

# **MateSim Manual v 1.0**

*Antonio Carvajal-Rodríguez*

*Departamento de Bioquímica, Genética e Inmunología*

*Universidad de Vigo, Vigo 36310, Spain*

*Email: [acraaj@uvigo.es](mailto:acraaj@uvigo.es)*

*Web: [acraaj.webs.uvigo.es/MateSim/matesim.htm](http://acraaj.webs.uvigo.es/MateSim/matesim.htm)*

## TABLE OF CONTENTS

Introduction.....	3
Version update.....	4
Process of pair formation.....	5
Discrete preference models.....	7
User-input discrete models.....	7
Population size and drift.....	8
Mating tables for discrete traits.....	9
Population phenotype frequencies.....	9
Matings.....	10
Bimodal populations for continuous models.....	12
Continuous preference models.....	13
Gaussian model.....	13
Logistic Model.....	14
Decreasing in time choosiness (impatience for mating).....	14
Aging: Weibull model.....	15
The Program.....	16
Download and ready-to-use executables.....	16
Compiling the program manually (linux/MAC).....	16
Run the program.....	16
User Input.....	17
Continuous models.....	17
Discrete models.....	18
Command line options.....	18
Examples.....	22
Discrete.....	22
Continuous.....	23
Output.....	23
Discrete models.....	23
Continuous models.....	24
References.....	24

## INTRODUCTION

MateSim (Carvajal-Rodríguez, 2018a) performs Monte Carlo simulation for generating mating tables for discrete or continuous traits. The aim of the software is to provide a tool that permits to explore the relationship between the individual mating preferences and the observed population mating pattern. MateSim would also be useful for testing different sexual selection estimation methods.

When the mating trait is discrete, the program can manage different models as random mating, female or male competition, and mate choice, or any user-defined mutual mating propensity model (sensu Carvajal-Rodríguez, 2018b).

For continuous traits the program may use a Gaussian (unimodal or bimodal) or a logistic (open-ended) preference function. It can also incorporate age structure to the model.

Both for discrete or continuous cases, the program performs distinct processes of pair formation (with or without replacement) and demography, to finally generate mating tables in the JMating (Carvajal-Rodríguez and Rolan-Alvarez, 2006) and InfoMating (Carvajal-Rodríguez, 2020) program formats for the discrete case, or a list of mating pairs and their correlation in the continuous case.

## VERSION UPDATE

### *Current version 1.0*

Added the possibility of defining only 1 type for any sex. Fixed a bug that happened under the discrete scenario with user-defined frequencies and no drift. In this case the frequencies from the file *MateSimUserFreqs.txt* were not used and the program failed.

### *Version 0.9*

New input tag *-MRmax* that works only for discrete traits and indicates the maximum number of matings per round under a mass-encounter model. By default is  $\min(n_1, n_2)$  where  $n_1$  and  $n_2$  are the population number of females and males, respectively. Note that if it is set to one (*-MRmax 1*), the mass-encounter becomes individual-encounter model.

Also, only for discrete traits, a new model number is added to run random mating under mass-encounter without requiring user-defined model. The new model has negative number -123321.

Therefore, for producing a mating table from a monogamous species under an individual encounter model we may use the following arguments:

*-continuous 0 -N 2000 -model -123321 -matings 400 -uniform 0 -drift 0 -MRmax 1*

The encounter model is mass-encounter but the maximum number of simultaneous matings at each encounter is set to 1 (*-MRmax 1*) so that the encounters become individual-encounters. The population size is two thousand, there are two phenotypic classes by sex and the population frequencies are not uniform ( $\neq 0.5$ ), there is no drift and the number of matings is 400.

### *Version 0.8*

The input tag *-incrfactor* changed to *-selfactor*. The input tag *-decrfactor* changed to *-choicfactor*. Previous tags are still valid but *-choicfactor* (instead of *incrfactor*) is required now for model 3.

NEW discrete models added (model numbers from 6-9, see Table S1). These models have separate parameters for intra-sexual competition and mate choice.

**Model 6:** female competition parameter ' $a$ ' generates female sexual selection; choice parameter ' $c$ ' generates assortative mating. For example for  $k_1 = k_2 = 3$ :

```
a  a  a
1  c  1
1  1  c
```

**Model 7:** male competition parameter ' $a$ ' generates male sexual selection; choice parameters ' $c$ ' generates assortative mating. It is obtained by transposition of the model 6.

**Model 8:** female competition and mate choice with compound parameters:

```
ac  a  a
1  c  1
1  1  c
```

**Model 9:** male competition and mate choice with compound parameters. It is obtained by transposition of the model 8.

## PROCESS OF PAIR FORMATION

When the availability of individuals is not affected by the matings that have already occurred, the process of pair formation (the encounter) can be viewed as sampling with replacement from the set of population types. On the contrary, when the matings are monogamous and a significant fraction of individuals of both sexes do actually mate, the proportion of available individuals for mating must be updated after each encounter event (sampling without replacement from the set of population phenotypes). Processes of the latter kind are the so called encounter-mating (EM) models (Gimelfarb, 1988).

MateSim implements both type of processes of pair formation, with or without replacement. The default program mode is panmixia which implies random pair

formation with replacement and no differential individual preferences or vigour among the phenotypes. This setting can be controlled by the user via the tag *-model* (see Table S1 and the command line section below).

After the pair formation, we introduce some restrictions, i.e., for a population with  $n_1$  females and  $n_2$  males, we consider that whatever the process of pair formation, the maximum number of mating attempts is  $n_1 \times n_2$ . Consequently, the sample size is not allowed to be higher than the maximum number of matings. In fact, if the pair formation is a mass-encounter type, the maximum possible number of simultaneous mating (without updating the frequencies) is  $\min(n_1, n_2)$ .

**Table S1. Options for defining the pair formation and the discrete preference model.**

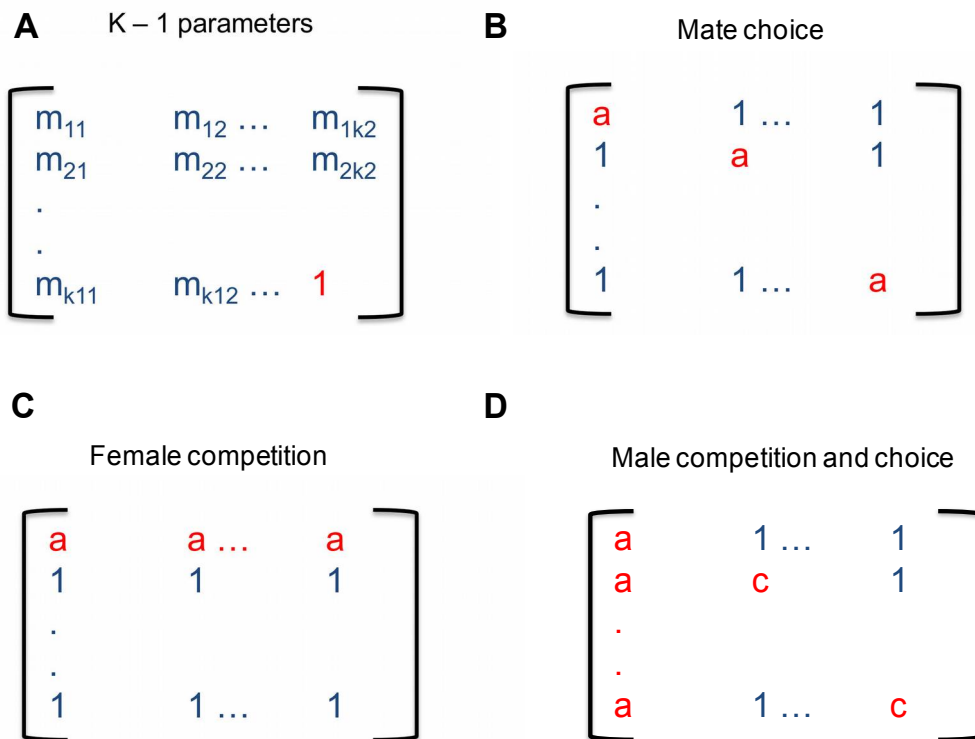
Model	Pair formation	Discrete preferences	# parameters
0	with replacement	random mating	1
1	with replacement	female competition	1
2	with replacement	male competition	1
3	with replacement	mate choice (positive assortative)	1
4	with replacement	negative assortative	1
5	with replacement	user-defined	user-defined
6 (7)	with replacement	female (male) competition plus choice	2
8 (9)	with replacement	female (male) competition plus choice	2 compound
-1	EM-individual	female sexual selection	1
-2	EM-individual	male sexual selection	1
-3	EM-individual	positive assortative	1
-4	EM-individual	negative assortative	1
-5	EM-individual	user-defined	user-defined
-6 (-7)	EM-individual	female (male) competition plus choice	2
-8 (-9)	EM-individual	female (male) competition plus choice	2 compound
-10	EM-mass	female sexual selection	1
-20	EM-mass	male sexual selection	1
-30	EM-mass	positive assortative	1
-40	EM-mass	negative assortative	1
-50	EM-mass	user-defined	user-defined
-60 (-70)	EM-mass	female (male) competition plus choice	2

-80 (-90)	EM-mass	female (male) competition plus choice	2 compound
-123321	EM-mass	random mating	1

EM: encounter-mating model. Model: value for the tag *-model* in the command line.

## DISCRETE PREFERENCE MODELS

These are the models by default, and imply a discrete trait involved in the mating process. The use of continuous or discrete traits, can be changed via the boolean tag *-continuous* (see the command line section below). In Figure S1 we appreciate different types of discrete models that can be defined by the tag *-model* (Table S1)



**Fig S1.** Discrete mating models as defined by the mutual mating propensities. The rows are female types and columns male types. A: The saturated model with  $K-1$  free parameters; this model produces mixed effects of assortative mating plus sexual selection. B: One-parameter model generating at least assortative mating pattern. C: One-parameter model producing at least female sexual selection pattern. D: Two-parameter model producing at least male sexual selection (parameter  $a$ ) and assortative mating (parameter  $c$ ) patterns.

## User-input discrete models

Models B, C and D in Figure S1, involve one or two parameters. If the user desire models with more parameters (as model A) this should be indicated to the program. This can be done when the tag *-model* has any of the following values {5, -5, -50} (each value corresponding to a different pair formation process, see Table S1). Then, the program assumes that a text-file called *MateSimUserModel.txt* exists in the same folder as the executable.

The format of the input file should be as in Fig S2, with the first non-commentary line indicating the number of female and male types and the following lines indicating the corresponding individual mutual mating propensities. Each row represents a female phenotype and each column a male phenotype. Thus, there should be as many rows as female phenotypic classes and as many columns as male classes.

```
# best model from some data

# num of fem and male types
2      2
# mutual propensities by rows (females)
0.81   1
1      0.63
```

**Fig S2.** Input file format for user-defined mutual propensity model.

Note that this option also permits to define random mating models under other pair formation types than with replacement (e.g. EM-i and EM-mass, Table S1). For example, defining an EM-i random mating model, requires setting the model tag to -5 and instantiating the model within the *MateSimUserModel.txt* file, with all mutual-propensities equal to 1.

Some command line examples are available [here](#).

### **Population size and drift**

In the discrete models we manage counts of different phenotypic classes. However, the effect of finite population size may has an impact on the observed mating patterns even if the pair formation occurs with replacement. Accordingly, when the population



number of individuals from any sex is less than 250, we model the effect of evolutionary sampling error (drift). In addition, the consideration of the drift effect can be defined by the user via the tag *-drift* (see the command line section).

The number of individuals of each sex is obtained by means of the population size parameter  $N$  and the operational sex ratio (OSR), which is the ratio of sexually competing males and females that are ready to mate (Clutton-Brock, 2007).

When drift is not considered, the observed population numbers of both sexes correspond exactly to the defined (default or user-defined) population frequencies. On the contrary, when drift is considered the observed population numbers are sampled following a Monte Carlo sampling algorithm.

## **MATING TABLES FOR DISCRETE TRAITS**

As a result of the process of pair formation jointly with the mating preferences, we finally get a table with the population phenotype frequencies and the observed mating counts.

### **Population phenotype frequencies**

For a given population of size  $N$ , we need to generate the number of pre-mating males and females of each class. As already mentioned, the total number of females  $n_1$  and males  $n_2$  is computed considering the operational sex ratio ( $OSR = n_2/n_1$ ), so  $n_1 = N/(1+OSR)$  and  $n_2 = N - n_1$ .

The phenotypic frequencies can be uniformly distributed or not, as defined by the tag *-uniform*. Therefore, for the computation of the phenotypic frequencies there are several options.

#### *1) Uniform frequencies without drift (-uniform 1 -drift 0)*

By default, the frequencies are assumed to be uniform. If there is no drift, this means that, given the phenotype  $A$  among  $k_1$  different female classes, the number of  $A$

females is  $n_{1A} = n_1 / k_1$ , and the number of A males  $n_{2A} = n_2 / k_2$  for  $k_2$  male phenotypes. The last female class (say the  $k_1$ th) is always computed as  $n_{1k_1} = n_1 - \sum_{(k_1-1)} n_{1i}$  and similarly for males,  $n_{2k_2} = n_2 - \sum_{(k_2-1)} n_{2i}$ . Then, the female population frequency for phenotype A is  $p_{1A} = n_{1A} / n_1$  and the male,  $p_{2A} = n_{2A} / n_2$ . Because there is no drift the initial population frequencies do not vary among files.

#### 2) Uniform frequencies with drift (-uniform 1 -drift 1)

A Monte Carlo sampling algorithm is applied, for which the expected frequency of each phenotype corresponds to the uniform expectations as computed in the previous case. The difference with the previous case is that the population frequencies vary among files.

#### 3) Non-Uniform non-user-defined frequencies without drift (-uniform 0 -drift 0)

In this case the frequencies are random values between 1 and  $n_1$  ( $n_2$ ) for females (males). Because there is no drift the initial population frequencies do not vary among files.

#### 4) Non-Uniform non-user-defined frequencies with drift (-uniform 0 -drift 1)

The expected frequencies are randomly obtained as in the previous case. Afterwards, a Monte Carlo sampling algorithm is used for obtained the observed population frequencies. The difference with the previous case is that the population frequencies vary among files.

#### 5) User-defined frequencies without drift (-uniform 0 -drift 0 -userfreqs 1)

When the tag *-userfreqs* is followed by the value 1 this indicates that the population phenotypic frequencies are user-defined from an input file (see input section below). In this case if there is no drift, the user frequencies are utilized directly as the population frequencies for each phenotypic class.

#### 6) User-defined frequencies with drift (-uniform 0 -drift 1 -userfreqs 1)

As in previous cases, a Monte Carlo sampling algorithm is applied, in which the expected frequency of each phenotype corresponds to the user-defined frequencies.

## Matings

Once the population phenotype frequencies have been computed and the encounters performed, we obtain a sample of  $n'$  matings.

If there is replacement for the mating phenotypes, the expected number of occurrences for each mating class  $i \times j$  is (Carvajal-Rodríguez, 2018b)

$$Q(i,j) = n' \times p_{1i} \times p_{2j} \times m_{ij} / M \quad (1)$$

where  $M = \sum p_{1i} \times p_{2j} \times m_{ij}$ .

The  $m_{ij}$  values correspond to the mutual mating propensities and will depend on the desired model (Figs. S1 and S2).

The format of the obtained tables is the same as the JMating (Carvajal-Rodríguez and Rolan-Alvarez, 2006) and InfoMating input files (Fig S3). The default format is 0 but this is automatically changed to format 2 (Fig S3-B) if the user defines a model with different number of female and male phenotypes ( $k_1 \neq k_2$ ).

<p><b>A</b></p> <p># format number 0</p> <p># num of types 2</p> <p>#premating male numbers 720 720</p> <p>#premating female numbers 720 720</p> <p># matings by rows (females) 611 87 247 759</p>	<p><b>B</b></p> <p># format number 2</p> <p># num of types 2 3</p> <p>#premating male numbers 500 500 500</p> <p>#premating female numbers 720 720</p> <p># matings by rows (females) 611 87 300 247 759 120</p>	<p><b>E</b></p> <p># format number 15</p> <p># num of types 2 3</p> <p>#premating male numbers 0.333 0.333 0.334</p> <p>#premating female numbers 0.5 0.5</p> <p># matings by rows (females) 611 87 300 247 759 120</p>
<p><b>C</b></p> <p># format number 1</p> <p># num of types 3</p> <p># matings by rows (females) 102 40 60 40 120 230 30 70 77</p>	<p><b>D</b></p> <p># format number 4</p> <p># num of types 2 3</p> <p># matings by rows (females) 611 87 300 247 759 120</p>	

**Fig S3.** Different output formats. A: Format number 0, the same as JMating. B: Format 2, the same as A but the number of types for females and males can be different. C: Format 1, the A format without premating information. D: Format 4, the B format without premating information. E: Format 15, the B format with the premating information as relative frequencies.

## BIMODAL POPULATIONS FOR CONTINUOUS MODELS

Consider the density of a two-component normal mixture where  $p$  is the mixing proportion:  $h(x; \mu_1, \mu_2, \sigma_1, \sigma_2, \pi) = \pi\phi(x; \mu_1, \sigma_1) + (1 - \pi)\phi(x; \mu_2, \sigma_2)$ .

The mixed distribution either bimodal or unimodal, has the mode(s) lying between  $\mu_1$  and  $\mu_2$  where without loss of generality  $\mu_2 > \mu_1$ . The distribution is unimodal or bimodal depending on the values of the three parameters  $\pi$ ,  $\mu = (\mu_2 - \mu_1) / \sigma_1$  and  $\sigma = \sigma_2 / \sigma_1$  (Robertson and Fryer, 1969). For example, if  $\pi = 0.5$  and  $\sigma_2 = \sigma_1 = s$  then  $h$  is unimodal only if  $|\mu_2 - \mu_1| \leq 2s$ . More generally, a mixture of two normal densities will not be bimodal unless there is a large difference between their means, typically larger than the sum of their standard deviations (Schilling et al., 2002). Besides, the number of modes that can be obtained by mixing two normal unidimensional components is 2 and in general for a  $D$ -dimensional mixture the maximum number of modes is  $D+1$  (Ray and Ren, 2012).

The modal features of a two component normal mixture remains unchanged under scaling and rotation, so we can generate a bimodal distribution by mixing a standard normal and a normal  $N(\mu, \sigma)$ . In this case, the condition for bimodality implies  $\mu > \sigma + 1$ . Therefore, if a continuous bimodal population is desired then the user may define the mixture proportion  $\pi$  ( $0 < \pi < 1$ ) and the mean  $\mu$  and standard deviation  $\sigma$  with the restriction  $\mu > \sigma + 1$  (see command line section below). Thus, the continuous population distribution will be bimodal provided that the following conditions hold:  $0 < \pi < 1$ ,  $\mu \neq 0$ ,  $|\mu| > (\sigma + 1)$ .

Note that under the Gaussian preference model even if the population distribution is bimodal the values can be rescaled to be between  $[\text{trait}_{\text{minval}}, \text{trait}_{\text{minval}} + d\text{max}]$  ( $[0, 1]$  by default). The default scaling may compact the values, so that the variance will be  $1 / (\text{max} - \text{min})$  (Carvajal-Rodriguez and Rolán-Alvarez, 2014) where  $\text{max}$  and  $\text{min}$  are the maximum and minimum values respectively, from any of the two components of the mixture. If such variance reduction is not desired, the  $\text{trait}_{\text{minval}}$  and/or  $d\text{max}$  values should be defined accordingly (see  $\text{mintrait}$  and  $d\text{max}$  tags in the command line section).

## CONTINUOUS PREFERENCE MODELS

We can define a continuous model by setting the tag *continuous* to 1

*-continuous1*

Once the continuous type is defined, the implementation of a Gaussian or logistic preference depends on the identifier for the *-model* tag (Table S2) which will make the difference on how we parameterize the general preference function (see below)

$$f = \exp(\alpha_0 + \alpha_1 Z) \quad (2)$$

Some command line examples for defining continuous models are available [here](#).

### Gaussian model

The use of FND (Carvajal-Rodriguez and Rolán-Alvarez, 2014) or BD03 (Bolnick and Doebeli, 2003) Gaussian similarity models depend on the model defined under the tag *-model* with {1, -1, -10} for FND and {2, -2, -20} for BD03 (Table S2).

The model parameters can be changed by the corresponding tags (see the command line section).

Considering equation (2) under the Gaussian model, we have that the value of  $Z$  is different depending on whether there is only one choosy sex or not. If only one sex is choosy (see tag *onechoosy* in the command line section) we have  $Z_{ij} = |d_{ij} - \text{bias}|^2$ . On the contrary, if both sexes are choosy, we have  $Z_{ij} = (|d_{ij}| - \text{bias})^2$ . The quantity  $d_{ij}$  is the difference between female and male phenotypes.

If the Gaussian model is FND then  $\alpha_0 = 0$  and  $\alpha_1 = (C/s)^2$  where  $C = \text{choice}$  and  $s = \text{tolerance} \times \text{dmax}$ . And  $\text{dmax}$  is defined as in the command line options below.

If the model is BD03 then  $\alpha_1 = (C^2 / \text{tolerance})^2$ .

**Table S2. Options for defining the pair formation and the continuous preference model.**

Model	Pair formation	Preference model
1	with replacement	Gaussian-FND
2	with replacement	Gaussian-BD03
3	with replacement	Logistic
-1	EM-individual	Gaussian-FND
-2	EM-individual	Gaussian-BD03
-3	EM-individual	Logistic
-10	EM-mass	Gaussian-FND
-20	EM-mass	Gaussian-BD03
-30	EM-mass	Logistic

EM: encounter-mating model. FND: FND preference function in (Carvajal-Rodriguez and Rolán-Alvarez, 2014). BD03: Bolnick and Doebeli preference function in (Bolnick and Doebeli, 2003).

### Logistic Model

The default values for the logistic model  $X = \alpha_0 + \alpha_1 Z$  are  $\alpha_0 = 0$  and  $\alpha_1 = 1$ . This setting can be changed by the tag *-logistic\_params*. For example:

*-logistic\_params*    -1    1.7

which defines  $\alpha_0 = -1$  and  $\alpha_1 = 1.7$ .

### Decreasing in time choosiness (impatience for mating)

As defined, both under Gaussian and logistic models, the choosiness characteristic of the individuals is constant through time, however, we could consider an scenario where single individuals become impatient as time passes and an increasing number of matings or “marriages” are performed, especially if neither polygamy nor divorce is allowed (Xie et al., 2015).

Therefore as time passes the individuals become less choosy proportional to the number of marriages already performed.

For the Gaussian preference this "impatience" is implemented by changing the definition of  $\alpha_1$  to be  $\alpha_1 = (C/s_t)^2$  where  $s_t = s \times (n_t + 1)$ , with  $s = \text{tolerance} \times \text{dmax}$  as previously defined and  $n_t$  is the number of existing marriages at time  $t$ . The default value is no impatient behavior and so  $s_t = \text{tolerance}$ , that can be changed by means of the tag *-impatience* which when different from 0 sets the impatient behavior.

For the logistic the impatience is defined by

$$X = \alpha_0 + cn_t + \alpha_1 Z$$

where  $c$  is constant and  $n_t$  stands for the number of existing marriages at time  $t$ .

The value of  $c$  is bay default 0 and can be changed, say to 0.001, by the tag

$$\text{-impatience } 0.001$$

Note that if impatience was defined, then as  $n_t$  increases, the mating probabilities tend to 1 independently of the mating trait, so the mating becomes less selective.

### **Aging: Weibull model**

The survival function  $R(t)$  represents the probability that at age  $t$  an individual is not yet sterile.

$$R(t) = \text{Prob}(T > t) = \exp\left[-\left(\frac{t-\gamma}{\alpha}\right)^\beta\right] \quad \text{if } t - \gamma \geq 0 \text{ or } 1 \text{ otherwise.}$$

The age is measured in terms of the natural logarithm of the mating attempts.

The continuous preference model, by default assumes no age structure, so that  $R(t) = 1$  for every age. The aging settings can be defined by the tag

$$\text{-age_params } 10 \quad 2 \quad 0$$

that introduces the values  $\alpha = 10$ ,  $\beta = 2$  and  $\gamma = 0$ .

If we don't want the aging process, we can simple put the alpha parameter to 0 or a negative sign to the gamma parameter so that both

```
-age_params 0    2    1
-age_params 10   2    -1
```

imply that there is not aging ( $R(t) = 1$  for every age).

An individual is considered sterile at time  $t$  when  $U \geq R(t)$  where  $U$  belongs to  $uniform(0,1)$ .

## THE PROGRAM

### Download and ready-to-use executables

The program can be downloaded from <http://acraaj.webs.uvigo.es/MateSim/matesim.htm>.

The zip file should include the windows and linux (Ubuntu) ready to use executables.

### Compiling the program manually (linux/MAC)

Navigate the console to the source folder that includes a file called Makefile, and just type make.

### Run the program


To execute the windows version of the program you can double click the MateSim.exe executable, which is equivalent to go to the command prompt (cmd.exe) and typing (default values)

```
MateSim.exe -continuous 0 -format 0 -model 0 -matings 100 -numfiles 1 -shift 0 -N
500 -femtypes 2 -maletypes 2 -osr 1 -uniform 1 -drift 0 -userfreqs 0 -userphenotypes 0
-selfactor 1.25 -choicefactor 1.25 -age_params 0 2 -1 -logistic_params 0 1 -impatient
0.0 -choice 0 -tolerance 0.08 -bias 0 -dmax 1 -writefiles 0 -fixseed 0
```

where the tag *-writefiles* is automatically set to 1 if the number of files is just one.



The meaning of each argument is given in the command line options section.

You can also access the Run command by pressing the Windows logo key +R, then drag and drop the .exe file from your folder and add the desired arguments, e.g if the MateSim.exe is in the folder MateSim then after drag and drop you will have

```
C:\MateSim \MateSim.exe
```

now add the desired arguments and then hit the Intro key, e.g.

```
C:\MateSim\MateSim.exe -continuous 0 -model 1 -N 10000 -matings 500 -numfiles 10
```

## USER INPUT

MateSim by default does not require an input file. However depending on the user-defined options, it could need different input files.

### Continuous models

Under continuous models an input file is only necessary when the user wants to provide the population phenotypic distribution. In this case, the file *MateSimUserPhenotypes.txt* must be present in the same folder as the program. For a population size of  $n$  ( $= n_1$  females +  $n_2$  males, depending on the operational sex ratio), the file format should consist in a list of  $n_1$  female values plus  $n_2$  male values e.g. if the population size is  $n=10$  and  $n_1= n_2=5$ :

```
// these are read as females
```

```
0.1
```

```
0.2
```

```
0.01
```

```
0.0067
```

```
0.32
```

```
// these are read as males
```

```
0.407
```

```
0.3
```

```
0.11
```

```
0.0987
```

## Discrete models

Under the discrete models and depending on the user-defined options, MateSim could require two different input files. The file for the **user-defined mutual mating propensity** model has already been explained (see user input discrete models subsection above).

The other input file contents the **user-defined frequencies** and is only necessary when the user wants to define the population frequencies. In this case, the file *MateSimUserFreqs.txt* must be present in the same folder as the program. The input file must have two non-commentary lines, the first for the desired female population frequencies and the second for males:

```
#females freq (k1=3)
0.4    0.1    0.5

#males freq (k2=3)
0.3    0.4    0.3
```

The number of frequency classes must coincide with the elsewhere defined female ( $k_1$ ) and male ( $k_2$ ) number of types. The frequencies for each sex must sum 1.

## COMMAND LINE OPTIONS

The following arguments can be passed to the program:

*-age\_params*<DOUBLE DOUBLE DOUBLE> For continuous (Gaussian or logistic) models only. The value of the parameters  $\alpha$ ,  $\beta$  and  $\gamma$  for the Beiwull age model.

- bias*<DOUBLE> For assortative positive Gaussian models only. This tag fixes the bias between the female and male preference. If the phenotypes are scaled (see the scaling tag) the bias is scaled to be in [0, dmax].
- bimodal*<DOUBLE DOUBLE DOUBLE> For continuous models only. The value of the parameters  $\pi$  ( $0 < \pi < 1$ , default 0.5),  $\mu$  (default 0) and  $\sigma$  (default 1).
- choice*<DOUBLE> For Gaussian similarity models only. Provided that a continuous Gaussian model was defined, this tag fixes the value for the choice preference parameter.
- continuous*<BOOLEAN> Indicates if the mating trait is discrete (0 by default), or continuous (1).
- choicefactor*<DOUBLE> For discrete models only. Defines the value of the mate choice preference parameter (discrete models 3, 4, 6, 7, 8, 9 and the EM counterparts). By default is 1.25. Should be  $> 1$  for model 3 (positive assortative) and  $< 1$  for model 4 (negative assortative).
- dmax*<DOUBLE> For Gaussian similarity models only. Provided that a continuous Gaussian model was defined, this tag fixes the maximum distance value between any two traits (independently of the sex). Default value is 1.
- drift*<BOOLEAN> For discrete models only. By default is 0. If 1 indicates that the algorithm for Monte Carlo sampling must be applied instead of using the deterministic expected values. The default value is changed to 1 always when the population number of any sex is below 250 individuals.
- femtypes*<INTEGER> For discrete models only. Number of female phenotypic categories.
- fixseed*<LONG LONG> The default value is 0 that means that the seed for the random generator is initialized to the unix timestamp. If a given positive value is passed, the seed will be initialized to that value.
- format*<INTEGER> For discrete models only. Defines the format for the output under discrete models. Default is 0 when  $k_1 = k_2$  or 2 otherwise ( $k_1 \neq k_2$ ).

- impatience*<DOUBLE> For continuous models only. The value of the parameter  $c$  for the decreasing choosiness logistic model.
- selfactor*<DOUBLE> For discrete models only. Defines the value of the preference parameter in discrete sexual selection models. Must be  $\geq 1$ . By default is 1.25.
- logistic\_params*<DOUBLEDOUBLE> For logistic models only. The value of the parameters  $\alpha_0$  and  $\alpha_1$  for the logistic preference formula.
- maletypes*<INTEGER> For discrete models only. Number of male phenotypic categories.
- matings*<INTEGER> Defines the number of matings (sample size).
- mintrait*<DOUBLE> For Gaussian similarity models only. Default value is 0. It defines the minimum value for performing the scaling of phenotypes to be between  $\text{mintrait}$  and  $\text{mintrait} + \text{dmax}$ .
- model*<INTEGER> Indicates the pair formation process and the preference model (Tables S1 and S2). By default is 0 which under discrete models means random mating (but see the model number -123321 for the version 0.9). Otherwise, the sign of the value indicates whether the model should be with or without replacement. If negative, the process is an encounter-mating (EM) model without replacement, while a positive value indicates models with replacement of the mating types. If the negative value is  $\leq -10$  the EM model implies mass encounters, if not, the EM model implies individual encounters (there is no simultaneous mating and the frequency update occurs after each mating). Besides the sign, the magnitude of the value indicates the kind of individual preference model. When the **model is discrete**, if the value is -10, -1 or 1, the preferences in the model produce at least female sexual selection pattern under idealistic conditions (Figure S1-C). If the value is -70, -7 or 7 the preferences in the model produce at least male sexual selection effects jointly with assortative mating (Figure S1-D). If the value is -50, -5 or 5 the preferences are user-defined by the values within the *MateSimUserModel.txt* file. See Table S1 for the complete set of discrete models. When the **model is continuous**, the values -10, -1 or 1, indicate that the preference function is the Gaussian-FND;

the values -20, -2 or 2, indicate that the preference function is the Gaussian-BD03. Finally, the values -30, -3 or 3 indicate that the preference function is the logistic (Table S2).

*-MRmax<INTEGER>* For discrete traits only. Indicates the maximum number of matings per round under a mass-encounter model. By default is  $\min(n_1, n_2)$  where  $n_1$  and  $n_2$  are the population number of females and males, respectively.

*-N<INTEGER>* The population size including males + females.

*-numfiles<INTEGER>* Specifies the desired number of mating tables (the number of files with one table per file, by default is 1).

*-onechoosy<BOOLEAN>* For Gaussian similarity models only. The default value is 0. If 1, it indicates that only one sex is choosy. If the phenotype distribution is user defined, the choosy sex corresponds arbitrarily to the second set (the ones read as males as explained in the section Input Continuous models). Note that if we desire the females to be the choosy sex we may simply consider the males as females and viceversa. For example, if we desire the females to be the choosy sex and the sex ratio (osr) is not 1, so we have  $n_1$  females and  $n_2$  males ( $n_1 \neq n_2$ ) then we set instead  $n_1$  males and  $n_2$  females.

*-osr<DOUBLE>* The operational sex ratio (default 1).

*-scaling 0<BOOLEAN>* For Gaussian similarity models only. The default value is 1. If 0 it does not perform scaling. If 1 it scales the initial phenotypes to be between mintrait and dmax +mintrait.

*-shift<INTEGER>* This parameter could be useful when we generate more than one file, it indicates the beginning index in the file name.

*-tolerance<DOUBLE>* For Gaussian similarity models only. Provided that a continuous Gaussian model was defined, this tag fixes the value for the tolerance preference parameter.

*-uniform<BOOLEAN>* For discrete models only. By default is 1 indicating that the phenotypic frequencies are uniform ( $1/femtypes$ ,  $1/maletypes$ , in females and males respectively).

- userfreqs*<BOOLEAN> For discrete models only. By default is 0. If 1 indicates that the phenotypic frequencies are user defined and should be read from the file *MateSimUserFreqs.txt* (see input section above).
- userphenotypes*<BOOLEAN> For continuous models only. By default is 0. If 1 indicates that the phenotype distribution is user defined and should be read from the file *MateSimUserPhenotypes.txt* (see input section above).
- writedistrib*<BOOLEAN> For continuous models only. The default value is 0. If 1 it writes the phenotypic distribution (after scaling) to a file called phenotypes.txt.
- writefiles*<BOOLEAN> For continuous models only. The value is fixed to 1 when the argument for the number of files is 1. However, if the program must perform more than one file e.g. -numfiles 2, the default value is 0, which implies that only one file is written with the correlations in as many rows as number of files defined by the -numfiles tag. Each row includes the number of the case and the mating correlation for that run. If we want a more exhaustive output we should define -writefiles 1 and then we additionally obtain as many as -numfiles, each one including the list of matings of the corresponding run.

## EXAMPLES

### Discrete

For obtaining 10 tables with random mating and sampling with replacement (model =0), population size 10000, sample size 500 with uniform phenotype population frequencies but subjected to drift. The list of arguments should be:

*-continuous 0 -numfiles 10 -N 10000 -model 0 -matings 500 -uniform 1 -drift 1*

If an individual-encounter with random mating is desired, it is necessary to have a user-defined model (model -5) and the *MateSimUserModel.txt* file must have propensities of 1 (should be in the same folder as the program)

*-continuous 0 -numfiles 10 -N 10000 -model -5 -matings 500 -uniform 1 -drift 1*

More command line examples for defining discrete models are also available [here](#).

## **Continuous**

To obtain 10 mating files (*-numfiles 10 -writefiles 1*) with Gaussian-FND preference with replacement, random mating (*choice =0*), population size 10,000 and sample size 400.

No impatience, no aging:

*-continuous 1 -numfiles 10 -N 10000 -model 1 -choice 0 -tolerance 0.08 -bias 0 -impatience 0.0 -age\_params 5 2 -1 -matings 400 -writefiles 1*

To obtain just 1 file (*writefiles 0*) with the correlations obtained in 10 mating simulations under a logistic individual-encounter mating with impatience and aging, population size 10,000, sample size 5000 (the whole population do mate).

*-continuous 1 -numfiles 10 -N 10000 -model -3 -logistic\_params -1 1.7 -impatience 0.0005 -age\_params 5 2 1 -matings 5000 -writefiles 0*

More command line examples for defining continuous models are available [here](#).

## **OUTPUT**

### **Discrete models**

The output folder and filename reflect the parameters of the simulation. Therefore, for the following command line

*-numfiles 2 -N 500 -model -30 -matings 100 -uniform 1 -choicefactor 2 -shift 0*

the output directory is *MateSim\_massEM\_3\_a2\_U\_sample100* and the two files are *MateSim\_D\_massEM\_M3\_a2\_U\_1.txt* and *MateSim\_D\_massEM\_M3\_a2\_U\_2.txt*.

Where the *D* after *MateSim* refers to the discrete case, and *M3* refers to the preference model type (Table S1), *a2* indicates that the value of the choice parameter *a* (Fig S1) is 2 and *U* stands for uniform frequencies. The ending number before the file extension indicates the number of file.

The content of each file is a table in one of the formats indicated in Fig S3.

### **Continuous models**

The output folder and filename reflect the parameters of the simulation. Therefore, for the following command line

*-continuous1-numfiles 2 -N 10000 -model -3 -logistic\_params -1 2 -impatience 0 -age\_params 5 2 -1-matings 1000*

the output directory is *MateSim\_iEM* and the two correlations obtained after generating two sets of matings (because *-numfiles 2*) are written in the file *MateSim\_C\_n1000\_iEM\_Log\_a-1\_b2\_Correl.txt*.

In the file name the "C" stands for continuous model while the lower case "n" refers to the sample size and "a" and "b" to the defined logistic parameters.

Because the number of files is more than 1, the files with the matings were not generated. If we want them we should set the tag *-writefiles 1* and then we obtain the files with the corresponding matings: *MateSim\_C\_n1000\_iEM\_Log\_a-1\_b2\_1.txt* and *MateSim\_C\_n1000\_iEM\_Log\_a-1\_b2\_1.txt*.

### **REFERENCES**

Bolnick, D.I., Doebeli, M., 2003. Sexual dimorphism and adaptive speciation: two sides of the same ecological coin. *Evolution* 57, 2433-2449.



- Carvajal-Rodríguez, A., 2018a. MateSim: Monte Carlo simulation for the generation of mating tables. *Biosystems* 171, 26-30.
- Carvajal-Rodríguez, A., 2018b. Non-random mating and information theory. *Theoretical Population Biology* 120, 103-113.
- Carvajal-Rodríguez, A., 2020. Multi-model inference of non-random mating from an information theoretic approach, *Theoretical Population Biology* in press.
- Carvajal-Rodríguez, A., Rolán-Alvarez, E., 2006. JMATING: a software for the analysis of sexual selection and sexual isolation effects from mating frequency data. *BMC Evol Biol* 6, 40.
- Carvajal-Rodríguez, A., Rolán-Alvarez, E., 2014. A comparative study of Gaussian mating preference functions: a key element of sympatric speciation models. *Biological Journal of the Linnean Society* 113, 642-657.
- Clutton-Brock, T., 2007. Sexual Selection in Males and Females. *Science* 318, 1882-1885.
- Gimelfarb, A., 1988. Processes of Pair Formation Leading to Assortative Mating in Biological Populations: Encounter-Mating Model. *The American Naturalist* 131, 865-884.
- Ray, S., Ren, D., 2012. On the upper bound of the number of modes of a multivariate normal mixture. *Journal of Multivariate Analysis* 108, 41-52.
- Robertson, C.A., Fryer, J.G., 1969. Some descriptive properties of normal mixtures. *Scandinavian Actuarial Journal* 1969, 137-146.
- Schilling, M.F., Watkins, A.E., Watkins, W., 2002. Is Human Height Bimodal? *The American Statistician* 56, 223-229.
- Xie, Y., Cheng, S., Zhou, X., 2015. Assortative mating without assortative preference. *Proceedings of the National Academy of Sciences* 112, 5974-5978.