

# Module 3: Populations with selection

Olivier Cotto  
Frédéric Guillaume  
Max Schmid

Tvärminne Zoological Station – University of Helsinki

March 2023

# Module 3

Basic forward-time simulations with selection

# OUTLINE

- Selection in WF populations: deleterious mutations
- Selection in non-WF populations: hard selection
- Demography in populations with a mutation load
- Eco-evolutionary feedback: the mutational meltdown
- Inbreeding depression
- Inbreeding depression and the evolution of dispersal
- Indirect selection: evolution of dispersal without mutation load

# Selection in Nemo

Selection acts on individual fitness. Fitness can be:

# Selection in Nemo

Selection acts on individual fitness. Fitness can be:

- **direct:** *genetic elements* code for a trait under direct selection with an explicit fitness function. Example: deleterious mutations, adaptive trait loci (QTL) directly affecting survival, fecundity, etc.

# Selection in Nemo

Selection acts on individual fitness. Fitness can be:

- **direct:** *genetic elements* code for a trait under direct selection with an explicit fitness function. Example: deleterious mutations, adaptive trait loci (QTL) directly affecting survival, fecundity, etc.
- **indirect:** *genetic elements* code for a trait under indirect selection for which we do not define an explicit fitness function. Example: dispersal, cooperation, any social behavior, which depends on *interactions among individuals* in a complex way (e.g., kin competition/selection).

# Selection in Nemo

Selection acts on individual fitness. Fitness can be:

- **direct:** *genetic elements* code for a trait under direct selection with an explicit fitness function. Example: deleterious mutations, adaptive trait loci (QTL) directly affecting survival, fecundity, etc.
- **indirect:** *genetic elements* code for a trait under indirect selection for which we do not define an explicit fitness function. Example: dispersal, cooperation, any social behavior, which depends on *interactions among individuals* in a complex way (e.g., kin competition/selection).
- **Evolution** of the *traits* in our simulations will depend on their *heritable variation* within populations.

# Selection in Nemo

## Selection LCEs (explicit fitness functions)

<code>viability_selection</code>	<code>#selection on offspring survival</code>
<code>breed_selection</code>	<code>#selection on fecundity or survival</code>
<code>breed_selection_disperse</code>	<code>#selection on fecundity or survival</code>

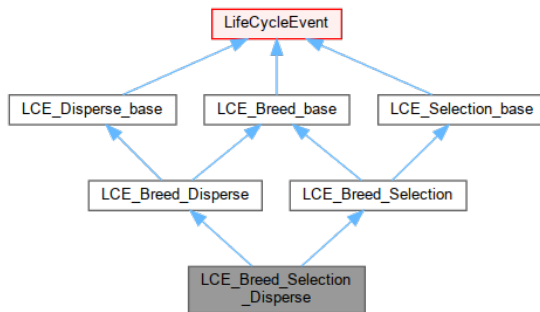


# Selection in Nemo

## Selection LCEs (explicit fitness functions)

viability_selection	#selection on offspring survival
breed_selection	#selection on fecundity or survival
breed_selection_disperse	#selection on fecundity or survival

Class hierarchy:



# Selection in Nemo

## Selection LCEs (explicit fitness functions)

```
viability_selection      #selection on offspring survival
breed_selection          #selection on fecundity or survival
breed_selection_disperse #selection on fecundity or survival
```

## Parameters

```
selection_trait          delet    # deleterious mutations
selection_model          direct   # trait's phenotype = fitness
(selection_fitness_model relative | absolute)
# more model-specific parameters
```

# Selection in Nemo

## Selection LCEs (explicit fitness functions)

```
viability_selection      #selection on offspring survival
breed_selection          #selection on fecundity or survival
breed_selection_disperse #selection on fecundity or survival
```

## Parameters

```
selection_trait          delet    # deleterious mutations
selection_model          direct   # trait's phenotype = fitness
(selection_fitness_model relative | absolute)
# more model-specific parameters
```

### Traits under selection:

- `delet`: deleterious mutations
- `quant`: quantitative trait loci (QTL)
- `dmi`: incompatibility loci

# Selection in Nemo

## Selection LCEs (explicit fitness functions)

```
viability_selection      #selection on offspring survival
breed_selection          #selection on fecundity or survival
breed_selection_disperse #selection on fecundity or survival
```

## Parameters

```
selection_trait          delet    # deleterious mutations
selection_model          direct   # trait's phenotype = fitness
(selection_fitness_model relative | absolute)
# more model-specific parameters
```

### Traits under selection:

- `delet`: deleterious mutations (direct model)
- `quant`: quantitative trait loci (QTL) (gaussian/quadratic model)
- `dmi`: incompatibility loci (direct model)

# Selection in Nemo

## Selection LCEs (explicit fitness functions)

```
viability_selection      #selection on offspring survival
breed_selection          #selection on fecundity or survival
breed_selection_disperse #selection on fecundity or survival
```

## Parameters

```
selection_trait          delet    # deleterious mutations
selection_model          direct   # trait's phenotype = fitness
(selection_fitness_model relative | absolute)
# more model-specific parameters
```

### Traits under selection:

- **delet**: deleterious mutations (direct model)
- **quant**: quantitative trait loci (QTL) (gaussian/quadratic model)
- **dmi**: incompatibility loci (direct model)

# Selection in Nemo-age

⇒ Works as in Nemo but adds stage-specific selection:

```
selection_at_stage
```

⇒ It acts on offspring by default (age-class 0)

⇒ can be set to act on other class(es):

```
selection_at_stage {{1,3}} #two age classes under selection
```

# Selection on deleterious mutations

## The delet trait:

- Encodes universally deleterious mutations which reduce fitness.

# Selection on deleterious mutations

## The delet trait:

- Encodes universally deleterious mutations which reduce fitness.
- Modeled as loci with directional mutation rate from a *wild type* ( $W_{wt,wt} = 1$ ) to a *mutant type* ( $W_{wt,mut} = 1 - hs$ ;  $W_{mut,mut} = 1 - s$ ). Back mutations exist.

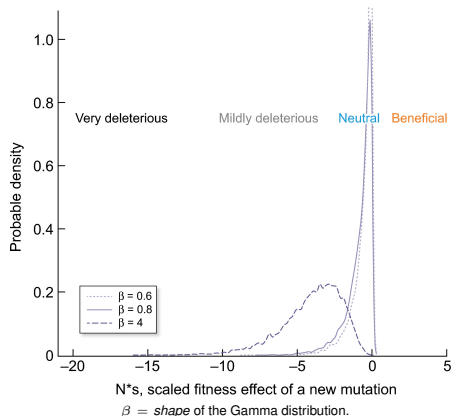


# Selection on deleterious mutations

## The delet trait:

- Encodes universally deleterious mutations which reduce fitness.
- Modeled as loci with directional mutation rate from a *wild type* ( $W_{wt,wt} = 1$ ) to a *mutant type* ( $W_{wt,mu} = 1 - hs$ ;  $W_{mu,mu} = 1 - s$ ). Back mutations exist.
- The distribution of fitness effects  $s$  (the DFE) of the mutations has been estimated in different species. It is a negative gamma with highest density close to  $s = 0$ . Most mutations are close to neutrality ( $s < 0.01$ ) with some mutations having mild effects  $0.01 < s < 0.1$  and rare lethal mutations ( $s > 0.1$ ).

## Shape of the DFE:



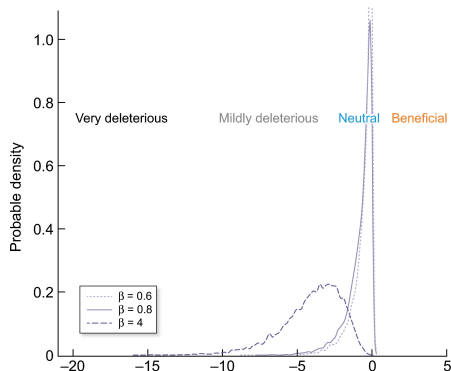
Chen, Bataillon, Glémin, Lascoux, 2022, *New Phytologist*

# Selection on deleterious mutations

## The delet trait:

- Encodes universally deleterious mutations which reduce fitness.
- Modeled as loci with directional mutation rate from a *wild type* ( $W_{wt,wt} = 1$ ) to a *mutant type* ( $W_{wt,mu} = 1 - hs$ ;  $W_{mu,mu} = 1 - s$ ). Back mutations exist.
- The distribution of fitness effects  $s$  (the DFE) of the mutations has been estimated in different species. It is a negative gamma with highest density close to  $s = 0$ . Most mutations are close to neutrality ( $s < 0.01$ ) with some mutations having mild effects  $0.01 < s < 0.1$  and rare lethal mutations ( $s > 0.1$ ).
- The distribution of dominance effects  $h$  is less well known. Convention is that  $h \propto 1/s$  with  $\bar{h} \simeq 0.37$ , and an exponential relationship:  $h = \exp(-ks)/2$ , with  $k = -\log(2\bar{h})/\bar{s} \Rightarrow$  mutations are recessive.

## Shape of the DFE:



$N*s$ , scaled fitness effect of a new mutation  
 $\beta$  = shape of the Gamma distribution.

Chen, Bataillon, Glémin, Lascoux, 2022, *New Phytologist*

# Selection on deleterious mutations

## The delet trait in Nemo:

### Parameters

```
# mandatory:
delet_loci                L
delet_mutation_rate       1e-4
(delet_backmutation_rate  1e-5 # < than mut rate)

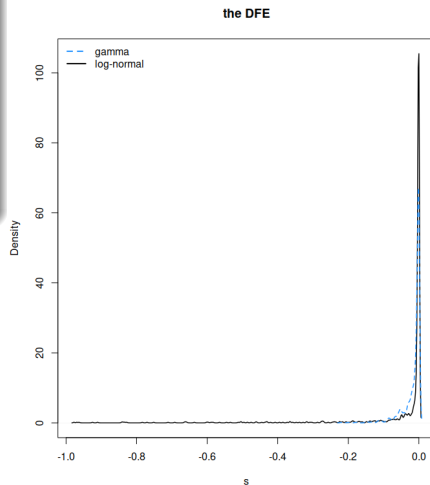
# mutation effects; the DFE:
delet_effects              {{hs: array size L}{s: array size L}}
# or
delet_effects_distribution gamma (constant | lognormal | exponential)
delet_effects_mean         s_bar
(delet_effect_dist_param1  shape)
(delet_effect_dist_param2  scale)
(delet_dominance_mean      h_bar)
...
```

# Parametrization of deleterious mutations: DFE

## Example

```
delet_loci                L
delet_mutation_rate        1e-4
delet_backmutation_rate    1e-6
# the DFE:
delet_effects_distribution gamma
delet_effects_mean          s_bar
delet_effect_dist_param1    shape (beta)
delet_dominance_mean        h_bar
```

## Shape of the DFE:



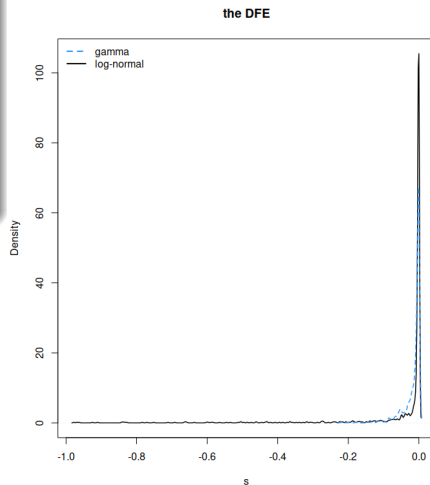
# Parametrization of deleterious mutations: DFE

## Example

```
delet_loci                L
delet_mutation_rate       1e-4
delet_backmutation_rate   1e-6
# the DFE:
delet_effects_distribution gamma
delet_effects_mean        s_bar
delet_effect_dist_param1  shape (beta)
delet_dominance_mean      h_bar
```

- genomic mutation rate:  
 $U = [0.1, 1] = 2L\mu$
- $L = 1000$
- mean DFE:  $\bar{s} \approx 0.05$
- shape DFE:  $\beta \approx 0.3$
- dominance:  $\bar{h} = 0.37$ ;  
(internally, we set:  $h_i = \exp(-51.1 s_i)/2$ )

## Shape of the DFE:



# Parametrization of deleterious mutations: DFE

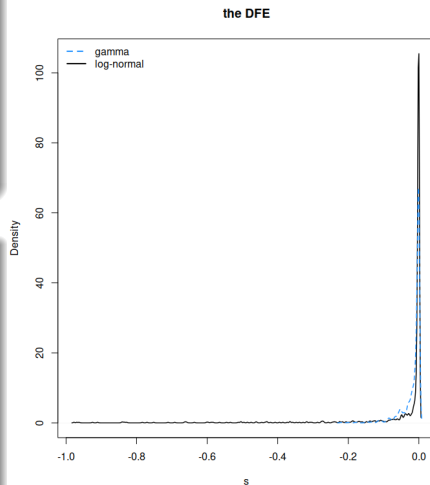
## Negative Gamma DFE

```
delet_loci          1000
delet_mutation_rate 1e-4
delet_backmutation_rate 1e-6
# the DFE:
delet_effects_distribution gamma
delet_effects_mean   0.05
delet_effect_dist_param1 0.3
delet_dominance_mean 0.37
```

## Alternative: negative truncated log-normal

```
delet_loci          1000
delet_mutation_rate 1e-4
delet_backmutation_rate 1e-6
# the DFE:
delet_effects_distribution lognormal
delet_effects_mean   0.05
delet_effect_dist_param1 -6.4
delet_effect_dist_param2 5.3
delet_dominance_mean 0.37
```

## Shape of the DFE:



# Fitness function for the deleterious mutation trait

In most cases, the fitness of an individual is calculated as:

$$w_j = \prod_{i=1}^L (1 - h_i s_i \phi_{i,j}^{het} - s_i \phi_{i,j}^{hom}),$$

with  $h_i$  and  $s_i$  the dominance and selection coefficients at locus  $i$ .  $\phi_{i,j}^{het}$  is 1 if locus  $i$  is heterozygote and 0 otherwise.  $\phi_{i,j}^{hom}$  similarly indicates the homozygosity of the locus.

# Fitness function for the deleterious mutation trait

In most cases, the fitness of an individual is calculated as:

$$w_j = \prod_{i=1}^L (1 - h_i s_i \phi_{i,j}^{het} - s_i \phi_{i,j}^{hom}),$$

with  $h_i$  and  $s_i$  the dominance and selection coefficients at locus  $i$ .  $\phi_{i,j}^{het}$  is 1 if locus  $i$  is heterozygote and 0 otherwise.  $\phi_{i,j}^{hom}$  similarly indicates the homozygosity of the locus.

In a WF pop, **fitness is relative**: the probability to reproduce of an individual is proportional to its fitness. The sum of relative fitness values in a patch is thus 1. It is the default behavior in SLiM. You will obtain it in Nemo with `breed_selection` LCE and `mating_isWrightFisher`.



# Fitness function for the deleterious mutation trait

In most cases, the fitness of an individual is calculated as:

$$w_j = \prod_{i=1}^L (1 - h_i s_i \phi_{i,j}^{het} - s_i \phi_{i,j}^{hom}),$$

with  $h_i$  and  $s_i$  the dominance and selection coefficients at locus  $i$ .  $\phi_{i,j}^{het}$  is 1 if locus  $i$  is heterozygote and 0 otherwise.  $\phi_{i,j}^{hom}$  similarly indicates the homozygosity of the locus.

In a WF pop, **fitness is relative**: the probability to reproduce of an individual is proportional to its fitness. The sum of relative fitness values in a patch is thus 1. It is the default behavior in SLiM. You will obtain it in Nemo with `breed_selection` LCE and `mating_isWrightFisher`.

The consequence of this model of implicit *relative* fitness is that fixed mutations do not contribute to variation in fitness and can thus be removed. It is the default in SLiM but not in Nemo. In Nemo, fixed mutations will contribute to depress mean fitness leading to a **mutation load**.

# Simulating the mutation load in a WF pop

## ① A single WF pop in Nemo

### Ini file

```
patch_number          1
patch_nbfem           1000
patch_nbmal            0

breed_selection        1
aging                  2
save_stats             3
save_files             4

mating_system          6
mating_isWrightFisher
selection_trait         delet
selection_model         direct

delet_loci             1000
delet_mutation_rate     1e-4
delet_backmutation_rate 1e-6
delet_effects_distribution gamma
delet_effects_mean      0.05
delet_effect_dist_param1 0.3
delet_dominance_mean    0.37
delet_recombination_rate 1e-3
```

## ② A single WF pop in SLiM

### Initializations

```
initialize()
{
  initializeMutationRate(1e-4);
  initializeMutationType("m1",0.37,"g",\
                        -0.05,0.3);
  initializeGenomicElementType("g1",m1,1.0);
  initializeGenomicElement(g1, 0, 999);
  initializeRecombinationRate(1e-3);
}
// Create the population
1 early() {
  sim.addSubpop("p1", 1000);
}
...
```

# Exercises

- 1 run the simulations in Nemo and SLiM
  - compare the end results, do they match?
  - look at the evolution of the mutation load through time, do we reach an equilibrium? (graph the allele frequency and the mean fitness)
- 2 compare the results from the gamma and log-normal DFEs (Nemo only)
- 3 (optional) craft a specific DFE with three classes of mutations:
  - 72% are of small effect (mean=0.01,shape=0.6),
  - 25% are of mild effect (mean=0.1,shape=4),
  - and 3% are recessive lethal ( $s=1, h=0$ );
  - save these values in a separate file (e.g., "myDFE.txt") and feed it to Nemo as `delet_effects &myDFE.txt`. Note that the parameter expects to get the values of  $h$ s and  $s$  in a  $2 \times L$  matrix.

# Selection in non-WF pop

## Demographic consequences of the mutation load.

- We will keep fitness absolute such that population size depends on mean fitness:

$$N_{off}^* = survival_{off} * N_{off},$$

where *survival* is the mean fitness of the offspring right after reproduction and  $N_{off}$  depends on mean fecundity  $\bar{f}$  and number of reproducing females  $N_{adlt}$ .

# Selection in non-WF pop

## Demographic consequences of the mutation load.

- We will keep fitness absolute such that population size depends on mean fitness:

$$N_{off}^* = survival_{off} * N_{off},$$

where *survival* is the mean fitness of the offspring right after reproduction and  $N_{off}$  depends on mean fecundity  $\bar{f}$  and number of reproducing females  $N_{adlt}$ . Alternatively, *fecundity* can depend on mean fitness, such that:

$$N_{off}^* = \bar{f} * fitness_{adlt} * N_{adlt}$$

The *effective fecundity* is then  $\bar{f} * fitness_{adlt}$ , with  $\bar{f}$  set by the `mean_fitness` parameter of the `breed` LCE.

# Selection in non-WF pop

## Demographic consequences of the mutation load.

- We will keep fitness absolute such that population size depends on mean fitness:

$$N_{off}^* = survival_{off} * N_{off},$$

where *survival* is the mean fitness of the offspring right after reproduction and  $N_{off}$  depends on mean fecundity  $\bar{f}$  and number of reproducing females  $N_{adlt}$ . Alternatively, *fecundity* can depend on mean fitness, such that:

$$N_{off}^* = \bar{f} * fitness_{adlt} * N_{adlt}$$

The *effective fecundity* is then  $\bar{f} * fitness_{adlt}$ , with  $\bar{f}$  set by the `mean_fitness` parameter of the `breed` LCE. It is faster to simulate fitness-dependent fecundity because less offspring are produced. It is provided by the `breed_selection` LCE.

# Selection in non-WF pop

## Demographic consequences of the mutation load.

- We will keep fitness absolute such that population size depends on mean fitness:

$$N_{off}^* = survival_{off} * N_{off},$$

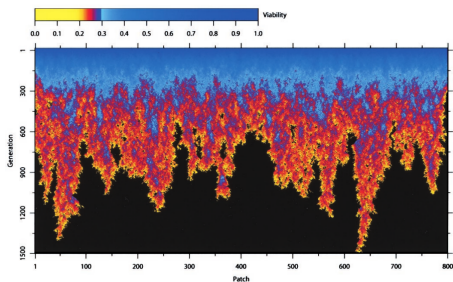
where *survival* is the mean fitness of the offspring right after reproduction and  $N_{off}$  depends on mean fecundity  $\bar{f}$  and number of reproducing females  $N_{adlt}$ . Alternatively, *fecundity* can depend on mean fitness, such that:

$$N_{off}^* = \bar{f} * fitness_{adlt} * N_{adlt}$$

The *effective fecundity* is then  $\bar{f} * fitness_{adlt}$ , with  $\bar{f}$  set by the `mean_fitness` parameter of the `breed` LCE. It is faster to simulate fitness-dependent fecundity because less offspring are produced. It is provided by the `breed_selection` LCE.

- A model where population size depends on average fitness is a **hard selection** model. It is said to be **soft selection** otherwise.

# Hard selection and mutational meltdown

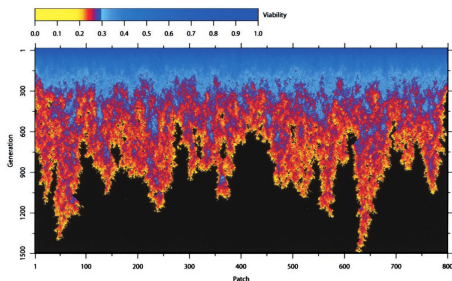


## Meltdown in Nemo

```
breed_selection      1
disperse             2
aging                3
save_stats           4
save_files           5
#selection on fecundity:
breed_selection_fecundity_fitness
```



# Hard selection and mutational meltdown



## Meltdown in Nemo

```
breed_selection      1
disperse             2
aging                3
save_stats           4
save_files           5
#selection on fecundity:
breed_selection_fecundity_fitness
```

## Exercise:

- 128 patches,  $N=64$ , 2 sex
- random mating
- mean fecundity = 5
- DFE:  
 $\mu=5e-4$ ,  $L=1000$  ( $U=1$ )  
constant  $s=0.01$ ,  $h=0.37$   
recombination rate= $1e-3$
- Dispersal:  
8x16 lattice as a torus  
range = 8 adjacent patches  
rate = explore values
- Stats:  
delet stats  
extinction rate  
fitness  
fitness per patch

# Consequences of the mutational load

## Inbreeding depression

One key consequence of the presence of deleterious mutations in our genomes is **inbreeding depression**, the decrease of fitness in inbred individuals relative to non-inbred individuals.

# Consequences of the mutational load

## Inbreeding depression

One key consequence of the presence of deleterious mutations in our genomes is **inbreeding depression**, the decrease of fitness in inbred individuals relative to non-inbred individuals.

$$\delta = 1 - \frac{W_I}{W_O}$$
$$\Rightarrow \delta \approx 0 \quad \text{if } W_I \approx W_O$$

# Consequences of the mutational load

## Inbreeding depression

One key consequence of the presence of deleterious mutations in our genomes is **inbreeding depression**, the decrease of fitness in inbred individuals relative to non-inbred individuals.

$$\delta = 1 - \frac{W_I}{W_O}$$
$$\Rightarrow \delta \approx 0 \quad \text{if } W_I \approx W_O$$

- **Inbreeding** is the increase of homozygosity in the genome caused by genetic drift in small populations or by mating among related individuals. Thus, inbreeding will depend on the **mating system**.

# Consequences of the mutational load

## Inbreeding depression

One key consequence of the presence of deleterious mutations in our genomes is **inbreeding depression**, the decrease of fitness in inbred individuals relative to non-inbred individuals.

$$\delta = 1 - \frac{W_I}{W_O}$$
$$\Rightarrow \delta \approx 0 \quad \text{if } W_I \approx W_O$$

- **Inbreeding** is the increase of homozygosity in the genome caused by genetic drift in small populations or by mating among related individuals. Thus, inbreeding will depend on the **mating system**.
- Inbreeding depression can be evaluated by comparing the fitness of related and unrelated individuals. This is easily done in Nemo.

# Consequences of the mutational load

## Inbreeding depression

One key consequence of the presence of deleterious mutations in our genomes is **inbreeding depression**, the decrease of fitness in inbred individuals relative to non-inbred individuals.

$$\delta = 1 - \frac{W_I}{W_O}$$
$$\Rightarrow \delta \approx 0 \quad \text{if } W_I \approx W_O$$

- **Inbreeding** is the increase of homozygosity in the genome caused by genetic drift in small populations or by mating among related individuals. Thus, inbreeding will depend on the **mating system**.
- Inbreeding depression can be evaluated by comparing the fitness of related and unrelated individuals. This is easily done in Nemo.
- Inbreeding depression is a key driver of the evolution of social behavior, dispersal, mating systems and is believed to play a key role in speciation.

# Exercises: inbreeding depression

- 1 Look at the fitness results of the meltdown simulations obtained with the `adlt.fitness` recorder. Compare the fitness of the different pedigree classes (`fitness.outb/.outw/.hsib/.fsib`). "outb" = outbred between patch (one parent is an immigrant), "outw" = outbred within patch, "hsib" = parents are half-sib, "fsib" = parents are full-sib.
  - Do you observe a difference between the most inbred (`fsib`) and most outbred (`outb`) categories?
  - Calculate inbreeding depression  $\delta$  using `fitness.fsib` ( $W_I$ ) and `fitness.outw` ( $W_O$ ).
  - Similarly, calculate **heterosis** by comparing `fitness.outw` and `fitness.outb`. You can also calculate the average within patch fitness for  $W_I$  and `fitness.outb` for  $W_O$ .
  - You may want to run multiple replicates of one simulation to estimate an average  $\delta$  across replicates, with its variation.

# Exercises: inbreeding depression

- 2 Simulate a species with a mixed mating system to vary inbreeding:
- keep same population structure and DFE as in the meltdown simulations
  - increase inbreeding with a mix of outcrossing and self-fertilization (mating system 4)
  - add a few neutral markers to monitor inbreeding with the  $F_{IS}$  and  $H_o$  coefficients

## Partial selfing:

```
#we need hermaphrodites
patch_nbfem      64
patch_nbmal      0
mating_system    4
mating_proportion 0.05 #selfing rate
mean_fecundity   5
```

## Neutral markers ( $\mu$ -sat):

```
ntrl_loci          20
ntrl_all           256
ntrl_mutation_model 2
ntrl_mutation_rate 1e-4
#unlinked loci, also from delet loci
ntrl_recombination_rate 0.5
#record Ho and Fis (and Fst etc.)
stat adlt.fstat
```



# Exercises: inbreeding depression avoidance

## Model the evolution of dispersal as an inbreeding avoidance strategy

You will now use two new components, the `disperse_evoldisp` LCE and the `disp` trait. The trait encodes a quantitative trait affecting the dispersal behavior of the individuals. The phenotype is the dispersal rate, or probability that an individual leaves its patch.

The `disperse_evoldisp` LCE inherits the parameters from the `disperse` LCE and adds the `dispersal_cost` parameter to model mortality rate during dispersal.

In this exercise, you will evaluate how dispersal evolves as a function of:

- dispersal costs: `dispersal_cost 0.005 0.05`
- inbreeding levels set by:
  - patch size: `patch_capacity 16 32 64`
  - mating system: random mating, polygyny with 50% random mating  
`mating_system 1 2 ; mating_proportion 0.5 ; mean_fecundity 4`
- the other parameters remain similar to the lattice model for the meltdown simulation (without selection on fecundity).

# Exercise: indirect selection on dispersal

## Model the evolution of dispersal without inbreeding

Dispersal is known to evolve also in the absence of inbreeding load, as a *kin competition avoidance mechanism*. By dispersing to another patch, an individual will not compete with its brothers and sisters, thus alleviating competition for reproduction in its natal patch. They will indirectly benefit from less kin competition.

# Exercise: indirect selection on dispersal

## Model the evolution of dispersal without inbreeding

Dispersal is known to evolve also in the absence of inbreeding load, as a *kin competition avoidance mechanism*. By dispersing to another patch, an individual will not compete with its brothers and sisters, thus alleviating competition for reproduction in its natal patch. They will indirectly benefit from less kin competition.

To model this, we do not need to define an explicit fitness function. The dispersal trait will evolve as an *emerging property* of the system. However, mathematical models of the system with an explicit expression of the selection pressure acting on the dispersal trait is possible.

# Exercise: indirect selection on dispersal

## Model the evolution of dispersal without inbreeding

Dispersal is known to evolve also in the absence of inbreeding load, as a *kin competition avoidance mechanism*. By dispersing to another patch, an individual will not compete with its brothers and sisters, thus alleviating competition for reproduction in its natal patch. They will indirectly benefit from less kin competition.

To model this, we do not need to define an explicit fitness function. The dispersal trait will evolve as an *emerging property* of the system. However, mathematical models of the system with an explicit expression of the selection pressure acting on the dispersal trait is possible.

Nemo parameters:

- Remove the `delet` trait and replace the `breed_selection` LCE with `breed`.
- Compare the male and female dispersal rates after 5000 generations with:

```
patch_capacity 8 64 128
```

```
dispersal_model 1 island model
```

```
dispersal_cost 0.01 0.1
```

```
mating_system 1 (random mating) (optionally, compare with polygyny)
```