



Instructions for MateSel

Brian and Sandy Kinghorn

Matesel is used to drive your breeding program. It makes selection and mating recommendations that can be used for a wide range of activities associated with breeding programs. By default, it gives mating lists that show Optimal Contributions Selection – balancing genetic gain and genetic diversity - but with deviations according to attention paid to potentially many other technical and logistical issues, as dictated by the user. See [An introduction to Mate Selection](#) for more detail on what MateSel does.

These instructions cover operation of the three versions of MateSel:

Windows version:	Windows desktop version
Web version:	Web browser version
Batch version:	Command line batch version for Windows and Linux

The first two of these provide a graphical user interface (GUI) that enables editing of parameters or “changing the goal posts” during the run, providing opportunity to discover what can be achieved across the many issues involved. They also help illustrate outcomes that are not desirable.

The Batch version is useful for widespread and/or very frequent runs, where parameter settings are thought to be appropriate and there is little time to spend on each run.

The interfaces for the Windows and Web versions are similar. Most of the illustrations in this help file come from the Windows version, and are similar to what you should see in the Web version.

Table of Contents

FIRST LOG IN AND ACCESSING THE SOFTWARE	6
EXPIRY DATE	6
RUNNING MATESEL ON YOUR WINDOWS DESKTOP	7
RUNNING MATESEL FROM YOUR WEB BROWSER.....	9
RUNNING THE BATCH VERSION OF MATESEL.....	12
STARTING THE RUN.....	12
DURING THE RUN.....	13
INPUT FILE REQUIREMENTS	14
THE MAIN DATA FILE	14
<i>File name</i>	14
<i>Fields</i>	14
<i>Missing information</i>	15
<i>Extra fields required for MateSel</i>	16
Index	16
Maxuse.....	16
Sex.....	17
MatingGroup.....	17
<i>Minimum use fields in the main data file (optional)</i>	17
Minuse	17
AbsMinuse	17
<i>Reserved field names</i>	18
<i>Trait fields in the main data file</i>	18
<i>Marker fields in the main data file</i>	19
Example field names	20
Missing genotypes	21
<i>Handling sex-linked markers</i>	21
<i>Handling multiple lethal recessives</i>	22
MATESEL.INI – INITIAL PARAMETERS AND CONDITIONS	23
<i>Parameters to manipulate Inbreeding, Coancestries, Traits and Markers</i>	23
<i>Automatic upgrading from a previous format</i>	24
<i>Example of MateSel.ini</i>	24
NRM limit	26
GROUPING FILES	26
<i>InpOneGroup.txt - the Not-Grouping file</i>	26
<i>InpGroups.txt - the Grouping file</i>	26
Why do this sub-grouping?	27
Targeting specific numbers of matings per group combination with a weighting.....	29
Forcing specific numbers of matings per group combination.....	30
Should I Target or Force specific numbers of matings per group combination?	31
Targeting minimum use of individual males within each female group	31
SETTING UP THE GRM	33
<i>How Matesel imputes relationships missing in the GRM supplied</i>	35
<i>Old method of setting up the GRM (still supported)</i>	36
IMPORTING YOUR OWN SOLUTION FOR PLOTTING	37
THE MATESEL USER INTERFACE	39
PARAMETER AND SETTINGS TEXTBOXES	39
<i>Balance Strategy</i>	39
<i>Max permissible coancestry, xAx/2</i>	41
<i>Target Degrees (TD)</i>	41
<i>Weight on progeny F</i>	41

Matings to be made	41
Grouping	41
Frontier degree gap.....	42
Frontier 0 degree generations.....	42
Frontier 90 degree generations.....	42
Frontier mid degree generations.....	42
Target % converged	42
% since last change	42
No change generations	42
Minimum generations.....	42
Maximum generations.....	43
CONTROL BUTTONS.....	43
THE TOOLBAR	43
THE MATESEL CONSOLE	43
THE FRONTIER GRAPH	45
THE PROGENY INBREEDING GRAPH	46
TOOLS FOR MANAGING OUTCOME DISTRIBUTIONS	47
Control Types for manipulating histograms.....	49
Tactical desired gains.....	49
Zooming into a chosen Trait or Marker histogram	50
VIEWING COMMITTED MATINGS AS WELL AS NEW MATINGS IN HISTOGRAMS.....	50
MANAGING THE PROGENY INBREEDING DISTRIBUTION	52
Interaction with Weight on Progeny F	53
MANAGING INDIVIDUAL COANCESTRY DISTRIBUTIONS	53
Why are there no	54
MANAGING TRAIT DISTRIBUTIONS	54
Missing trait information	55
MANAGING GENETIC MARKERS.....	56
Missing genotypes.....	58
Panning across marker loci	58
Hiding histograms	58
THRESHOLD MANAGEMENT FOR PROGENY INBREEDING AND OTHER FACTORS	59
Dynamic Threshold Management for Progeny Inbreeding	59
To invoke Dynamic Threshold Management in Matesel.ini.....	61
Direct Threshold Compliance across Traits, Markers and more.....	61
THE MATE SELECTIONS TAB	62
OUTPUT FILES	62
HOW DO I ... ?	64
GET WHAT I WANT – TACTICAL DESIRED GAINS	64
BOOSTING FEMALE REPRODUCTION: MOET AND IVF	65
MULTIPLE ENDUSES: HANDLING DIFFERENT TARGET END-USES	66
Results.....	68
Some ‘EndUse’ items to note:	68
ACCOMMODATING BISEXUALITY.....	69
Avoid matings that are duplicate, or reciprocal, or selfings.....	70
MANAGING BACKUP SIRES	70
Backup all matings of a sire as a block.....	71
Restricting backup sires to those that have not been selected	72
Use a male for backup even if it is at its MaxUse limit.....	73
Print backup matings to file(s) even if the Print Matings button is clicked.....	73
COMMITTED MATINGS: PRIOR CONTRIBUTIONS AND PRE-SET MATINGS.....	73
Background	73
Operation	76
Automatic setting of Maxuse - set Committed Matings switch to value 1	76
Manual setting of Maxuse - set Committed Matings switch to value 2.....	77

<i>Some rules</i>	77
<i>Trait and marker distributions under Committed Matings (from 6.9.11.5)</i>	78
<i>Other outputs</i>	78
MANAGE LOSS-OF-FUNCTION (LOF) LOCI	78
MANAGE COST OF SEEDSTOCK PURCHASE	78
MANAGE COST OF ASSEMBLING MATING GROUPS.....	79
<i>Hard implementation</i>	79
<i>Soft implementation</i>	79
<i>Invocation</i>	82
RANK THE NUMBER OF MATINGS ALLOCATED PER FULL SIB.....	82
TIPS AND TRICKS.....	83
HANDLING LARGE GROUPS OF INDIVIDUALS AS SINGLE “DUMMY INDIVIDUALS”.....	83
OPTIMAL CONTRIBUTIONS SELECTION WITHIN ONE SEX ALONE.....	84
ALLOCATE FEMALES TO GROUPS OF MALES FOR MATING – “MULTI-SIRE JOINING”	84
TEST THE IMPACT OF KEEPING MAXUSE VALUES LOW TO SAVE GENETIC DIVERSITY	85
DICTATE THE NUMBER OF SIRE TO BE USED	86
SPREAD SIRE USE MORE EVENLY OVER WEEKS IN PIGS AND OTHER SPECIES	86
CONSIDER ATTRIBUTES OF CANDIDATES OTHER THAN THEIR TRAITS.....	86
CONSTRAIN MATINGS ACCORDING TO PHENOTYPE – EG. HIGH CALVING EASE BULLS TO SMALL HEIFERS.....	87
MANAGE SIRE AND DAM INDICES OR EBVS DURING A RUN	89
MANAGE COST OF SEMEN AND OTHER SEEDSTOCK COSTS	90
TEST DIFFERENT PRESELECTION POLICIES	90
MANAGEMENT OF MULTIPLE GENE EDITS.....	90
INVOKE GENOMIC MATE SELECTION.....	91
TARGET SPECIFIC GENOMIC REGION(S) FOR MANAGEMENT OF DIVERSITY AND INBREEDING	92
REDUCE PROGENY INBREEDING WITH LITTLE OR NO IMPACT ON PROGENY INDEX AND PARENTAL COANCESTRY	93
MATE SELECTION ... OR SELECTION FOLLOWED BY MATE ALLOCATION?	93
MANAGE DISTRIBUTION OF MALES IN A SIRE REFERENCING SCHEME	93
GET MORE SPEED.....	94
COMMON MISTAKES	95
USE MATE SELECTION, NOT JUST MATE ALLOCATION!	95
“THE TWO INBREEDINGS”	95
SETTING AND WEIGHTING THRESHOLDS	96
INSUFFICIENT CONVERGENCE	96
DIAGNOSIS OF CONVERGENCE.....	97
CRITERIA FOR STOPPING	97
A PREDICTED ASYMPTOTIC MAXIMUM SOLUTION	98
IF THE PROGRAM CRASHES.....	101
IF YOU CANNOT CLOSE MATESEL WINDOWS.....	102
USING A REDSERVER.....	102
NON-REGISTERED PROJECTS.....	103
REGISTERED PROJECTS	103
SUBMITTING A PROJECT FOR REGISTRATION.....	103
SETTING UP MATESEL ON A WINDOWS SERVER	105
RUNNING DEMONSTRATION OVER THE WEB.....	105
THE SECURITY OF YOUR DATA	105
AN INTRODUCTION TO MATE SELECTION	106
REFERENCES.....	109
MATESEL LIMITS	111

NUMBER OF INDIVIDUALS IN THE PEDIGREE	111
NUMBER OF CANDIDATES	111
LENGTH OF ID	111
NUMBERS OF TRAITS, GENETIC MARKERS, GROUPS AND ENDUSES	111
VERSIONS OF MATESEL	112
VIDEO TUTORIALS AND DISCUSSIONS	112
NEW FEATURES AND BACKWARD COMPATIBILITY	115

First log in and accessing the software

The Matesel team will supply you with a username and password to log in at Matesel.com*. You can change your password once logged in.

According to your licence conditions, you will be granted access to one or more of the three versions: “Web”, “Windows” and “Batch”. After logging in, click the Download button, and download the zip file(s) that you need for the Windows and/or Batch version. Unzip these to a folder of your choosing on your computer.



Web Version: Once logged in you can proceed directly – see section [Running MateSel from your web browser](#).

Windows Version: Once logged in, click on the “Download” button and download the appropriate zip file to a folder of your choosing on your computer. Extract the contents of this folder and run Matesel.exe – see section [Running MateSel on your Windows Desktop](#).

Batch version: Once logged in, click on the “Download” button and download the appropriate zip file to a folder of your choosing on your computer. Extract the contents of this folder. See section [Running the Batch version of MateSel](#).

When you make a run, the [Matesel console](#) (or the file Console.txt under Batch mode) reports when the software you have downloaded will expire. This is governed by the licence that you operate under as well as the version of Matesel that you have downloaded. You will be prompted to download a new version of the software, as recommended or required.

Note that all versions of MateSel require Internet connection to operate.

- * If you do not have a Matesel account to log into, you can still download this Instruction file plus examples and exercises, by clicking on the “Documentation” button on the login screen at Matesel.com.

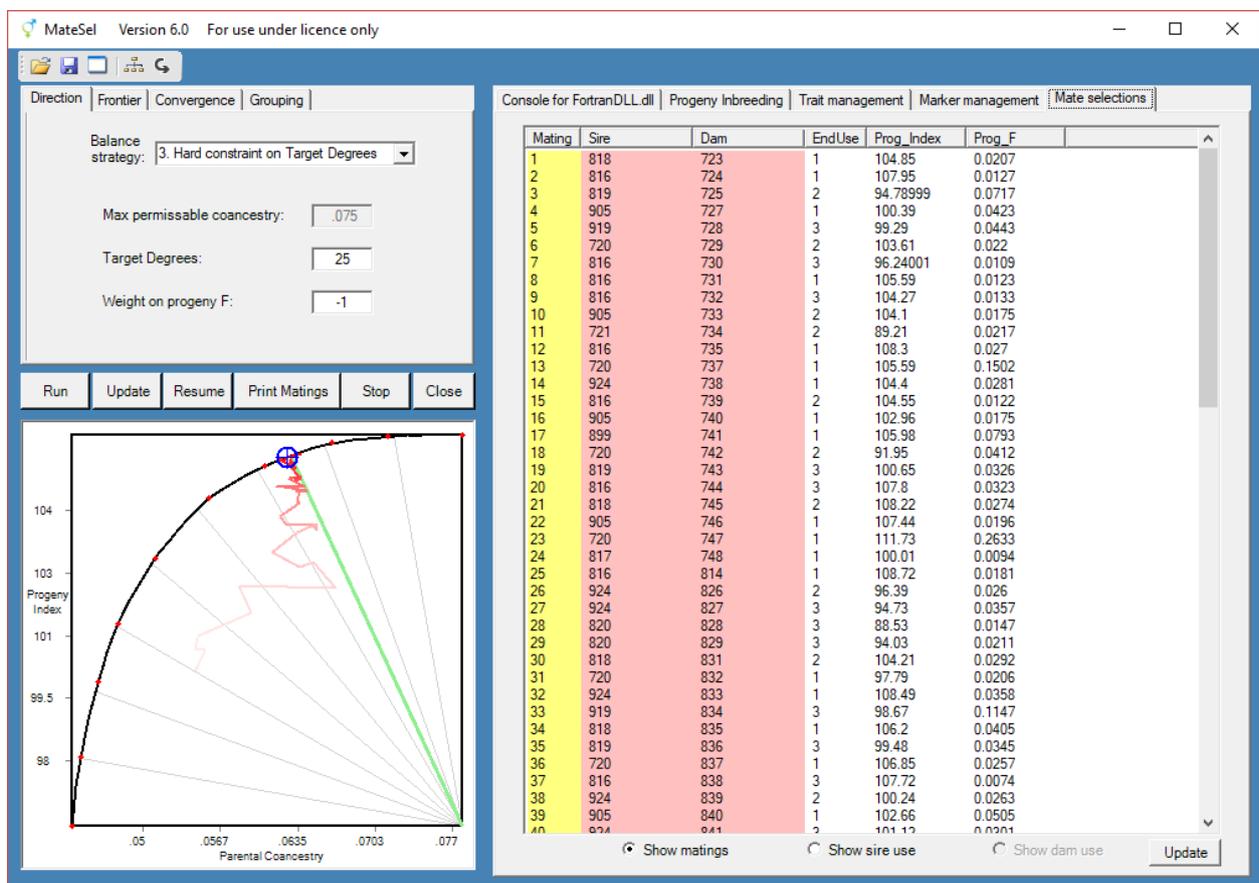
Expiry date

Expiry date is shown near the top of the Console tab, or Console.txt for batch versions. The expiry date is the earlier of your licence expiry and the expiry of the version of the software you are currently using.

Running MateSel on your Windows Desktop

Getting started quickly

An example input file called Matesel.txt is included in your installation. You can use this file to try out MateSel. Run Matesel.exe, enter your username and password on the login window, then open Matesel.txt, then click the “Run” button.



The screenshot shows the MateSel software interface. On the left, there is a control panel with the following settings:

- Balance strategy: 3. Hard constraint on Target Degrees
- Max permissible coancestry: .075
- Target Degrees: 25
- Weight on progeny F: -1

Below the control panel is a graph showing Progeny Index (Y-axis, ranging from 98 to 104) versus Parental Coancestry (X-axis, ranging from .05 to .077). The graph displays a curve and several lines representing different mating scenarios.

On the right, there is a console window titled "Console for FortranDLL.dll" with tabs for Progeny Inbreeding, Trait management, Marker management, and Mate selections. The console displays a table of mating data:

Mating	Sire	Dam	EndUse	Prog_Index	Prog_F
1	818	723	1	104.85	0.0207
2	816	724	1	107.95	0.0127
3	819	725	2	94.78999	0.0717
4	905	727	1	100.39	0.0423
5	919	728	3	99.29	0.0443
6	720	729	2	103.61	0.022
7	816	730	3	96.24001	0.0109
8	816	731	1	105.59	0.0123
9	816	732	3	104.27	0.0133
10	905	733	2	104.1	0.0175
11	721	734	2	89.21	0.0217
12	816	735	1	108.3	0.027
13	720	737	1	105.59	0.1502
14	924	738	1	104.4	0.0281
15	816	739	2	104.55	0.0122
16	905	740	1	102.96	0.0175
17	899	741	1	105.98	0.0793
18	720	742	2	91.95	0.0412
19	819	743	3	100.65	0.0326
20	816	744	3	107.8	0.0323
21	818	745	2	108.22	0.0274
22	905	746	1	107.44	0.0196
23	720	747	1	111.73	0.2633
24	817	748	1	100.01	0.0094
25	816	814	1	108.72	0.0181
26	924	826	2	96.39	0.026
27	924	827	3	94.73	0.0357
28	820	828	3	88.53	0.0147
29	820	829	3	94.03	0.0211
30	818	831	2	104.21	0.0292
31	720	832	1	97.79	0.0206
32	924	833	1	108.49	0.0358
33	919	834	3	98.67	0.1147
34	818	835	1	106.2	0.0405
35	819	836	3	99.48	0.0345
36	720	837	1	106.85	0.0257
37	816	838	3	107.72	0.0074
38	924	839	2	100.24	0.0263
39	905	840	1	102.66	0.0505
40	924	841	2	101.12	0.0201

File locations

You can install the MateSel software in any folder.

All input files, output files and error reporting files reside in the same folder as the main data file. This means that you can have a separate folder for each job, if you wish, and run each from the same instance of MateSel.exe.

First be sure that you have the following files in the folder where your datafile resides:



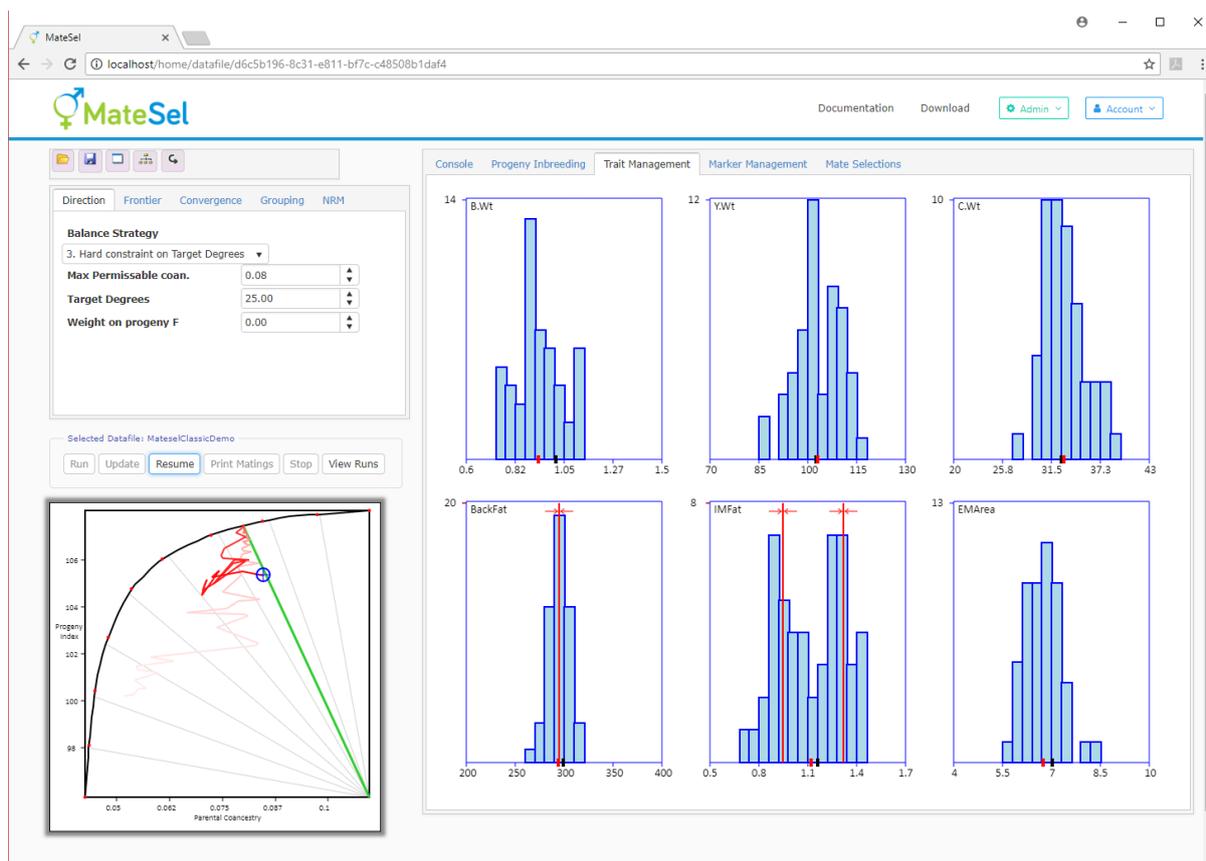
<any filename >	Your main data file
InpOneGroup.txt	If you are not using grouping
InpGroups.txt	If you are using grouping
Matesel.ini	Default start-up values for this run
EndUses.txt	(Optional) To supply information for Multiple EndUses
ImportSolution.txt	(Optional) Your own solution to view on the interface
CommittedMatings.txt	(Optional) Matings you insist must be made.

These files are described in detail in following sections.

To run the program, launch Matesel.exe. You are now ready to login, then open a main data file and edit parameters or simply hit “Run”.

Running MateSel from your web browser

[Service providers: See [Setting up Matesel on a Windows server](#). Please note that hosting of the Web version of MateSel is only carried out by BSK and those customers that enter a special agreement for service provision.]



First prepare a zip compressed file containing the following files:

- | | |
|----------------------|---|
| <ProjectName>.txt | The main data file, where <ProjectName> is your chosen project name |
| InpOneGroup.txt | If you are not using grouping |
| InpGroups.txt | If you are using grouping |
| Matesel.ini | Default start-up values for this run |
| EndUses.txt | (Optional) To supply information for Multiple EndUses |
| ImportSolution.txt | (Optional) Your own solution to view on the interface |
| CandGRM.txt | (Optional) Genomic relationships among candidates |
| CommittedMatings.txt | (Optional) Matings you insist must be made. |

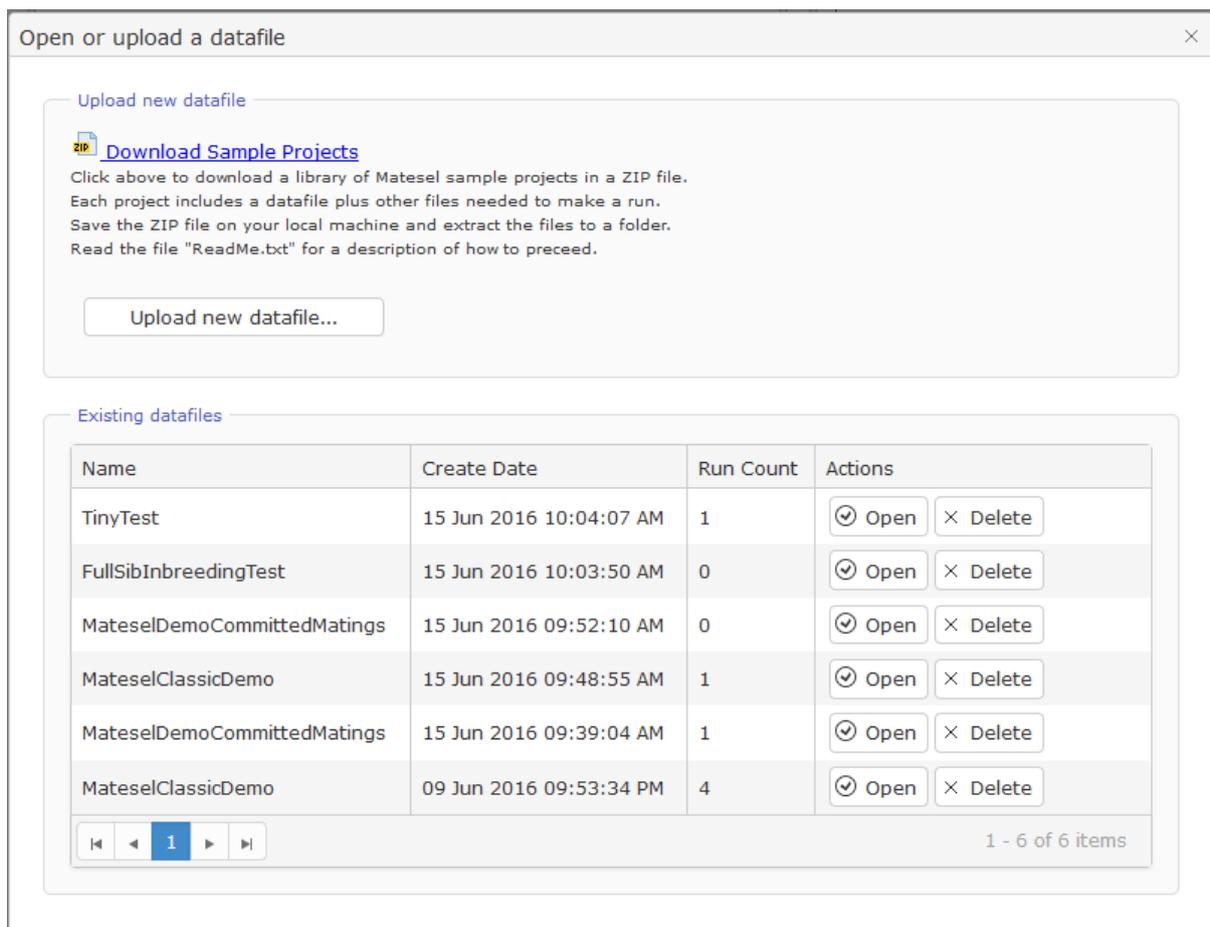
Note that for the Web version, unlike the Windows and Batch versions:

- All these files must be present except for those marked optional. Eg. Even if you are not grouping you need InpGroups.txt. It could be an empty file as it will not be read, but in the current version it must exist in your zip file.
- <ProjectName> is the name you allocate to this project, and this must be named exactly the same as the zip file name. For example MyMateselDatafile.txt would be contained in MyMateselDatafile.zip
- Name the zip file with your chosen project name, as indicated in the item above.
- If you are using a Research, Education and Demonstration server (a REDserver), please read the section [Using a REDserver](#).

If you do not already have such files, you can download a library of sample projects (see “Download Sample Datafiles” in the next illustration).

Your service provider will give you a web address to access MateSel. Logon using the information that has been provided to you.

When the MateSel screen appears, click on the Open Folder icon . You should see a floating window something like this:



Open or upload a datafile

Upload new datafile

 [Download Sample Projects](#)

Click above to download a library of Matesel sample projects in a ZIP file. Each project includes a datafile plus other files needed to make a run. Save the ZIP file on your local machine and extract the files to a folder. Read the file "ReadMe.txt" for a description of how to proceed.

Upload new datafile...

Existing datafiles

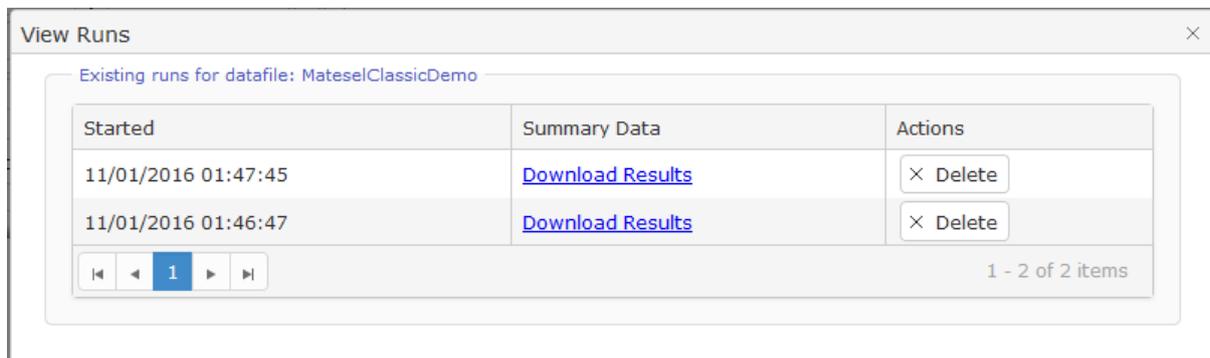
Name	Create Date	Run Count	Actions
TinyTest	15 Jun 2016 10:04:07 AM	1	<input checked="" type="checkbox"/> Open <input type="checkbox"/> Delete
FullSibInbreedingTest	15 Jun 2016 10:03:50 AM	0	<input checked="" type="checkbox"/> Open <input type="checkbox"/> Delete
MateselDemoCommittedMatings	15 Jun 2016 09:52:10 AM	0	<input checked="" type="checkbox"/> Open <input type="checkbox"/> Delete
MateselClassicDemo	15 Jun 2016 09:48:55 AM	1	<input checked="" type="checkbox"/> Open <input type="checkbox"/> Delete
MateselDemoCommittedMatings	15 Jun 2016 09:39:04 AM	1	<input checked="" type="checkbox"/> Open <input type="checkbox"/> Delete
MateselClassicDemo	09 Jun 2016 09:53:34 PM	4	<input checked="" type="checkbox"/> Open <input type="checkbox"/> Delete

1 - 6 of 6 items

If there are no datafiles to select (or if you want a new datafile to be used), then click on “Upload new datafile ...”, then navigate to and select your zip file containing the datafile and associated files, previously prepared on your computer. Then click “Upload files” to send these files to the Web Server. You can “Upload new datafile ...” many times for the same Project Name, and these will be listed using the same name but distinguished by different create dates. This way you can store different run conditions as dictated by files such as Matesel.ini and Groups.txt.

You can now click “Open” for your chosen datafile, and this will bring you back to the Web GUI ready to edit parameters or simply hit “Run”. During the run, do not close the browser tab that Matesel is running in, and if you click a link on the page during a run, The Right-Click and “Open in a new tab”, to avoid closing your Matesel Run. If this does happen, Matesel will attempt to reconnect you automatically.

When you finish your run you can click “View Runs” and then “Download Results” to show the output files in a file named Out.zip. This file contains all the output files as listed under section [Output Files](#).

A screenshot of a web application window titled "View Runs" with a close button (X) in the top right corner. Below the title bar, it says "Existing runs for datafile: MateselClassicDemo". The main content is a table with three columns: "Started", "Summary Data", and "Actions". There are two rows of data. The first row has "11/01/2016 01:47:45" in the "Started" column, a blue hyperlink "Download Results" in the "Summary Data" column, and a button with an "X" icon and the text "Delete" in the "Actions" column. The second row has "11/01/2016 01:46:47" in the "Started" column, a blue hyperlink "Download Results" in the "Summary Data" column, and a button with an "X" icon and the text "Delete" in the "Actions" column. Below the table is a pagination control with a blue box containing the number "1" and the text "1 - 2 of 2 items" on the right.

Running the Batch version of MateSel

Starting the run

Ensure that all the installation files are placed in the same directory or folder. For Linux, the Linux executable file is the only installation file required. In both cases, the Licence file RunSeeds.txt should also be placed in the installation folder.

You can put the Matesel installation folder in the system PATH, so that you can run Matesel from wherever your data folder is. The following are illustrative examples of launching the batch version (Windows then Linux; colours are for ease of reading alone):

Run from Installation folder when your project files are in that same folder:

```
C:\Installation>MateSelBatch.exe MateselClassicDemo.txt  
~/Installation$ Matesel113Linux MateselClassicDemo.txt
```

Run from Installation folder when your project files are in a different folder:

```
C:\Installation> MateSelBatch.exe C:\Project\MateselClassicDemo.txt  
~/Installation$ ./Matesel113Linux /home/Project/MateselClassicDemo.txt
```

Run from your project folder when the installation is in a different folder:

```
C:\Project>C:\Installation\MateSelBatch.exe MateselClassicDemo.txt  
~/Project$ /home/Installation/Matesel113Linux MateselClassicDemo.txt
```

Run from your project folder when the executable folder is in the system PATH:

```
C:\Project>MateSelBatch.exe MateselClassicDemo.txt  
~/Project$ Matesel113Linux MateselClassicDemo.txt
```

If no filename is supplied as an argument (such as MateselClassicDemo.txt above), then name Matesel.txt will be taken by default.

Windows users can create their Licence file simply by logging into the Windows version using Matesel.exe launched from the same folder as MateSelBatch.exe. Without this Licence file, you will get an error message in Console.txt, and no results. Otherwise you will see your software expiry date in Console.txt. The expiry date depends on your licence renewal status and the version of the software you have. Under normal circumstances, your authentication will be refreshed once per year.

Linux users should request a licence file “RunSeeds.txt” from us. [Or possibly: Follow the method for Windows above, running the Windows Desktop version using a Windows emulator on your Linux machine. Copy the generated RunSeeds.txt to the Linux executable’s folder.]

Keep an eye on your expiry date at the top of the Console.txt file, and renew your Licence file as needed.

Unexpected expiry notice? ... In some cases, Matesel has to search the system Path for the first instance of RunSeeds.txt, and the resulting location is shown near the top of the console. If you get an unexpected expiry notice, then this could be an old Licence file that should be deleted or removed from the path.

During the run

You will see some progress of the run on your screen. If you want more detail, open the file Console.txt in a non-locking viewer/editor such as TextPad (non-locking so that MateSel can still write to this file) and view it as it updates. The run will come to an end when convergence criteria have been met.

You can stop the run early by executing the provided file MateSelBatchStop.exe (or the corresponding Linux executable file MateselBatchStopLinux {enter ./MateselBatchStopLinux at the command prompt}) ***in the same folder as the main data file that you are currently running*** – ie. copy the MateselBatchStop executable to your project folder and run it from there.

This stop is not immediate ... you can keep an eye on Console.txt to see the stop taking place. Matesel only picks up the signal to stop after the Frontier has started to build, and only when it reports a result to the console, every n_{see} generations (n_{see} can be edited in Matesel.ini), and even then Matesel needs to take some time to stop gracefully.

Output files all start with “Out ..”. You can use OutSeqFile.txt to match results from the Fortran DLL with your ID’s.

Input file requirements

The main data file

Your data file contains the candidates for selection and as much of their pedigree as you choose to handle (However, see the section [Matesel limits](#)). You need to include individuals that are not candidates but that are ancestors of candidates if you want to fully account for coancestry when managing genetic diversity. However, even with no pedigree, coancestry is accommodated as much as is possible - equivalent to using classical concepts of effective population size.

MateSel uses any standard Pedigree Viewer file format, as described here. However, there are at least three extra fields required for Matesel, as described further below.

File name

For the Windows and Batch versions the main data file can have any name and extension, but the Windows open file dialog box by default shows files with extensions txt, csv, dat and ped (eg. Example.ped, Example.txt, Example.dat and Example.csv). For the Web version the main datafile is to be named <ProjectName>.txt (see [Running MateSel from your web browser](#))

Fields

The minimum requirement is a text file with free format fields for:

Individual identification (ID)

Sire ID

Dam ID

<Other fields>

The ID fields can be numeric or alphanumeric, and they must be the first three fields, in the order shown above. However, you can name them as you like, as in the examples below. The other fields can be in any order, and the minimum requirement for these is described under [Extra fields required for MateSel](#). The examples given below do not include these compulsory fields, for simple illustration of formatting.

The maximum length for any ID field is 20 characters. However, this constraint only applies when “Committed Matings” is being used, or a GRM file is being read in.

A single header line is needed to declare the contents of all fields.

Fields can be delimited by commas or tabs or spaces. The program parses the header line to discover what delimiter is being used. Use only one type of delimiter throughout the file - commas OR tabs OR spaces.

Many spaces together are read as one field separator, whereas many commas together (or many tabs together) are read as many fields, with null fields between delimiters.

For space delimited files, any one record containing legitimate spaces must be bounded by double quotes eg. "Body weight", and missing character information is represented as "".

Here is example file, using (multiple) spaces as delimiters:

```
Individual Father Mother Weight
George          0      0      67
Daisy          George   Jean    72
Karen          George   Jean    43
Butch          George   Daisey  143
Fred           Butch    Karen   69
```

And here is a similar file, which would show a similar pedigree diagram:

```
ID           Sire   Dam   "Body weight"
12           0     0     67
23           12    2     72
5            12    2     "Not much"
"B 2"        12    23    Heavy
"Young Fred" "B 2"  5     69
```

Notice that these free-format fields can be numeric, alphanumeric, or a mixture of both - each field can contain both numeric and character records. Any number of fields can be added.

Here is the same file, but with commas as delimiters

```
ID, SireID, DamID, Body weight
12, 0, 0, 67
23, 12, 2, 72
5, 12, 2, Not much
B 2, 12, 23, Heavy
Young Fred, B 2, 5, 69
```

For all fields, leading and trailing spaces are ignored. Character fields are case sensitive, such that "bert" and "Bert" are recognised as different individuals. These fields are detected by the program, which then disables the facility for shading on merit for these fields.

The program detects the number of records (lines) in the data file.

Missing information Unknown parents are denoted as 0 or # or *. They can also be noted as a space or a null entry, which is, with adjacent delimiters:

```
Tab delimited file:   <tab><space><tab> or <tab><tab> .
Comma delimited file: ,<space>, or ,,
Space delimited file: <space>""<space>""<space> or <space>""<space>
```

Base parents not included as individuals (eg. Jean in the first example), are recognised and entered in the pedigree. If an individual's parent appears as an individual in the file, then it should ideally appear before (above) the individual in the file, as this marginally improves the speed of loading the file. However, the program will automatically re-order the pedigree file as required.

Missing numerical trait and index information should be given as a decimal point (.), as the value 0 will be taken as an observed value.

Extra fields required for MateSel

Index

A field called “Index”. This should contain the selection index value for each individual, which will typically be a multi-trait BLUP index value or other similar selection index value that is a measure of genetic merit across traits. It can be simply the EBV for the single trait of interest.

Maxuse

A field called “Maxuse”. (“Cand”, “Candidate” and “CandStat” are also read as Maxuse, for backward compatibility). This is the maximum value for number of matings permitted for each candidate. For example, Maxuse = 1 mating for natural mating females, Maxuse =30 matings for natural mating bulls, and possibly Maxuse =1,000 matings for AI bulls, or the number of semen doses left for a deceased bull.

The Maxuse field should always contain a zero (“0”) for each individual that is not a candidate for selection. Candidates can be entered simply with a “1” in this column – just to signify candidacy. Alternatively, a custom Maxuse value can be entered for each candidate. In either case behaviour is dictated by the parameter file [InpOneGroup.txt](#), or [InpGroups.txt](#) if Grouping:

1. Read from the parameter file: Maxuse is set for each sex and Group, such that all individuals within a sex/Group combination have the same Maxuse.
2. Read from main datafile: If a Maxuse value entered in the parameter file is set to -1, then the Maxuse values for individuals in that sex and Group are read from the datafile. This means that you can set different Maxuse values for each individual within one or more nominated sex and Group combinations.

[Minuse and AbsMinuse](#), described later, are read in this same way.

Backward compatibility:

The old way of handling Maxuse, Minuse and AbsMinuse (Fortran DLL version 12.1 and before) is still supported – it is invoked if there is no “-1” in this line of [InpOneGroup.txt](#), or [InpGroups.txt](#), as would be the case for all previous projects:

```
Male ABS-Minuse numbers (Enter -1 to read from the data file)
```

If you do not have a “-1” in that line then the old method will be invoked and you will likely have a problem!

It is *not* recommended that you set Maxuse values low with the intention of conserving genetic diversity. Let Optimal Contributions Selection manage that through your chosen [Balance Strategy](#). Better to set Maxuse according to reasonable biological and/or logistical limits. For discussion see [Test the impact of keeping Maxuse values low to save genetic diversity](#).

The treatment for bisexual candidates is a bit different. See [Accommodating bisexuality](#).

Sex

A field called “Sex” or “Gender” that shows the sex of individuals, with “1” or “m” or “M” for male, and “2” or “f” or “F” for female. These values are inconsequential for non-candidates.

MatingGroup

ONLY REQUIRED IF GROUPING IS SET TO TRUE: A field called either “MatingGroup or Mgroup”. This field must contain integers that represent group membership. Entries for non-candidates can be set to zero – they are not used. But do not use zero as a Group number for candidates. It is reserved for internal use. Groups are different for each sex, even if they have the same label (eg. Group 6 for males is treated differently from group 6 for females), so you can use the same numbers across sexes.

The fields names noted above are not case-sensitive.

Minimum use fields in the main data file (optional)

Minuse

A field called “Minuse”: The minimum value for number of matings given that there will be some use. For example, if a bull is to be selected for natural mating, we might specify a minimum female group size of Minuse= 15 for that bull, as mating groups of less than this size are not acceptable to the breeder, for good reasons. In this case number of matings = 0 is permitted, as are number of matings ≥ 15 .

AbsMinuse

A field called “AbsMinuse”: The absolute minimum number of matings. This is generally zero, but may be set higher, for example where a breeder has a given number of doses of semen available for a favoured bull, and insists that these should all be used.

These two field names are not case-sensitive, and they can be simply left out of the main data file if desired.

You can set Minuse and AbsMinuse values in the data file or in one of the parameter files [InpOneGroup.txt](#) and [InpGroups.txt](#), in the same way as is done for Maxuse values, above. If entries for these parameters are “-1” (read from data file) in the parameter file, and they have no columns in the datafile, their values will all be set to zero.

Warning notes:

- It does not make sense to specify non-zero values for both Minuse and AbsMinuse for any one individual or group. Eg. If AbsMinuse is 1 and Minuse is 10, then this effectively makes AbsMinuse=10, as 1 mating is illegal under Minuse=10. If you do

specify both, then one will take precedence, but which one depends on several factors, including grouping strategy.

- Take care if setting Minuse equal to Maxuse. For example, if for males you set Minuse=Maxuse=10 and you ask for 55 matings, there is no valid solution and an error will be reported. Moreover, by setting equal contributions, you will be losing much of the potential benefit of optimal contributions selection.

The treatment for bisexual candidates is a bit different. See [Accommodating bisexuality](#).

Reserved field names

The following field names are reserved and cannot be used to denote Traits:

[Index](#)

[Maxuse](#) (Cand, Candidate and CandStat are also reserved)

[Minuse](#)

[AbsMinuse](#)

[Sex](#) (Gender is also reserved)

[MatingGroup](#) (Mgroup is also reserved)

Any field starting with [g_](#) (g<underscore>)

Any field starting with [gp_](#)

[Location](#)

[LethalA](#)

[LethalG](#)

With the exception of the last three above, reserved fields names are not case-sensitive. Do not use more than one fieldname in a grouping (eg. if you have a field “Sex”, you cannot have a field “Gender”).

Trait fields in the main data file

All fields are taken to be Trait fields with the exception of:

- The first three ID fields
- Fields recognised as genetic marker fields (See [Marker fields in the main data file](#))
- Fields with [reserved field names](#).

Do not use Trait names starting with LethalA or LethalG. See [Handling multiple lethal recessives](#).

This means that fields such as Age, DoB, SemenDoseCost and YoB are treated as traits. But this is OK – it can be useful for information, and also for [intentional manipulation](#). For example, you can make constraints on YoB and DoB of selected individuals by using the [Trait Management tool](#) to manage trait distributions. See [here](#) for more details.

Marker fields in the main data file

The field names for marker information have to be constructed very carefully, because the core code uses these names to understand what data are being presented to it. Each field name has a number of sections, separated by underscores.

For example, the field `gp_2_4_3_M4-1_2` conveys that the field contains **genotype probabilities** for a **2**-allele marker with **4** genotypes considered (genotypes 11, 12, 21 and 22), and these probabilities are for the **3rd** genotype, marker name is “**M4-**” and the allele names are “**1**” and “**2**”.

The following table describes these sections, and how to use them:

Section	Content
Type	G or g for a genotype presented as an integer. In this case there is one field per marker. GP or gp for a genotype probability. In this case there are multiple fields per marker.
Number of alleles, nA	Usually 2, but any number > 2 is accepted.
Number of genotypes [This is the number of genotypes that could be present, including lethals, even if not all genotypes are in your dataset. So if you have no 22 genotypes in your dataset, and you are blocking 12 and 21 together as one genotype, then these are the genotypes in this list: {11, (12 or 21), 22}, and the number of genotypes is 3, not 4.]	<p>This has to be either nA^2 (all genotypes) if direction of inheritance is handled, or $nA*(nA+1)/2$ if heterozygotes are not split (ie. One probability for a heterozygote, not two (eg. p12 and p21)).</p> <p>For this and the previous section together, the legal entries denoting genotype in the data column are as shown below:</p> <pre> 2_3 1 2 3 2_4 1 2 3 4 3_6 1 2 3 4 5 6 3_9 1 2 3 4 5 6 7 8 9 </pre> <p>So for 2 alleles and 3 genotypes (2_3) the genotypes to enter as data are 1 for 11, 2 for (12 or 21) and 3 for p(22). For 2 alleles and 4 genotypes (2_4) the genotypes are 1 for 11, 2 for 12, 3 for 21 and 4 for 22.</p>
Genotype number	This is only included if type is GP. This is the number of the genotype to which the probability belongs, as in the examples above for 2 and 3 alleles and the two possible numbers of genotypes for each of these.

	<p>So for 2 alleles and 3 genotypes (2_3) the genotype numbers are 1 for p(11), 2 for p(heterozygote) and 3 for (22), and the data columns contain probabilities ranging 0 to 1.</p> <p>Multiple fields for a given marker should be grouped together in the data file, and presented sequentially according to genotype number. Genotype number 1 can be omitted, as noted below.</p>
Marker name	I append a dash, as in the examples, because marker name is usually followed directly by the name of one of its alleles in graphical output. Do not use a space in a name. It is critical that the multiple genotype probability fields for a single marker locus use exactly the same marker name.
Allele 1 name	This might usually be an integer, but alphanumeric of any length are OK. Do not use a space in a name.
Allele 2 name	As above
<Other allele names >	As above, name all nA alleles.

For genotype probabilities (gp_ ...), where the number of genotypes is n , there can be n fields for that marker, with values summing to 1 for each individual (a checksum is carried out, with some allowance for rounding error), but it is also possible to omit the field for the first genotype, giving $n-1$ fields for the marker. In the latter case, the code detects the omission, and the probability for the first marker is generated in the code.

The core MateSel code converts all genotype and genotype probability information to handle genotype probabilities for all nA^2 genotypes at each marker locus. This is appropriate, as even where candidate genotypes are specified, the results for prospective progeny have to be given as probabilities (eg. a mating 12 x 12 gives progeny probabilities of $\frac{1}{4}$ for each genotype: 11, 12, 21, 22). Moreover, even if supplied candidate genotypes or genotype probabilities do not show direction of inheritance for heterozygotes (eg. 12 versus 21), such information can still be generated for prospective progeny (eg. progeny from a 11 x 22 mating are all 12, and none are 21).

Example field names

Two biallelic markers presented as genotypes:

g_2_3_M1-_1_2
and
g_2_4_M2-_1_2

M1- has just three genotypes presented (data entries of 1, 2 and 3 for genotypes 11, 12 or 21, and 22).

M2- has all four genotypes presented (data entries of 1, 2, 3 and 4 for genotypes 11, 12, 21 and 22).

Two biallelic markers presented as genotype probabilities:

gp_2_4_1_M3-_1_2
 gp_2_4_2_M3-_1_2
 gp_2_4_3_M3-_1_2
 gp_2_4_4_M3-_1_2

and

gp_2_4_2_M4-_1_2
 gp_2_4_3_M4-_1_2
 gp_2_4_4_M4-_1_2

For M3-, all four genotype probabilities are included, but for M4- the first probability is not included, such that it will be deduced in the core code.

Missing genotypes

Genotypes denoted as 0 are treated as missing. Also, genotype denoted as any number between the maximum number of genotypes and 9 are treated as missing (eg. for 2 alleles and 3 genotypes, the following are treated as missing: 0, 4, 5, 6, 7, 8, 9). In addition, a period . is recognised as a missing genotype. The maximum number accepted for a genotype number is 9 (but ask BK if you need to change this, for >3 alleles). Where a full set of genotype probabilities is provided and they do not sum to 1 (strictly $\text{abs}(\text{sum}-1.) > 0.000001$) then that locus is treated as missing for that individual.

Where an incomplete set of genotype probabilities is provided (all but the first) and the first probability is deduced to be negative (strictly < -0.00001) then an error message will appear on the interface or console.

An individual with missing genotypes or genotype probabilities is allocated the raw mean of genotype probabilities for current candidates (ie. excluding those who are only listed as candidates in CommittedMatings.txt) that have legal genotypes or genotype probabilities.

Handling sex-linked markers

This can be accommodated by using an extra allele to denote the missing allele on the Y chromosome in males – or on the W chromosome in female birds. It is good to not list the missing allele as the first allele – as this would be reported in the top histogram (frequency of first allele) for that marker, and would contain the certain information of a frequency of 0.5 in progeny, being the expected proportion of the heterogametic sex.

With reference to the table above, and for two “real” alleles segregating, you can take this approach:

Use field name, for example, g_3_6_MS1-_1_2_0 . Here I name the locus MS1 and I take the two alleles to be named 1 and 2, and the missing allele is denoted 0. The genotypes are laid out as follows:

3_6	11	12	10
		22	20

So that the genotypes in the datafile will be as follows:

3_6	1	2	3
		4	5
			6

Genotype 6 (00) cannot exist. Genotypes 3 and 5 (10 and 20) are for the heterogametic sex, and 1, 2 and 4 (11, 12, and 22) are for the homogametic sex.

As this is set up as a 3-allele scenario, you will see 10 histograms on expected progeny outcomes for this marker – one for the frequency of allele 1, and 3x3 for the 9 ordered genotypes.

Handling multiple lethal recessives

Moves are underway to discover hopefully many low-frequency recessive lethals by looking for loci with ‘missing homozygotes’ in the population(s) of interest.

You can enter these loci as markers, following the section above, using the genotype or the genotype probability format. Of course the latter is useful where not all candidates are genotyped, and eg. Geneprob is used to extend the information on hand.

However, with possibly hundreds of these markers in your main data file, we need a special way to manipulate them. This is done as follows:

Create two extra Trait fields. These are basically LethalA (for Lethal Alleles) and LethalG (for Lethal Genotypes). However we hitch-hike on these field names to also point to which markers are to be included in this way. Thus

LethalA1 and LethalG100 means that marker numbers 1 to 100 are to be included.

LethalA23 and LethalG49 means that marker numbers 23 to 49 are to be included.

The data in these columns is of no consequence (so maybe set them all to zero). The program will populate them on the fly according to the selections and mate allocations of the current solution, and you will see histograms for these “traits” that show:

LethalA: Use this for longer-term objectives, by selecting against recessive lethal alleles in the population. LethalA for each progeny is the probability of mortality in their offspring (i.e. in grand-progeny of the current candidates), given that the progeny will be randomly mated to the population at large, with recessive allele frequencies equal to those observed (or implied from genotype probabilities) among the current candidates.

LethalG: Use this for short-term objectives by selecting against the incidence of lethally affected progeny from the current matings. LethalG for each progeny is the probability of mortality due to lethal homozygosity at one or more of the loci concerned. Given that the parents have zero probability of being homozygous lethal at any locus, the minimum predicted probability of *survival* is $(1 - 0.5 * 0.5) = 0.75$ for each locus, where both parents are heterozygotes. Across n loci, the minimum probability of survival is thus 0.75^n , giving a maximum value for LethalG of $1 - 0.75^n$. So LethalG ranges 0 to $1 - 0.75^n$ for n loci considered.

You can manipulate LethalA and LethalG using their histograms under Trait Management. As frequencies are generally very low, the scales shown will be well below the maximum possible. Rescaling will occur as you find success in reducing allele and genotypes frequencies to very low levels. But look out for the impact on other issues!!

After the run (or after clicking “Print Matings”) you can view response for these ‘traits’ in OutSummary.txt, as for all other traits.

Some rules:

You must include either both these fields, or neither of them.

Column order is free except for the first 3 ID columns.

Whatever your column order, these field names must be LethalA<from> and LethalG<to>

These two column names are *case sensitive*.

You can hide histograms for the full block of these lethal loci, in order to increase operation speed, especially in the Web version of Matesel.. See [Hiding histograms](#) for details.

Matesel.ini – initial parameters and conditions

This file contains default parameters and conditions that are read by MateSel in order to populate its various textboxes and other controls. You can edit it to avoid having to make many edits each time you come back to a job. There is an instance of this file in the same folder as MateSel.exe, simply to start it with controls filled. However, when you read in your main data file, Matesel looks in that same folder for your possibly customised version of Matesel.ini.

For the Windows version, if Matesel.ini does not exist, MateSel creates it for you when you make a run. For other versions you must include Matesel.ini in your set of files.

After editing the MateSel interface, you can save the current parameters and conditions into your MateSel.ini by clicking the disk icon in MateSel’s [toolbar](#).

Parameters to manipulate Inbreeding, Coancestries, Traits and Markers

This latter part of MateSel.ini is optional. Your last line can be this line, or the line before it:

```
For a new job leave the rest of this file blank. Run then click Save in Windows MateSel. Then you can edit defaults here if required.
```

This latter part of MateSel.ini contains switches and weightings for management of the distributions illustrated by histograms.

If the number of traits and markers in the datafile do not match the numbers in MateSel.ini, Matesel will inform you that it will ignore what you have supplied and use default settings and zero weightings for the Traits and Markers it has detected in your main datafile.

However, you will not have this problem if you take the suggested route: Leave the latter part of the file out, make a run, possibly including some trait and marker manipulations, and then save MateSel.ini by clicking on the save icon (Windows interface). You do not have to wait for the run to finish, but make your desired settings before saving the file.

NOTES: Matesel.ini is read when you open a datafile, not when you hit “Run” (else edits you make on the interface would be lost!). So, if you edit MateSel.ini and want to use those edits, be sure to open your datafile again. (Or simply close then restart your MateSel session).

It is recommended for a new job that you let Matesel create the Histogram control parameter part of Matesel.ini. However, if you need to do this directly by yourself, note that the descriptions for each Item (eg. “ProgInb”, “Trait1”, etc) MUST NOT exceed 10 characters.

Automatic upgrading from a previous format

Simply open the main datafile in your old project and click Run. Matesel will read your old Matesel.ini and save your parameters **except for the histogram control parameters described above – be sure to first keep a copy of any settings you need.** When you click the Save icon after the run has started, MateSel.ini will be saved in the new format.

Example of MateSel.ini

```
Integer parameters:
3           , Balance Strategy
0           , Grouping (1, else 0)
166        , Matings to be made if not grouping
10         , MinNewGens (No improvement for this number of generations helps to stop)
5000      , AbsMinNewGens (No fewer than this total number of generations are ...
100000    , MaxNewGens (Stops on first improvement or nsee after this)
1000      , n_see (report status every n_see generations at least)
2000     , Front0Gens (Generations to find 0 degree point on Frontier)
2000     , Front90Gens (Generations to find 90 degree point)
1000     , FrontMidGens (Generations to find each intermediate point)
12        , Threshold Management (see manual) (was nGensBackMany)
1         , UseColleau for (usually) faster x'Ax calculation (NRM, not GRM) (1,else 0)
1         , SeedStart (change to change random number sequence)
2         , Verbose (Reporting level on console)
0         , ReadGRM 0:NRM. 1:GRMcoan. 2:GRMprogF. 3:GRMcoan and GRMprogF.
0         , Switch to control reproductive boosting: MOETing = 1, IVF = 0
0         , Switch to control Multiple EndUses (1, else just nucleus EndUse only)
0         , Switches to control backup matings
0         , Switches to control avoidance of duplicate, reciprocal and self matings
0         , Switch to control cost of assembling mating groups
0         , Switch to control ranking of full sibs
1         , Switch to control Committed Matings
Real number parameters:
25        , TargetDegrees (Used under BS3, BS4 and BS5 only)
0.075    , xAx2Limit Coancestry limit (Used under BS1 and BS2 only)
0        , WeightOnxAx2 (Use histogram control of progeny coancestry instead.)
-1       , WeightOnF (a negative Weighting to reduce Progeny Inbreeding)
0        , Reserved for later use (was F_Threshold)
0        , Reserved for later use (was WeightOnF_T)
99.9     , ConvergePercentTarget (Helps to stop run. See manual)
15       , NoChangePercent (No improvement over ~this percent of generations)
```

```

10          , FrontDegreeGap (For Frontier drawing)
0          , xNRMLimit (Values below this are treated as zero in NRM, for speed.)
0          , BiSexCands (Set to 1 where a candidate can be male AND female, else 0)
-999999    , Missing (This value is dataset is treated as missing for traits, etc.)
0          , Reserved for future use
0          , Weighting for avoiding duplicate, reciprocal and/or self matings
0          , Weighting for soft implementation of Location issue
0          , Reserved for future use
0          , Reserved for future use
For a new job leave the rest of this file blank. Run then click Save in Windows MateSel.
6          , Number of traits
4          , Number of marker loci
Parameters to manipulate Inbreeding, Coancestry, Trait and Marker histograms ...
Item      Invoked      ControlType      Weighting      Target1      Target2      Target3
ProgInb   0                8                1              0.258062     0.258062     50
SireCoan  0                1                1              0.04740112   0.04740112   50
DamCoan   0                1                1              0.04668888   0.04668888   50
ProgCoan  0                1                1              0.04704498   0.04704498   50
B.Wt      0                1                1              1.01309      1.01309      50
Y.Wt      0                1                1              102.2804     102.2804     50
C.Wt      0                1                1              32.96264     32.96264     50
BackFat   0                1                1              298.2869     298.2869     50
IMFat     0                1                1              1.165046     1.165046     50
EMArea    0                1                1              7.067358     7.067358     50
M1-1      0                1                1              0.3537192    0.3537192    50
M1-11     0                1                1              0.1248447    0.1248447    50
M1-12     0                1                1              0.2123645    0.2123645    50
M1-21     0                1                1              0.2453844    0.2453844    50
M1-22     0                1                1              0.4174062    0.4174062    50
M2-1      0                1                1              0.5827712    0.5827712    50
M2-11     0                1                1              0.3395171    0.3395171    50
M2-12     0                1                1              0.2535062    0.2535062    50
M2-21     0                1                1              0.2330019    0.2330019    50
M2-22     0                1                1              0.1739748    0.1739748    50
M3-1      0                1                1              0.7034851    0.7034851    50
M3-11     0                1                1              0.4948652    0.4948652    50
M3-12     0                1                1              0.2137259    0.2137259    50
M3-21     0                1                1              0.2035141    0.2035141    50
M3-22     0                1                1              0.0878951    0.0878951    50
M4-1      0                1                1              0.6235959    0.6235959    50
M4-11     0                1                1              0.3888144    0.3888144    50
M4-12     0                1                1              0.2271994    0.2271994    50
M4-21     0                1                1              0.2423635    0.2423635    50
M4-22     0                1                1              0.1416224    0.1416224    50

```

Target1, Target2 and Target3 in the above list have these generic names because their meaning depends on what [ControlType](#) is set for the item concerned. They are initially populated by candidate means for the item concerned (Target 1 and Target2) and 50 (% in the upper tail) for Target 3, which is only used for Bimodality. Of course, you can edit this whole section so that MateSel adopts these controls across items for your run. When using Batch MateSel in simulations, you can generate this whole MateSel.ini file in your simulation software, such that, for example, the mean value for Trait1 matches your desires set for each year of breeding.

The description for most of the content of Matesel.ini is covered under [The MateSel user interface](#) and other relevant sections, but two orphaned components are covered here:

UseColleau

Set this to 1 to use the fast method of Jacques Colleau (<https://doi.org/10.1186/1297-9686-34-4-409>) to calculate mean parental coancestry, rather than using simple matrix manipulations on the non-zero elements of the relationship matrix. This is considerably faster, except perhaps for few candidates and a deep pedigree. This option is overwritten to 0 when using a [GRM](#).

NRM limit

[Use only if needed, else leave at zero.] For large problems you can speed up operation by setting to zero any numerator relationship value (between two individuals) that is below this value. Elements with a value of zero are not addressed in calculating $x'Ax$, and so setting low values to zero speeds up operation.

Grouping files

InpOneGroup.txt - the Not-Grouping file

If you are not grouping (ie. Grouping is set to 0 in Matesel.ini), then you have just one group of males and one group of females with no grouping constraints. You need to include the file InpOneGroup.txt. Here is an example (Note that the descriptive lines of text are all single lines – they are wrapped around here because of space limitation only):

```
Max number of backup sires reported per mating
3
Male preselection. 1 for no preselection, <1 for proportion
preselected, >1 for number preselected ...
1
Female preselection. ... Any preselection is on the main (first)
Index only.
1
Male ABS-Minuse numbers
0
Male Minuse numbers
0
Male Maxuse numbers
50
Female ABS-Minuse numbers
0
Female Minuse numbers
0
Female Maxuse numbers
1
```

Links: [Maxuse](#), [Minuse](#), [AbsMinuse](#)

InpGroups.txt - the Grouping file

If you are grouping (ie. Grouping is set to True under the grouping tab at the time you hit Run), then you have at least two groups of males or at least two groups of females with grouping constraints (such as don't mate a bull from farm A with a cow from farm B). Here is an example (Note that the black descriptive lines of text are all single lines – they are wrapped around here because of space limitation only. Also, there are no blank lines in the file itself – splitting here is only carried out to insert helpful comments, in blue font. In general, 1 means yes and 0 means no).

Numbers in a row can be delimited by commas (only one at a time!) or by any number of spaces. You should see an information message if you have made a mistake in constructing this file.

Grouping example

		Female group		
		Farm 1	Farm 2	Farm 3
Male group	Farm 1	120	0	0
	Farm 2	0	110	0
	Farm 3	0	0	75
	A.I.	210	180	125

Kingham, B.P. 2011. An algorithm for efficient constrained mate selection. *Genetics Selection Evolution* 43:4

53

PLEASE NOTE THAT THE GROUP NUMBERS YOU ENTER IN YOUR DATAFILE ARE ACTUALLY SUB-GROUPS, AS SHOWN HERE ...

```

2           (# of Male Groups)
3 100 101 103 (Number of sub-groups followed by sub-groups)
2 120 122     (These are for the second male group)
3           (# of Female Groups)
2      100 102
4      101 120 121 122
1      150
  
```

This format involves subgroups, allowing you to amalgamate group numbers for the purposes of the analysis. Here, the first of two male groups is made up of 3 (“3” in first column) subgroups (100, 101 and 103). This means that groups 100, 101 and 103 for male candidates in the input file will be treated as belonging to the same group for the purposes of the current mate selection run.

Why do this sub-grouping? Let’s say you have Natural Mating bulls (Group 1 in the data file) and AI bulls. However, you have identified high calving ease AI bulls for possibly use on heifers, and these are identifies as Group 3, with normal AI bulls being group 2 in the data file. So, if you do not want to use the calving ease information you would use this ...

```

2 (# of Male Groups)
1  1
2  2 3
  
```

... but if you want to use the calving ease information, (eg so heifers an only be mated by group 3 bulls) you would use this ...

```

3 (# of Male Groups)
1      1
1      2
1      3

```

Use number of matings (1) or selection proportions (0) to describe matings per female group
1

In this case, the numbers (30, 40, 50) in the following line are the numbers of female matings to make in each female group.

Number of matings per female group, or proportion selected out of preselected females. Numbers can be > #candidates, or proportions>1, to drive MOET and IVF/JIVET
30 40 50

This lets you define the amount of selection pressure for each female group, but ...

Invoke Female mating numbers or proportions (Enter 1, else enter 0 to use above line to calculate total #matings only)
1

Entering zero means that female groups compete with each other for selection. Notice that under grouping, the number of matings to be made is calculated based on the female mating numbers or proportions.

Male group could be backup sires
1 0
Max number of backup sires reported per mating for each female group
5 5 0

Up to 5, 5 and 0 backup males (in addition to the recommended male mate) ranked in order of appropriateness will be nominated for each female in the three groups.

Male preselection. 1 for no preselection, <1 for proportion preselected, >1 for number preselected ...
40 40

In this case the best 40 males in each group will be preselected, on the basis of their index value. The only reason to do this is to achieve faster analysis and/or better convergence, and is only warranted for large jobs. (Preselection down to 40 males in each of two groups would not be required!)

Female preselection. ... Any preselection is on the main (first) Index only, Ignores maxuse. Set these values to 1 in a 'pre-run' to see candidates per group.
100,120,150

Male ABS-Minuse numbers (Enter -1 to read from the data file)
 0 0
 Male Minuse numbers (Enter -1 to read from the data file)
 0 0
 Male Maxuse numbers (Enter -1 to read from the data file)
 -1 30

Links: [Maxuse](#), [Minuse](#), [AbsMinuse](#)

Female ABS-Minuse numbers (Enter -1 to read from the data file)
 0 0 0
 Female Minuse numbers (Enter -1 to read from the data file)
 0 0 0
 Female Maxuse numbers (Enter -1 to read from the data file)
 1 1 1

Links: [Maxuse](#), [Minuse](#), [AbsMinuse](#)

Male natural mating vector (1 = constrain to one female group)
 0 0

Set to 1 if each individual male in the group can only be used in one female group.

As examples: If male group 2 consists of bulls from which semen can be collected, then we set the above to 0 for male group 2, and each individual bull in that group can then be used in both female groups 2 and 3 (see permission matrix below). However if male group 2 consists of bulls to be used for natural mating only, then we set the above to 1 for male group 2, and then each individual bull in group 2 can be used either in female group 2 or 3 **but not both**, because he must stay in the group (typically farm or paddock) where he is being used.

Mating permission matrix: Male x Female ...
 1 1 1
 0 1 1

0 means no matings permitted between members of the groups concerned. 1 means permission granted.

Optional Extras:

Targeting specific numbers of matings per group combination with a weighting

[May the Force be with you: Don't miss [Forcing specific numbers of matings per group combination](#) - this could be your preferred option.]

This is simply triggered by having a text description line (Such as "Overall weighting for GrpGrp #matings", below) immediately after the last line of the permissions matrix.

Note: the old method of changing any of the "1" entries in the mating permission matrix "-1" is still supported, but not required.

Given this signal, you must append further lines as in this example:

```
Mating permission matrix: Male x Female ...
1 1 1
0 1 1
Overall weighting for GrpGrp #matings
1
Targets for GrpGrp #matings
30 20 10
  0 20 40
```

In this case a single weighting of 1 unit is applied *towards* achieving the desired outcome. If you are not happy with the result you need to edit this weighting (a real number) and try a new run. If you find you need a weighting much above 1 to get the sort of result you want, then you are most probably better to use [Forcing specific numbers of matings per group combination](#) below.

The last two rows of this example show the targets. We are looking for a big representation of male group 2 x female group 3 matings. These values are read as real numbers, and do not have to sum to the number of matings required (although this is sensible practice). Values are over-ridden to zero where the corresponding Mating Permission = 0, and the remaining values are used to set target numbers of weightings that sum to total number of matings.

These target figures relate to “new” matings, in addition to any Committed Matings already made or otherwise committed to.

In this example, the number of matings per female group (30, 40, 50), set earlier in the file, will be fully adhered to as `Invoke Female mating numbers = 1` and the grouping algorithm is “hard”. But the current section merely pushes the result towards a particular outcome, and this part is “soft”.

NOTE: Using this approach, the [Frontier](#) is built *without* any emphasis placed on GrpGrp targets.

WARNING: If you have some weighting applied, and the GrpGrp targets are all set to zero (or possibly some other unreachable state), you might find that the optimal outcome is very poor on the frontier. This was found to be the case for a non-evolving bad result – a weighting applied and GrpGrp targets all set to zero.

Forcing specific numbers of matings per group combination

You might use this feature if “Targeting specific numbers of matings per group combination with a weighting” results in slow performance, and/or you see messages such as “Instances of: Target not achievable due to Minuse constraints”.

To invoke this Forcing feature *instead* of the Targeting feature, simply start the header for the Targets with the word “Force”, such as:

```
Force Targets for GrpGrp #matings
```

Or

Force GrpGrp Targets

When this feature is invoked, you will see the following on the Console:

```
Forcing of Group x Group mating numbers invoked.
```

If using this feature, the Overall weighting for GrpGrp #matings is not used, but the two lines involved must be populated. A weighting of 0 does not switch of this Forcing feature – remove the word “Force” to do that.

NOTE: Using this approach, the [Frontier](#) is built *with* GrpGrp targets forced.

WARNING: If you specify numbers of matings per group combination that cannot be attained for the dataset you are using, some sort of error will occur, such as “k/=MatingsNew Male”. This is not the case when “Targeting specific numbers of matings per group combination”. ALSO, if you set `Invoke Female mating numbers = 1` and these numbers do not match the column totals in Force GrpGrp Targets, then you will get a warning message on the console and you will need to abort the run to fix this inconsistency.

Should I Target or Force specific numbers of matings per group combination?

If you are dogmatic about the numbers that you want, use the Force option for its much more fast and robust convergence. The possible downside is that by deviating a bit from these desires, you might achieve useful extra performance for other issues.

What you can do it try a run each way, and [import](#) each solution into the opposite run. This is probably better than just inspecting the graphical results for the two runs, as the Frontiers differ between the two runs: the Forcing of your desires is reflected on the Frontier, but the Targeting is not (grouping numbers run free, unweighted, during the building of the Frontier).

If the weighting approach is not much better for other issues, then probably use the Forcing approach.

Targeting minimum use of individual males within each female group

This is simply triggered by having a text description line (Such as “Filter and weighting for Male Minuse per group”, below) immediately after the last line of the GrpGrp Target matrix.

Note: this GrpGrp Target matrix must exist in the file for Targetting Minuse to be invoked. However, you can, if you wish, give the GrpGrp Overall weighting a value of zero, as in the example below.

Given this signal, you must append further lines as in this example:

```
Mating permission matrix: Male x Female ...  
1 1 1
```

```

0 1 1
Overall weighting for GrpGrp #matings
0
Targets for GrpGrp #matings
30 20 10
  0 20 40
Filter and weighting for Male Minuse per group
1 1.5
Targets for Male Minuse per group
  5  5 10
  0 10  8

```

There are two mechanisms for targeting these Group Minuse figures shown in the last array:

A translation filter: Enter 0 (off) or 1 (on). In the example above, the filter is on. This filter translates the Mate selection set to one that conforms (as closely as the algorithm can achieve) to the Minuse per group requirements. This is done within the evolutionary algorithm – it is not merely a “fix-up” of the final result – and so evolution proceeds for other issues affected by mate allocations. Activating this filter causes an increase in the computing time per generation of the evolutionary algorithm, but this is very likely to be very well compensated for in terms of fewer generations required for convergence.

An overall weighting: This is set to 1.5 in the example above. [We suggest you start with a weighting of 1 – we use 1.5 for identification here]. This weighting is used to penalise solutions for which the filter cannot give full conformation to the target Minuse array, which is more liable to happen eg. at higher [Target Degrees](#) where more sires are used, each more thinly spread across female groups. As with other weightings, this one is applied *towards* achieving the desired outcome. If you are not happy with the result you need to edit this weighting (a real number) and try a new run. However, be patient and let the job converge well before increasing weights – high weights can reduce rate of convergence for other issues considerably, even though you may satisfy the current issue quickly.

It is recommended that you start by using a value of 1 for both the filter and the weighting. Be sure to set both values to zero if you do not want to invoke any Group Minuse control!!

The Group Minuse figures in the last array relate to “New” matings, in addition to any Committed Matings already made or otherwise committed to.

Let us say that we have a Group 1 male with a Minuse of 15. Then, given the matrix above (5 5 10 for Group 1 males), and that it is used, then the following example patterns of use of this male over female groups will attract penalty as follows:

No penalty applied			Penalty applied		
Female Grp 1	Female Grp 2	Female Grp3	Female Grp 1	Female Grp 2	Female Grp3

>=5	>=5	>=10	4	5	10
0	>=5	>=10	6	0	9
0	0	>=15	7	3	12

The overall value of the penalty is: the overall weighting times a linear scaling factor times the average deficit in number of matings below the Group Minuse set for each group, with the exception that there is no penalty for zero matings made in a female group.

If a male group is only permitted to mate with one female group, then that Group x Group combination will still be considered when setting penalties, even though normal Minuse values would handle that situation. This is done to give you more flexibility when using this feature.

You can monitor sire use across groups by keeping an eye on the “Show sire use” part of the “Mate selections” tab. Your final solution may not fully conform to the target minuse values you set.

Please Note: Be careful how you set Minuse figures. Of course, Individual Minuse figures must equal or exceed Group Minuse figures as logic dictates. If Individual Minuse for males is set close to (some of) the Group Minuse figures, then there is little room for the filter and weighting methods to move – and indeed there may not be fully conforming solutions possible. If your results do not look satisfactory, try reducing Individual Minuse values for males, and/or the Group Minuse figures. Maybe try a test run with Individual Minuse values set to zero. **When using the filter, under “tight” circumstances, Individual Minuse figures may not be met for some individuals.**

Setting up the GRM

A Genomic Relationship Matrix (GRM) among your candidates can be used to help manage coancestry and progeny inbreeding. As you supply the GRM, as described below, you can introduce any type of relationship matrix to Matesel by this method, whether or not it involves genomic information. See Meuwissen et al 2020 “[Management of Genetic Diversity in the Era of Genomics](#)” for a discussion on different types of GRM.

In addition, your GRM might include blends between pedigree and genomic relationship matrices, genomic relationship matrices pertaining to specific genomic region(s) (see [Target specific genomic region\(s\) for management of diversity and inbreeding](#)), and even matrices that include dummy individuals that represent groups of real individuals, and their predicted average relationships with all other candidates (see [Handling large groups of individuals as single “dummy individuals”](#)).

[Note: As of Fortran DLL version 8.2 (see [Versions of Matesel](#)), a much easier way of setting up the GRM has been invoked, as described immediately below. However this is backward compatible with the previous method, which you can still use after a small alteration to Matesel.ini. This old method is described at the end of this section.]

Make a file called CandGRM.txt and fill it with GRM elements in one of three ways, as in the three following examples:

Full	Bert	George	Maisey	Daisey	Fred	Roger
Bert	1.043	0.053	0.070	0.070	0.059	0.068
George	0.053	1.017	0.041	0.041	0.327	0.033
Maisey	0.070	0.041	1.015	0.546	0.042	0.064
Daisey	0.070	0.041	0.546	1.015	0.042	0.064
Fred	0.059	0.327	0.042	0.042	1.064	0.043
Roger	0.068	0.033	0.064	0.064	0.043	1.016

Top	Bert	George	Maisey	Daisey	Fred	Roger
Bert	1.043	0.053	0.070	0.070	0.059	0.068
George		1.017	0.041	0.041	0.327	0.033
Maisey			1.015	0.546	0.042	0.064
Daisey				1.015	0.042	0.064
Fred					1.064	0.043
Roger						1.016

Bottom	Bert	George	Maisey	Daisey	Fred	Roger
Bert	1.043					
George	0.053	1.017				
Maisey	0.070	0.041	1.015			
Daisey	0.070	0.041	0.546	1.015		
Fred	0.059	0.327	0.042	0.042	1.064	
Roger	0.068	0.033	0.064	0.064	0.043	1.016

The Top-Left string in this file must be either “Full”, “Top” or “Bottom” (case sensitive), for a full-stored or half-stored (top or bottom) GRM. These files are read free-format, and do not have to be spaced as above, which is only done to help with illustration. GRM elements can have any number of decimal places shown.

For your CandGRM.txt file:

- The individual IDs used (Bert, George, etc., in the examples above) should be exactly the same as for the corresponding individuals in the main datafile used for the run.
- The overall order of individuals is not important (George could come before Bert), but order within the file should be consistent – for example in the Full file, an individual should have the same row number and column number, giving a symmetric GRM.
- You can include individuals in the GRM that are not candidates and even not in the main data file.
- You can have individuals that are candidates missing from the GRM file (but see [How Matesel handles candidates missing in the GRM supplied.](#))

Now run Matesel with ReadGRM set as follows in Matesel.ini :

```
0    Pedigree coancestry and Pedigree F for progeny inbreeding
1    Genomic coancestry and Pedigree F for progeny inbreeding
```

- 2 Pedigree coancestry and Genomic F for progeny inbreeding
- 3 Genomic coancestry and Genomic F for progeny inbreeding

Genomic coancestries and F values are calculated from your GRM elements, just as if they were NRM elements (ie. no scaling in Matesel).

With this method, it is possible for the one GRM file to remain unchanged over some time, across several Matesel runs. Each run cherry-picks what it needs from the file, and whatever elements are missing are imputed, if possible, within Matesel. Of course, the GRM file should be updated whenever new DNA info is available, or a new blending is made. But from a practical point of view, the mating process and the genomic information can be somewhat asynchronous. Of course this can all work with the updating of GRM or equivalent for the genetic evaluation – but it can be managed separately if required for some reason.

Note: Use of the GRM for genomic EBVs can take place prior to Matesel runs, by supplying gEBVs in the “Index” field.

How Matesel imputes relationships missing in the GRM supplied

PLEASE NOTE: The following is a non-rigorous recipe for imputing values for missing GRM elements. You are welcome to not hold it in favour. In this case you can to use your own approach to impute missing elements, and build a CandGRM.txt file that includes all candidates and no missing elements. If your pedigree information is dubious, then these imputations of missing elements are affected accordingly.

Matesel in fact works with missing GRM elements, whether they are missing in the file CandGRM.txt, or they are present in the file but set to a missing value of -99999.

The supplied GRM is first augmented by appending rows and columns for the missing candidates, and these rows and columns are populated with the missing value code of -99999.

The approach is to seek the four parents of the two individuals that each missing element relates to. For individuals 1 and 2 the parents are S1, D1, S2 and D2. The missing element is taken as the mean of the following four elements S1S2, S1D2, D1S2 and D1D2, if all four of these elements are present and not missing (Note that they can become non-missing following successful imputation).

The method progresses from ancestors to descendants, “painting” the full ordered-pedigree matrix from the top-left corner towards the bottom right corner, to impute parents before progeny. After a full pass seeking all 4 parental elements for each missing element, a full pass is made where at least 3 parental elements can be found, and the mean of these 3 taken to impute the missing element, then passes at 2 and 1 parental element.

Higher stringency was sought in a test dataset by resetting the minimum requirement to 4 parental elements after each new imputation of an element, and also by iterating this whole process. However, neither of these strategies increased the number and quality of imputations made.

Missing elements that cannot be imputed by this method are allocated a value equal to the mean element plus one unit of standard deviation among elements.

Here is the sort of output you will see on the console. This is from a test dataset with relatively sparse pedigree information available – another reason to use a GRM for coancestry and inbreeding management:

```

ReadGRM is set to      3
    281 individuals in CandGRM.txt
    21 candidates not represented in CandGRM.txt.
    1575 GRM elements among candidates not represented in CandGRM.txt.
Imputing these elements from relatives:
    75 GRM elements imputed at          4 parental matches
    6 GRM elements imputed at          3 parental matches
    445 GRM elements imputed at        2 parental matches
    466 GRM elements imputed at        1 parental matches
    807 GRM elements still not represented in CandGRM.txt.
        Setting these to mean + 1*SD.
Mean of GRM elements:    0.00067
SD of GRM elements:     0.14719
Missing elements set to: 0.14786

```

Old method of setting up the GRM (still supported)

First run Matesel with ReadGRM set to -1 (minus one) in Matesel.ini.

This run will stop automatically with the message “ReadGRM=-1. See CandHalfNRM.txt for template file to help set up GRM. Stopping now.”

Observe the newly created file “CandHalfNRM.txt”, in the same folder as your Matesel data file, with fields McandID, FcandID, NRMelement. This is a top-right half NRM with the diagonal elements at the end. McandID and FcandID are the internally generated sequential IDs for the male and female candidates. [Note: this file is created on the server when using the Web version, but it cannot currently be downloaded (it will be big), so use the Batch version for this if you can.]

You need to build a similar file “CandHalfGRM.txt”, with the last field being GRMelement, as calculated separately from your genomic data. This file should be the same as its NRM counterpart with just the last field differing (ie. same order of lines).

To get from McandID and FcandID to your original identities, use the file OutSeqFile.txt that is written to the same folder as your Matesel data file. This is a 2-column text file that lists the sequential ID and the original identities in the left and right columns.

Use your own data system to build “CandHalfGRM.txt”, in the same folder as “CandHalfNRM.txt”.

Now run Matesel with ReadGRM set as follows in Matesel.ini (note these values differ from what you used to use for this old method):

```

0    Pedigree coancestry and Pedigree F for progeny inbreeding
11   Genomic coancestry and Pedigree F for progeny inbreeding

```

- 12 Pedigree coancestry and Genomic F for progeny inbreeding
- 13 Genomic coancestry and Genomic F for progeny inbreeding

Note: Be careful when using preselection of candidates in conjunction with ReadGRM under this older method. Preselection changes the numbers of candidates, and so changes the indexing of the relationship files. Use the same settings for preselection for both file preparation and running modes. It is wise to check the numbers of preselected candidates as reported on the console – they should be identical between your two runs.

Importing your own solution for plotting

Matesel can read in a solution that you construct “on the outside”. Importing a solution like this just interrupts the current running job to show results for the imported solution, then it carries on with the original run (not the imported run). Importing has no effect on the job you import these results into. It is only done to make a graphical and numerical comparison. You can import many times in a run, whenever you want to compare.

Here is a suggested way of doing this, using a MateSel-derived mating file for simple illustration:

1. Run the job with all relevant grouping and constraints in place. Guide the solution to a memorable place (eg. send it to the Frontier at 60 Target Degrees). Either press “Print Matings” or “Stop” after the main run has started.
2. Find the file OutMatings.txt and rename it to ImportSolution.txt.
3. [Alternatively create your own ImportSolution.txt from a mating list generated outside Matesel. You only need fields “Mating”, “Sire” and “Dam” – all following fields will be ignored (but see note below for Multiple EndUses). In the Mating column, enter mating number (1, 2, 3, ...), as seen in OutMatings.txt. Order is not important, but be sure to have the exact same number of matings as the job you will import this file into for comparison. This file should be delimited with spaces or commas.]
4. Run the job again, and whenever you are ready ***after the frontier has built***, click in the Import Solution button in the toolbar. You can make the click while the job is running. You can also hit Pause, then the Import button, then Resume.
5. The job will interrupt to show your solution on the Frontier, and on the Progeny Inbreeding, Individual Coancestry, Trait Management, Marker Management and Mate Selections tabs, as well as numerically on the console.
6. You will see a message if your solution breaks constraints that are tested for, and a report will go to OutSolutionError.txt
7. To resume the main job, follow the instructions on-screen: “Click Pause then Resume to continue your run” – but be sure to dismiss the message

Notes:

- If you are using Multiple EndUses, then you will need extra columns in ImportSolution.txt. You need the same six fields as the first six fields in OutMatings.txt: (Mating Sire Dam GrpS GrpD EndUse). This is done so that you can still copy OutMatings.txt to ImportSolution.txt and have a valid import file to use. The fields GrpS and GrpD (Group numbers for Sire and Dam) are not in fact used, but they are included to maintain ability to copy from OutMatings.txt. If you are

constructing your own ImportSolution.txt, you can enter any values in these two fields (eg. make all entries zero). But of course, you must be sure that each mating has an EndUse nominated by you. The number of matings for each EndUse must match the numbers reported (“Matings by EndUse:”) near the top of the console.

- For projects involving [Committed Matings](#), ImportSolution.txt should contain all committed plus new matings.

The MateSel user interface

Run MateSel.exe then open your data file. Now you can edit the user interface before clicking “Run”. Once your job is running, you can do many things to alter the course it takes. Note that for some settings, MateSel shows values related to your current mouse position, and you can Point-and-Click to set some parameters.

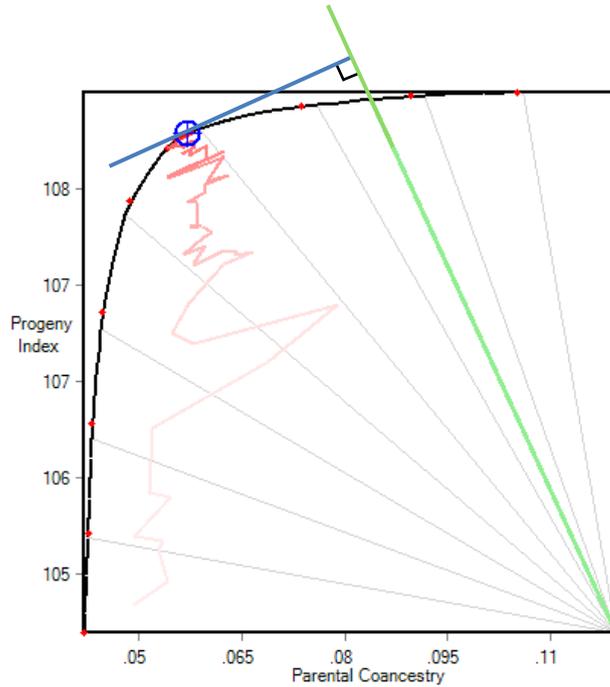
Parameter and settings textboxes

Balance Strategy

This relates to management of the balance between genetic gain (predicted index value in progeny) and parental coancestry, whose outcome is illustrated in the graph to the lower left of the MateSel window.

0	Weighting λ on coancestry: $xG + \lambda * xAx/2$	This is the simplest approach, but it offers little control – adding other issues can have a big impact on the balance between gain and coancestry. Set λ in MateSel.ini.
1	Hard constraint on coancestry WARNING: If your constraint cannot be met, all other issues (Progeny Index etc.) will suffer dramatically while a legal solution is sought.	This maintains the solution at or below a given value of parental coancestry. Set the limit at textbox “Max permissible $xAx/2$ ”. You should specify a value that is greater than the minimum value – this is the minimum value on the horizontal axis of the Frontier Graph, that only appears after “Run” has been clicked.
2	Soft constraint on coancestry WARNING: As for 1 above.	As above, except that exceeding the limit by a small amount is permitted, but heavily penalised. This can help with speed of convergence.
3	Hard constraint on Target Degrees	This maintains the solution at or “above”* a given value of Target Degrees (TD). In fact, this component simply aims to “maximise gain”*, but also severely penalises solutions that break the TD constraint. * For TD > 45 degrees, substitute “below” and “minimise coancestry” respectively.
4	Soft constraint on Target Degrees	As above, except that undercutting the limit is permitted, but heavily penalised beyond a 2 degree deviation. This can help with speed of convergence.
5	Project to Target Degrees line	This aims for a solution that is most deviated from the origin when projected to the TD line (see illustration below). [Hold a ruler perpendicular to the TD line and move it out to where it last coincides with the frontier – that is where a solution should go with no other issues

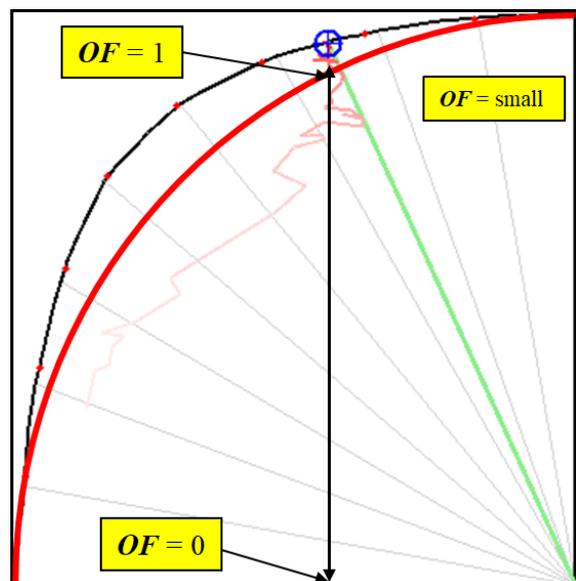
included.] For non-circular frontier shapes, this projected point can be outside the frontier, with the solution exploiting the prevailing shape of the frontier to maximise the index implied by the TD line.



- 6 Projection plus deviation penalty As above, except that a weighting is applied against deviations of more than 5 degrees from the TD line. [Note: Some danger of settling on a local optimum. You can check this by moving to strategy 5, waiting for good convergence, then change back to strategy 6]

For method information only:

For all Balance Strategies other than strategy 0, progeny index (xG) is scaled on a range of OF=0 (OF = contribution to the Objective Function value) at 90 degrees, to OF=1, at the point on the frontier where the TD line or the constrained coancestry line intersects with the red circular arc, which is a proxy for the Frontier (see diagram). This renders xG independent from the trait(s) used, such that the weightings used for other issues (such as progeny inbreeding) have a more consisted impact across different datasets. Note that the OF value is set VERY small for illegal solutions to the right of the Target degree line.



The Direction Tab in the Matesel Input Groupbox (near to the top left of window) contains textboxes for you to edit the Maximum permissible $x_{Ax/2}$ and [Target Degrees](#), as described below. Balance strategy 0 seems to be not used these days, so its Weight on Coancestry must be edited in Matesel.ini before you make a run.

The following lists the edits you can make in the Matesel input section:

Max permissible coancestry, $x_{Ax/2}$

This is the limit on mean parental coancestry, which is only used in Balance Strategies 1 and 2.

Target Degrees (TD)

[See [Frontier Graph](#)] TD is used to target a relationship between Genetic Gain (Progeny Index, x_G) and Parental coancestry ($x_{Ax/2}$). It is only used for Balance Strategies 3 to 6.

Weight on progeny F

This the weighting value used to penalise inbreeding (F) in the progeny that derive from the matings to be made. Use a negative value to put downward pressure on short-term inbreeding. Suggestion: start with a value of -1. [Note that this has little impact on long-term inbreeding. Further discussion [here](#).]

Even a tiny weighting can be effective: Note that both Progeny Index and Mean Parental Coancestry are unaffected by the pattern of mate allocation – they only depend on which candidates are used as parents and their genetic contribution (~number of matings). This means that you can put a very low weighting on Progeny F and yet get a good impact on Progeny F, if there are no other issues affected by mate allocation ... Set weight on Progeny F to zero then wait till the run is very well converged. Then apply a tiny weighting (perhaps 0.001) and get good impact on Progeny F. This results because, *once optimal contributions have been arrived at*, the only way for Matesel to improve the solution is to reduce Progeny F, and even the tiniest of weighting values will drive the solution in that direction. Essentially this means that Matesel keeps the same selections (because the weight on Progeny F is sufficiently low to not change them), but manipulates the pattern of mate allocation among those selected individuals to lower Progeny F.

See also [Interaction with Weight on Progeny F](#).

Matings to be made

This is the number of matings to be made. It is typically the number of breeding females, but for IVF matings it is the number of oocytes fertilized. [Under Grouping, the numbers of matings is calculated from grouping information provided, overriding what you enter here.]

Grouping

This feature gives ability to define group membership for each candidate, set mating numbers for each group, set mating permissions on a male group by female group basis,

and other grouping functions. See section [InpGroups.txt - the Grouping file](#) for more details.

Frontier degree gap

This is the gap between points generated to draw the frontier of possible responses in Index and Coancestry.

Frontier 0 degree generations

The number of generations of the Differential Evolution algorithm to be used to generate the 0 degree point, which reflects full emphasis on high Progeny Index. [This is only used when Grouping = True, else a truncation selection approach is used.]

Frontier 90 degree generations

The number of generations of the Differential Evolution algorithm to be used to generate the 90 degree point, which reflects full emphasis on low Coancestry.

Frontier mid degree generations

The number of generations of the Differential Evolution algorithm to be used to generate each intermediate degree point.

NOTE: The number of generations for 0 and 90 degrees should be kept reasonably high (maybe about as high as is required to get some reasonable convergence during the main run). This is because these locations are used to “anchor” the frontier, and to help define what directions are taken for policies that are specified in degrees. However, if these anchor points are overshoot during a run, the frontier is recalculated on the fly, and a message made on the console. The number of generations for intermediate degrees is of cosmetic value (to make a nice-looking frontier), and will not affect the solution.

Target % converged

This is a component of the overall stopping criterion. The percentage of the currently predicted asymptotic maximum fitness that has been reached is calculated each time a better solution is found, and also each 1000 generations. When the value in this textbox is exceeded, then this component criterion has been met. [Note that it can latterly be no longer met if the predicted asymptote increases sufficiently.] See [Diagnosis of Convergence](#) for details.

% since last change

This component criterion is met if there has not been a better solution over the last x% generations of the last generation when there was an improvement in DE algorithm fitness.

No change generations

This component criterion is met if there has not been a better solution over the last x generations.

Minimum generations

This is the absolute minimum number of generation that has to be completed before stopping of the DE algorithm.

Maximum generations

This is the maximum number of generation that the algorithm will run for. However, this number can be overrun if “Do not stop till “Stop” clicked” is checked.

Control Buttons

- Run:** Click this button to start a run.
- Update:** Click this button to interrupt the run and inject new settings made in Balance Strategy and/or the textboxes under the Directions tab: Max permissible coancestry or [Target Degrees](#), plus Weight of progeny F.
- Pause:** Click this button to pause a run.
- Print Matings:** Click this button to print matings from the current solution. They appear under the Mate Selections tab. You can click on “Sire Use” to show the pattern of use of sires. Results are also delivered to the file OutMatings.txt.
- Stop:** Click this button to stop a run and print matings. Matings will be found both under the Mate Selections tab and in the file OutMatings.txt. Other results are printed in other files that start with “Out ...”.
- Close:** Click this button to close the MateSel window.

The Toolbar

	Open a new main data file
	Show/Hide the file reading pane (for diagnostics)
	Display the pedigree
	Save current settings to Matesel.ini
	Import a user solution from ImportSolution.txt

The MateSel console

The tab “Console for FortranDLL.dll” shows the output from the Fortran. At any stage you can click the “Copy console to clipboard” button at the bottom of the console, then paste the contents into Notepad or a word processor.

Much of the MateSel program is written in Fortran, which gives good speed. This can be seen in three parts:

1. Preparation

Reading data
 Processing candidates and groups
 Calculation of relationships

2. Frontier point calculations

The first point is at 0 degrees and the second at 90 degrees. These are required for scaling to make correct trigonometric calculations for the remaining points, which progress from 90 towards 0 degrees. If insufficient generations have been provided for convergence, the frontier can be exceeded during the main run. When this happens, the frontier is re-calculated.

3. The main run

Results for the best solution in each generation are shown in the following fields:

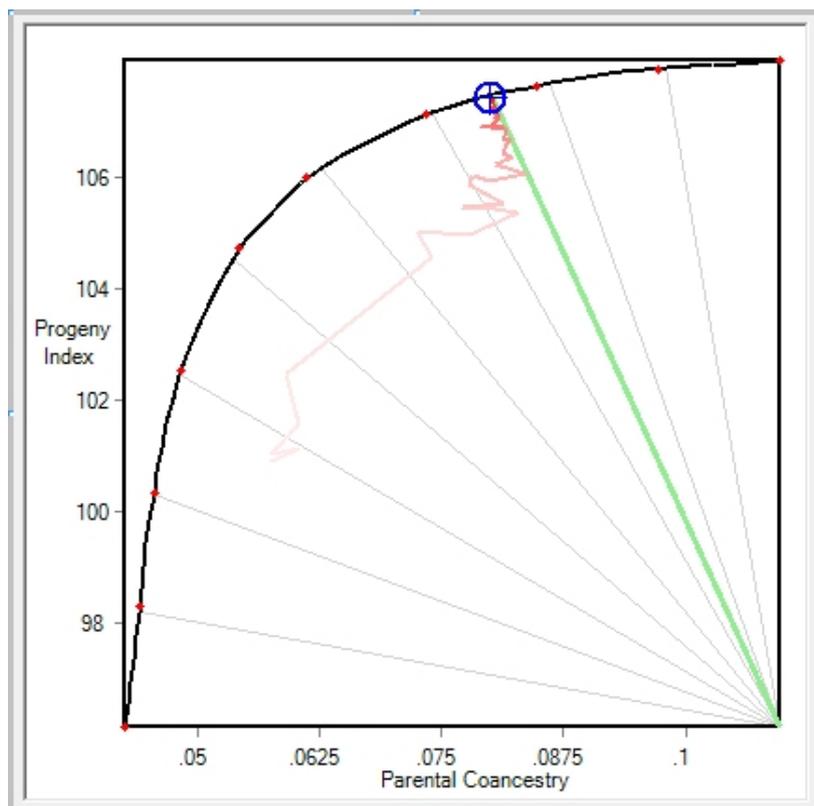
Gen	The total number of generations since the start of the main run
NewGen	The number of generations since the last update of parameters/conditions
Sires	The number of sires involved in the current solution
xG	The population average Progeny Index
xAx/2	The parental coancestry
Deg	The degrees on the frontier graph for the current solution (marked by a blue circled cross).
MeanF	The mean progeny inbreeding coefficient
MaxF	The inbreeding coefficient of the most inbred progeny
Fitness	The value of the current solution on the prevailing objective function, across issues. An asterisk (*) at the end of a fitness figure indicates that the objective function has changed (change in direction or possibly a change in Frontier coordinates at 0 or 90 degrees) and fitness cannot be compared to that of the previous generation.
Conv%	The perceive percentage converged (as described above)
Seconds	The number of seconds since the start of the main run.

The Frontier Graph

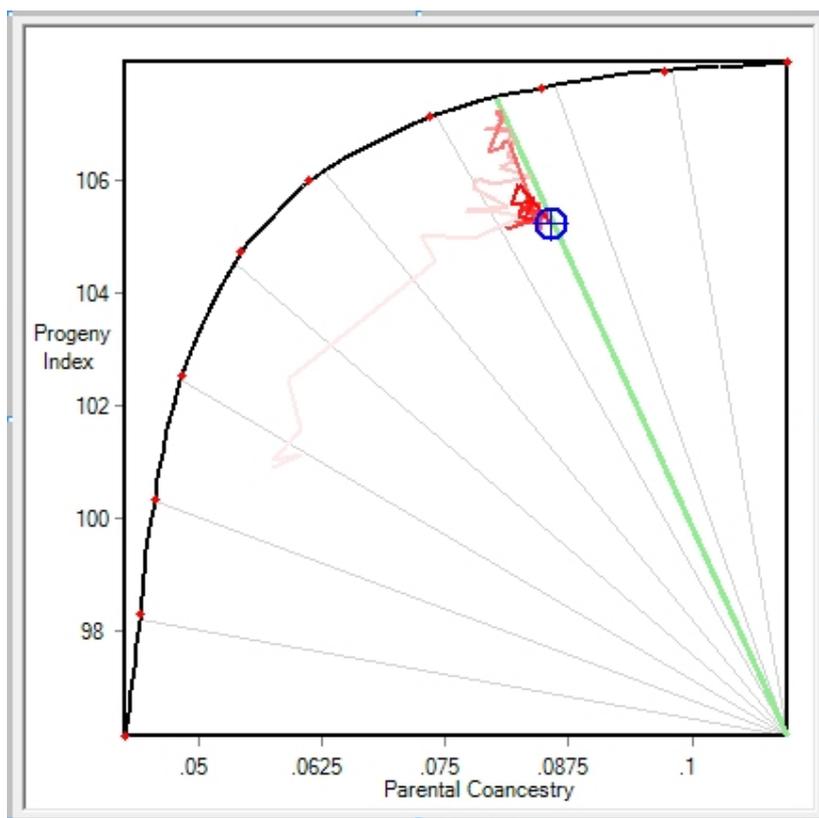
This graph shows the most extreme responses possible when all emphasis is shared by just two component objectives: Progeny Index (xG) and Parental Coancestry (xAx/2). It is the frontier of Optimal Contributions Selection (OCS).

The favourable direction is towards the top left. The frontier line is drawn between points that are found by running the DE algorithm multiple times, each with a different relative emphasis on each of these issues. [In fact it is run to target solutions that are graphically closest to calculated points on the graph.]

The “origin” is at the bottom right, and a web of 10-degree spacing is drawn for orientation. In the diagram below the current solution indicator is sitting on the frontier at 25 degrees (using Balance Strategy 3). The red trail shows the pathway that the algorithm has made during convergence.



In the next diagram, strong emphasis has been placed on progeny inbreeding, and some compromise has resulted in both lower Progeny Index and higher Coancestry:



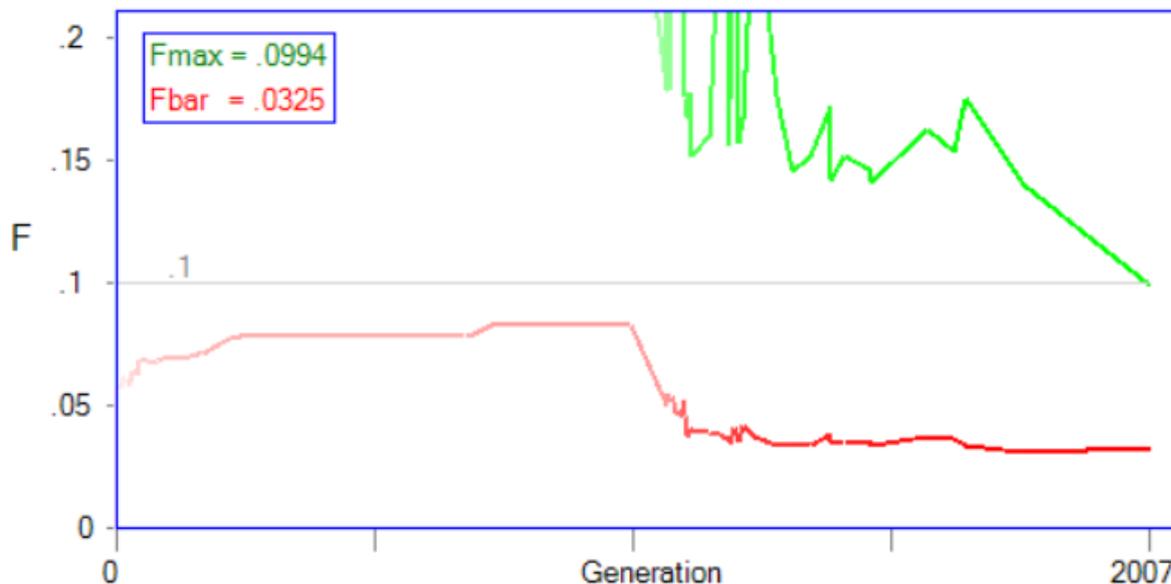
For $TD \leq 45$ degrees the TD line is green in color, as shown in these diagrams, whence Balance Strategies 3 and 4 aim to increase gain while keeping the solution above the chosen [Target Degrees](#) (to the left/below the TD line, as in the diagram above).

For $TD > 45$ degrees a magenta color is shown to illustrate that Balance Strategies 3 and 4 operate differently – here they aim to minimise coancestry while keeping the solution below the chosen Target Degrees (to the right/above the TD line).

Notes: At $TD=0$, negative degrees (to right of green line) is permitted. Likewise, at $TD=90$, >90 degrees is permitted (below the magenta line). When the solution is off the scale, the current solution indicator stays on the nearest axis and it makes an arrow pointing to where the current solution lies.

The Progeny Inbreeding Graph

This graph, shown on the Progeny Inbreeding tab, plots the value of the inbreeding coefficients of the progeny that would arise from the matings in the best solution, graphed over generations of the DE algorithm. Its value is in tracking the impact of policy changes during the current Matesel run.



The mean inbreeding coefficient (**Fbar**) is shown in **red**.

The inbreeding coefficient for the most highly inbred progeny (**Fmax**) is shown in **green**.

The current value of a threshold constraint placed on Progeny Inbreeding is shown by a horizontal grey line, unless it is above the current scale for F.

Inbreeding coefficients are pedigree-derived or GRM-derived, depending on your choice for [ReadGRM](#).

This graph shows the impact of changing [Weight on progeny F](#) and [Managing the progeny inbreeding distribution](#).

Tools for managing outcome distributions

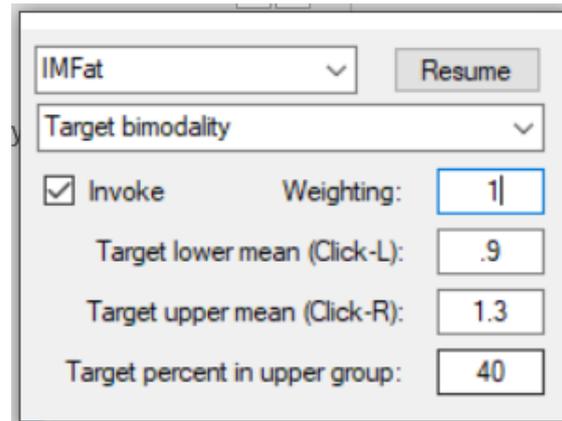
Matesel shows the impact of the current solution in histograms for many items:

- Progeny Inbreeding
- Individual Coancestries: For Sire, Dam and Progeny resulting from each mating
- Traits: Including all non-key fields, such as “Age” and “Paddock”
- Markers: Typically SNPs

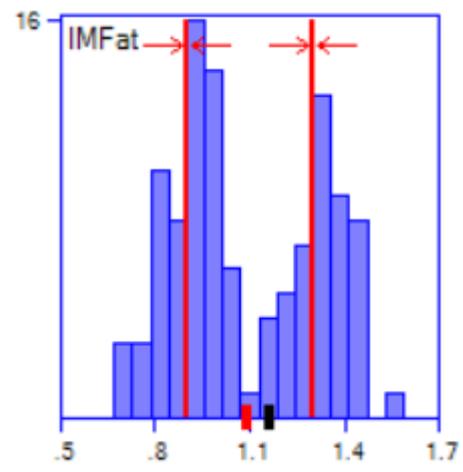
The four sets of histograms are under four tabs on the right-hand output side of the Matesel window.

Click on any one of these histograms to get an extra small *Histogram Control* window at the top-left, as here:

In this case we are heading for bimodality of IMFat with 40% of matings targeting the upper part of the distribution. This is because we have clicked on the IMFat graph under [Trait Management](#) and chosen the *Control Type* “Target Bimodality”. Be sure that “Weighting” is non zero and invoke is ticked if you want to go ahead. Then click “Resume”. You can enter the targeted means in their textboxes, or just click left and right on the histogram distribution to set these values.

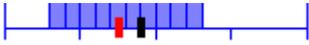


The histogram graphs themselves show threshold or target values as vertical lines, and other control types used are shown in text (“M” and “V” for Mean and Variance) to remind what you have chosen. These lines and text will be shown in red when the prevailing control type is Invoked, otherwise grey. The invoke feature is supplied so that you can keep your settings but switch them off as desired for testing.



If you click on a graph while the Histogram Control window is **not** displayed, you will open that window with the clicked item selected.

If you click on a graph while the Histogram Control window is displayed, you will change the position of any threshold values (vertical lines) on the graph. So, if you have carefully chosen these values, take care not to click on the graph and lose that information!

The horizontal line on each histogram shows two tick marks: 

Black tick: Sex-balanced mean of all candidates. This should remain fixed for the duration of your run.

Red tick: Mean of progeny for the current solution. This will change as your settings change and as the solution evolves.

For [Traits](#), [Markers](#) and [Coancestries](#), the black tick also accommodates candidates only represented in [Committed Matings](#). The [Progeny Inbreeding histogram](#) also sports a green tick, and its black tick is treated differently. You can read about that [here](#).

Control Types for manipulating histograms

The following Control Types are available for histogram:

1. Increase mean
2. Decrease mean
3. Increase variation
4. Decrease variation
5. Increase variation about optimum
6. Decrease variation about optimum
7. Set minimum value at boundary
8. Set maximum value at boundary
9. Target bimodality
10. Tactical Desired Gains upwards
11. Tactical Desired Gains downwards
12. Tactical Desired Gains stabilising

You can edit your own default Control Types, Weightings and Targets in Matesel.ini for each item (see [Matesel.ini – initial parameters and conditions](#)). The easiest way to do this is to make all the selections and weightings you want, as described above, then click the [Save icon on the Toolbar](#) to save all parameters in Matesel.ini.

You can set any such conditions for any number of traits (and/or genetic markers). You will need to play around with Weightings to get the results you want.

The Histogram Control window will show fields that need to be filled in for your chosen control type – but some of these are automatically filled in by clicking on targets or limits in the corresponding Histogram, as described above.

The extent to which your desired outcome is met depends on the Weighting that you impose for each item, in the Histogram Control window. Achieving exactly the outcome that you want for an item may cause too much compromise in other factors (traits, markers, Index gain, coancestry, progeny inbreeding etc.). So you may have to play the balancing game for a while.

More complex manipulations than those provided by these Control Types can be achieved for traits and markers by replicating fields in the main data file so that you have two or more Histograms to work with for each single trait or marker. See [here](#) for some detail on this.

Tactical desired gains

This Control Type aims to move the progeny mean up to (or down to) the specified target value for the item in question (trait, marker, inbreeding or coancestry). This is done by penalising solutions that do not meet the target value, with greater penalty for greater deviation below (or above) the desired target. These are three options:

Tactical Desired Gains upwards: Use this option if you want to target the progeny mean upwards from the current value to at least your chosen minimum value. There is no specific

emphasis on exceeding this value, but if other factors cause your minimum to be exceeded, you are happy with that.

Tactical Desired Gains downwards: Use this option if you want to target the progeny mean downwards from the current value to at most your chosen maximum value. There is no specific emphasis on undercutting this value, but if other factors cause your maximum to be undercut, you are happy with that.

Tactical Desired Gains stabilising: Use this option if you want to target the progeny mean towards your chosen value with no deviation upwards or downwards from that value.

So, under “upwards” and “downwards”, there is zero benefit allocated for overshooting the target. Thus, with a sufficiently strong weighting applied, and a logistically achievable target, the histogram mean marches towards the desired mean, and does not go past it unless other factors drive it to do so.

If the “neutral” value for the item you wish to control is already in the region of your target, and you want to keep it there while you make other manipulations, then you can use the “stabilising” option.

The Tactical Desired Gains approach is not equivalent to independent culling, as, for traits, the target trait and the Index are considered simultaneously when evaluating each candidate mating set.

For more discussion, see comments in [this section](#).

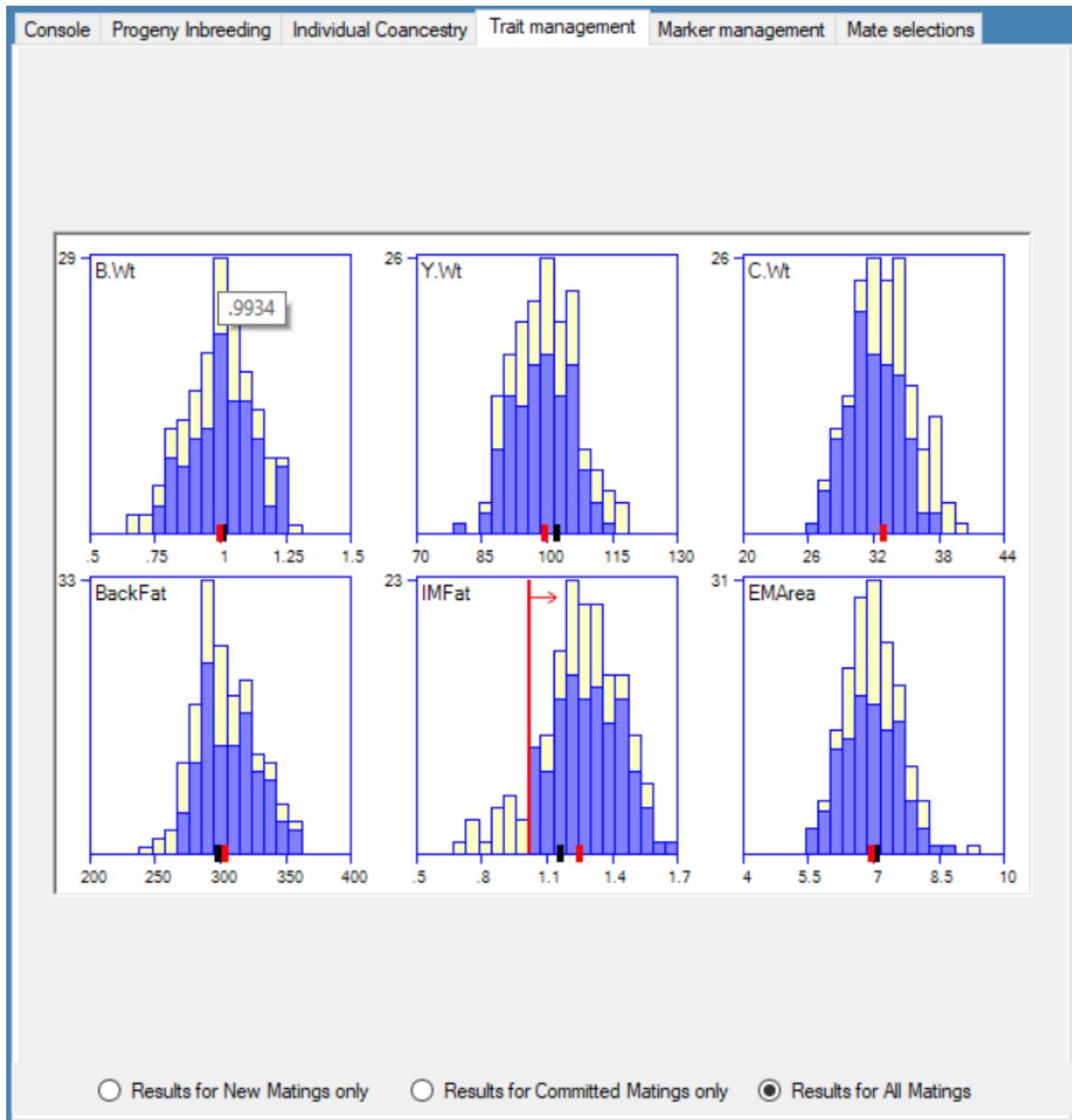
Zooming into a chosen Trait or Marker histogram

If you have many trait or markers on a display, then each histogram may be too small to work with accurately. To zoom into a given histogram, hold down the Shift key while clicking on it. You can continue to manipulate its settings as described above.

To Unzoom, hold down the Control key while clicking on the Zoomed histogram.

Viewing Committed Matings as well as New Matings in Histograms

When your run includes [Committed Matings](#) that are forced to be part of the solution, you can view these in histograms by choosing the appropriate option button on the histogram displays:



The darker bars are results for New Matings, and these change as the solution evolves. The light-yellow bars are for the Committed Matings, and these are fixed ([except for Individual Coancestries](#)).

When viewing Results for All Matings, the Committed Matings bars are not “behind” the New Matings bars, but stacked on top of them, so that the total height in any one column shows the total number of matings in that value range.

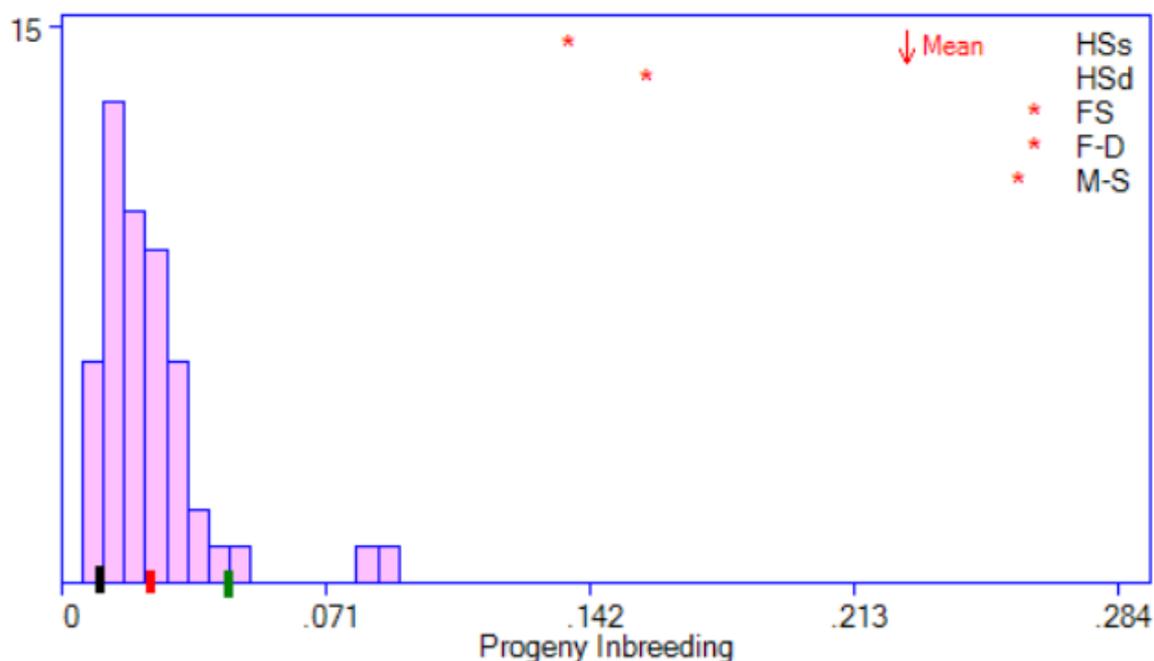
The Red ticks show means for the set of matings chosen (New, Committed or All). Note that [histogram manipulations](#) operate on New Matings alone – so the red ticks will only show how well these manipulations have worked when showing “Results for New Matings only”.

Managing the progeny inbreeding distribution

Inbreeding coefficients are pedigree-derived or GRM-derived, depending on your choice for [ReadGRM](#).

The distribution of Progeny Inbreeding for the current solution is shown in the Progeny Inbreeding tab. You can use the full range of [Tools for managing outcome distributions](#) to manipulate outcomes for Progeny Inbreeding.

To make it easier for you, the Progeny Inbreeding histogram includes red asterisks * to show the minimum level of inbreeding of key related matings. This is described in more detail at [Threshold Management for Progeny Inbreeding](#).



You can see three ticks on the Progeny Inbreeding axis:

Black: This is the mean F predicted for random mating among *candidates*. This is just a guide, as it will generally give a low value, possibly unattainable.

Red: This is the mean F among progeny for the current solution and [set of matings chosen](#).

Green: This is the predicted mean F for random mating among the *selected parents* in the current solution and [set of matings chosen](#), weighted for the numbers of matings they have been allocated.

The difference between the **Red** and **Green** ticks shows the contribution of Mate Allocation to lowering (or raising) Progeny Inbreeding.

Unlike the Trait, Marker and Coancestry histograms, where the **Black** tick always shows the candidate mean for All Matings, the **Black** tick here shows the candidate mean for the set of matings chosen ([New, Committed or All](#)). If all parents in CommittedMatings.txt are represented as current candidates for New Matings, these three means will be equal.

Interaction with Weight on Progeny F

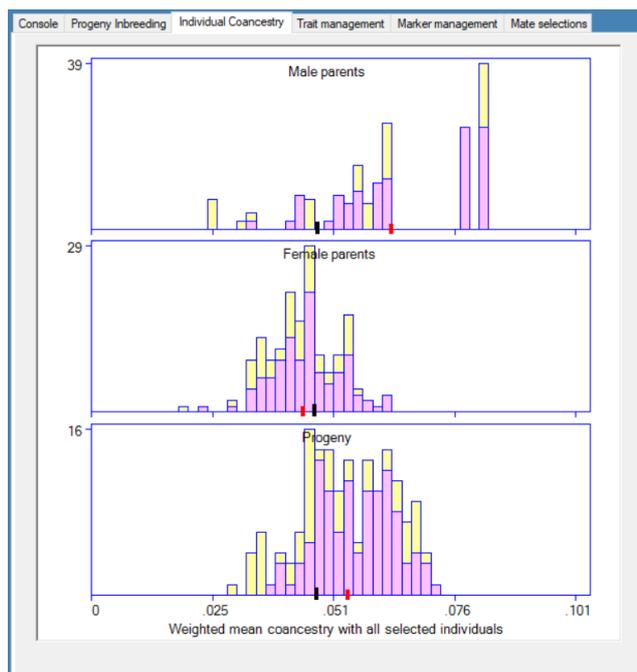
The scaling of weightings to increase and decrease Progeny Inbreeding using Control Types 1 and 2 are the same as for [Weight on Progeny F](#), and so their effects are balancing. You can leave the weighting on Progeny F at -1 (or any other value), and still make any other additional manipulations on the progeny inbreeding distribution.

Managing individual coancestry distributions

Most users will probably not manipulate these distributions, but they do provide insights to some aspects of genetic variation in your population.

The weighted mean Progeny Coancestry is equal to the Parental Coancestry on the [Frontier](#) – but the variation about that mean tells you a bit more. High variation in Coancestries is suggestive of greater differentiation in your population (and/or your current solution) – possibly bloodlines, lineages, dairy pathways, etc.

For many animal species, Male parent distributions might show a small number of sires with high coancestries - they are strongly related to the selected population.



Which sires are these?? ... Click “Print Matings” then “Show sire use” then sort on “Coan.” and you will see them.

Applying downward pressure on the Male distribution does not necessarily reduce the use of these sires ... reduction can also come from reduced use of more closely related dams and other sires. The same sires become less related to the *selected* population. Mate allocation would not play a role in that.

Another consideration is to [reduce the variation](#) in Progeny Coancestry. This will tend to allocate high coancestry sires to low coancestry dams (not to be confused with aiming to reduce progeny inbreeding). If these are elite sires, then you would be hitch-hiking their elite genes on the diverse genes of low coancestry females, possibly helping to ensure the contribution of these elite genes in the longer term. You might take other steps to discourage involvement of females with index values that are too low. Such thoughts remain to be tested.

As Progeny Coancestry is equal to Parental Coancestry you have a different way to control outcome on the Frontier. For example, set to zero [Target Degrees](#) so that Coancestry is not affected by Frontier Balance Strategies, and then manipulate coancestry on the Frontier using

histogram management for mean Progeny Coancestry. You might have to use very strong weightings to get very low Coancestries.

Small Note: When you set TargetDegrees>45 then this places a hard constraint on reduced progeny index, and this overrides any downward pressure on Progeny Coancestry.

Why are there no Committed Matings option buttons for Individual Coancestry histograms?

These are Coancestries with all *selected* individuals. The group of individuals selected changes as the best solution evolves, and so these coancestries change for the Committed Matings, as well as for New Matings. Viewing New Matings alone would thus miss an important part of the impact of your actions. Moreover, the [Frontier](#) shows results for All Matings, and so the mean of the Progeny Coancestry histogram equals the mean Parental Coancestry on the Frontier when All Matings are shown in these histograms.

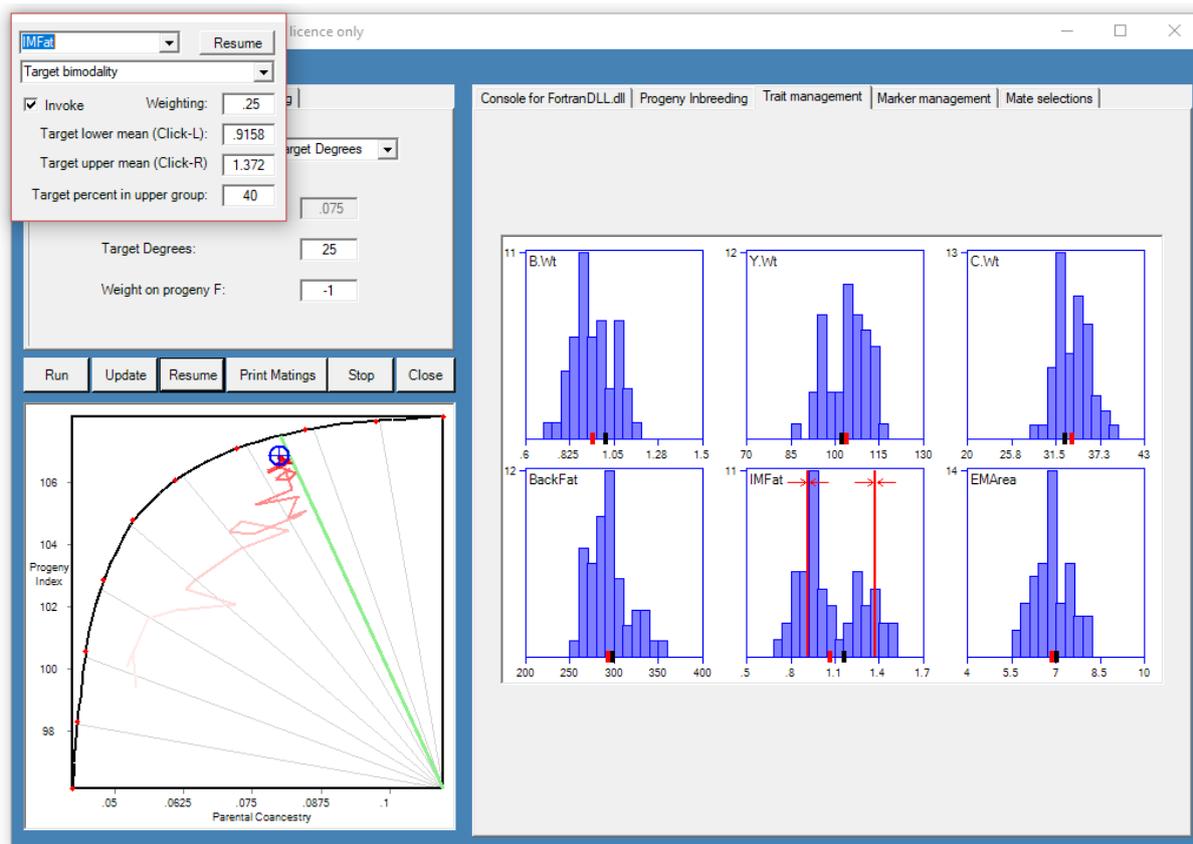
Managing trait distributions

Click on the “Trait Management” tab to manage trait distributions, after your job has passed the frontier building stage.

With the exception of [reserved field names](#), the three ID fields, and fields recognised as genetic marker fields (See [Marker fields in the main data file](#)), *all other fields are taken to be Trait fields*.

This means that you might end up displaying fields such as “Age” and “Index2” as if they are traits. But that is OK – it can be useful for information, and also for intentional manipulation (See [here](#) for some discussion on this).

Once the run is underway, and the frontier has been built, you should see trait distributions under the Trait management tab. (NB. These are distributions of predicted genetic merit in progeny!!).



Missing trait information

Use a period (.) to denote missing trait information in your datafile. Matings that include either one or both parents with missing information for a trait are not included in the histogram for that trait.

PLEASE NOTE: Such matings are ignored by Matesel not only in the drawing of histograms, but in the targeting of outcomes when using the Trait Management control tools. For example, if you set Tactical Desired Gains Upwards to value 10, Matesel will show the target achieved (red tick meets red target line at value 10) when any set of [New Matings](#) with known parental EBVs meets that outcome. The other matings are ignored.

[If individuals with missing trait EBV information are pedigree connected to individuals with records, a trick is to fill in the missing EBVs by running a BLUP analysis (eg. in Pedigree Viewer) fitting these EBVs as the ‘phenotypes’ of interest, setting heritability to one, and fitting no fixed effects other than the mean. Check your results – eg. A newly calculated “missing” progeny EBV is mean of “non-missing” parents’ EBVs]

Predicted progeny trait and index means reported in OutSummary.txt are calculated from all the non-missing parental values – they are the average of the sexes for the means calculated from non-missing values. They are not simply the mean of matings where both parents have non-missing values. Thus, for a mating between a sire with a trait EBV in the input file, and a dam with a missing value for that trait EBV, the sire contributes to the reported progeny mean, but the dam does not.

Managing genetic markers

Click on the “Marker Management” tab to manage marker allele and genotype probability distributions, after your job has passed the frontier building stage.

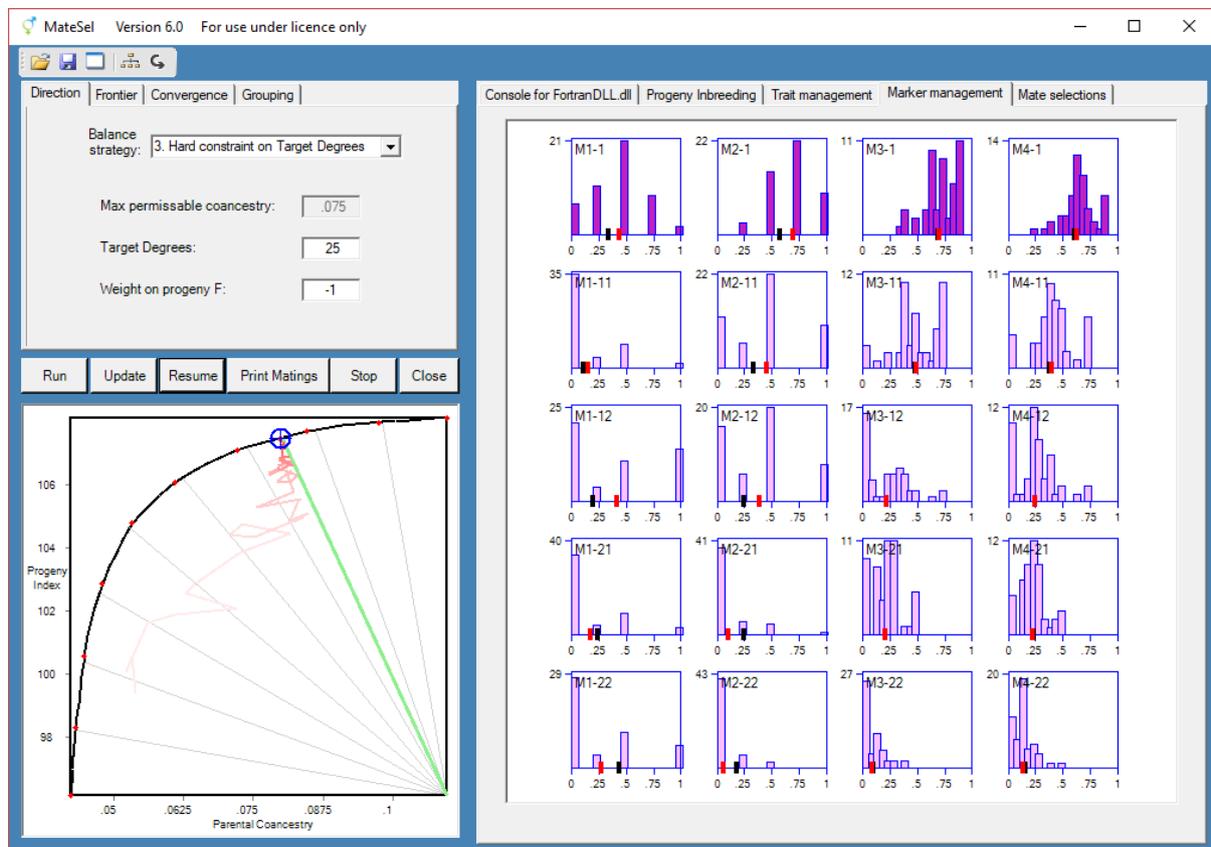
Each distribution on the display can be manipulated as described under “Managing trait distributions”, above.

Each column of distributions relates to one marker locus. A maximum of 6 marker loci are shown at one time. See [Panning across marker loci](#).

The top row of distributions shows the allele frequency distributions in progeny for the first allele at each locus (column).

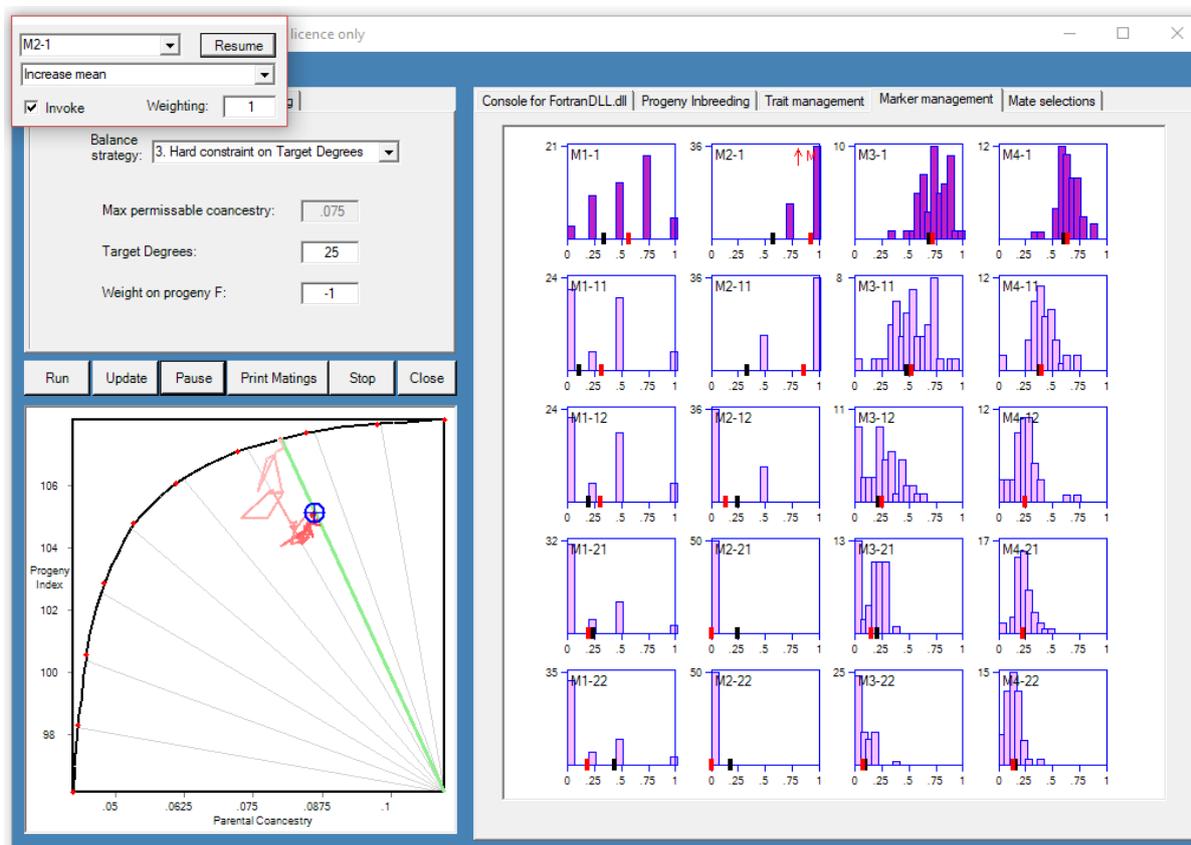
The remaining rows show the genotype frequencies for the 4 ($n*n$) genotypes for the 2 (n) alleles at each marker locus. The system can handle any number of alleles, but of course the number of rows of graphs becomes high for $n>2$.

In the example below, the first two markers are presented as genotypes in the input file, and the second pair of markers have genotype probabilities presented in the input file. Input file requirements will be described elsewhere. For markers 1 and 2, prospective progeny do not always have certain genotypes – only their parents have known genotypes.



Marker outcomes are managed in the same way as trait outcomes. In the following example, strong positive emphasis has been placed on allele 1 of marker locus 2 (M2-1). One result is

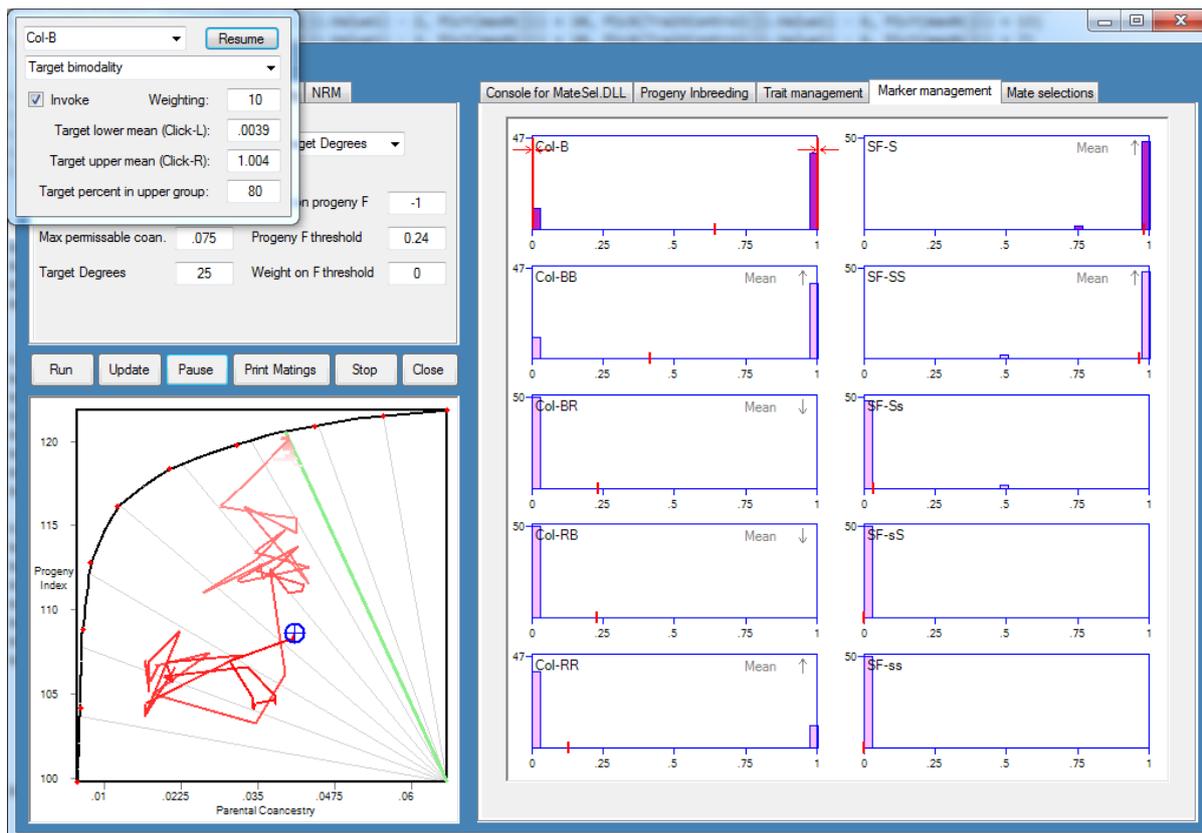
that only genotype 11 sires are used, as the probability of 21 and 22 genotypes are zero for all progeny.



The next example uses a red/black color locus and pattern locus. The one direction set in this example is for bimodality in allele frequency at the color locus, aiming for 80% Col-BB and 20% Col-RR – as might be the case for a breeder wanting to become pure black (no carriers) but with some red calves to sell off. This outcome is met perfectly (given a high weighting of 10 for this outcome), but at considerable cost in genetic gain. A better outcome can be found with some compromise on the color desires.

Weightings on the lower graphs (genotype frequencies, rather than allele frequencies) will be important for imprinted loci – eg. Callipege, where only one of the two heterozygote genotypes shows the effect. Such genotypes can be targeted specifically.

This set up gives power to target a wide range of desires (maybe all?). It takes a bit of time playing around to see the different manipulations that can be made. Probably not for the average breeder without a bit of training.



Providing further output to reflect eg. numbers in each phenotypic category would be quite a big job (to make general across markers), and would make output more complex. But it could be considered. As it stands, such outcome can be seen by inspecting the graphs and knowing the phenotypes associated with each genotype.

However, for longer term objectives beyond the next calf crop, the key graphs are simply the allele frequencies in the top row. Maybe that will be the biggest use.

Missing genotypes

An individual with missing genotypes or genotype probabilities is allocated the raw mean of genotype probabilities for current candidates (ie. excluding those who are only listed as candidates in CommittedMatings.txt) that have legal genotypes or genotype probabilities.

Panning across marker loci

A maximum of 6 marker loci are shown at one time. To move to the next set of 6 marker loci, hold down the Alt key and click of the right side of the Marker management frame (eg. click anywhere in the last column of histograms). To move to the previous set of 6 marker loci, hold down the Alt key and click of the left side of the Marker management frame.

Hiding histograms

If you have many histograms available to be displayed, then speed can be slow in the Web version. This is notable for many recessive lethal loci, as described under “Handling multiple lethal recessives”. For example, 100 biallelic markers makes for $100 \times (1+2 \times 2) = 500$ histograms to be updated each time the solution improves – and that makes for a lot of data transmission.

The hard work has been done to hide marker histograms in a flexible manner, but this has only been implemented in a simple manner so far, targeting multiple lethal markers. This has been done because these markers are handled collectively using the LethalA and LethalG “trait” distributions, and there seems no need to manipulate them individually.

Hiding markers is simply controlled by the case of the first “L” in the fields LethalA and LethalG in your main data file. Here are the rules:

1. To show all collectively managed lethals use a capital L in LethalA, else use a small l.
2. To show all other marker loci use a capital L in LethalG, else use a small l.

For example, if you have 100 marker loci and collectively managed recessive lethals are in a block from 10 to 95:

LethalA header	LethalG header	Markers 1 to 9	Markers 10 to 95	Markers 96 to 100
LethalA10	LethalG95	Show	Show	Show
lethalA10	LethalG95	Show	Hide	Show
LethalA10	lethalG95	Hide	Show	Hide
lethalA10	lethalG90	Hide	Hide	Hide

So, the most sensible option here, when using the Web version, is the second one (use trait headers lethalA and LethalG), so that you hide histograms for the block of recessive lethals, but show the other marker loci, which are presumably of individual interest (coat color etc.).

Of course, you will need to keep your collectively managed recessive lethal loci in a block. That block does not have to be “in the middle” of the markers set, as in the example above, but it can be.

Note that all marker loci are always included in Matesel.ini, and the weightings etc. in that file are invoked during the run. This way you can manipulate the distributions of hidden markers, albeit non-interactively. However, at least for testing purposes, you can show all marker histograms and manipulate them interactively.

Threshold Management for Progeny Inbreeding and other factors

The Threshold Management setting controls two features to do with applying threshold constraints: Dynamic Threshold Management and Direct Threshold Compliance.

Dynamic Threshold Management for Progeny Inbreeding

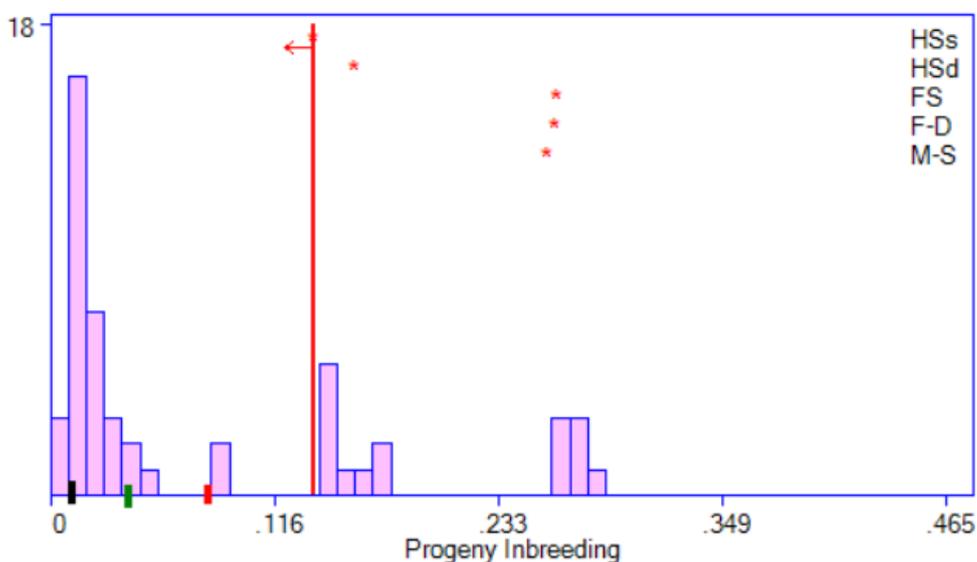
This feature discovers where to make a threshold on Progeny Inbreeding in order to help avoid full-sib (FS) matings, half-sib (HS) matings and parent-offspring (PO) matings. [This replaces the old user-fixed threshold for inbreeding, which led to problems when it was left fixed over years.]

The Matesel console will show you the minimum inbreeding values wherever such relationships are possible for your dataset, as in this example:

```

Minimum progeny inbreeding for key related matings:
Half sib via sire:      0.1366
Half sib via dam:      0.1580
Full sib               0.2633
Father-daughter:      0.2623
Mother-son:           0.2581
Maximum progeny inbreeding threshold set at:    0.1366
  
```

You will also see these values denoted by red asterisks under the Progeny Inbreeding tab in the GUI versions:



Just for illustration in this example, the [Weight on Progeny F](#) was set to +1 to ensure that inbred matings can be seen on the histogram ([Red tick > Green tick!](#)), and the [Resume button](#) was not yet clicked for setting the maximum progeny inbreeding at 0.1366, to persuade against these matings! The vertical red line at 0.1366 is automatically set by Matesel. You can see the two groups of HS matings and FS+PO matings.

Using the NRM for Progeny Inbreeding: The threshold values will be at least 0.125 for HS matings and at least 0.25 for FS and PO matings. However, these values will be higher in most cases due to a history of inbreeding

Using the GRM for Progeny Inbreeding: The threshold values will be determined by the genomic relationships you [supply](#). These may or may not be similar to the NRM thresholds, depending on your genomic resources and methods of calculating and scaling relationships.

To invoke Dynamic Threshold Management in Matesel.ini

Enter 1 to avoid matings with 1 or more parents in common between the two candidates
 Enter 2 to avoid matings with 2 parents in common between the two candidates.

Option 1 covers FS and HS matings, but also PO matings.

Option 2 covers FS and PO matings, but not HS matings.

For these options to take effect, you must ensure that you have a suitable Weighting and Invoked “on” for Progeny Inbreeding. For Batch runs this must be done in Matesel.ini:

Number	Invoked	ContolType	Weighting	Target1	Target2	Target3
One line for Progeny Inbreeding histogram control:						
ProgInb	1	8	1	0.04475679	0.04475679	50

For GUI runs you can also set these inside the application at run time.

According to the settings for Threshold Management in Matesel.ini, the ControlType for ProgInb in Matesel.ini will be overwritten at runtime with 8 ([Set maximum at a boundary](#)). Also, the criterion for ControlType 8 (Target1) will be reset at runtime to the maximum progeny inbreeding level possible that is still small enough to avoid the types of mating that you have specified. Of course, the success in avoiding such matings depends on weightings that are used for this and other factors.

Direct Threshold Compliance across Traits, Markers and more

[Note: Substantial improvements to the performance of the core optimisation algorithm in Matesel as of Fortran DLL version 12.3 probably make Direct Threshold Compliance redundant. We suggest that you do not use this feature except perhaps for some detailed testing, and possible use if you find that it does in fact make a significant improvement for your scenarios.]

Direct Threshold Compliance is designed to improve speed where you set threshold conditions in jobs with very large numbers of candidates and/or matings (typically several thousand matings).

This feature invokes an algorithmic method to more quickly arrive at competitive solutions that comply with your threshold constraints (if these are possible, and sufficiently weighted). It works across all components with a histogram, except for Sire and Dam Coancestries, ie. Traits, Markers, Progeny Inbreeding and Progeny Coancestries.

Background and recommendation: If there is just one mating that is breaking a threshold constraint, then, using the evolutionary algorithms alone, the dice must be rolled across all candidates and matings in order to find complying solutions that are also competitive. This can be very slow for jobs with many candidates and matings, as most new solutions have been perturbed in such a way that they break a delicate balance that has been established across other factors. Direct Threshold Compliance uses a hybrid approach that helps maintain this established balance while making progress on threshold compliance. It does not force compliance in your results, but it takes a direct route to creating a higher proportion of candidate solutions that are more likely to

comply, using meiotic drive. It is recommended that you also try making runs with Direct Threshold Compliance switched off, to compare results across all factors.

Add 10 to the Threshold Management setting in Matesel.ini to additionally invoke Direct Threshold Compliance across all factors with a histogram. This gives the following:

Threshold Management entry in Matesel.ini	Avoid HS matings	Avoid FS and PO matings	Direct Threshold Compliance
0	No	No	No
1	Yes	Yes	No
2	No	Yes	No
10	No	No	Yes
11	Yes	Yes	Yes
12	No	Yes	Yes

NOTES

- Direct Threshold Compliance takes typically 20% more time per generation of evolution, but this is probably more than compensated for by a faster rate of evolution per generation.
- Direct Threshold Compliance does not work with [Hard implementation of the Location feature](#) – whatever setting you make in MateSel.ini, DTC will not be used.

The Mate Selections tab

The contents of the Mate Selections tab shows the current or final mate selection solution, together with predicted mean Index and Inbreeding for the progeny of each mating. It is not functional until the frontier has been built and the main run started. It is populated whenever the Print Matings button is clicked, the Stop button is clicked, or its own update button is clicked (ie. It does not update automatically except after the Stop button is clicked).

You can choose “Show sire use” to show sire index, mean coancestry with all candidates including self (CoanCand), mean coancestry with all selected individuals including self (CoanSel), and number of matings per sire. This is very helpful for getting a feel for the diversity properties of your solution.

You can click the column headings to sort columns. For an hierarchical sort, click on the minor columns first.

Output files

Output files all start with “Out”, and are generated in the same folder as your main data file. You should find:

OutSummary.txt		
OutMatings.txt	and	OutMatings.csv
OutResults.txt	and	OutResults.csv
OutBackups.txt	and	OutBackups.csv
OutMarkers.txt		

OutGroups.txt
 OutFrontier.txt
 OutSeqFile.txt (can be censored for security purposes)

All these files are produced each time you click ‘Print Matings’ (but possibly not OutBackups.* ... see [here](#)) or click ‘Stop’. The exception is OutSeqFile, which only needs to be printed at the start of the run, as it does not change after that.

You might find other files such as OutSolutionError.txt (in this case, if your [manually entered solution](#) breaks constraints that you have specified), and OutErrorMessage.txt, if you have had a [crash](#).

Output files that contain “DLL” in their name are produced directly by the Fortran DLL, which does not know the IDs of the individuals in your datafile. They contain sequential IDs (1, 2, ..., n) instead. Not so useful for you.

Mating results are printed to the file OutMatings.txt (and .csv). There is one line per mating, with fields as follows:

SireID	The male to use as the sire of this mating
DamID	The female to use as the dam of this mating
GrpS	The sire’s group
GrpD	The dam’s group
IndexS	The Index value of the sire
IndexD	The Index value of the dam
F	The inbreeding coefficient of the progeny from this mating

The OutResults files contain rather full information, including what you find in the OutMatings files, plus trait and marker results, plus more detailed backup mating results. For the latter, n-ID, n-Index, n-Crit, n-% and n-Br all refer to results for the nth backup male, and are its ID, its Index value, its Criterion value (value of whole mating set when it replaces the primary male), its Percent value, and whether its use causes a breaking of its [Maxuse](#) value (1, else 0).

OutFrontier.txt gives the Final Frontier, not the cowboy one, but the points used to draw the MateSel Frontier, including points added during the main run. This may be useful for you to help illustrate results outside MateSel, for example in Excel.

OutSeqFile.txt gives the key between sequential ID as used in the Fortran DLL, and the ID as supplied in the main data file.

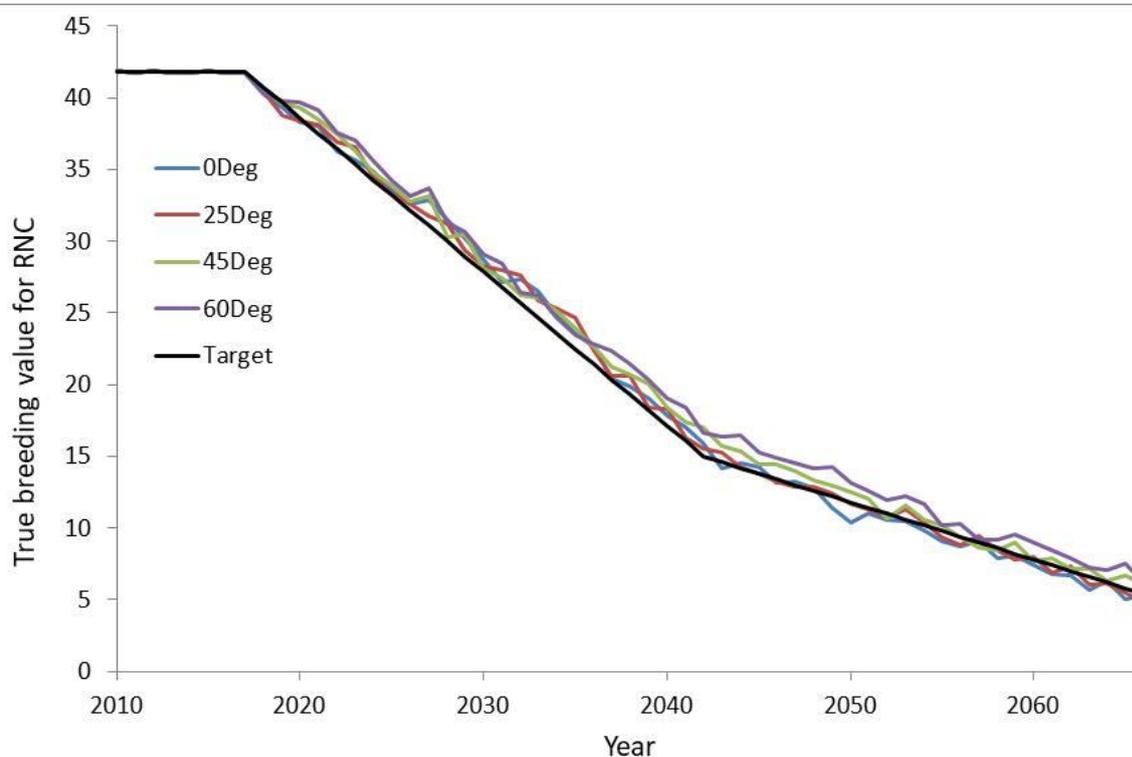
How do I ... ?

Get what I want – Tactical Desired Gains

Sometimes we want some control over the genetic gains made in each trait. We can build selection indices (eg. to make Index values for Matesel) using a Strategic Desired Gains tool such as [Desire](#). This gives some indirect control over the genetic gains made in each trait through use of “implied economic weights” that are predicted to give the responses desired when using the resulting Index. However, the actual outcome depends more directly on the candidates involved and their EBVs. The tactical approach used in Matesel uses these EBVs themselves.

How to do this is described [here](#).

The following shows an example of using Tactical Desired Gains over years in a computer simulation, from Suontama et al. (2018):



The simulation program PopSim has used the Tactical Desired Gains feature in Matesel to breed a simulated population of pine trees over many years, with a desired path for resistance to a disease being the black Target line. Various treatments (difference values for [Target Degrees](#)) give good adherence to the target. [TDG works on EBVs, of course, and this response is shown in TBVs, which probably explains some deviations seen.]

It should be noted that the best route to achieving desired gains is probably to get the selection index performing at least approximately as you want – giving something close to your desired pattern of response across traits. Then you can use Tactical Desired Gains to refine the outcome.

However, in some cases the use of Tactical Desired Gains will be important – e.g. to track a required pattern of response in resistance to a disease, tolerance to climate stress, or looming market demands.

Boosting female reproduction: MOET and IVF

The file Matesel.ini has a line that looks something like this:

```
0          , MOETing
```

Enter a 0 (as above) for IVF/JIVET, or a 1 for MOET (*in-vivo* fertilisation, Multiple Ovulation and Embryo Transfer). If you are doing neither (all female [Maxuse](#) values =1) then it does not matter what is entered here.

Under MOET, all progeny from any one female have the same sire. Under IVF/JIVET, it is possible for all progeny to have a different sire (although they could conceivably all have the same sire, eg. when that results in the lowest progeny inbreeding values and there is a high weighting on that).

Set female [Maxuse](#) numbers to 2 or more, this being the maximum number of *planned pregnancies in recipient females* for the donor females concerned. You can set these figures globally in InpOneGroup.txt or InpGroups.txt, or set them for individual females in the main data file. You might also set values for female Minuse, so that you have a standard value of range of number of planned pregnancies (or recipients) per female.

Note that the total number of matings input to Matesel (when not grouping) or reported by Matesel (when grouping) is the total number of planned pregnancies. So, an insemination into a female involving eight predicted embryo transfers is in fact eight matings, not one.

Take account of this when setting the number of matings. In the file InpGroups.txt, note that for matings per female group: “Numbers can be > #candidates, or proportions>1, to drive MOET and IVF/JIVET”. Under IVF, you can reduce or [avoid the incidence of duplicate matings](#). But note that duplicate matings can be nominated by Matesel for good reasons!

Please Note: Under MOETing=1, use constraints for males operate differently. An insemination into a super-ovulated female is taken *in the code* as a single use. However, the contribution of this to the reported Use value for that male is the number of matings (planned recipient pregnancies) for that female. Moreover, AbsMinUse and MinUse constraints can be violated for males under MOETing=1. These things are a consequence of how MOETing=1 works: Matings are allocated to a female as for IVF/JIVET, but then, before evaluating the solution, the first allocated sire replaces all the other allocated sires for that female, and this is done *after* use constraints have been applied.

The MOETing switch operates across the whole dataset – it applies to all females with Maxuse>1, either all IVF or all MOET. There could be a need to let this operate at the Group level – so that for example one Group of females are to be available as IVF donors, and a different Group in the same analysis are to be available as MOET females. We have never had such a request, but we may provide such functionality if sufficient demand arises.

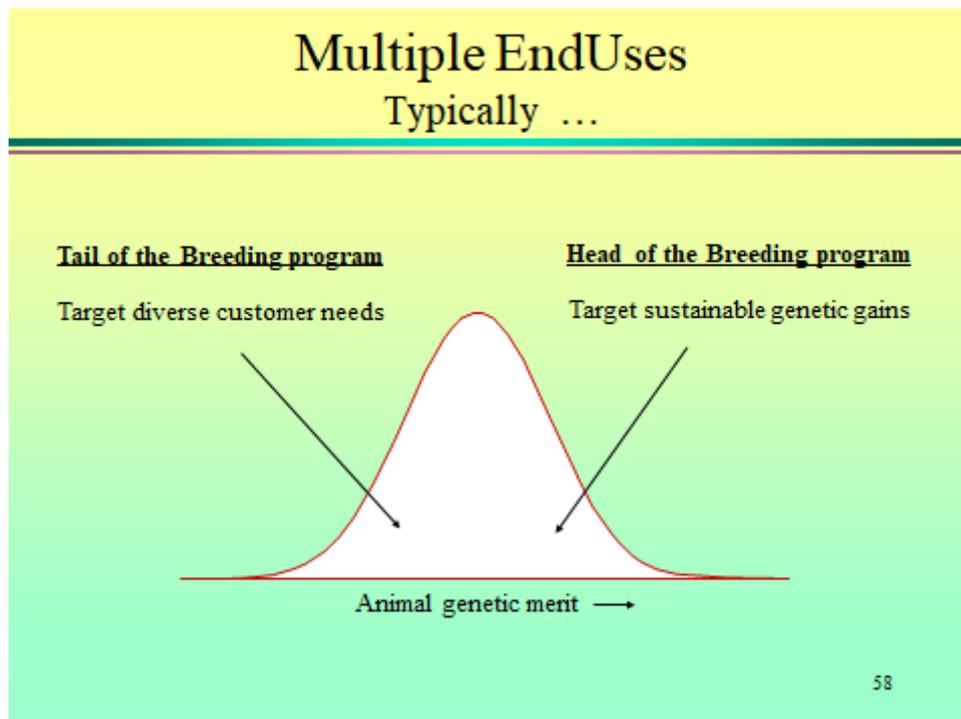
Multiple EndUses: Handling different target end-uses

In addition to matings targeted at long-term nucleus gains, you might want to make matings to give progeny that are targeted towards other uses such as:

- To target the production of seedstock for sale to a range of customers with differing needs: eg.
 - Terminal versus maternal bulls
 - A specific disease important to some but not to others
 - High versus low demand for marbling in the carcass
 - Targeting a specific set of “trait leaders” in dairy cattle.
 - Customers in different environments
- To contribute to a multiplier tier with the same breeding objective
- To contribute to a multiplier tier with a different breeding objective
- To target a specific product niche
- To make resources for specific testing in some way – eg. high versus low EBV groups for a specific trait.

Generally, you will want to use the best stock for the nucleus (most suited to the nucleus objective). And genetic diversity will generally be important only in the nucleus (in fact this is assumed). So in some ways, this is all about targeting the head of the breeding program at the nucleus, and the tail of the program somewhat towards diverse customer needs.

However, in using Multiple EndUses you are not committing to the different directions you might choose ... what you are doing is “loading the dice” in a manner that will give you better opportunities when deciding the fate of the resulting progeny – almost always after they have records of their own, and their EBVs have deviated from their parent-average predictions. You should be able to make groups for selection, disseminations and culling that are better suited to these different destinations. See Kinghorn et al. (2015) for an example.



Unlike Grouping, Multiple EndUses does not pre-allocate which division each individual or mating is assigned to. This comes out as a result of the optimisation. This means that the main datafile is not affected.

You invoke Multiple EndUses by setting the appropriate value in Matesel.ini to 1:

```
1           , Multiple EndUses (1, else one EndUse (main breeding program) only)
```

... and by populating the file EndUses.txt (Note that the descriptive lines of text are all single lines – they are wrapped around here because of space limitation only):

```
Number of EndUses, including nucleus breeding program as EndUse number 1:
3
Relative Trait Weightings on the Index then each trait. Enter #EndUses rows
of 1+nTraits columns:
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
Relative number or proportion of Matings in each EndUse (enter in one row):
20 15 15
Relative EndUse Weighting of each EndUse for genetic merit (enter in one
row):
1 0.75 0.5
```

The weightings in the last row can differ in magnitude substantially. In this case, this has the effect of making EndUse 1 (the main breeding program) most probably capture all the best stock for its objectives, and, other than that, EndUse 2 gets the pick over EndUse 3. In this case the weightings only serve to essentially rank the EndUses on their priority for good stock, appropriate to their own needs.

If the latter two EndUses were of equal importance (eg., possibly, speciality wool production and speciality meat production) then they might be allocated equal weights in the last row. To

help make these weightings work appropriately, each EndUse index is scaled within the program to range 0 to 1. This helps avoid eg. the EndUse with the most extreme Relative Trait Weightings having undue influence on outcome. However, all reporting uses unscaled indices.

Results

On the Windows interface, click the Mate Selections tab, and click Update if needed. You will see the EndUse allocation for each mating to be made. Clicking on the column header will sort by that column.

Look at the end of the file OutSummary.txt, which is created each time you click “Print Matings” or “Stop”. You will see the number of matings, mean Index (this is the EndUse 1 index reported for each EndUse), and value for each trait within each EndUse. You will also find the mating list in file OutMatings.txt.

Some ‘EndUse’ items to note:

- The actual number of Matings in each EndUses will be calculated from the total number of matings, using the relative numbers or proportions entered in EndUse.txt as weightings. Notice that the total number of matings may not be as entered in Matesel.ini (or on the interface when “Run” is clicked), as this can be overwritten by the program where, for example, there are insufficient candidates, or where you have Grouping set to true and number of Matings is set by the program according to your settings in the file InpGroups.txt.
- If you specify Index and Trait weightings for EndUse 1 as being anything other than a weight of 1 on Index and a weight of 0 on each trait (ie. 1 0 0 0 0 ...) then you will be re-defining the index, and this new index will be the one reported as “Index”.
- Trait weightings should be ordered according to the order of appearance of traits in the main datafile, even if they are separated by other fields, but with the weighting on Index always coming first.
- Take care about how many “traits” you actually have. For example, Yob (Year of Birth) will be treated as a trait, and so you must include it in EndUses.txt. Probably with a weighting of zero, but maybe not if you want to manipulate it!!
- If using Multiple EndUses, the Frontier will relate to EndUse 1 alone, and Progeny Index and Parental Coancestry will relate to EndUse 1 alone. However, weightings etc. related to progeny inbreeding will act across all EndUses.
- All reporting of progeny merit and coancestry, both on the interface and in output files, relates to EndUse number 1, the main breeding program, even for matings allocated to other EndUses. It is suggested that you add other EndUse indices as extra columns in the main datafile, if you want to track these explicitly.
- All reporting of progeny inbreeding relates to all EndUses collectively.

- If you strongly emphasise genetic diversity (eg. high [Target Degrees](#)), then it is possible that EndUse 1 (the nucleus) will be of lower Progeny Index, or lower merit for key traits, compared to other EndUses that are not held back by emphasis on diversity.
- The frontier building now depends on mate allocations, as well as selections. This is because it is mating pairs that compete to enter the different EndUses. This can make convergence slower for Frontier building. However, potentially lower speed may be offset by the fact that calculations are smaller for frontier building – as only EndUse 1 matings are considered. Nevertheless, it could be a good idea to increase Front0Gens and Front90Gens in Matesel.ini (these being the important anchor points for tracking Target Degrees), as well as FrontMidGens to get a cleaner looking Frontier at least to start with (cosmetic value only).
- If, for example, you weight all the EndUses very low compared to the first EndUse, then Matesel could feel it is quite well converged at an early stage – as all those lower EndUses have relatively little impact on overall fitness of solutions. Accordingly, you should probably test different relative weightings on EndUses, to be sure that the lower-weighted EndUses have the sorts of outcomes you might want. And of course, be sure to let your run go for a sufficiently long time to get reasonable convergence. ***When using Multiple EndUses, convergence can be very slow, even when the solution looks converged on the frontier. Give the job a lot of time!***
- The Frontier is built with an objective that pays full attention to EndUse 1, “the nucleus”. This means that a fully converged main run may not reach the Frontier because the other EndUses are detracting somewhat from that simple objective.
- You can use both Multiple EndUses and Grouping simultaneously, such that Grouping constraints and different Multiple EndUse weightings will both be applied. However, they will be disjointed. This means that, for example, you cannot specify that EndUse 2 matings must come from Group 2 males mated to Group 3 females. HOWEVER, you can get around this sort of thing quite easily ... Eg. to force a given group of individuals into Enduse 1, make an extra trait column with only these individuals having positive values. Put a big weight on that “trait” for EndUse 1 and these individuals will tend to be used Enduse 1. Use very big weights to force the issue, given that other constraints do not preclude this.
- When you see EndUse = -1 in output, then the mating concerned is a committed mating that was included in the file CommittedMatings.txt. Committed Matings are not allocated to any of the EndUses in EndUses.txt.

Accommodating bisexuality

If your main data file has individuals that are candidates to be used as both male and female parents (including candidates for selfing) then set BiSexCands to 1 in MateSel.ini. [Note, if this value is set not equal to zero, then ‘normal’ main data files will not work properly, so be sure to revert to zero as appropriate.]

In the data file, the column that sets [Maxuse](#) (and [Minuse](#) and [AbsMinuse](#) if included) needs to contain 10000*male use + female use for all candidates. For example, if the maximum use for

an individual is 63 times as a male and 42 times as a female then its entry for Maxuse should be 630042. [This restricts the maximum 9999 uses as each [sex](#).]

Reading the usage parameters (Maxuse, Minuse and AbsMinuse) is essentially the same as for no bisexuality: Values are read from the grouping file ([InpOneGroup.txt](#) or [InpGroups.txt](#)), except when the value for a sex (or sex/Group) in that file is set to -1, in which case the values for individuals of that sex (or sex/Group) in the datafile will be used. In this way it is possible that, for example, male usage is taken from the parameter file, and female usage is taken from the values for individuals in the datafile.

Avoid matings that are duplicate, or reciprocal, or selfings

Where at least some members of each sex have [Maxuse](#) > 1 there is the possibility of duplicate matings. Under bisexuality, there is also the possibility of reciprocal matings and/or self matings. To avoid these, set the “Switches to control avoidance of reciprocal, duplicate and self matings” in Matesel.ini as follows:

To avoid duplicate matings, set to value 1 *in its hundreds column* and ensure that the parameter MOETing is set to 0 in Matesel.ini.

To avoid reciprocal matings, set to value 1 *in its tens column*

To avoid self matings, set value 1 *in its units column*

For example, to avoid all three types of matings, use this:

```
111      , Switches to control avoidance of reciprocal, duplicate and self matings
```

And also enter a single weighting on these issues, a few lines below in Matesel.ini, eg:

```
1.0      , Weighting for avoiding reciprocal, duplicate and/or self matings
```

A positive weighting avoids the type of mating(s) concerned. If you still see duplicate, reciprocal or self matings in the converged result, increase this weighting until you get the outcome you desire. The downside of using a stronger weighting is possibly slower convergence, but this may be a minor issue for relatively large problems where such matings will probably occur at lower frequency. The compromises made in Progeny Index and parental coancestry can be large. You must ask yourself if these compromises are worth it!

Note that the feature to avoid duplicate matings also operates on normal (‘Sire-Dam’) matings, typically in IVF programs where duplications are possible.

Managing backup sires

A note up front: Is there a better way to do this? Probably yes, if it fits your operations:

When a backup sire is nominated for a given mating, the calculations involved are done in ignorance of which other matings will need a backup sire to be used – as the broken legs

and failed libido that remove the primarily nominated sires from use have yet to take place. So, at this stage, the result will be unavoidably sub-optimal. For optimality, we want to know which matings need to have a backup sire actually used. To do this is easy, if it fits logistically:

1. Do the Matesel run without invoking backup sires.
2. At the stage when you are ready to use backup sires, add all the successful matings (those that do not require a backup sire) to the file [CommittedMatings.txt](#).
3. Set the [Committed Matings switch](#) to 2 (manual setting of use constraints) in Matesel.ini. Option 2 means that eg Maxuse for a sire covers these committed matings plus backup matings made at step 6. Use Option 1 to set usage values for step 6 matings alone.
4. Remove the failed sires from candidacy (Maxuse=0 for these sires), if appropriate.
5. Add chosen backup sires to candidacy (Maxuse>0) if needed.
6. Re-Run Matesel. You should now get an optimal result for new backup sire allocations.
7. Repeat steps 2-6 above for later stages of failed matings detected, if and as required.

You can nominate the number of backup sires to report per mating in InpOneGroup.txt and InpGroups.txt. In InpGroups.txt you can also nominate which groups can provide backup sires.

Results are shown in OutBackups.csv (and more information in OutResults.csv). This is how it works: For each mating, the selected sire is replaced one-at-a-time by all other legal candidate males. For each of these the full solution is recalculated, including all issues that you have set up in the run (including impact on trait distributions etc).

Please Note: For the full solutions calculated in order to rank backup males, Balance Strategies that involve hard constraints (numbers 1 and 3) are replaced by the equivalent balance strategies that involve soft constraints (numbers 2 and 4 respectively). This is because, in some cases, all alternative males are rejected as backups because their use would result in breaking the hard constraint, incurring a very negative value of no use for ranking.

The percentage figures reported are on a linear scale of 100% for the best selected sire down to 0% for the worst legal replacement. If a backup has a percentage figure of >100%, that shows that your run has not fully converged (the selected sire is not the best sire), and/or that by using that backup, you are breaking its Maxuse constraint, if allowed (see [Use a male for backup even if it is at its MaxUse limit](#)). Minuse constraints are not considered when Backup sires are evaluated.

Use the percentage figures a bit like this:

I have a 98% backup right here next to me, and a 99% backup two paddocks/barns away from me, so I'll use the 98% one.

I have a 92% backup right here next to me, and a 99% backup two paddocks/barns away from me, so I'll make the extra effort and go get the 99% one.

Backup all matings of a sire as a block

To invoke this, set the switch integer to value 1 *in its thousands column* in Matesel.ini, eg:

```
1001           , Switches to control backup matings
```

If this is done, then you will find that for all females in the same female Group that have been allocated to a given sire, the backup sires for these females will all be the same, and in the same order.

This may be useful for paddock mating, where you want to keep the group of females intact when choosing a backup male.

It is also likely to be invoked when you use information on location of females to allocate males to sequential groups of females (See [Manage cost of assembling mating groups](#)).

The default (0 in the thousands column) is to find the most suitable backup males on a female-by-female basis.

Restricting backup sires to those that have not been selected

To invoke this, set the switch integer to value 1 *in its hundreds column* in Matesel.ini, eg:

```
101           , Switches to control backup matings
```

In the event that no unused sire is available in a sire group that is legal for the female concerned, the best sire that is already used will be chosen. Thus, for multiple backup sires requested, already-used sires may be included in the list, but they will be ranked at the end of the list - but still with the better ones nearer the front of the list. In effect this is done by subtracting 100 from their percentage values. Where no grouping-legal sire is available to fill a position, then no sire is nominated.

Here is some information extracted for one mating:

Backup#	ID	nUSEs	%Value
1	719	0	89.1
2	903	0	85.3
3	926	0	84.0
4	900	0	76.8
5	906	0	70.0
6	927	0	69.4
7	925	0	65.7
8	901	0	58.6
9	721	0	49.0
10	815	0	44.9
11	896	0	43.1
12	825	0	33.5
13	813	0	24.6
14	918	0	24.1
15	930	0	13.6
16	921	0	4.0
17	895	0	0.5
18	897	0	0.0
19	819	1	-6.9
20	818	2	-14.3
21	919	2	-15.1
22	824	2	-18.7
23	816	9	-18.7
24	894	1	-19.2

25	898	1	-19.4
26	822	1	-24.3
27	924	3	-24.4
28	904	1	-24.9
29	823	2	-25.3
30	902	2	-26.8
31	905	2	-27.1
32	928	2	-28.5
33	899	2	-34.3
34	817	4	-35.3
35	820	3	-39.0
36	720	3	-43.0
37	920	2	-44.8
38	722	2	-46.1
39	821	2	-79.1

Notice that the Percent values for backups that have already been selected are negative. Also note that the worst unused sire has a percentage value of zero.

Use a male for backup even if it is at its MaxUse limit

To invoke this, set the switch integer to value 1 *in its tens column* in Matesel.ini, eg:

```
110           , Switches to control backup matings
```

You might choose to do this when the [MaxUse](#) value you specify is in fact quite flexible. Note that nomination as a backup male does not affect its Use value, so it is possible that a key male will be represented many times in the backup file – many more than its Maxuse number. You can always choose a backup male from further down the list.

In file OutResults.txt, the field n-BR contains a 1 if the n-th backup sire is breaking its Maxuse limit. Percent value figures for such sires can be substantially greater than 100%.

Print backup matings to file(s) even if the Print Matings button is clicked

To invoke this, set the switch integer to value 1 *in its units column* in Matesel.ini, eg:

```
011           , Switches to control backup matings
```

In some cases you can save a lot of speed if you set this to OFF (0), especially if you have a lot of male candidates. This is because, for every mating, all male candidates are tested as a backup – replace the primary male and evaluated for the full prevailing objective.

Matesel will always print backup matings to file(s) when the Stop button is clicked.

Committed Matings: Prior contributions and pre-set matings

Background

This feature accommodates matings that have been committed to. There can be many reasons to pre-decide matings - and many of these may relate to the preference of the breeder.

However, the key technical reason for doing this is to account for breeding contributions that have already been made, but are not represented in the main data file. For example, consider a male that has recently been used for many matings, but with the resulting progeny not yet born or otherwise represented in the main data file. Such a male (and indeed his close relatives) should be subjected to some negative pressure on allocation of matings in the current Matesel run. This can be done by including his previous “committed” matings in the current Matesel run.

For the sake of simplicity, the following description takes it that that Committed Matings are matings that have already been made in a previous session. However, the approach for Committed Matings to be made at the same time as uncommitted matings is the same.

Before the “Committed Matings” feature, this problem has been tackled by adding “embryos”, representing these prior matings, to the list of candidates, and mating these in separate groups within the same analysis as the active matings to be made in the current mating session. Juveniles that are not sexually mature can be included in the same way, and for the same reason.

Under Committed Matings, matings that have already been made (or otherwise decided on) are presented in a separate file, as noted below, and the analysis will include these matings in its solution. This means that the total number of matings to be derived for any one analysis/cycle/session includes those of previous sessions. Candidate status ([Maxuse](#)) must be set accordingly, either by the operator, or automatically by Matesel, as described below. So Maxuse values must be sufficiently high to cover at least the committed matings, but only the New Matings will be acted on by the breeder for the current session.

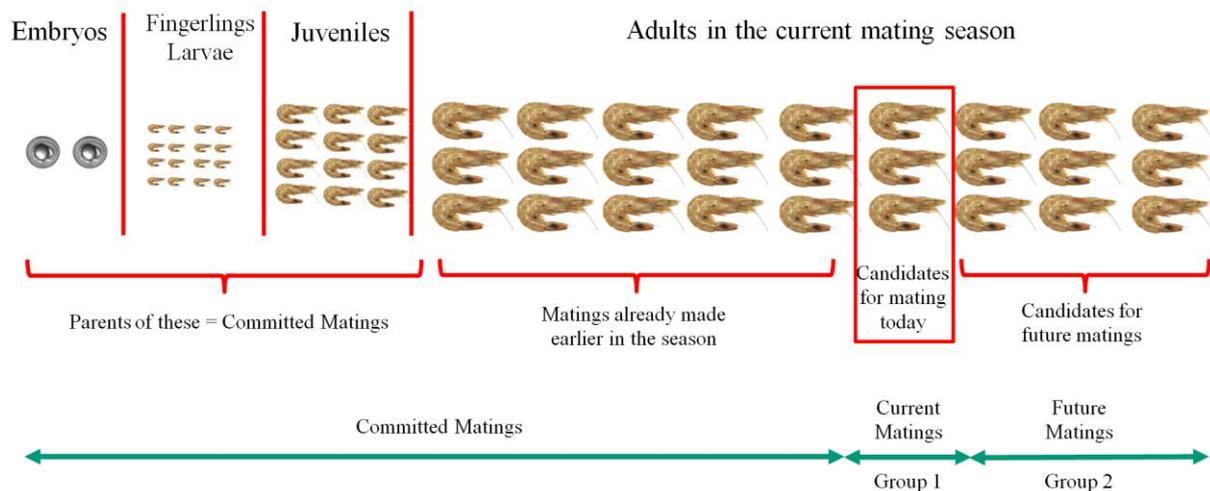
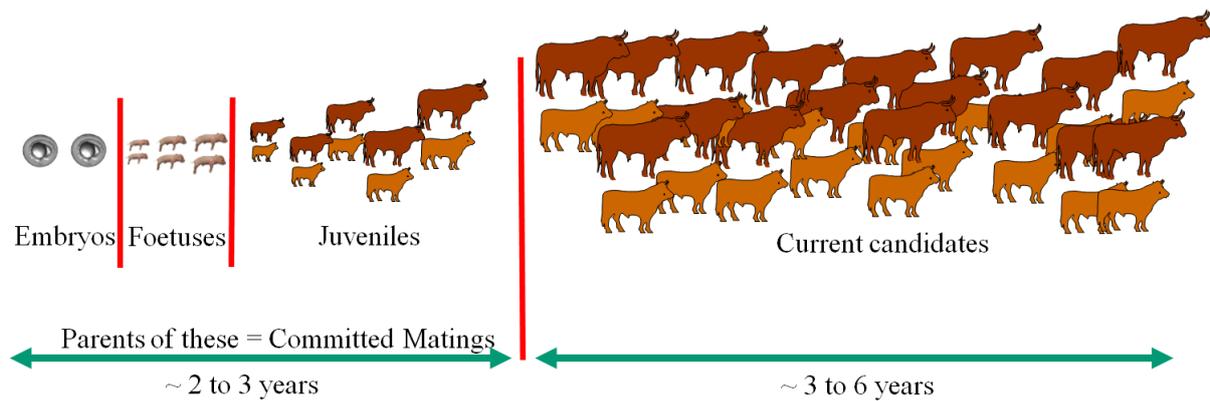
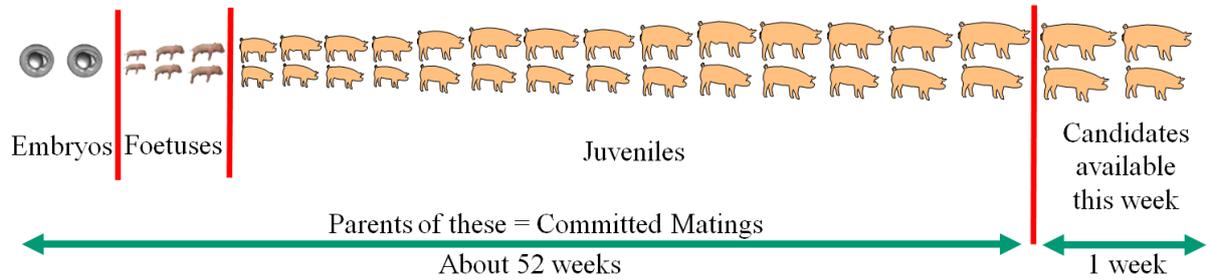
Accordingly, the candidate status of individuals that have already been mated should continue to be appropriately represented (either in the main datafile or automatically managed by Matesel, see below) until they have no more progeny that are not yet represented as candidates in the main data file, but have the prospect of becoming candidates. For species that have juvenile individuals available at mating time (most species), there are two ways of handling this:

Include juveniles as candidates: Juvenile individuals can be included as candidates in the main data file, with juvenile male x juvenile female Grouping set up such that juvenile mating results can be used for selection purposes, but without actual mating, as they are not sexually mature. In this case, Committed Matings for adult selection candidates need only cover matings that have not yet resulted in birth of progeny entered in the main data file as juvenile candidates.

Do not include juveniles as candidates: If juvenile individuals are not included as candidates in the main data file, Committed Matings for adult selection candidates should cover all matings that have not yet resulted in adult candidates, or culling from the breeding program. This does not give the benefit of selection among juveniles in a manner that accommodates adult selections and matings made simultaneously.

The Committed Matings approach is superior to the old approach of adding “embryos” to the main datafile, because it makes a proper assessment of contributions made across sessions, and does not rely on choice of numbers of “embryos” (or numbers of contributions each embryo can make) in order to handle contributions from previous mating sessions.

The following illustrations give some concept of how Committed Matings can be used. See Kinghorn (2018) for more detail.



Operation

To invoke Committed Matings you need to set the Switch to control Committed Matings to value 1 or 2 (the difference between these is described below) in Matesel.ini. You also need to include a file called CommittedMatings.txt in the same folder as the main datafile. This file is space or comma delimited, with a single header line and content as in the following 8-matings example (Note: Just two columns!!!):

Sire	Dam
George	Daisy
Jim	Jessica
Harry	Gloria
Harry	Michelle
Harry	Fatima
Tom	Gertrude
Tom	Susan
Hein	Karen

If these committed matings lead to a breaking of any constraints, such as [Maxuse](#) constraints or [group mating permissions](#), then this will be reported on the Matesel console, and the run may have to abort.

Automatic setting of Maxuse - set Committed Matings switch to value 1

In this case, you keep [Maxuse](#) values in the main datafile (or in InpOneGroup.txt or InpGroups.txt) as appropriate for the new uncommitted matings to be made in the current mating session/analysis alone. Matesel will automatically add to these Maxuse values according to the contents of CommittedMatings.txt. Matesel will also turn non-candidates in the main datafile into candidates, according to the contents of CommittedMatings.txt.

Implementation where mating sessions are ongoing, as in pigs and poultry, will relate mostly to management of CommittedMatings.txt. Here the Committed Matings switch will be set to 1, and Maxuse values in the main datafile (or in InpOneGroup.txt or InpGroups.txt) will relate to the current Mating session alone. The Maxuse values used by Matesel will be automatically set, and:

- Reduced according to removal of matings from CommittedMatings.txt as the resulting progeny are either culled or included as candidates (juvenile or adult) in the main datafile. If just some of the progeny from a single mating are perhaps not yet trait-recorded, and so not yet either deceased, culled or candidates, a judgement needs to be made as to whether the mating should remain in CommittedMatings.txt, probably according to the number of progeny in each category (culled, candidate, not yet considered).
- Increased according to the addition to CommittedMatings.txt of matings made in the previous mating session.

Manual setting of Maxuse - set Committed Matings switch to value 2

You must set the candidate status of individuals in the main data file (or in InpOneGroup.txt or InpGroups.txt) such that [Maxuse](#) is sufficient to cover the Committed Matings previously made *plus* the *maximum* number of new uncommitted matings that can be made in the current mating session. If a parent in the Committed Matings list is no longer available, its Maxuse value should be set to its number of matings in that list.

Implementation will probably be quite simple where there is a number of mating sessions within one mating season, as in some fish species and some sheep and cattle systems that use reproductive boosting. Here the main datafile can remain unchanged over the whole season, with Maxuse set to cover maximum use of each individual over that whole season, and the Committed Matings switch set to 2. The content of CommittedMatings.txt will be expanded by the breeder after each mating session. And of course Maxuse can be temporarily reduced in the main datafile for individuals that are not yet sexually mature or become unavailable for any reason.

Some rules

In all cases, [Minuse](#) and [AbsMinuse](#) relate to new uncommitted matings alone, whether these are specified in the Main data file or in InpOneGroup.txt or InpGroups.txt. This is done because logistical factors that impact on Minuse are most likely to relate to the current mating decisions.

As an example: If a male already has 5 committed matings, and has a Minuse value of 20 and a Maxuse value of 40, then his total number of matings will be in the domain (5, 25 to 45) under Committed Matings switch 5 set to 1, or in the domain (5, 25 to 40) under Committed Matings switch set to 2.

In all cases, exclude the Committed Matings from the numbers of matings entered in Matesel.ini, InpOneGroup.txt and InpGroups.txt. With the exception of the Committed Matings switch in Matesel.ini, these files should be set up as if for the new or uncommitted matings alone. However, there is one exception: Under Preselection, selection is first made across all candidates, including those used in committed matings, using the selection numbers or proportions specified by the user. After this, culled candidates that are used in committed matings are reinstated as candidates. For Natural Mating, the constraint applies only to New Matings.

If you edit the number of matings in the Windows or Web interface, enter the number of uncommitted matings. However, once the run is underway, Matesel will change this value to the total number of matings, including committed matings – so you must reset the number of New Matings before re-running. The console will report how Matesel has calculated total matings.

EndUse mating numbers or proportions declared in EndUses.txt cover only the new uncommitted matings to be made. During analysis, Committed Matings are allocated to EndUse 1 (however, Committed Matings are reported in output as EndUse = -1, for simple identification). This is the only EndUse for which parental coancestry plays a role in the analysis. It is suggested that any committed matings that belong to a different EndUse should not be included in CommittedMatings.txt, as their contributions should not impact on the future breeding program and its genetic diversity.

[ImportSolution.txt](#) must include the Committed Matings that are included in its hosting run.

Issues to do with the Location of individual candidates (see [Manage cost of assembling mating groups](#)) relate to new uncommitted matings alone.

Trait and marker distributions under Committed Matings (from 6.9.11.5)

When you make a run using Committed Matings you will see the following options at the bottom of the appropriate output tab windows:

Results for New Matings only Results for Committed Matings only Results for All Matings

Choose your desired option - it will apply to the graphs and histograms under Progeny Inbreeding, Trait Management and Marker Management outputs. (See [Why are there no Committed Matings option buttons for Individual Coancestry histograms?](#)) These options only affect what information you see on the screen ...

Please Note: When you apply trait and marker distribution manipulations through Matesel.ini and/or by clicking on the trait and marker histograms, these apply to New Matings only, whatever results are being shown. So, in order to see clearly the impacts of your manipulations, use the default “Results for New Matings only”. However, viewing “Results for All Matings” might help you to, for example, choose values about which to reduce variation, or choose bimodal means and volumes.

Other outputs

All “Out*.*” result files give results for Committed plus uncommitted matings, except for OutLocation.txt, which relates to new uncommitted matings alone.

Committed matings in output are identified by allocation of an EndUse value of -1. Sorting output files on EndUse can thus be used to group your New Matings for action lists.

The Frontier graph gives results for Committed plus uncommitted matings.

Manage Loss-of-Function (LOF) loci

This is described here: [Handling multiple lethal recessives](#)

Manage cost of seedstock purchase

There is no specific feature for this, and yet some key aspects of such costs can be managed using [Trait Management](#) tools. See [Manage cost of semen and other seedstock costs](#).

Manage cost of assembling mating groups

The motivation here is to make it easier to assemble the group of mates to be allocated to each male. For this we need information on the location of each female candidate.

To invoke this feature, include a field called “Location” in the main data file and use it to enter a meaningful location number (integer) for each candidate. This feature currently uses only location information on females, and a linear representation on location is currently assumed, such as a long row of enclosures holding one female each.

Hard implementation

In this case the location issue is handled as a hard constraint, meaning that the solution will adhere perfectly to what is asked of it, without the need to use any weighting factors.

The constraint imposed is to make the location of the females allocated to any one male be uninterrupted by any female allocated to a different male. For example, here is such a solution, with position in the sequence reflecting female location, numbers indicating male allocated to, and – meaning not selected:

```
6--6-66-6--5-5-558-88--8-83--3---2-2-2-2
```

Invoke this hard implementation by including a field called “Location” in the main data file and setting “Switch to control cost of assembling mating groups” to value 1 in Matesel.ini, as follows:

```
1      , Switch to control cost of assembling mating groups
```

Soft implementation

[Note: Soft implementation is not recommended, because of slow convergence. However, it does seem to operate correctly, and you may wish to test it for your situation.]

Here we aim to *reduce* either Distance or Overlap, as described below.

The weighting factor for how much emphasis to place on soft implementation of the Location issue is entered under “Weighting for soft implementation of Location issue” in Matesel.ini, for example:

```
-1      , Weighting for soft implementation of Location issue
```

You choose just one of two ways to manage location information:

Distance: Invoke by setting “Switch to control cost of assembling mating groups” to 2 in MateSel.ini. MateSel calculates the average absolute difference in location numbers within the female mating group of each sire and averages this over sires. The aim is to reduce this number in balance with other issues.

For scaling purposes, this figure is divided by the average absolute difference in location numbers between female candidates within female groups, pooled over groups. This means that the criterion for this component ranges from:

0: Where each female mating group is sourced from just one location

through

1: For a random allocation, or approximately so depending on grouping constraints and other such factors.

to

?: Some higher figure that you can discover by using a strong positive weighting

There is some danger that high emphasis on Distance will encourage too many sires to be used, which is conducive to lowering average absolute difference in location numbers among females to be mated. Keep an eye on number of sires on the console output, and coancestry etc. Ask BK if this becomes an issue and you want to try to scale this effect out.

Overlap:

Invoke by setting “Switch to control cost of assembling mating groups” to 3 in MateSel.ini. This is soft implementation of the criterion used under “Hard implementation” above. Matesel ranks female candidates on location number, and then moves along this list noting which male mate has been allocated (unselected females are ignored). Each time the male is different from the previous one, a counter is increased by one. The minimum value for this counter will be achieved when there is no overlap in males used, eg: 66666555588888332222. Lower emphasis on this issue will mean that there will be overlaps, and separation of females allocated to each male, eg: 66555858888666332222.

The counter value of a solution is linearly scaled, with the minimum value ($nSires-1$) scaled to zero and the random expectation for the number of selected sires scaled to 1. This random expectation is important, as we do not want this issue to drive the number of selected sires to just one in order to get no overlap!! [But ask BK if you want to scale independently of nSires, so that you will reduce number of sires if you put strong emphasis on this issue.] Using m_i as the number of matings for sire i and M as the total number of matings:

There are $M - 1$ opportunities for a change in sire, and these will be assumed independent from each other. For each opportunity we multiply the probability of the previous sire being i by the probability that the following sire is not i , and sum that over sires:

$$(M - 1) \cdot \sum_i \frac{m_i}{M} \frac{(M-1-(m_i-1))}{(M-1)} = \frac{\sum_i m_i (M-m_i)}{M}$$

And so this value of the counter is scaled to one. Note however that with exactly one sire used, both this value and $nSires-1$ will equal zero, whence the criterion is set to zero [Ask BK if you want this set to a higher value, eg 0.5]. However the treatment above inhibits the *progression* to just one sire.

The scaling for Distance and Overlap attempts to conform to MateSel's goal of having the criterion for each key issue range 0 to 1, so that weighting factors applied at "Weighting for soft implementation of Location issue" are within some typical range, and are more robust for application across different runs and scenarios.

Output

The mean value of the Location criterion will be displayed under the heading "Dist" or "Over" in the Console tab, once the Frontier has been drawn and the main run is underway. (These replace "Deg", the effective degrees realised, in runs not invoking Location).

Location will also be displayed as a trait under the Trait Management tab. This is the distribution of "progeny mean" location – the mean of the parental location numbers. This will generally not convey information on *differences* in locations, so this is probably not useful in the current context. But it may provide you with other information and ability to manipulate for other reasons.

The file OutLocation.txt will be written in the datafile folder each time you click Print Matings, Update in the Mate Selections tab, or Close. This file lists the location numbers of selected females (as integers), the male candidate number (a small number) and its sequential ID (a bigger number). This should help you to judge outcome for the Location issue.

Possible Strategy

Just as we use a negative weighting to reduce progeny inbreeding we must use a negative weighting to reduce Distance or Overlap. Try a value of 0 to start with, to see what no emphasis does to Dist or Over on the console, then try a massive weighting of say -1000 to see what can be done. Then try -1 as a starting point to seek a useful outcome. It is also a good idea to try [Hard implementation](#), which should give full adherence to the Location issue.

Slow convergence

Convergence will be slower when using the Distance or Overlap features. This is especially so for runs that use Overlap, because it will typically involve big valleys in the fitness landscape. Eg. to move the 6's together in 6655585888866632222 means a fair bit of movement required with no reward (some penalties) until you actually join the two groups. Test for your scenario by running and waiting for a long period to see if you get improvements late in the process. You may need to disturb the goalposts (eg change [Target Degrees](#)) to kick the solution into new areas, to help it get over these valleys. This factor alone might make you choose the Distance option instead.

Rules and Notes

For this feature you must have a maximum of one male mate per female.

If you have big differences in Location number that do not reflect distance apart then you might want to change these if using Distance. For example, if most locations are around 500, but you have just a few around 5000, this will affect scaling, especially when none of the females around 5000 get used – Distance values will be tiny.

Invocation

Include your field named “Location” in the main data file. Set “Switch to control cost of assembling mating groups” in Matesel.ini as follows:

- 0: Switch off and treat field “Location” just as a trait
- 1: Hard implementation
- 2: Soft Distance implementation
- 3: Soft Overlap implementation

Reminder: Matesel.ini is read when the main datafile is read. So if you Stop a run, edit Matesel.ini, then hit Run, your edits will NOT have been taken. Simply hit Stop, edit Matesel.ini, then Open the same datafile, then hit Run.

Rank the number of matings allocated per full sib

[Note: Substantial improvements to the performance of the core optimisation algorithm in Matesel as of Fortran DLL version 12.3 probably make this feature redundant. We suggest that you do not use this feature except perhaps for some detailed testing, and possible use if you find that it does in fact make a significant improvement for your scenarios.]

This is only a potential issue for very large jobs that have groups of full sibs of the same sex (males and/or females). If convergence is slow then we might get suboptimal solutions involving for example more matings allocated to one individual than to its full sib that is identical in gender, group, lack of progeny and genotype. We can rectify this directly:

Take care that such observations are not necessarily suboptimal if weightings have been put on trait EBVs, or other criteria treated as traits. Full sibs are unlikely to have identical EBVs unless they have not been measured.

If this feature is invoked, the program detects all full sib groups that are identical for gender, group, lack of progeny and genotype (Noted as “Number of equivalent (Fe)male Full Sib groups” on the console.

This is then handled one of two levels:

Candidate level: This is invoked throughout the optimisation process. Numbers of allocations within each solution tested are reallocated highest to lowest according to Index of the

candidates, highest to lowest. [It is possible that this affects how well the solution space is searched, which is why this is not done by default.] This mode respects the usage constraints (eg [MaxUse](#)) applied to each candidate (the next one does not).

Selected individual level: This is invoked only at the solution printout stage. The highest Index full sib is allocated the matings of the most allocated full sib, and so on down to fewest matings for the least allocated full sib. Note that “Candidate re-ranking” also invokes this “Selected individual re-ranking” by default. WARNING: This mode can result in constraints on use being broken, eg if a male with few doses of semen available is allocated the matings of his brother with many doses available.

To invoke these features, work with “Switch to control ranking of full sibs” in Matesel.ini:

Switch value	Actions
0	No re-rankings
1	Re-rank candidate males only (also re-ranks selected males).
2	Re-rank candidate females only (also re-ranks selected males).
3	Re-rank candidate males and females (also re-ranks selected individuals).
4	Re-rank selected males only
5	Re-rank selected females only
6	Re-rank selected males and females

Tips and Tricks

Some of these Tips and Tricks constitute ~research, and some lead to strategies that enable more effective implementations. It can take a fair bit of work to make and test such enabling strategies. However, once you get them right you can embed them in your database automation system to make the data and parameter files, and spend less time at the keyboard and more time at the pub.

Handling large groups of individuals as single “dummy individuals”

There are many reasons for using this trick. Many of these relate to speed of operation.

Typically, all members of a group of animals (all the same sex, but possibly also all on the same farm or in the same age class) to be mated have their candidate status removed (set e.g. their [Maxuse](#) value to zero in the main datafile). They are then replaced by adding a single dummy individual to the main datafile, with the same sex as the removed candidates, and an appropriately high Maxuse value, at least as high as the actual number of matings to be carried out in that group. You might also use an appropriate AbsMinuse value. You might allocate the dummy individual with average values for trait EBVs and Indices in the group.

If you want to properly reflect the relationships between the group of animals and the rest of the candidates, you can construct a relationship matrix with the dummy individual having a relationship to each other candidate that is the average of the group's relationships to each other candidate. You can then import this new relationship matrix using [Setting up the GRM](#).

Such a dummy individual might represent, for example, all the commercial cows in a given region or all the female pigs in a multiplier tier. In such cases, Matesel can manage both nucleus breeding and dissemination to commercial or multiplier herds in a single analysis.

This approach has been used with knowledge of the history of pedigreed sires used unpedigreed commercial herds, together with a modified geneflow approach, to give the predicted average relationship between each current candidate male and all current unpedigreed females in the commercial herd. This helps manage progeny inbreeding at the commercial level. It could also be used together with segregation analysis to reduced expression of genetic defects at the commercial level.

Optimal Contributions Selection within one sex alone

The need for this has arisen a few times. One example is a breeder with a number of pregnant females, from an overly successful reproductive boosting program, wanting to identify a given number to be sold off.

There are two approaches to this. One is to make a single dummy member of the opposite sex (see [Handling large groups of individuals as single “dummy individuals”](#)). It could be acceptable to let this dummy individual be unrelated to the candidates for selection, especially if the latter are male. This is because males are usually the “weak link” in accumulation of parental coancestry. Otherwise, an appropriate relationship between the dummy animal and the real candidates can be predicted, based on existing candidates of the dummy's sex.

A second approach (thanks to Vera) is to make the candidates for selection [bisexual](#). Of course they have to be allowed at least one use as a male plus one use as a female, but the number of uses allocated by Matesel will at least reflect optimal contributions.

Allocate females to groups of males for mating – “multi-sire joining”.

Some breeding operations mate each group of females with a group of males, leaving no control on who mates with who in the resulting mating group. It is likely that the numbers of matings per male is rather skewed, with one of a few males mating most of the females. Realised allocations can be detected by pedigree allocation using genetic markers.

Matesel does not handle this situation explicitly, however you can use Matesel to help get a result. Note that the following is not a recommendation *per se*, and that there may be better ways of doing this.

1. Allocate females to individual males using Matesel, with [Minuse=Maxuse](#) for males, to give equal allocations per individual male. (Note – make total number of matings a multiple of Maxuse in order to get a solution!).
2. Group the selected males into “syndicates” for group matings. This may be done in a manner that maximises, for example, relationship among the males within each group, or similarity for genotype at a chosen locus. You now have a mating list that could be used.
3. (Optional) Add a dummy male for each syndicate as the only male candidates in the main data file (ie. set individual male Maxuse values to zero), with Maxuse = number of uses across the group in the original Matesel run. Use the GRM facility to add these dummy males into the relationship matrix, with entries being the average relationship between the males in the group and each female candidate. Run this new Matesel job to allocate females to the male groups.

The last step may make little difference – it may be worth a test to investigate that. Take care to accommodate the fact that realised coancestries will likely be higher than reported in the Matesel runs, due to non-equal mating success among males within each group.

Test the impact of keeping Maxuse values low to save genetic diversity

This has been a common practice, but it is usually misguided. Letting Optimal Contribution Selection have freedom to make decisions is the better way – as long as constraints on usage (notably [Maxuse](#)) do not provide biological or logistical problems. To make a test on overly constraining Maxuse in males (with figures being for example only):

Make a Matesel run with a low Maxuse=10. [When you look at the results, you might find that most selected males are used 10 times, especially if you have a low TargetDegree policy. This would essentially be a wide equal contributions solution where Maxuse=10 was chosen to manage diversity.]

Save this result for later importing (see [Importing your own solution for plotting](#)).

Now set Maxuse \geq number of matings, or even 9999 for all males. Make that run to get the frontier (where you aim on the frontier is not critical). After importing your Maxuse=5 solution you will most likely see that it can be well beaten by Maxuse =9999. More Progeny Index gain at same Parental Coancestry and also less Parental Coancestry at the same gain. Then move your solutions to approximately the same Parental Coancestry as the Maxuse=5 solution. Look at the distribution of sire use - it might be surprising!

Now set Maxuse = 25, or whatever is a reasonable biological maximum for your scenario. You should still see that your solution at Maxuse=5 can be beaten. Have a similar investigation as with Maxuse=9999. You might even find a similar result, especially if you choose to keep Parental Coancestry not very high.

This test should illustrate that you should let OCS have freedom to work, and it will then give you good benefits. Don't use artificial limits on usage as we used to do. Biological/logistical limits are the important ones.

Dictate the number of sires to be used

This could be important if for example you have 4 mating paddocks and you want one bull to be used per paddock. Matesel does not have a feature to directly set the number of males to be used, but the following should do it for you:

Eg for 100 cows across four paddocks, you can set Male Minuse to 21 and Male Maxuse to 33. This way Matesel will not give a solution with 5 or more bulls as $5 \times 21 > 100$, and it will not give 3 or fewer bulls as $3 \times 33 < 100$. All solutions will be for 4 bulls. Variation in number of cows allocated will exploit Optimal Contribution Theory to give you a generally better result than setting Minuse=Maxuse=25. The choice is yours.

If you do something like this, you should also try some runs with less constraint on the number of bulls used (and their pattern of use) so you can see the value of changing the number of mating paddocks (and accepting more variation in cows per bull).

Spread sire use more evenly over weeks in pigs and other species

This can also be relevant to other species such as poultry, or even beef and sheep under female reproductive boosting programs.

Without attention to this issue, the number of matings allocated to a male over a number of weeks can be quite erratic. For example, 10 matings in week 1, followed by 1 mating in week two, then 12 matings in Week 3 ... that is frustrating for operators.

To keep selection and use of males more consistent over their usable lifetime, then, once selected and used in a given ~week, a male can have its AbsMinUse value set to an appropriate value or values in future weeks in order to even out its use over time. This helps logistics, with a technical compromise that is probably rather be small.

You could use Minuse (where 0 uses is permitted) rather than AbsMinuse, and cull the male as of the week that it is allocated 0 matings. Or possibly a combination: Eg. for last week's allocation being ≥ 8 , use AbsMinuse = 6, else use Minuse = 6.

Consider attributes of candidates other than their traits

All fields that are not [reserved fields names](#), [ID fields](#), or [genetic marker fields](#) are treated as traits. This means that you might end up seeing Trait histograms for fields such as "Age" and "ProportionAngus". But this is OK – it can be useful for information, and also for intentional manipulation. You can use the [Trait Management](#) feature on such traits to aim at, for example:

- Mate younger ewes with older rams
- Aim for 30% progeny > 90% Angus, 30% progeny < 10% Angus and 40% progeny being in the range 40%-60% Angus. This might reflect the product descriptions of bulls you aim to sell.
- Allocate high calving ease bulls to small heifers (see [Constrain matings according to phenotype](#))
- Aim at mating flock 1 rams to flock 2 ewes, and vice-versa.

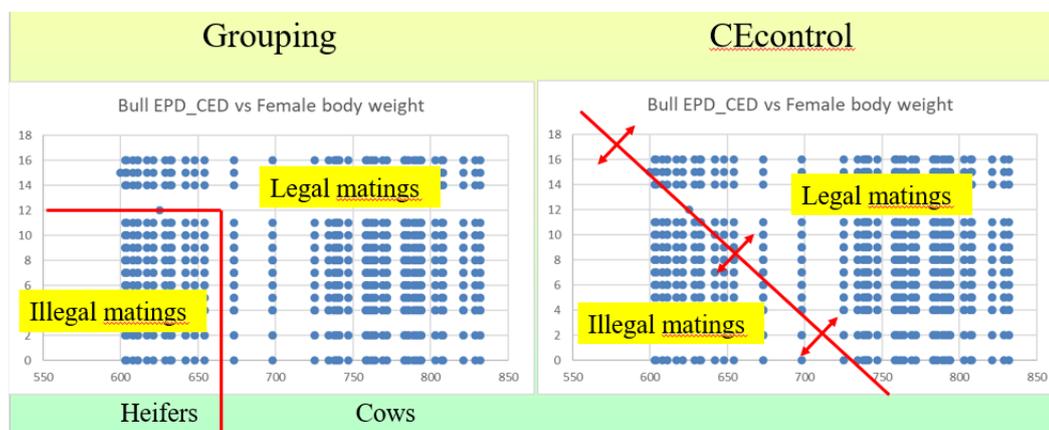
The last of these examples could be achieved by use of Grouping. However, if Grouping is complex or slow, some key aspects of it can be handled in this “soft constraint” manner. This is likely to give much more speed in tightly constrained scenarios.

To achieve more complex manipulations than the [Control Types](#) provided, you might have to clone columns of the information in the main data file so that you can impose more than one manipulation on a single trait or factor. For example, you could have columns ProportionAngus1 and ProportionAngus2 in the main data file, with identical information in each. Then you will have a “Trait” histogram for each of these under the Trait Management tab, and you can impose different operations on each of them (eg Set boundary at a minimum AND Set boundary at a maximum).

Constrain matings according to phenotype – eg. high calving ease bulls to small heifers

This section will be described in relation to the example – but you can probably think of other scenarios relevant to you. Maybe a cup of coffee helps here.

Aside: You could use Grouping to allocate only good calving ease (CE) bulls to heifers, as described in an example [here](#), and in the slide below. However the approach presented below is more powerful, as it customises allocations according to the bodyweight of each heifer, with smaller heifers getting better CE bulls (see the slide below). In addition, you can alter the overall emphasis on calving ease during a run, trading it off against all other issues. This approach is perhaps a bit more fussy the first time you use it, but it is probably worth the effort.



Make a new column in your data file, and give it a non-reserved name, such as CEcontrol (Calving ease control), or BWTcontrol (Birthweight control), depending on what EBVs are available for ease of calving. Enter values in the CEcontrol column as follows:

For each female candidate: Enter her current phenotype for body weight (or some such criterion).

For each male candidate: Enter -1 times the smallest weight of female that this bull should be allowed to mate, given its EBV for Calving Ease Direct (or some such EBV). This is described in more detail below.

With this setup, a mating between any bull and any cow gives a predicted progeny value for CEcontrol that needs to be at least 0 to satisfy the breeder’s desires in relation to calving ease for that female.

We can use [Trait Management](#) on CEcontrol to manage the distribution of this “trait”, for example setting a minimum boundary at 0 so that all matings satisfy the breeder’s desires.

Example calculation for male entries under CEcontrol:

Female body weight	Minimum Calving Ease Direct EBV chosen by breeder	Bull threshold criterion ($Intercept+Slope*EBV$)	Progeny value for CEcontrol reported in Matesel if using a threshold bull
275Kg Heifer	+10	$-400+12.5 \times 10 = -275$	0
300Kg Heifer	+8	$-400+12.5 \times 8 = -300$	0
400Kg Cow	0 (Extrapolated from above)	$-400+12.5 \times 0 = -400$	0

The breeder only has to choose the four figures 275Kg, 300Kg, +10 and +8. This represents the breeder’s attitude to bull requirements for calving ease EBVs depending on female body weight. The third row (400Kg Cow) is only included for illustration.

Asides: Notice that the fourth column (values = 0) is the average of the first and third columns, just as progeny predictions are the average of dam and sire EBVs.

The four driving figures (275Kg, 300Kg, +10 and +8 here) should cover the bodyweight region where calving ease is an issue. If there is little benefit from high calving ease bulls in heavy cows, this is not critical, as you can use Trait Management in a way that gives no reward for high calving ease in heavy cows, eg. by only avoiding matings below the 0 threshold.

So, what we enter for each bull in column CEcontrol is $Intercept+Slope*EBV$ where EBV is the bull’s Calving Ease Direct EBV. *Slope* and *Intercept* are calculated from the four figures chosen by the breeder, plus the target Progeny value threshold ($Thresh = 0$ here, but you can choose a different value for cosmetic reasons), using the following recipe:

$$Slope = \frac{300-275}{10-8} = 12.5 \text{ (This is actually the negative of the simple regression slope)}$$

$$\text{Intercept} = (2 \times \text{Thresh}) - 275 - (\text{Slope} \times 10) = 0 - 275 - 125 = -400$$

Or, same thing ...

$$\text{Intercept} = (2 \times \text{Thresh}) - 300 - (\text{Slope} \times 8) = 0 - 300 - 100 = -400$$

Once you have made the CEcontrol column, run Matesel and constrain progeny CEcontrol to a threshold of 0, using [Set minimum value at boundary](#), as in the following graphic:

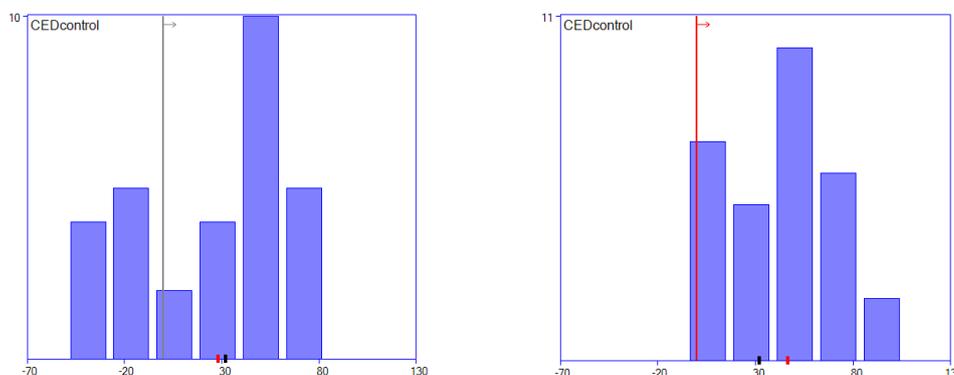


Figure: Setting CEDcontrol boundary at zero to ensure that all matings conform to the breeder's calving ease policy with respect to female body weights. Left: Boundary not invoked (9 matings do not conform). Right: Boundary invoked (all matings conform). This is a small example with just 30 matings.

Try changing the minimum boundary upwards from 0 to give more overall emphasis on Calving Ease. [Note that this is a dynamic policy with smaller heifers always attracting more attention, whatever boundary you set.] Is the extra calving ease worth the compromises you see in other issues?

Just to help with this exercise ... what is the impact of increasing the boundary from 0 to say 10? Well, to get 10 in the right hand column of the top row in the table, still using the EBV=+10 bull, we would use a 295Kg heifer (as the average of 295 and -275 is 10).

This means that going to a minimum of 10 now applies the same conditions to a 295Kg heifer that were previously applied to a 275Kg heifer – a fair bit stronger emphasis on calving ease.

If you prefer, you could scale this approach differently, so that a +1 progeny value represents an increase of +1 in EBV units, such that by invoking such a minimum value the breeder is setting the bar higher by that amount of EBV. Alternatively, if you have a good predicted relationship between EBV and % calving difficulties, you could let the breeder operate directly at the level of predicted % calving difficulties.

These considerations are important if you want to deliver this approach as a service provider to many breeders.

Manage sire and dam indices or EBVs during a run

To manage the Selection Index as if it were a trait (see [Trait Management](#)), simply duplicate the Index column in the main data file, and call it something other than Index, such as Index2. This addresses predicted average Index in progeny. However, you can use the following trick to manage the Index values and trait EBVs of the parents:

Make 2 extra fields in your datafile. Call these eg IndexS and IndexD.

Fill IndexS with TWICE the index value for all males, and 0 for all females

Fill IndexD with TWICE the index value for all females, and 0 for all males

These will now appear as “Traits” under the Trait Management tab, and as these show the average of parental values, they show sire and dam Index values respectively.

Now you can manipulate the Index of each parental sex separately using eg “Set boundary at a minimum” in [Trait Management](#), with enough weighting to have some impact on outcome.

There could be many reasons to do this. One is to investigate compromises that preselection on Index within each sex will make. Of course, you can do this for trait EBVs or any other field, such as Date of Birth.

Manage cost of semen and other seedstock costs

Add a column in your main data file that contains the cost associated with each mating made. You can then manage the outcome for this using [Trait Management](#) tools. You can also make two columns to manage male and female costs separately, using the trick shown above for [managing sire and dam indices](#). You can use [Tactical Desired Gains](#) to bring overall costs down to a target level. But you can also do other things, such as set a maximum cost per mating for each sex.

Where there is a cost at the candidate level, such as the cost of buying a natural mating bull, you might express his cost as the likely cost per mating – with the number of matings taken as, for example, the average of his [Minuse](#) and [Maxuse](#) numbers.

Test different preselection policies

In addition to testing preselection the last tip, you can also do more stringent preselection in the main data file itself (eg in Excel) in a separate run, and then [import](#) that solution into a run with no preselection, or less stringent preselection. You should find strong preselection gives approximately the same results at lower [Target Degrees](#). The best preselection policy gives essentially the same results at the Target Degrees adopted (and maybe also at a bit higher target degrees, to be safe).

Management of multiple Gene Edits

With current technology, when Gene Edits have been made in a population of reasonable size it seems most unlikely that all candidates will homozygous for the desired effects across all

loci. Of course, methods that can yield homozygous individuals *de novo* will be much more effective, but the need to do this across multiple loci, and to have a reasonable effective population size (not always required) leaves us with a considerable breeding challenge. Without such generation of homozygous individuals, the task is more challenging.

The simplest way to accommodate gene edits is by letting their effects be represented in the EBVs and Index used in Matesel. However, this leaves no direct control, which is likely needed to help manage specific targets related to the gene edits and their marketing in seedstock.

You can represent gene edits as marker loci and manage them through [Managing genetic markers](#). You can set weightings on Gene Edit ‘gene’ frequencies and/or genotypic frequencies through the Matesel interface, or as default weightings in [Matesel.ini](#). You can apply differential weightings across Gene Edit loci according to effects and breeding objectives. However, you may want to alter the overall emphasis on Gene Edits, in competition with other issues. This would be tedious to apply manually, and you would probably want to populate Matesel.ini with weightings generated ‘automatically’, for example from a spreadsheet that itself alters your weightings ‘in concert’.

Multiple loci can be collectively managed within Matesel using [Handling multiple lethal recessives](#). This would only be useful for managing Gene Edits under restricted circumstances: with objectives more related to targeting homozygosity (true breeding), rather than trait merit. However, such a collective approach could possibly be developed for more usefully managing Gene Edits, in a future version of Matesel.

Invoke Genomic Mate Selection

“Genomic Mating uses information in a similar fashion to genomic selection but includes information on complementation of parents to be mated” (Akdemir and Sanchez, 2016. <https://doi.org/10.3389/fgene.2016.00210>). These complementation effects can include predicted effects on progeny genetic merit (typically due to heterozygosity/inbreeding/epistasis) as well as effects on genetic variance in progeny. They could also include phenotypic effects on mating/pregnancy/birth performance (eg. A small male might not reach a big female, or a big bull on a small cow could lead to dystocia), and possibly imprinting effects.

The trick here is to capture the Genomic Mating merit predicted for each possible mating. This can be done through Matesel’s framework for using the [Genomic Relationship Matrix](#).

However, in the current case you need to populate CandGRM.txt with the predicted merit of prospective progeny, based on information from genetic markers, phenotypes and possibly pedigree, following Akdemir and Sanchez (2016), or something similar. For the method of Akdemir and Sanchez (2016), this merit is the parental mean EBV plus a “*progeny deviation*” that represents the value of segregation variance predicted in the progeny – recognising the future selection opportunity that this brings.

You may also (or alternatively) include components in this “GRM” to reflect the predicted impacts of heterozygosity/heterosis and progeny inbreeding (calculated from pedigree, markers, or both) and any other factors that affect the mean performance of progeny (or their descendants).

In doing this, you need to consider what you use for [the selection index field “Index”](#) – it seems sensible to include genomic EBVs in Index and progeny deviations alone in the “GRM” matrix. Here is the suggestion:

1. Follow the section [Setting up the GRM](#) to populate the “GRM” with the Genomic Mating values you have calculated outside Matesel. These values can include consideration of all the effects discussed above.
2. In [Matesel.ini](#), set ReadGRM to 2 (Pedigree coancestry and Genomic F for progeny inbreeding). This means we use the NRM to manage [coancestry](#), and the weighting on [Progeny F](#) is “borrowed” to manage emphasis on Genomic Mating.
3. In Matesel.ini or on the Matesel interface, set the [Weight on Progeny F](#) to an appropriate value. Under [Balance Strategy](#) 0, if you set this weighting to 2 (as $F=NRM/2$), then the progeny deviations will simply be added to the mean parental gEBVs (Assuming the later have been placed in the field “[Index](#)”).
4. Do some simple test runs on a small dataset to check that you have set this up to give the results that you want. Use OutMatings.txt and CandGRM.txt to evaluate Genomic Mating components for yourself.
5. Proceed with runs for your real dataset. You can track Genomic Mating values or progeny deviations in the [Progeny Inbreeding Graph](#), and numerically on the [console](#), while dynamically controlling them with Weight on Progeny F. However, all reporting of predicted progeny merit in graphs and tables will relate to mean parental EBV/Index values alone.

Note that this trick “piggybacks” on the GRM facility, such that it can no longer be used to manage genomic coancestries. In this case we have to use the NRM for that. If this issue becomes sufficiently important, we will make a special facility for Genomic Mating, so that both Genomic Mating and management of Genomic Coancestries can be undertaken simultaneously.

With sufficient weighting on these Genomic Mating values, Matesel will alter selections as well as mate allocations in order to exploit the effects involved. That is why this section is headed Genomic Mate Selection. The method of Akdemir and Sanchez (2016) is in fact Genomic Mate Selection.

Target specific genomic region(s) for management of diversity and inbreeding

Some concern has been voiced that selection involving major QTL will result in too much homozygosity or lack of diversity in the genomic regions surrounding them, and that steps should be taken to conserve that diversity. But there are potential benefits from doing the opposite, at least in the short term, by reducing diversity management in QTL regions to further promote their selection.

(see: <https://gsejournal.biomedcentral.com/articles/10.1186/s12711-019-0475-5>)

Control across specific genomic regions can be set up outside Matesel by preparing genomic relationship matrices (GRMs) that are calculated with attention paid to specific region(s). It is likely that you will have a consensus GRM that has placed varying levels of emphasis on different regions across the genome. You can test the impact of using such a GRM as follows:

- Make a run using the consensus GRM and parameters/settings appropriate to your desires.
- Rename the mating list to ImportSolution.txt
- Now you can make different runs using GRMs that target specific regions, and then Import the consensus results into these runs to observe how the latter performs at each region of interest (see [Importing your own solution for plotting](#)).
- You can also do this the other way around – observe how policies that target a single region (or a given set of regions) perform under the consensus GRM.

Reduce Progeny Inbreeding with little or no impact on Progeny Index and Parental Coancestry

This can be made to happen for scenarios in which Progeny F is the only issue affected by mate allocation, and where very good convergence can be achieved for Progeny Index and Parental Coancestry. See [here](#) for discussion.

Mate Selection ... or Selection followed by Mate Allocation?

It has been suggested that, at least for one scenario, we should first optimise selections, and then, as a separate step, optimise mate allocations based on these selections ([Fernandez et al., 2021](#)). This can be made to happen in Matesel by setting the weightings for issues affected by mate allocation to very small values. This follows the last tip ... See [here](#) for discussion. However, in this case the *relative* values among these tiny weightings for mate allocation issues is of course important in managing the balance among them. [Note that for the example cited, different components were included in the objective functions of the two selection-and-mating strategies tested. Appropriate use of full Mate Selection seems appropriate for both objective functions.]

Manage distribution of males in a sire referencing scheme

This applies to sire referencing schemes or other schemes involving distribution of males for matings to be made at different geographical locations.

There can be several motivating factors here. One is the desire to have a practical solution with respect to natural mating group sizes and numbers of doses of semen per male per farm.

Two Matesel features can be useful here:

Committed Matings

As illustrated in the figure showing shrimp under [Committed Matings](#), species with frequent mating sessions are characterised by some matings to be made at each session, from a relatively small pool of candidates. This may be relevant to your program in your species if, for example, males have to be used sequentially at different locations at different times, due to natural mating requirements.

In this case, set all matings already mate as Committed Matings, and optionally make groups for including future matings (virtual matings), and include the current mating period as the main matings to be made.

Targeting minimum use of individual males within each female group

This relates to distribution of semen for simultaneous mating across multiple locations (across female groups). Even with Minuse set for individual males, it could be that only one mating is listed for a given male in a given location – and this proved to be a problem in a number of such schemes. Use this feature to target a minimum number of matings for each male in each location. You can set this target to be different for each male x female group combination.

Get more Speed

Runs with many candidates and/or complex grouping structure can be slow. This may be acceptable for a final run to get a mating list that is to be used. However, low speed is less acceptable when you are in discovery mode – aiming to surf the overall response surface to see what can be achieved, to learn about your scenario and eventually to choose policy and parameters for final or routine use.

Some of the tips given here cut corners that may be acceptable for the discovery stage, while others are expected to have little or no impact on the solution and can be adopted for all runs.

Corner cutting speed-ups

1. Use relatively few generations for the [90 degree frontier point](#) unless you aim for very high Target Degrees. This only speeds Frontier building.
2. Use [NRM limit](#)
3. Set [NRM generations](#) to a value that is lower than the number of generations in your main data file. Use this one with caution.
4. Use [Dummy individuals](#) if appropriate to your scenario.
5. If Grouping, you might choose to switch off [Grouping](#) for some scoping runs.

Robust speed-ups

1. Prune the main datafile to remove all individuals that are neither a candidate nor the ancestor of a candidate. Pedigree Viewer can do this for you.
2. Use [Soft constraints](#).
3. Switch off [printing of backup matings](#), except after Stop.
4. If handling multiple lethal recessives, then [hide marker histograms](#).
5. Use few generations (say, 100) for [intermediate degree frontier points](#). You can also increase the [Frontier degree gap](#), to plot fewer points inbetween 0 and 90 degrees. These steps only makes for a less pretty frontier.
6. Use [Trait Management](#) to achieve some aspects of the Grouping strategy you want to implement. See the last example at [Consider attributes of candidates other than their traits](#). See also: https://www.youtube.com/watch?v=X38blqxp7g&list=PLy8khhn_GxCbACAD0VtZ_ZSIWkC1iGnUtv. This strategy will be effective if your Grouping scenario is sufficiently complex, and robust only if effectively implemented.

Common mistakes

Use Mate Selection, not just Mate Allocation!

Some users have already implemented selection before running Matesel by making the sum of Maxuse for each sex equal the number of matings to be made. It is strongly recommended to *not* do this – it gives zero opportunity for Optimal Contributions Selection to work for you. Most of the value in using Matesel is usually due to selection decisions. And of course, the best pattern of mate allocation depends on which individuals have been selected.

If you really want to do this, then copy your mating list into ImportSolution.txt, and [import](#) it into a run where you have let Matesel do the selections. Then you will see what you are missing.

Also note that Frontier drawing is not possible if there is no room for selection. The Frontier is unaffected by the pattern of mate allocation *per se.*, and with no selection, 0 degrees and 90 degrees give the same point on the Frontier, as both involve the same individuals as parents with the same contributions per individual.

“The Two Inbreedings”

There are two very different ways of controlling inbreeding in Matesel. The difference is critical!

Progeny F for Short-Term inbreeding: This aims to manage the inbreeding level in progeny of the current mating. It is affected by mate allocation for reduced relationship between the two individuals used in each mating. Progeny F is also affected by selection, as some otherwise unselected individuals will be lowly related to at least some selected members of the opposite sex.

Parental Coancestry for Long-Term inbreeding: This aims to manage the inbreeding level in future descendants by selecting more individuals and less-related individuals as parents. Mean Parental Coancestry is affected by selection but not by mate allocation.

Here are two illustrations that these two things are very different:

- Consider aiming for the ***lowest possible Progeny Inbreeding*** when there is very high potential fecundity, as in an IVF program. You might choose just one male plus one female as parents because they are totally unrelated, giving non-inbred progeny. Although you have done well for the short term, these parents’ ‘genes’ will meet each other in future generations, giving high inbreeding levels in the long term.
- Consider aiming for the ***highest possible Progeny Inbreeding*** in a species such as pigs. For each female, you (or MateSel) will look for a full-sib male to mate with to get high progeny inbreeding. This means that many more males will be used than under normal

circumstances. This makes for a high effective populations size, low Parental Mean Coancestry, and *low inbreeding in the long term*.

It is suggested that reducing Parental Coancestry is usually much more important than reducing Progeny Inbreeding, and that you should allow little or no influence of Progeny F on the outcome on your [Frontier Graph](#). Read [here](#) for why you might choose a small weighting on Progeny F.

Setting and weighting thresholds

[Thresholds can be set on any item with a histogram displayed (plus items with histograms hidden but listed in Matesel.ini.) using [Set minimum at boundary](#) and [Set minimum at boundary](#). In each case you supply a weighting (default=1) to give emphasis according to your desires.]

Common problems/mistakes arise from one or more of these:

- Setting too many thresholds across many items. However, you can get away with rather many if the next two are OK.
- Being too ambitious in what can be achieved. Eg. trying to shift a significant proportion of matings over a threshold.
- Using weightings that are too high – Maybe go up a bit on key issues, but generally keep it moderate.

These issues can cause major compromises on the Frontier. This can be especially the case if grouping and other constraints give less room to move, so that really bad individuals (low index or high coancestry) are used to satisfy thresholds. In our experience, **these compromises are almost always not worth the threshold compliance achieved**. Note that for traits, the distributions you are working with are of predicted genetic merit in progeny, and miss Mendelian sampling and environmental effects that will affect progeny merit.

Spend some time investigating strategies. In general, if you are not getting the result you want, relax your threshold value(s) rather than pumping up the weighting(s). Try moving the mean of the distribution, as an alternative. You can try out [Direct Threshold Compliance](#), which may give faster convergence.

Insufficient Convergence

Don't just stop after a few thousand generations. See the next section.

Diagnosis of convergence

How the user can manage convergence in Matesel is described elsewhere. This section gives some technical detail on how convergence is diagnosed in MateSel.

When will we ever get there? The trouble is that for most practical problems we would not know when we have arrived.

With some searching around the region of the current solution we might be able to detect that we are either at or very close to an optimum. But it could be a local optimum, with a valley to be crossed to get to the global optimum.

In practical situations, we have to stop evolving and accept the result at some stage. In most cases it is not critical to find the exact best solution – one that is pretty close to that will be sufficient – a “satisficing solution”. “Satisficing is a decision-making strategy which attempts to meet criteria for adequacy, rather than to identify an optimal solution” <http://en.wikipedia.org/wiki/Satisficing> .

If your job is important, it is probably a good idea to let it run for a long time – perhaps overnight, having checked the box: “Do not stop till “Stop” clicked”. That way you will find out if there was more improvement to be had!

Criteria for stopping

There are many ways of deciding when there has been sufficient convergence. This chapter suggests a small number of approaches for accepting convergence. Some or all of these can be applied simultaneously in practice. They are listed in the order that they appear in the demonstration program that will be used for illustration:

Criterion 1: `ConvergePercentTarget` in `Matesel.ini`. The solution must exceed a specified percentage of the current predicted asymptotic maximum solution. This is described below.

Criterion 2: `NoChangePercent` in `Matesel.ini`. No improvement over the last p_{nc} percent of n_{last} generations, where n_{last} is the last generation in which the best solution improved on the best solution in the previous generation, and p_{nc} is a percentage. It can be >100%.

It is sensible to make p_{nc} a function of n_{last} . For example, if we fixed p_{nc} at 20%, then we would say “Stop!” at generation 12 if the last improvement was at generation 10. This is not sensible, whereas we would say “Stop!” at generation 24,000 if the last improvement was at generation 20,000. This is more sensible. Here is a suggestion that ‘tunes’ p_{nc} to 20,000 generations:

$$p_{nc}(\text{corrected}) = p_{nc} * \text{sqrt}(20,000/n_{last})$$

If we use this, then we would say “Stop!” at generation 100 if the last improvement was at generation 10 – more sensible.

Criterion 3: `MinNewGens` in `Matesel.ini`. No improvement for a specified fixed number of generations. This is an extra safeguard against premature stopping.

Criterion 4: `AbsMinNewGens` in `Matesel.ini`. No fewer than n_{min} generations in total.

Criterion 5: `MaxNewGens` in `Matesel.ini`. No more than n_{max} generations in total.

Notes:

These criteria are not assessed at every generation – only at reported generations, in which fitness has improved, or at generations that are divisible by the parameter `n_see` in `Matesel.ini`. This means that choice of `n_see` can affect diagnosis of convergence.

The actual decision to stop can depend on meeting various combinations of these criteria, as described later.

The control of convergence noted here only applies to Batch versions, plus the Windows desktop version when the “**Control and track convergence**” checkbox is checked. In all other cases the program will stop at `MaxNewGens`, as long as “**Do not stop till Stop clicked**” is checked in the GUI versions.

A predicted asymptotic maximum solution

The concept is quite simple here. Fit a non-linear regression of best solution in each improved generation against generation number, and use this to predict the asymptotic maximum solution. An ‘improved generation’ here is one in which the best solution is better than that in the previous generation. However, in practice we should include the most recent generation, even if it does not give an improvement.

The non-linear fit is a bit of an art, because we never have a “correct” model for the form of the non-linear function to be used. You might find a better recipe – but here is one that has worked well for me:

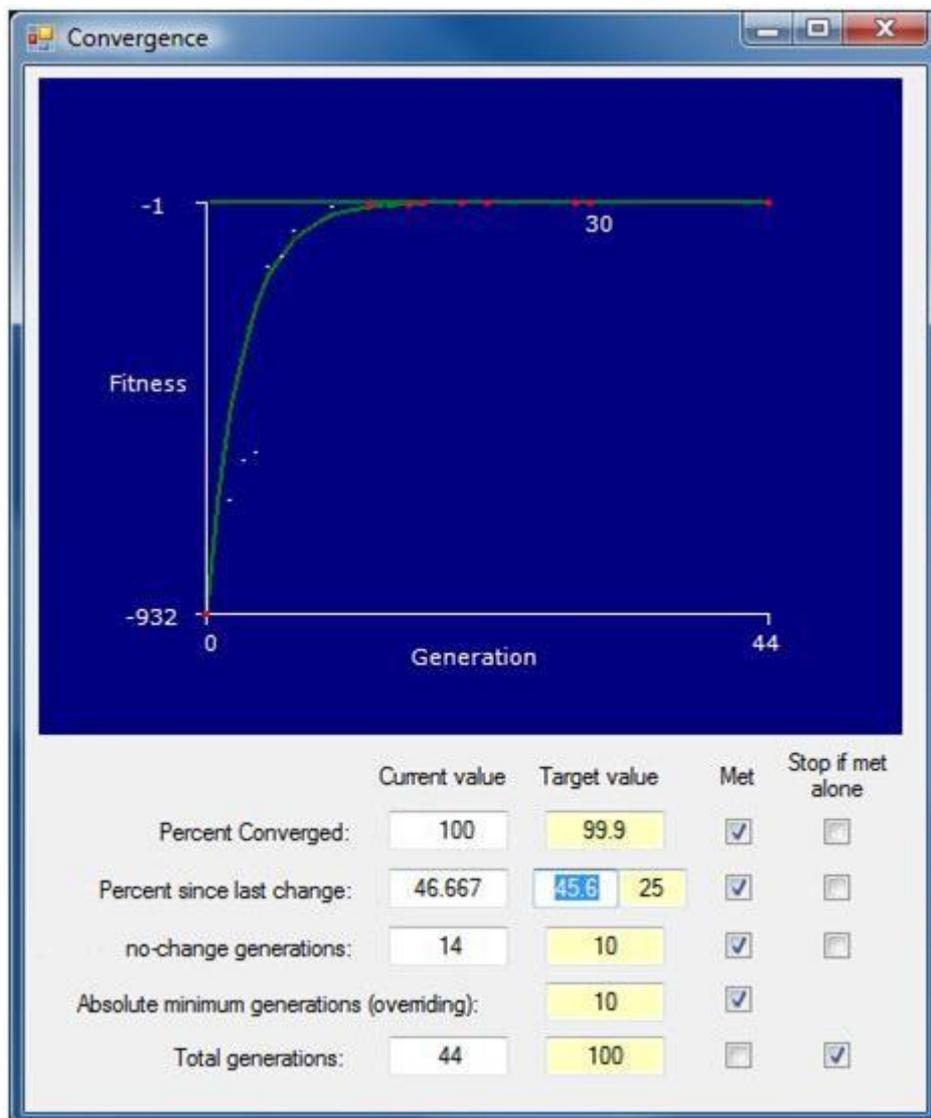
$$S_g = S_{max} * (1 - e^{(-k.g^b)}) + \text{error}$$

where S_g is the best solution at generation g , S_{max} is the asymptotic maximum solution, k is a rate constant, and b is a bender of the exponential function, which I treat as a fixed constant. I find that $b=1/2$ or 1 has been reasonable – choose a value that seems to give a good fit for your own scenario, easily seen if you have graphic output, as below. S_{max} and k are to be found to give the best fit to the data (g and S_g), probably by minimizing $\text{Sum}(S_g - S_{g\hat{}})^2$. The fit is facilitated by forcing the regression to pass through both the origin and the last point. This defines S_{max} for a given k , such that we only have one degree of freedom to adjust (k) to minimize $\text{Sum}(S_g - S_{g\hat{}})^2$.

Because the fit of the curve is most critical towards the asymptote, I only use (or ‘track’) the last $nTrackJumps = 8$ number of improvements (or ‘jumps’), plus the origin, in the exponential fit. [This is why you see a small column of “0.000”s under Conv% on the console at the start of the main run and after each change of direction. There is not yet enough information to fit the curve.] The most recent generation tested can be an exception here – it may not constitute a jump, and as generation number increases with no further jump in fitness, the predicted asymptote reduces to approach this stable fitness value.

Example

The next diagram shows an example of the convergence module that is attached to the DE_Demo program. In this example run, convergence is deemed to have been achieved in generation 44, and generation 30 is the last generation in which an improvement in fitness occurred.



Note that there are $nTrackJumps = 8$ red points in addition to the red point at the origin. These 9 points have been used to make the exponential fit (the green curve) and hence the asymptotic prediction of maximum fitness (the green line at Fitness = -1).

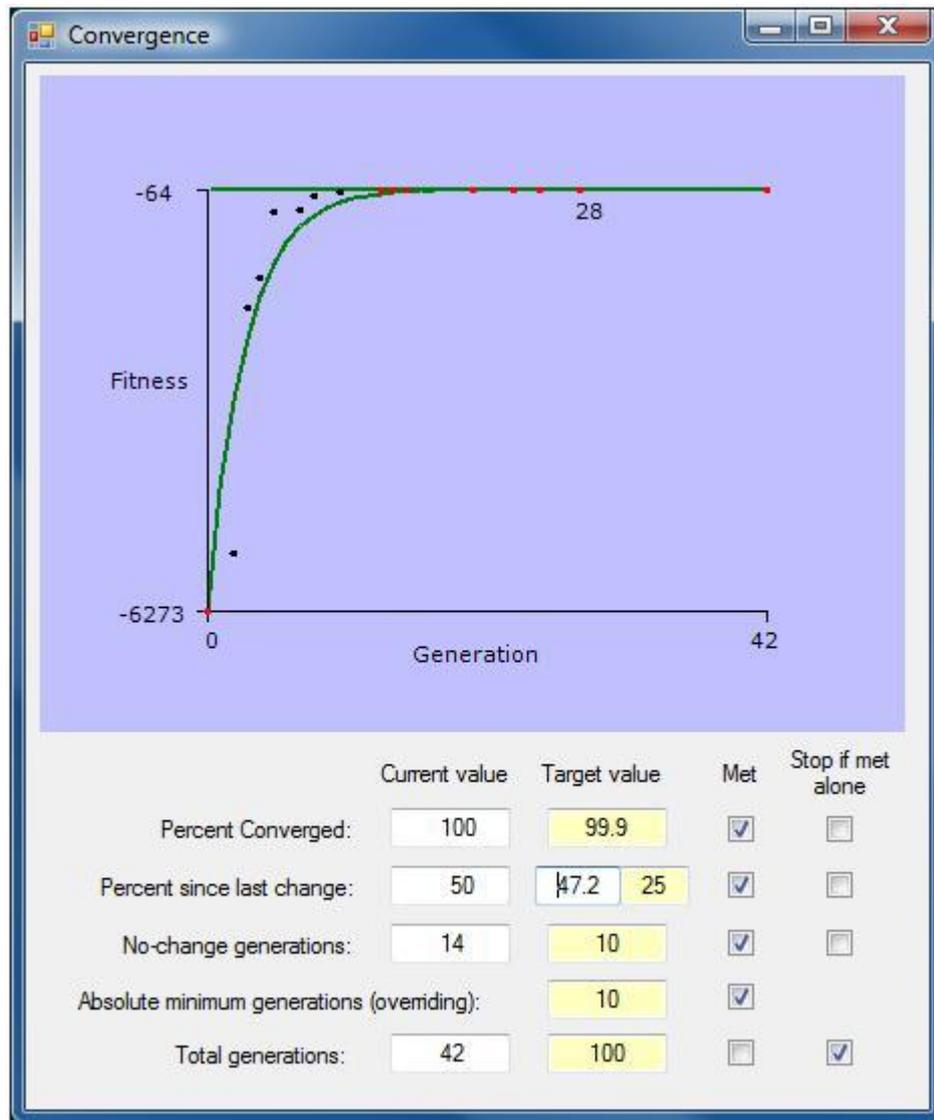
Note also that four component criteria have been met – four ticks under “Met”. Looking at the last column of tick boxes, you can see that “Total generations” is the only individual criterion that on its own will stop the program (when $\text{Generation} = n_{max} = 100$ is reached). However, in this case n_{max} is not reached, because all other criteria have been met. In the case of “Percent since last change”, the target value has been scaled to:

$$p_{nc}(\text{corrected}) = p_{nc} * \text{sqrt}(n_{max}/n_{last}) = 25 * \text{sqrt}(100/30) = 45.6\%$$

“Percent converged” is rounded up to 100% from a value $\geq 99.9995\%$. Such figures can give false confidence, which is why a mix of criteria is important.

The first three criteria are volatile, in that they can be met, and then not met as evolution progresses, because of the finding of new better solutions. This is why scaling of the “Percent since last change” criterion is important, to help avoid premature stopping.

NB: When changing the objective function (“Changing the goal posts”) on the fly, the generation counter is reset (NewGen on the console) for the purposes of diagnosing convergence.



No guarantee!! There is no guarantee that convergence has been truly met, unless you know the maximum from an algebraic approach or an exhaustive search. What can help is to observe the best solution achieved, and realize (if true) that it has all the properties that you think it should have – that you are satisfied with it. That it is a “Satisficing Solution”.

If the program crashes

Most errors are trapped and reported. However, Matesel may crash without an error message shown on the interface. If this should happen, **look for warning messages on the console output** (or Console.txt if running in batch mode) and **see the file OutErrorMessage.txt**.

Read the Console for evidence of what your problem could be – just a bit of detective work. For example, you might have a number of matings required that is equal to the sum of Maxuse for females, and possibly also for males (see [here](#)). Most such issues result in a warning or an error message – but there are many ways to do something wrong.

Some error messages need just a bit of thought. For example:

This line in InpGroups.txt unexpectedly contains a non-integer:
4 (# of Female Groups)

... could be because you overstated the number of male groups on the first line of InpGroups.txt, and Matesel is expecting four male subgroups to follow after the number “4”, but instead it finds “(# of Female Groups)”.

Setting the parameter “Verbose” in Matesel.ini to a value of 10 or more causes some debug information to be printed to a file named Debug.txt. This information may not mean much to you, but we might ask you to send it to us. Please note that this can slow down processing considerably under some scenarios, so only use it when needed.

If you launch Matesel from a script or from a remote computer, then you may encounter problems if you do not first Change Folder to that in which the Matesel batch executable resides before launching it.

If all else fails, you can email mateselservices@gmail.com, attaching the main data file plus all associated files required to replicate your run. We can have a look for you.

If you cannot close Matesel Windows

While Matesel is running a job, the ControlBox (including the red “X” at the top-right of the Matesel window) and the Close button are disabled. This is done so that the Matesel job running in memory is not left running after you close the window, think you have finished, and go to bed.

When the run stops, either because it has converged, or some time after you have clicked the Stop button, then the ControlBox and Close buttons are enabled so that you can close the Matesel window.

However, in case you cannot stop the Matesel run, and the ControlBox and Close buttons are not available, you can make a definite stop to close the Matesel window as follows:

Overriding stop: Click-Right on the Stop button while holding down the Control (Ctrl) key.

Using a REDserver

A REDserver is a Windows server that has been set up for Research, Education and Demonstration use. At the time of writing there is one such server: <http://matesel.une.edu.au>

Such a server is not to be used for runs aimed at making breeding decisions in commercially-operated programs. However, it can be used for “RED” purposes on an example dataset from a commercial program.

To this end, projects run on a REDserver fall into two categories: “Non-registered” and “Registered”.

Non-registered projects

You can go ahead and use a REDserver in exactly the same way as described under “Running MateSel from your web browser”. However, unless you run a registered project, the analysis will terminate soon after the start of the main run, but it will provide the full set of graphics and output files.

These outputs may not be very usable, as virtually no progress will have been made towards a converged solution. However, with this “open” facility, you can make the changes to your project files required to set up an error-free project run, with the type of output that you require. You can then submit that project zip file to be registered.

Registered projects

Registered projects have been accepted by one of the administrators of the REDserver. They are placed on the server itself, and are usually available for other users to run. These projects do not terminate early and a full run can be made with termination determined by the user.

Registered projects are run exactly as if the server were not a REDserver – the user does not even have to know that there is anything different about this server. However:

- The project name used has to be exactly the same as the registered project.
- You have to upload the project after it has been registered, even if you had uploaded it before that time.
- The datafile uploaded as part of the project zip file is overwritten by the version that is stored on the server.
- All other files in the project zip file that the user uploads remain unchanged by the server.

This latter feature means that when you submit a project for registration, you can include strategies for Grouping, EndUses, Imported solutions etc that are perhaps useful for the education of others, but do not reveal strategies that you wish to keep secret. You can then use such secret strategies in the registered project without making them public. Conversely, users of your registered project are free to apply any strategies they are able to by manipulating the other files in the project, mostly parameter files, and by manipulating the Matesel interface itself.

Submitting a project for registration

Please choose a project name (the name of the .zip and .txt files) that is informative and likely to remain unique. For example you might include the species, something functional (such as “PTmatings”) and maybe even the number of lines of data in the datafile. Your administrator may be able to help.

You can choose to sanitise your datafile by showing only sequence numbers for the pedigree (eg use “Save sequential ID file” in Pedigree Viewer), changing trait and marker names, and rescaling indices and EBVs. Unless agreed otherwise, your project will be made available to



others via the “Download Sample Projects” link on the Web version of Matesel. Please try to make the project useful in some way, for example to be used in a classroom of students. You might even include an example exercise.

To submit your project, please email it to an administrator. For the UNE REDserver at <http://matesel.une.edu.au>, these are:

Brian Kinghorn (bkinghor@une.edu.au)

Sam Clark Sam.Clark@une.edu.au

Julius van der Werf jvanderw@une.edu.au

Robert Banks rbanks@une.edu.au

John Gibson jgibson5@une.edu.au

[A separate manual is available for REDserver administrators]

Setting up Matesel on a Windows server

This is now covered by a separate document “Instructions to Service Providers for hosting a MateSel web service”. Please note that hosting of the Web version of MateSel is only carried out by BSK and those customers that enter a special agreement for service provision.

Running demonstration over the Web

Matesel has some ability for a single run to be viewed by many users in different locations. If you are interested in this feature, please contact the makers.

The security of your data

The only occasion that your data leaves your computer is when you are operating the Web version on an external server, specifically when you click on “Upload new datafile” and choose data to upload from your local computer. Of course, if you are running the server within your organisation, your data remains within your organisation.

Related to this: Our document “MateSel Web hosting Instructions” has a section “Keeping pedigree and data on the server alone”, describing how you can maintain data security in-house when you host Matesel and serve it to many external clients. Of course, in this case, your data is also secure from the providers of Matesel software, as noted below.

All versions of Matesel software need an internet connection to operate, including server-side Matesel running on a server. This is for licence purposes only, with permission to operate being granted from our Matesel server for each run. Your IT department can probably confirm the very small and identical amount of traffic involved for each run made – none of your data, parameter settings etc, are transmitted.

An introduction to Mate Selection

Mate Selection is the simultaneous choice of individuals to select for breeding and the pattern of mate allocation. The output is a mating list.

The outcome of this process is aimed at a balanced achievement across a range of objectives, or desires, including direction, magnitude and pattern of genetic change for different traits, avoidance of genetic defects, avoidance of progeny inbreeding, managing longer-term genetic diversity, use of reproductive boosting (eg. superovulation), satisfying diverse customer needs, and a range of logistical constraints on what is practical to implement.

A key task for breeding programs is to balance the relative emphasis on genetic gain and maintenance of diversity. This has now been done quite widely in the context of an objective function that includes measures of genetic merit and coancestry among selected parents. Wray and Goddard (1994) and Brisbane and Gibson (1994) included negative weighting on individual relationships in selection indices in order to help control the rate of inbreeding, and hence genetic diversity. Meuwissen (1997) formalized the optimisation of individual contributions, differentiating a function that includes genetic merit and individual coancestries, accommodating individual contribution constraints in an iterative manner. Hinrichs et al. (2006) and Dagnachew and Meuwissen (2016) have improved speed in this method considerably, and Pong-Wong and Woolliams (2007, 2018) have illustrated a method that handles individual contribution constraints directly, and thus gives a guaranteed optimal solution.

The results from such methods generally give optimal contributions as proportions on a continuous scale. The practitioner uses these to help make selections and allocate numbers of matings to each breeding individual. This represents a powerful way to allocate more matings to higher-EBV individuals in a manner that properly manages genetic diversity.

Progressive breeding programs are motivated by many other issues in addition to genetic gain and diversity, including progeny inbreeding, management of genetic marker frequencies and genotypes, management of trait distributions, multi-stage selection, various types of costs, and various logistical constraints, including constraints on pattern of use of individuals across groups. For such cases, contributions can be specified in numbers of matings to be allocated, rather than proportional contributions, and indeed the actual mate allocation set (e.g. Kinghorn 2000), but the optimal solution must be discovered, rather than derived, due to the complexity of the resulting objective function. Such problems can be solved using Evolutionary Computation (EC), and adaptations of Differential Evolution (DE, Storn and Price 1997) have been used for this. The central idea behind EC is to create populations of candidate solutions of a problem and evolve these populations by selection based on an objective function which emulates natural selection (eg. Bäck 2003).

The key to making this work relates to how the problem is presented to the solving algorithm. Kinghorn and Shepherd (1999) show how a vector of numbers can be transformed to derive a pattern of individual selection and mate allocation – Mate Selection. This makes the optimisation problem simple – the DE algorithm has only to find the vector of numbers that maximizes the objective function.

For each mating set tested, the component outcomes evaluated constitute the overall Mate Selection Index (MSI). Ideally, each component should be evaluated on the same scale, typically the scale of the breeding objective in units of, for example, dollars profit per breeding female per year. The MSI is a function of the selections and mate allocations chosen across all mating individuals in the population - it is used to form a mating set for the specified number of matings in the breeding program. The MSI can include predicted genetic merit in progeny, long-term inbreeding as reflected by parental coancestry, progeny inbreeding, heterosis (if relevant) and various costs and logistical constraints associated with the breeding program (Banks 2000; Kinghorn 2000). The objective function can be changed (weightings changed, or variables added or deleted) to suit the desired outcomes of the breeder.

Comprehensive use of a Mate Selection Tool (MST) involves decision making at various stages of the life cycle, including selection/culling at various juvenile and adult stages, selections for main round matings and, in some cases, selection of sires for backup matings. The MST can be made accommodate all members of the population, across different groups, including putative embryos in utero, immature juveniles, virgin females and pregnant females, as well as candidates for active matings. An effective way to do this is to force the MST to include recent actual matings in its solution – this being preferable to including the juvenile fruits of these matings as candidates for ‘virtual matings’ that will not be acted on. These approaches help to accommodate overlapping generations, and to account for contributions already made as reflected by juveniles and embryos in the analysis – sires that have already been used extensively, and their close relatives, will be somewhat inhibited from further use.

To implement such a scheme, there is a requirement to constrain the solution such that individuals can only be allocated to members of the opposite sex (unless bisexuality is accommodated) that are in an appropriate group, with the two key groups being the male and female active mating groups (although these may be sub-grouped in turn, for example by location). One way to handle such complex grouping constraints is to penalize non-conforming solutions in the objective function. However, this has proved to result in slow computation speed, as most evolutionary pressure is used up maintaining legality of solutions. However, a method was developed to translate a vector of numbers to a legal mate selection solution that satisfies complex constraints, not previously handled by Kinghorn and Shepherd (1999), related to grouping and mate allocation among groups. This has resulted in considerably higher speed, giving much increased flexibility (Kinghorn, 2011).

Part of the need for high speed in such MSTs relates to the need to explore the solution space to find a result that gives a satisfactory balance among the wide range of issues that can be represented in the objective function. It is difficult to be economically rational across issues such as level and cost of migration, pattern of trait distributions, marker genotype frequencies and number of mating paddocks required. Experience shows that iterations of quickly finding an optimal solution for the prevailing objective function followed by adjustments to weighting factors and/or thresholds set in the objective function gives the practitioner a good feel for the trade-offs involved in putting differential emphasis on component objectives. In decision-making theory these solutions are sometimes referred to as satisficing – they are at least close to optimal given all of the information at hand. This leads to some understanding of what can be achieved, and acceptance of a final mate selection solution.

In practice, there is a trade-off between what rigorous scientific effort might determine to be the “best” overall direction, and the direction that is chosen by the practitioner. The latter has the advantage that it is in fact implemented, and not left hanging as a mere recommendation.

In addition, of course, scientific method gets well implemented ‘around’ the constraints and weightings applied by the practitioner - for example, the theory of optimal contributions (Meuwissen, 1997) is effectively invoked by default, but subject to hard and soft constraints related to issues other than genetic gain and genetic diversity.

Breeding decisions are made on a frequent basis in, for example, the pig and poultry industries, and so this discovery process needs to be captured in a manner permitting essentially automatic running of the MST with little or no user intervention. This can be achieved using algorithms to maintain certain key outcomes, such as keeping the relationship between parental coancestry and predicted progeny merit within a narrow trajectory, so that other issues cannot displace the final solution from that path. This is analogous to a desired gains selection index (e.g. Brascamp 1984), but where the ratio of gains between two traits (issues in the MST case) is softly constrained within certain bounds. Overall response surfaces can be very flat – giving good opportunity to move one issue in a favourable direction, with little compromise in other key issues.

MSTs that are available include Ani-Mate (Amer, www.abacusbio.com/projects/animate/), TGRM and X’Aim (X’Prime, www.xprime.com.au), GENCONT/OCSELECT (Meuwissen, 2002; Hinrichs et al., 2006), EVA (Berg et al., 2006) and MateSel. These are now being used in breeding programs for pigs, poultry, meat and wool sheep, beef and dairy cattle, fish, prawns, trees and plants.

Implementation level ranges from providing target numbers of matings for each candidate individual, with the aim of maximizing genetic gain at a defined rate of inbreeding, through to dictation of all stages of selection, culling, semen collection, mate allocation and backup matings, with comprehensive attention to the full prevailing range of technical and operational constraint issues. More complete implementations can take a considerable amount of setting-up. However, once established they provide for routine optimal implementation of progressive breeding programs. They also constitute an appropriately competitive framework for implementation of new technical opportunities, including a range of genomic opportunities, reproductive boosting, and engineering extra genetic variation in key traits well before the physical splitting of a breeding line.

Look ahead mating strategies (Hayes et al 2002, Shepherd and Woolliams, 2004) have been proposed, aiming to exploit non-additive components some generations ahead. This includes the setting up of “investment matings” (such as F1 crosses) in order to capitalize on “realization matings” (such as 3-breed cross using F1 females). This constitutes short-term management of genetic variation for longer term gains. A simulated example has been given by Li et al. (2006) where an MST operating on cohorts rather than individuals was implemented for simultaneous decision making over generations, resulting in the setting up of divergent lines to exploit dominance at known loci. This approach has the potential to give tactical targeting of a suitable route through genotypic changes across multiple interacting loci, should we ever have sufficient understanding of the genotypic effects involved.

References

- Banks, R., 2000. Genetic improvement in the animal breeding industries. Chapter 21 in “Animal Breeding – Use of New Technologies”, Kinghorn, B.P., Van der Werf, J.H.J. and Ryan, M. (Eds.). The Post Graduate Foundation in Veterinarian Science of the University of Sydney. ISBN 0 646 38713 8.
- Bäck, T., Ed, 2003. Handbook of Evolutionary Computation. Institute of Physics Publishing, Bristol.
- Berg, P., Nielsen, J. and Sørensen, M.K. 2006. EVA: Realized and predicted optimal genetic contributions. Proc. 8th World Congress on Genetics Applied to Livestock Production [CD-ROM communication 27-09].
- Brisbane, J. R., and Gibson, J. P., 1994. Balancing selection response and rate of inbreeding by including genetic relationships in selection decisions. Proc. 5th World Congr. Genet. Appl. Livest. Prod. (Guelph) 19:135-138.
- Dagnachew, B.S. and T H. E. Meuwissen 2016. A fast Newton–Raphson based iterative algorithm for large scale optimal contribution selection. Genetics Selection Evolution, 2016, 48:70 <https://doi.org/10.1186/s12711-016-0249-2>
- Hayes, B.J., Shepherd, R.K and Newman, S., 2002. Look ahead mate selection strategies for multi-breed populations. Animal Science 74:13-24.
- Hinrichs, D., Wetten, M. and Meuwissen, T.H.E., 2006. An algorithm to compute optimal genetic contributions in selection programs with large numbers of candidates. J. Anim Sci. 84:3212-3218.
- Kinghorn, B.P. 2000. The tactical approach to implementing breeding programs. Chapter 22 in “Animal Breeding – Use of New Technologies”, Kinghorn, B.P., Van der Werf, J.H.J. and Ryan, M. (eds.). The Post Graduate Foundation in Veterinarian Science of the University of Sydney. ISBN 0 646 38713 8. Pages 291-308.
- Kinghorn, B.P. and Shepherd, R.K., 1999. Mate selection for the tactical implementation of breeding programs. Assoc. Advmt. Anim. Breed. Genet. 13:130-133.
- Kinghorn, B.P., Meszaros S.A. and Vagg, R.D., 2002. Dynamic tactical decision systems for animal breeding. Proceedings of the 7th World Congress on Genetics Applied to Livestock Production. 33: 179-186.
- Kinghorn, B.P. 2011. An algorithm for efficient constrained mate selection. Genetics Selection Evolution. 43:4. <http://www.gsejournal.org/content/43/1/4>
- Kinghorn, B.P., 2018. Accommodating recent contributions under optimal contribution selection. Proceedings of the 11th World Congress on Genetics Applied to Livestock Production. Paper 217.

Kinghorn, B.P., Van Eenennaam, A.L., MacNeil, M.D. 2015. Targeting Multiple End-Uses in Beef Cattle Breeding Programs. Proceedings of the 2015 Applied Reproductive Strategies in Beef Cattle conference, August 17-18, 2015, Davis, CA. <http://appliedreprostrategies.com/2015/documents/proceedings/17Kinghorn-VanEeen-pg250-257.pdf>

Li, Y., van der Werf, J.H.J. and Kinghorn, B.P., 2006. Optimisation of a crossing system using mate selection. *Genetics, Selection, Evolution*. 38:147-165

Meuwissen, T.H.E., 1997. Maximizing the response of selection with a predefined rate of inbreeding. *J. Anim Sci*. 75: 934-940

Meuwissen, T.H.E., 2002. GENCONT: An operational tool for controlling inbreeding in selection and conservation schemes. *Proc. 7th World Congress on Genetics Applied to Livestock Production*. 28:769-770

Meuwissen, T.H.E., and Sonesson, A.K., 1998. Maximizing the response of selection with predefined rate of inbreeding: Overlapping generations. *J. Anim. Sci*. 76: 2575–2583.

Meuwissen Theo H. E., Sonesson Anna K., Gebregiwergis Gebreyohans, Woolliams John A. 2020. Management of Genetic Diversity in the Era of Genomics. *Frontiers in Genetics*. <https://www.frontiersin.org/article/10.3389/fgene.2020.00880>

Pong-Wong, R. and Woolliams J.A., 2007. Optimisation of contribution of candidate parents to maximize genetic gain and restricting inbreeding using semi-definite programming. *Genet. Sel. Evol*. 39:3-25

Pong-Wong, R. and Woolliams J.A., 2018. A general quadratic programming method for the optimisation of genetic contributions using interior point algorithm. 11th World Congress on Genetics Applied to Livestock Production, paper 714.

Shepherd, R.K. and Woolliams, J.A., 2004. Minimising inbreeding in small populations by rotational mating with frozen semen. *Genet. Res., Camb*. 84:87–93.

Suontama, M., B.P. Kinghorn, W.A. Cowling and H.S. Dungey, 2018. Tactical desired gains for control of red needle cast in radiata pine under optimal contributions selection. Proceedings of the 11th World Congress on Genetics Applied to Livestock Production. Paper 375.

Wray, N. R. and Goddard, M.E., 1994. Increasing long term response to selection. *Genet. Sel. Evol*. 26:431-459

Matesel limits

Number of individuals in the pedigree

This is only limited by addressable memory.

[However, large runs can be very slow indeed. A job with over 68,000 in the pedigree and 46,000 candidates and no grouping took between 2 and 5 minutes per 1,000 generations on a 2018 Dell Latitude, 16Gb RAM, with the higher speed being at later stages of convergence. This probably makes for some hours to “visual convergence”, and probably at least a day to accept a result. See also [Get more Speed.](#)]

If your Matesel job runs out of memory at the stage of building the NRM, you should see the last line on the console reporting the number of individuals in NRMmem. This will be the number of individuals in your main datafile, after adding parents that were not listed as individuals.

If this happens, all is not lost! Three options:

1. Prune the main datafile to remove all individuals that are neither a candidate nor the ancestor of a candidate. Pedigree Viewer can do this for you.
2. Use a computer with more memory.
3. Include a file named `InpNRMmem.txt` in the same folder as your main datafile, with a single number as its sole content. This will be the number of individuals whose relationships are stored memory. This number will be lower than the number shown on the console – sufficiently low that your program will run. So you may need some trial-and-error here. However, you should keep this number as high as permitted for your computer’s memory and the job in hand, as the algorithm that builds the NRM will have less information stored in memory and will have to recalculate relationships that it has already calculated. In extreme cases, the increase in computing time can be so high as to be impractical.

Number of candidates

This is only limited by addressable memory. However, large numbers of candidates can slow operations considerably.

Length of ID

The maximum length for an individual identification is 20 characters. However, this constraint only applied when Committed Matings is being used, or a GRM file is being read in.

Numbers of Traits, Genetic Markers, Groups and EndUses

These are only limited by addressable memory.

Versions of Matesel

As Matesel is modular, there are different version numbers for different components. If you receive a package of components, typically in a zip file, you should be sure to use all the components in that package, and not mix them with earlier versions.

The Windows desktop interface version number is in the Title bar.

The Web Browser version is at the bottom of the login window.

The MateselBatch launcher version is the same as the Fortran DLL (Dynamic Link Library) version, below.

The Fortran DLL version is shown at the top of the console (Console tab in Windows and Web versions, and top of Console.txt for the Batch version).

Video tutorials and discussions

Videos without an active link in the **Title** column are planned, and have not yet been produced.

[It is easy to update the manual for each new version, but not so the videos. We will try to point out which bits of these videos have become out-of-date, and point to newer videos that might help. The manual should always be up to date at each software release.]

Series	Title	Notes
Introduction	General Introduction	These give a basic introduction to some core functions and features.
	Overview	
Preparing your main data file	Preparing your main data file	Comprehensive, except no detail on handling genetic markers.
Preparing your parameter files	1. Matesel.ini	Please also see Introduction to version 7.0.12.0 from about 6min, which shows the new format for Matesel.ini, including automatic upgrading.
	2. InpOneGroup.txt	Setting constraints and other settings across all candidates. Includes basic discussion on what Groups are in MateSel.
	3. InpGroups.txt - Standard operations	Everything in this file up to the permission matrix, except for 'natural mating'. Example in beef cattle across US states.
	4. InpGroups.txt - Advanced operations	Features below the permission matrix in this file, plus 'natural mating'. Some discussion on method used for Grouping.
The MateSel user interface	1. Windows Desktop	
	2. Web browser	

	3. Batch versions	
The MateSel Frontier	1. Frontier fundamentals	Includes discussion of what parental coancestry is, and why relationships are important. Simple spreadsheet example. Desired outcomes discussion, including programs Desire and Diversity.
	2. Where should I aim?	First, be sure to make a correct Frontier – by example. Your corporate position. Measuring and predicting inbreeding rate/diversity loss and its effect on long-term gains.
	3. Working with Balance Strategies	First 4 mins is introduction – you could fast forward that. Walking through the different ways of balancing Progeny Index and Parental Coancestry, with discussion on what strategy might be most suitable.
Managing Trait Distributions	1. Standard operations	What are these trait distributions? Worked example of using the different control types for manipulating traits.
	2. Advanced operations	This includes a number of unusual tricks, eg showing distributions for sires and for dams, as well as for progeny. Also using dummy traits to manage ages, breed crossing, locations etc. Please also see Introduction to version 7.0.12.0 for the desired gains options when Committed Matings are being displayed, and also for some relevant updates to format of Matesel.ini.
	3. Should I manage my traits?	Changing distribution means (responses), and shapes (eg variances). Managing genetic variation for individual traits, and planning for the splitting of breeding lines. Discussion on compromises caused by mating for desired trait distribution shapes.
Genetic markers	1. Coding markers in the main data file	
	2. Managing marker/QTL/gene edit effects	
	3. Handling multiple recessive lethals	
	4. Setting up and using the Genomic Relationship Matrix	
Effective use of MateSel	1. Working with weightings to achieve the best outcomes	
	2. Working dynamically with output files	Examples in many videos, including 4. InpGroups.txt - Advanced operations

	3. Import your own solution	For an example see from about 3 minutes here: 2. Where should I aim?
	4. Diagnosis of convergence	
	5. Shortcut commands	
	6. Troubleshooting	
Other features	1. Multiple EndUses	Target customer needs as well as your own. Conceptual examples in beef and trees. Worked example of the Demo Project. Discussion on treating whole farms as single individuals. Multiple EndUses and Grouping.
	2. Committed matings	Manage diversity across the full lifecycle. Force MateSel to make specified matings – usually to accommodate matings made recently. Examples in pigs, beef, aquaculture. This can handle sporadic ripening of aquaculture broodstock (also flowering in plants!). Favourable simulation results. Worked example using the Demo Project, with some emphasis on management of individual coancestries.
	3. Accommodating bisexuality	How to represent usage constraints when an individual can be used as both a male and a female. Controlling the levels of self, duplicate and reciprocal matings. For plant breeding, and maybe some animals. But you can use this feature to trick Matesel into selecting in one sex alone.
	4. Managing backup sires	
Other Videos	Introduction to version 7.0.12.0	Including: Comprehensive control of Progeny Inbreeding and Individual Coancestry, Dynamic Threshold Control, View Committed Matings in histograms, more speed etc.

New features and backward compatibility

Date	Version Set*	Key Changes
16/7/2021	7.1 2.2 12.3	Substantial improvement in speed of convergence. Improved reporting of input errors. Bug fixes: “Record overflow”, and Matesel Windows freeze on resizing form before Run is clicked.
2/7/2021	7.1 2.2 12.2	Reading of Maxuse, Minuse and AbsMinuse simplified, with independent choices on read from datafile or set globally for each sex/Group combination. [Old format still supported, but read the new manual entry ... you will need to use newly supplied grouping files or make a very small edit to existing files in order to use this improved method!!] Duplicate matings can now be inhibited/avoided under IVF programs. Frontier drawing more robust for unusual starting solutions, and smoother transition of optimal solution above and below 45 target degrees. CoanCand and CoanSel now both shown under Sire Use. Better reporting of mismatched target matings in your InpGroups.txt. ImportSolution now handles Multiple EndUses in a single importation. Sire Usage list consolidated for Multiple Enduses. Improved speed of convergence in runs that require many generations. Matesel.Helpers no longer required, and not installed. We thank them for their help over the last two years.
19/02/2021	7.0 2.2 12.1	Committed Matings no longer printed to OutBackups. Reporting of use constraint breakage in OutSolutionError.txt now accommodates use of Committed Matings Tab delimiters now accepted in ImportSolution.txt (in addition to spaces and commas). Improved targeting of low coancestry under Hard constraint on coancestry Bug fix: Progeny index scaling for 90 TargetDegrees while weighting on coancestry Bug fix: Candidate count under Bisexuality plus Committed Matings 12.1a 25/2/21: Dynamic Threshold for inbreeding now set just below targeted maximum, to inhibit matings at that exact level.
12/11/2020	7.0 2.2 12.0 (Note: Web version to be updated soon)	Controllable Histograms now given for Progeny Inbreeding and Individual Coancestries with the selected population for Sires, Dams and Progeny. “Dynamic Threshold Control” for Progeny Inbreeding. Committed Matings can now be separately identified in Histograms. Mouse-over values shown, plus point-and-click setting of Frontier and Histogram parameters. Histogram and Frontier plotting improved. Histogram numbers shown for 1-pixel bar heights. (For v. large jobs). Windows form controls now available during run. Delay in resizing graphics fixed. Faster output file production. “Direct Threshold Compliance” across all distributions (Beta test release). Improvements to core algorithms:

		<p>Greater speed for most issues affected by mate allocation (inbreeding, trait thresholds, avoid self matings, etc.) Speed of accessing tight solutions. Sundry other improvements.</p> <p>Note the following for compatibility with your pipeline scripting: No Histogram result files in Batch mode, for speed. Marker results removed from OutGroups.txt (probably never accessed from there) Some changes to the content and format of Matesel.ini, See here.</p>
16/06/2020	6.9 2.2 11.4	<p>Frontier drawing improved. Various new error traps with messaging. Fix for extended character sets causing login issues. Note the following for compatibility issues with your pipeline scripting: *.csv as well as *.txt files now made for key output files File formatting matched between GUI and Batch versions OutData.* renamed to OutResults.* Order of fields changed in OutResults.* New output file OutFrontier.txt</p>
31/05/2020	6.8 2.1 11.3	<p>New feature: Stabilising option for Tactical Desired Gains. Missing trait values now handled in histogram displays. (This eases operation, but analysis always handled these missing values.) Genotypes of candidates that are only listed in CommittedMatings.txt are now accommodated in reporting genotype frequency results. Extra space in format of OutDataDLL.txt to accommodate missing EBV information - please note this for pipelined operations. Matesel batch versions can now be launched from outside their own folder, either explicitly or from being in the system PATH. Logon connection diagnostics improved. Bug fix: Desired Gains no longer over-riding.</p>
13/03/2020	6.7 2.1 11.2	<p>New feature to force specific numbers of matings per group combination: Dramatic speed improvement for highly constrained problems. Improved filter algorithm for Group Minuse. Incorrect sex field entries reported, and Maxuse set to zero. Zero variation in EBVs now handled in trait Management. Method described for Genomic Mating/Genomic Mate Selection (p77). Matesel.Helpers: Matesel now works under Windows 7 (computability was lost in Version 10.9) and server connection improved.</p>
10/12/2019	6.6 2.1 11.1	<p>Detection of GrpGrp Targets all set to zero when weighting. Expiry date bug fix.</p>
19/11/2019	6.6 2.1 11.0	<p>Warning on duplication of some key fields in input header. Better reporting of fields that are missing. Fix bug when using preselection under Committed Matings. Formatting adjustment to Effective weight on Coancestry and Fitness in OutSummary.txt Licence file expiry notification improved</p>
12/06/2019	6.5 2.1 10.9	<p>Fix for "Error 65475". Little need for this update unless you see this error. Software expiry warning one month before the event.</p>
03/05/2019	6.5 2.1 10.8	<p>Improved trapping of blank ID and zero matings issues. Improved logon exception handling.</p>

14/03/2019	6.5 2.1 10.7	Reporting of construction errors for files InpOneGroup.txt, InpGroups.txt and EndUses.txt. Allow a whole Group to have no mating permissions. Security testing more robust. Fix indexing of nUse in RankSibsToPrintF. Warning trap for wrong number of Trait and/or Marker datalines in Matesel.ini. Close button disabled till very end of run. Targeting minimum use of individual males within each female group using both filter and weighting. Web version includes Service Provider security and tools described in “Instructions to Service Providers for hosting a MateSel web service”. [Supersedes beta versions 10.7xx]
14/11/2018	6.3 2.0 10.6	ControlBox and Close button disabled during run to ensure graceful exit of DLL. Use “Stop” to first stop the run. Ctrl Right-Click Stop for overriding stop. Timeout on zero CPU availability extended to 10 mins. Fixed: GetZeroDegrees bug; Click Run when running could hide current Frontier in tiny datasets.
26/09/2018	6.02 2.0 10.5.	Now reporting all duplicate IDs in CandGRM.txt. Set verbose \geq 10 in Matesel.ini to generate Debug.txt. Foundation generation now adheres to Minuse; Max length of ID increased to 20.
16/08/2018	6.02 2.0 10.4	Improvements in speed and handling Minuse constraints under Grouping. Improvements in speed under no-Grouping. New warning messages for 4 different issues to do with meeting usage and grouping constraints.
01/08/2018	6.02 2.0 10.03	Improved reporting on insufficient male Maxuse under complex grouping. Chinese character operating system now supported. Non-standard date format issue fixed. Matesel.ini folder issue fixed. Proxy login issue fixed.
04/04/2018	6.0 2.0 10.0	<u>New features:</u> This is a major new upgrade. Tactical Desired Gains Control of self matings Much more extensive trapping and reporting of logical errors in the main data file and the parameter files. Improved frontier drawing (Frontier bounds keep previous outer points in case main run oversteps in one dimension while at intermediate degrees). Software upgrade from www.matesel.com, Logging in for Windows version. Substantial improvement in the Instruction Manual. <u>Backward compatibility:</u> This version set is backward compatible with data and parameter files used for all versions released since 13 April 2014 (Windows 2.6; FortranDLL 4.8) except for the following new changes:

		<p>Soft implementation of the Location control feature now has its own line in Matesel.ini for applying weights, rather than borrowing the weight entered for WeightOnxAx2.</p> <p>The weighting for Avoiding Duplicate and Reciprocal matings is now managed differently, following addition of the feature to avoid self matings.</p>
--	--	---

* Windows/Web/FortranDLL