaleSelCrit* ‘genomicblup’ and *MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist &EVA, variable EvaSelection, and (iii) and phenotypes and/or genotypes realised since breeding values last predicted

***gMatrix.dat***

Description: Genomic-relationship matrix contructed by Guosheng's program, gmatrix. It is used as input by DMU to predict genomic-breeding values.

Criteria for output: (i) *selection\_criterion* ‘genomicblup’ and *RunBlup* 1 in namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* or *FemaleSelCrit* ‘genomicblup’ and *MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist &EVA, variable EvaSelection, and (iii) and phenotypes and/or genotypes realised since breeding values last predicted

***gMatrix.lst***

Description: Log-file from Guosheng's program, gmatrix, when constructing genomic-relationship matrix

Criteria for output: (i) *selection\_criterion* ‘genomicblup’ and *RunBlup* 1 in namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* or *FemaleSelCrit* ‘genomicblup’ and *MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist &EVA, variable EvaSelection, and (iii) and phenotypes and/or genotypes realised since breeding values last predicted

***genotyped.dat***

Description: Data file with list of genotyped animals. It is used as input by DMU to predict genomic-breeding values.

Criteria for output: (i) *selection\_criterion* ‘genomicblup’ and *RunBlup* 1 in namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* or *FemaleSelCrit* ‘genomicblup’ and *MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist &EVA, variable EvaSelection, and (iii) and phenotypes and/or genotypes realised since breeding values last predicted

***dmu.lst***

Description: Log-file from DMU when predicting genomic-breeding values

Criteria for output: (i) *selection\_criterion* ‘polyblup, ‘genomicblup’, or ‘gas’ and *RunBlup* 1 in namelist &SELECTION, variable *selection\_scheme*, or (ii) *MaleSelCrit* and/or *FemaleSelCrit* ‘polyblup, ‘genomicblup’, or ‘gas’ and *MaleRunBlup* and/or *FemaleRunBlup* 1 in namelist &EVA, variable EvaSelection, and (iii) and phenotypes and/or genotypes realised since breeding values last predicted

[LD\_Hayes]

***ldSummary.r***

Description: R-script used to generate summary data and plots from LD-analysis

Criteria for output: (i) *printLdHayesFiles* ‘yes’ in namelist &REPORT and (ii) *geneticModel* ‘genomic’ in namelist &CONTROLPARAMETERS and *useStoredHaplotypes* ‘no’ in namelist &LD or *geneticModel* ‘ld\_Only’ in namelist &CONTROLPARAMETERS

***ldSummary.rLog***

Description: Log-file from execution of *ldSummary.r* by R

Criteria for output: (i) *printLdHayesFiles* ‘yes’ in namelist &REPORT and (ii) *geneticModel* ‘genomic’ in namelist &CONTROLPARAMETERS and *useStoredHaplotypes* ‘no’ in namelist &LD or *geneticModel* ‘ld\_Only’ in namelist &CONTROLPARAMETERS

[Report]

***geneticTrendsPlots.r***

Description: R-script used to generate output file, *geneticTrends.pdf,* plots of genetic trends as functions of time

Criteria for output: R-script always generated

***geneticTrendsPlots.rLog***

Description: Log-file from execution of *geneticTrendsPlots.r* by R

Criteria for output: Log-file always generated

***biasAccuracyMeans.r***

Description: R-script used to generate output file, *biasAccuracyMeans.res,* mean biases and accuracies for each trait at each time step, selection stage, and sex

Criteria for output: *printBiasAccuracy* ‘yes’ in namelist &REPORT

***biasAccuracyMeans.rLog***

Description: Log-file from execution of *biasAccuracyMeans.r* by R

Criteria for output: *printBiasAccuracy* ‘yes’ in namelist &REPORT

***biasAccuracyPlots.r***

Description: R-script used to generate output file, *biasAccuracyPlots.pdf,* plots of biases and accuracies as functions of time

Criteria for output: *printBiasAccuracy* ‘yes’ in namelist &REPORT

***biasAccuracyPlots.rLog***

Description: Log-file from execution of *biasAccuracyPlots.r* by R

Criteria for output: *printBiasAccuracy* ‘yes’ in namelist &REPORT

[genomicInbreedingCoefficients]

***map.gimatrix***

Description: Input-data file with marker loci. The file is used as input for Guosheng's program, invgmatrix, to construct genomic-relationship matrix with markers

Criteria for output: *geneticModel* ‘genomic’ in namelist &CONTROLPARAMETERS. New data files replace older files written to output directory.

***marker.gimatrix***

Description: Input-data file with genotypes of all animals. The file is used as input to Guosheng's program, invgmatrix, to construct genomic-relationship matrix with markers

Criteria for output: *geneticModel* ‘genomic’ in namelist &CONTROLPARAMETERS. New data files replace older files written to output directory.

***par.gimatrix***

Description: Parameter file used by Guosheng's program, invgmatrix, to construct genomic-relationship matrix with markers

Criteria for output: *geneticModel* ‘genomic’ in namelist &CONTROLPARAMETERS. New data files replace older files written to output directory.

***giMatrix.dat***

Description: G-matrix; genomic-relationship matrix contructed with markers. The matrix is output from Guosheng's program, invgmatrix. It is used by ADAM to calculate breeding coefficients for all animals in the population.

Criteria for output: *geneticModel* ‘genomic’ in namelist &CONTROLPARAMETERS. New data files replace older files written to output directory.

***giMatrix.lst***

Description: Log-file from Guosheng's program, invgmatrix, when constructing genomic-relationship matrix with markers.

Criteria for output: *geneticModel* ‘genomic’ in namelist &CONTROLPARAMETERS. New data files replace older files written to output directory.

giGMatrix.dat \*.lst par.

giBlendedHMatrix.dat \*.lst par.

# References

Goddard 2009

Grundy et al. 1999. JABG 115:39-51

Henryon et al. (2010)

Jannink 2010 GSE 42:35

Meuwissen 2000

Meuwissen 2001

Wu & Schaeffer 2000. JABG 117:361-374

See Grundy et al. (1998) JABG 115:39-51 and Wu & Schaeffer (2000) JABG 117:361-374 for details

Faux et al. 2016 AlphaSim

Stift 2010 tetrasomic