

Getting started with EVA v 3.0

Software written by
Peer Berg

Program to select mating set with optimal contributions
& to maximise function of merit and relationships.
Optionally minimises inbreeding conditional on genetic contributions.

Disclaimer

The free software program EVA may be freely distributed, provided that no charge above the cost of distribution is levied, and that the disclaimer below is always attached to it.

The program is provided as is without any guarantees or warranty.
Although the authors have attempted to find and correct any bugs in the free software program, the authors is not responsible for any damage or losses of any kind caused by the use or misuse of the program.

It will be our aim to correct any bugs reported as well as providing regular updates. But the author is under no obligation to provide support, service, corrections, or upgrades to the free software programs.

To receive information on updates, send an e-mail to Peer Berg:
<mailto:Peer.Berg@nmbu.no?subject=Information on updates>

Table of Contents

Quick start	3
What can EVA do for me?.....	3
Installation.....	3
<i>Windows</i>	3
<i>Mac OS X</i>	3
<i>Linux</i>	3
Running EVA.....	3
<i>Running EVA using R</i>	4
Input files	4
Data input file - mandatory.....	4
EVA Parameter file - mandatory.....	5
<i>Overview of Parameters</i>	6
<i>&DataParameters</i>	10
<i>&PopulationHistory</i>	10
<i>&Contribution</i>	11
<i>&RelationshipMatrix</i>	11
<i>&OCSPParameters</i>	11
<i>&AlgorithmParameters</i>	12
<i>&Mating</i>	12
Relationship file - optional.....	12
Tips.....	12
Output files	13
<i>Log file</i>	13
<i>F_summary.txt</i>	13
<i>f_coeff.txt</i>	13
<i>max_gc.txt</i>	14
<i>gen_cont.txt</i>	14
<i>Eva_conv.txt & Eva_best.txt</i>	15
<i>eva_MatingList.txt</i>	15
<i>male_list.txt</i>	16
<i>Candidates.txt</i>	16
<i>Optional output files</i>	17
Introduction to Evolutionary Algorithms	17
Optimisation criteria	19
Citation	19

Quick start

What can EVA do for me?

The EVA software can be used to:

- Describe the history of a population in terms of
 - Individual inbreeding coefficients and completeness of pedigree
 - Average inbreeding, coancestry, pedigree completeness and generation equivalents per cohort
 - Genetic contributions
 - All founders
 - Most contributing ancestors
 - Any user-specified individuals to any individual or cohort
- Optimize genetic contributions
 - To optimize a linear function of genetic merit and average additive relationships
 - Conditional on optimal contributions to mate individuals randomly or minimizing inbreeding in offspring

Installation

The software is available for Windows, Linux and Mac OS X.

Windows

Use the windows installer to complete the installation.

Mac OS X

We recommend to run the following commands in a terminal window, from the bin directory

```
sudo mkdir -p /usr/local/gfortran/lib
sudo cp libquadmath.0.dylib /usr/local/gfortran/lib/.
sudo cp eva /usr/local/bin/.
```

These commands are also in the script file `install_eva`, run by typing

```
./install_eva
```

Linux

The executable needs to be in your path, for example in `/usr/local/bin` .

Running EVA

EVA is run from the command line.

For Windows, you can open a shell by running the `EVA.bat` script in the program list.

- An EVA shell is opened by the `eva.bat` script.
- This opens a shell in which you can run command line arguments.
- Opening a shell via the `eva.bat` also ensures that the executable is in your path.
- You can change directory to anywhere you want and run EVA.
- You run EVA by typing the command

```
EVA eva.prm
```

Where `eva.prm` is the filename of your parameter file.

The progress will be shown on screen as well as written to the log file.

For a list of useful Windows command line arguments see for example:

<http://www.digitalcitizen.life/command-prompt-how-use-basic-commands>

If the run is aborted with an error, this log file is located in the submission directory. Otherwise it is located in the directory with results files. See [ResultsDirectory](#)

Running EVA using R

Eva can be called by application of global functions. You will find a R-script `evaR.R` as well as a set of functions `evaR_functions.R` together with the material you downloaded with the programme. Use the R-script to load the functions, execute EVA and receive output in your R environment. There are also few examples of plots of your data available. The functions are under development. The current version is tested on Mac OS X and Windows systems.

Input files

For EVA to run it needs a data input file and a parameter file that specifies what the program should do. Optionally the user can supply a relationship file.

Data input file - mandatory

The data file has to be a plain ASCII file with the following columns:

Column	Short description
Individual ID	Integer ID for individual
Sire ID	Integer ID for sire of individual
Dam ID	Integer ID for dam of individual
Sex	Sex of individual (1 if male, 2 if female). Integer. Must take values 1 or 2 for individuals that are candidates (where value of <code>Max.Matings>0</code>). For other individuals, this variable is ignored and can thus take any value.

BirthTime	Time when individual is born. In units of time period. See below for further explanation. Integer
MaxMatings	Maximum number of matings for this individual. The value 0 indicates that the individual is unavailable as a candidate for selection. Integer.
BreedingValue	Genetic merit/Predicted breeding value, e.g. BLUP of individuals breeding value. Real.
Text	Optional field with a text string of up to 15 characters to be printed in the mating list. Can be any combination of letters and numbers used for additional identification of individuals.

The file contains no headers and the columns are space- or tab-separated.

IDs must be standard integers in the interval -2,147,483,648 (two billion....) to 2,147,483,647. These IDs are internally recoded from 1 to number of individuals. These recoded IDs can be written to a file.

Generally, all individuals should be included in the file. Individuals must be in the file prior to their offspring. Parents that do not show up in the file as individuals, can be inserted by the program depending on what the parameter *IgnoreParentalPedigreeErrors* is set to (see below).

BirthTime is the time of birth (cohort/timestep) of the individual. These should be specified such that parent and offspring are not born in the same cohort. They should be consecutive, in increments of 1. Founders are all individuals for which no parental information is given.

If *maxage* is the maximal age (in timesteps) of a parent, the data file should have complete records of parent contributions in the last *maxage*+1 time period. This means that there should be records for all individuals born in the last *maxage*+1 time period. This is required to compute past contributions in cases with overlapping generations.

It is advantageous (for computational efficiency) that the number of time steps are minimized. This means that in for example cattle and horses a time step of 1 year is natural, in pigs 6 months, and in fish 3 or 4 years depending on time of sexual maturation.

The program assumes that individuals from matings to be planned will be born at a time step advanced by 1 relative to the maximal value observed in the data.

EVA Parameter file - mandatory

The parameter file gives directives to EVA on what to do. Most parameters have default values, which are used if not specified otherwise in the parameter file.

An example parameter file with comments is provided in the documentation folder, where this manual can also be found. Use the example parameter file as a starting point and delete and modify the parameters as needed. If you want to insert comments to yourself in the parameter file, you can use the # symbol to tell EVA that the following text is just a comment. Anything on a line starting with # will not be read by EVA.

Parameters are specified in a number of NAMELISTS with parameters for the different parts of the program. The general format for these NAMELISTS is

```
&SECTION
    Parameter1=xxx ,
    Parameter2=yyy ,
    ...
/
```

Where &SECTION is the name of the parameter list, individual parameter and their value are separated by commas, and the parameter list is closed with “/”

If e.g. Parameter 1 is not specified, it will take its default value.

The following list is a short summary of the parameters. Click on the links to get a more detailed description.

Overview of Parameters

Parameter	Description	Default value
&DataParameters		
DataFile	Filename (and path if not in directory where EVA is run).	eva.dat
ResultsDirectory	Prefix on all results files or directory path to where result files are placed.	"EVA_"
IgnoreParentalPedigreeErrors	If .true. or T then parents are inserted in the pedigree, if not found previously.	.false. / F
RecodeFile	Filename for a file containing original and recoded IDs. If RecodeFile is not given a value or given the value “_null_” no file will be written.	'_null_'
&PopulationHistory		
PCI_nngen	Number of generations for computation of Pedigree Completeness Index.	5
&Contribution (as many of these NAMELISTS as needed can be specified)		

Ancestor	ID of ancestor for whom to compute genetic contributions. If 0 it means all founders in the pedigree.	0
Descendant	ID of descendant whose genetic contributions from ancestors are computed.	0
Group	Timestep (BirthTime) for individuals for whom the genetic contributions are computed.	0
&RelationshipMatrix		
Source	'pedigree': compute additive genetic relationships from the pedigree in the data file. 'file': read additive genetic relationships from file.	'pedigree'
GFile	Filename. If source='pedigree', then relationships are written to this file, unless it is '_null_'. If source='file' relationships are read from this file.	'_null_'
TimeSteps	Number of timesteps to include. All individuals born in the last timesteps will be included in the numerator relationship matrix as well as all their ancestors. If 0 this is determined from data by the maximum age of parents. Timesteps<0 ends the program after computing inbreeding and genetic contributions, ignoring request for writing or reading the relationship matrix.	0
&OCSPParameters		

Nmatings	Number of matings to be selected by Optimal Contribution Selection (OCS). If -1 OCS is aborted.	-1
Optimise	Criterion for optimization, possible values: "penalty" or "constraint" or "merit"	penalty
Wmerit	Weight on genetic merit (estimated breeding value or index).	0.0
Wrelationship	Weight on average additive genetic relationship of the last generation including the proposed matings. Only values ≤ 0.0 is accepted	-1.0
dFconstraint	Constraint on rate of inbreeding. Only read if Optimise="constraint" This is the constraint on rate of inbreeding per generation.	1.0 (no constraint)
LimitMaleMatings	Males can only be used in a multiplum of the number specified. If 2 then males can only be used 0, 2, 4 .. up to MaxMatings (as specified in the data for that male).	1
W_nMales ¹ ¹ not yet available in v3.0	The value of each selected males. A negative value specifies a cost for each male selected.	0
NSelectedMales ¹ ¹ not yet available in v3.0	Desired number of selected males. 0 results in optimizing the number of males. Use with caution as sub-optimal solutions are generated.	0
&AlgorithmParameters		
generations	Number of generations the Evolutionary Algorithm is running.	10000

NGenerationsNoImprovement	Number of generations without improvements, that should stop further iterations	10000
PopSize	Size of population of solutions that evolves	100
N_offspring	Number of new possible solutions produced per generation.	10
Restart_interval	If the best solution has not improved for this number of generations then more variance is generated by increasing the mutation variance for one generation.	1000
Exchange_algorithm	Interval between using an exchange algorithm to iteratively optimize solutions.	500
Mutate_probability	Probability of mutating an individual in a solution (randomly exchanging that individual).	$1/(4*Nmatings)$
Crossover_probability	Probability of crossovers when generating a new solution from two parental solutions.	0.2
Directed_mutation_probability	Probability of mutating an individual in a solution (exchanging that individual with a better).	$1/(2*Nmatings)$
Seed_rng	Seed for random number generator.	0 (use computer clock to sample seed)
&Mating		
MatingsStrategy	<p>'mai' : maximum avoidance of inbreeding. Minimises inbreeding conditional on the optimal genetic contributions.</p> <p>'random': Individuals selected are mated randomly.</p>	'random'

RepeatedMatings	<p>If .true. (T) then repeated matings between the same two individuals are allowed.</p> <p>If .false. (F) then repeated matings between the same two individuals are avoided.</p>	.true.
-----------------	--	--------

&DataParameters

DataFile and *RecodeFile* are exhaustively described in the overview table.

ResultsDirectory

Specifies either

- A prefix on all results files
- A directory where results files are placed

Examples:

`ResultsDirectory=' 2012Pig'`

Will add a prefix on all result files, such that all files are named 2012Pigxxx, where xxx indicates the default result file name.

`ResultsDirectory=' .\Pig2012\'`

Will place all result files in a subdirectory in the current directory (where EVA is run) named Pig2012. In this subdirectory result files will take their default names.

`ResultsDirectory=' c:\users\mydata\Pig2012\'`

Will place all result files in a directory c:\users\mydata\Pig2012. In this directory result files will take their default names.

IgnoreParentalPedigreeErrors

If the value is .false. or F (the default) then all parents should also be in the file with their own record. Otherwise an error message is given and the program is aborted.

If the value is .true. or T, then a parent that has not been represented as an individual previously in the file will be inserted. If this parent is then later in a record as an individual, then an error message is given and the program is aborted (the individuals are not sorted properly – parents before offspring).

Use the value .true. with caution.

&PopulationHistory

PCI_ngen

Pedigree completeness indexes are computed as described by the following two articles:

Sigurdsson, A. & Jonmundsson, J.V. 1994. Proceedings of the 5th WCGALP, vol 17: 140-143

MacCluer J.W.; Boyce A.J.; Dyke B.; Weitkamp L.R.; Pfennig D.W. & Parsons C.J: 1983. Journal of Heredity 74:394-399

Essentially this is a measurement of the proportion of combinations of maternal-paternal ancestors up to PCI_nngen generations back, that is known and thus could contribute to the inbreeding coefficient of an individual or group of individuals.

&Contribution

Any number of NAMELISTs of this type can be specified. They are processed consecutively. Only two of the three parameters can be specified.

Examples:

```
&Contribution
ancestor=10,
descendant=25
/
```

Computes the genetic contribution of 10 to 25 (and the additive relationship between them).

```
&Contribution
ancestor=10,
group=2010
/
```

Computes the genetic contribution of 10 to all individuals born in timestep 2010 as specified in the data file.

```
&Contribution
ancestor=0,
group=2010
/
```

Computes the genetic contribution of all founders (individuals with unknown parents) to all individuals born in timestep 2010 as specified in the data file.

&RelationshipMatrix

Individual inbreeding coefficients are computed with the algorithm described in:

Meuwissen, THE & Luo, Z.; 1992. Computing Inbreeding Coefficients in Large Populations. Genet. Sel. Evol. 24:305-313.

Additive genetic relationships and average relationships between groups of individuals are computed with the algorithm proposed by:

Colleau, J.-J. 2002. An indirect approach to the extensive calculation of relationship coefficients. Genet.Sel.Evol. 34:409-421

&OCSPParameters

NSelectedMales and **w_nMales** can be used to control the number of males. Used alone (NSelectedMales=0) w_nMales specifies the value or cost of selecting males (negative values indicate costs). If NSelectedMales >0 then a number of males deviating from the desired number is penalized by w_nMales (default $1/(2*Nmatings)$). Avoid too large values of w_nMales as this negatively

influences convergence. Use this option with caution as it can result in sub-optimal solutions relative to the optimal frontier.

Please note that the options *NSelectedMales* and *w_nMales* are not available in Version 3.0.

&AlgorithmParameters

Be cautious in changing these parameters. It might affect the rate of convergence.

&Mating

With EVA, proposals for new matings can be obtained, conditional on the optimized genetic contributions, either:

- a) Mating at random
- b) Mating to minimise inbreeding in the resulting offspring (maximum avoidance of inbreeding – mai)

Relationship file - optional

An external relationship file can be provided, containing for example genomic relationships between individuals. See [&RelationshipMatrix](#) in the EVA parameter file section for more information.

The format of the relationship file is one line for each (non-zero) element containing

ID_i: id of individual i in the same form as used in the data file.

ID_j: id of individual j in the same form as used in the data file.

G_{ij}: element of the relationship matrix for individuals i and j.

Symmetry is assumed so G_{ij} needs only be provided once (G_{ji} is not required). If provided multiple times (as either G_{ij} or G_{ji}), the last element read will be used.

Diagonal elements have i=j. If a diagonal element is not provided, then the corresponding individual cannot be a candidate. If it is a candidate then MaxMatings=0 will be set for that individual.

A summary of expected vs. realized number of individuals read from the file will be printed.

Off-diagonal elements not read from the file are set to 0. Thus, only non-zero elements need to be specified in the file.

Tips

- If you think you have detected a bug, please report it
- If the Optimal Contribution Selection is not running as expected or results look suspicious, start by checking the descriptive statistics. Examples
 - Check that birth times are properly defined
 - Check that pedigrees are complete. A few individuals with incomplete pedigrees can seriously affect the results obtained.
 - Check age profile of candidates.
- Ensuring convergence, see [Iteration history](#)

Output files

All output files are in ASCII format and can be opened by a variety of editors, imported into spreadsheet software such as Excel or imported into R, which has good graphical plotting capabilities.

Log file

EVA prints a log file - `eva.log`. This gives information on the parameters used, error messages and progress of the computations.

F_summary.txt

Summarises information on average inbreeding, coancestry, pedigree completeness and generation equivalents.

Summarises information on the candidates and the best mating set produced.

Contains a list with averages for individuals born in each timestep

- GroupBirth time timestep
- N Number of individuals born in this group
- N_inbred Number of inbred individuals born in this group
- avg.F Average inbreeding coefficient
- max.F Largest inbreeding coefficient
- avg.coa Average coancestry incl. self
- exp.F Expected inbreeding under random mating
- Alpha Deviation from random mating
- PCIn Pedigree Completeness Index for n generations
- Gencoef Average generation coefficient. Number of complete generations including the base population.
- GenInt Generation interval. Average age of parents.

Alpha is negative when inbred matings have been avoided, and positive when inbred matings have been favoured. There is a dependence among the parameters such that

$$(1-\text{avf.F})=(1-\text{exp.F})(1-\text{alpha})$$

Alpha thus describes deviations from the level of inbreeding that would have been obtained by random mating, due to the actual matings done.

Contains a summary of the data. If relationships are read from an external file a summary of inbreeding and co-ancestries are printed as well as a summary of the expected and realized number of records read from the file.

Contains a summary of the best mating set found (if run).

Contains a summary of the inbreeding expected in the new cohort after the matings are optimized as specified.

f_coeff.txt

A file with one record for each individual in the same format as the input

For each individual its inbreeding coefficient, pedigree completeness and generation equivalent is printed:

Variable	Description	Format
Individual ID		integer
Parent ID	missing=0	integer
Other parent ID	missing=0	integer
Sex	1: males, 2: females	1 or 2
Birth_time	Birth year code or equivalent from input file)	Integer
Inbreeding coefficient	real between 0 and 1	real
Generation coefficient	Generation equivalent is 1 for founders and updates with one for complete each generation away from the founders.	real >1
Pedigree Completeness Index n generations back	n specified by the user	

[max_gc.txt](#)

A list of ancestors with the 50 largest genetic contributions to the last cohort in the data (individuals born in the last timestep). Genetic contributions are not corrected for the contributions of other ancestors and thus sums to more than 1.

[gen_cont.txt](#)

A list of genetic contributions and additive genetic relationships

- of founders (individuals with unknown parents) to the last cohort in the data (individuals born in the last timestep).
- Of any combinations asked for by the [&Contribution](#) namelist

Note: When genetic contributions of founders are written, individuals are marked as either “founder” or “½ founder”. Founders are individuals with both parents unknown. ½ founders are individuals with one parent unknown. Such individuals are only considered half a founder and the genetic contribution is only half its contribution, reflecting the contribution of its unknown parent. This is done to ensure that genetic contributions of founders sum to 1. This also implies that if you ask specifically for the contribution of such an individual you will get its full genetic contribution. The additive genetic relationship is not affected by founder status.

Eva_conv.txt & Eva_best.txt

The iteration history of the Evolutionary Algorithm is written to two files

Eva_conv.txt Gives convergence of the population of solutions

Eva_best.txt Gives information on the evolution of the best solution

These files give indications whether EVA has converged to the optimal solution. If the best solution is found in a recent generation, then there is reason to believe that the algorithm might not yet have converged.

Generally, the convergence follows a pattern where initially there are large improvements in the current best population, but these improvements get gradually smaller, meaning that after a few generations the current best solution is likely close to the optimal solution.

To test convergence two approaches can be used, either separately or preferably in combination.

1. Run EVA with a large number of generations (parameter `Generations` in `Namelist &AlgorithmParameters`).
2. Run several independent runs and compare results. If they differ then the algorithm has not converged, and parameter `Generations` should be increased.

Generally, convergence is specific to the data structure. So once a conservative value for `Generations` has been found, this value can be used in data with the same structure, such as running the same data with different constraints on inbreeding.

eva_MatingList.txt

The first line in the mating list is the header: 'Sire','Dam','N.mating','Nm','Nf.m.','EBV','F','vF','Sire txt','Dam txt'. A short description of the columns is given below.

Variable	Description	Format
Sire	Sire id	Integer
Dam	Dam id	Integer
N Mating	Number of mating, numbered from 1 to number of matings	Integer
Nm	Mating number for sire	Integer
Nf(m)	Mating number of dam within sire	Integer
EBV	Estimated Breeding value	Real
F	Coefficient of inbreeding of offspring from mating between sire and dam	Real

vF	Mendelian sampling variance factor for offspring, that is mendelian variance is $v_f \cdot V_a$, where V_a is genetic variance. VF is between 0 and $\frac{1}{2}$	Real
Sire Text	field with an optional text string from the input file	Characters
Dam Text	field with an optional text string from the input file	Characters

male_list.txt

A list of males, the number of matings per male and the optional information on males (given in the input data file).

Variable	Description	Format
Sire	Sire id	Integer
N.matings	Number of matings for sire	Integer
Prev.offspr	Number of offspring in last and previous timesteps	Integer
Merit	Genetic merit (breeding value) of candidate	Real
Rel.males	Average relationship to male candidates	Real
Rel.females	Average relationship to female candidates	Real
Rel.males.sel	Average relationship to selected males	Real
Rel.females.sel	Average relationship to selected females	Real
Sire.txt	Field with an optional text string from the input file	Character

Candidates.txt

A list of all candidates, their previous contributions, contributions to the next cohort and their relationships to other candidates.

Variable	Description	Format
ID	Candidate id	Integer

Sex	Sex of candidate	Integer
N.matings	Number of matings for candidate	Integer
Prev.offspr	Number of offspring in last and previous timesteps	Integer
Merit	Genetic merit (breeding value) of candidate	Real
Rel.males	Average relationship to male candidates	Real
Rel.females	Average relationship to female candidates	Real
Rel.males.sel	Average relationship to selected males	Real
Rel.females.sel	Average relationship to selected females	Real
Txt	field with an optional text string from the input file	Character

Optional output files

In addition, output files are produced if requested in the parameter file

- File with recoded IDs
- File with additive genetic relationships

Introduction to Evolutionary Algorithms

An Evolutionary Algorithm mimics the way natural selection shapes populations by favouring the fittest individuals. Thus, the vocabulary is the same as used in evolutionary genetics.

The principle is that one has a set of possible solutions, called a population. Individual solutions from this population can be evaluated in terms of the criterion that is to be maximized. This gives their fitness. Individuals are then selected to be parents proportional to their fitness. The possible solutions then generate new solutions mixing two parental solutions by recombination. In addition, additional variation is generated by mutation. This generates new offspring that then competes with the parents to survive until the next generation. The probability of survival is proportional to their fitness.

This ensures incremental adaptation relative to the criterion to be maximized. The best solution observed is always maintained in the population.

Evolutionary algorithms are a class of algorithms based on biological evolution. For a thorough description of evolutionary algorithms see:

Michalewicz, Z. 1996. Genetic Algorithms + Data Structures = Evolution Programs. Springer Verlag.

Alternatively, take a look at the Wikipedia entry:

http://en.wikipedia.org/wiki/Evolutionary_algorithm

Optimisation criteria

Genetic contributions that optimise two different criteria can be computed by EVA.

The first (default) criterion is a linear function of genetic merit and average relationship of the last generation of which the next cohort is a part.

With discrete generations this simplifies to

$$H_1 = w_{merit} \cdot c' \hat{a} + w_{relationship} \cdot c' A c$$

Here w_{merit} and $w_{relationship}$ are input parameters (weights), \hat{a} is a vector of estimated breeding values (merit) provided in the input data, A is a matrix of relationships based on either

- a) Pedigree, computed from the input data file
- b) Relationships read from file (e.g. genomic relationships)

And c is the vector of genetic contributions that are optimized.

The second criterion maximizes genetic merit conditional on the rate of inbreeding being lower than a specified threshold. With discrete generations this simplifies to

$$H_2 = \max\{c' \hat{a} \mid c' A c < C_t\}$$

where C_t is a function of the user specified constraint on rate of inbreeding and average co-ancestry in the previous generation. Thus, for both criteria the long-term rate of inbreeding is controlled (penalized or constrained).

With overlapping generations, the above criteria extends to include elements of relationships in and contributions to previous cohorts to account for their contribution to the last generation.

For a more thorough description see:

Woolliams JA, Berg P, Dagnachew BS and Meuwissen THE. 2015. Genetic contributions and their optimization. *Journal of Animal Breeding and Genetics* 132(2):89–99. DOI: 10.1111/jbgs.12148

Henryon M, Ostersen T, Ask B, Sørensen AC and Berg P. 2015. Most of the benefits from optimum-contribution selection can be realised with restrictions imposed. *Genetics Selection Evolution* 2015, 47:21. doi:10.1186/s12711-015-0107-7

Grundy, B., Villanueva, B. and Woolliams, J.A. (2000) *Anim. Sci.* 70: 373-382

Meuwissen, T.H.E. (1997) *J. Anim. Sci.* 75:934-940

Meuwissen THE & Sonesson A. 1998. *J. Anim. Sci.* 1998. 76:2575–2583

Citation

Please cite the following paper, if you use EVA:

Henryon, M., Ostersen, T., Ask, B. Sørensen, AC., Berg, P. Most of the long-term genetic gain from optimum-contribution selection can be realised with restrictions imposed during optimisation. *Genet Sel Evol* 47, 21 (2015) doi:10.1186/s12711-015-0107-7