Module 1

Intro to Nemo

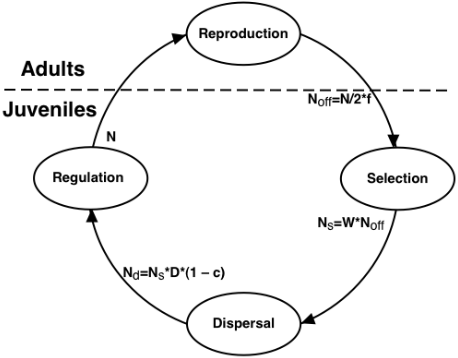
OUTLINE

- Overview of NEMO – Simulation components
- User's interface: the parameter file & its syntax
- Population component
- Representation of space in Nemo
- Life cycle events (LCE)
- Traits, and their genetics
- References
- Resources

Overview of Nemo's features

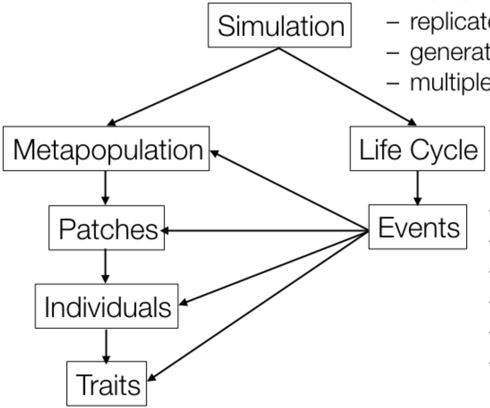
Population	Spatially explicit meta-population framework Discrete space, ind. move between patches $\rightarrow$ space modeled via the migration matrix
Mating systems	Random mating/promiscuity, monogamy, polygyny, selfing, cloning, hermaphroditic, assortative
Life Cycle	Complete flexibility in the order of the events
Traits & genetics	• neutral markers (μ-satellites, SNP) • deleterious mutations (di-allelic) • BDM incompatibility loci • quantitative traits: QTL with pleiotropy (*n*-traits)
Selection	spatially & temporally varying, multi-trait fitness
Demography	stochastic, stage-structured, d.d. regulation, fitness dpdt, etc.

Nemo: Life Cycle



- breed
- disperse
- selection
- aging
- extinction/harvesting
- cloning
- crossing (on a pedigree)
- mutator
- stats & files I/O

Simulation Components:



- file name (simulation name)
- replicates
- generations
- multiple parameter sets

- mating/breeding
- dispersal
- selection/survival
- aging
- extinction
- etc.

Nemo: components

Simulation Components:

- Each component defines a set of parameters
- A component is included in a simulation only if its parameters are set
- Two components are mandatory: `simulation` and `population`
- A minimum of one life cycle event (LCE) is required
- Individuals without trait (no genetics) can be simulated

Nemo: components

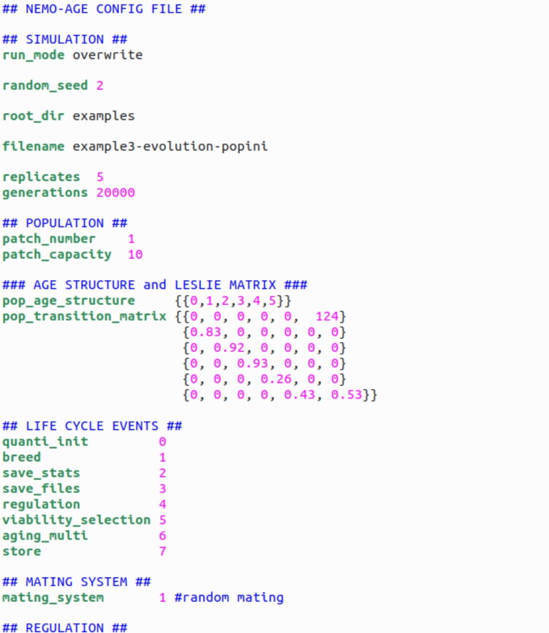
Simulation Components:

- Each component defines a set of parameters
- A component is included in a simulation only if its parameters are set
- Two components are mandatory: `simulation` and `population`
- A minimum of one life cycle event (LCE) is required
- Individuals without trait (no genetics) can be simulated

Program interface

A simple text file with the simulation's parameters and values

Nemo interface: a simple parameter file



```
## NEMO-AGE CONFIG FILE ##

## SIMULATION ##
run_mode overwrite

random_seed 2

root_dir examples

filename example3-evolution-popini

replicates  5
generations 20000

## POPULATION ##
patch_number    1
patch_capacity  10

### AGE STRUCTURE and LESLIE MATRIX ###
pop_age_structure     {{0,1,2,3,4,5}}
pop_transition_matrix {{0, 0, 0, 0, 0,  124}
                        {0.83, 0, 0, 0, 0, 0}
                        {0, 0.92, 0, 0, 0, 0}
                        {0, 0, 0.93, 0, 0, 0}
                        {0, 0, 0, 0.26, 0, 0}
                        {0, 0, 0, 0, 0.43, 0.53}}

## LIFE CYCLE EVENTS ##
quanti_init         0
breed               1
save_stats          2
save_files          3
regulation          4
viability_selection 5
aging_multi         6
store               7

## MATING SYSTEM ##
mating_system       1 #random mating

## REGULATION ##
```

Syntax of the parameter file

Basic structure of the parameter file:

→ a list of `parameter argument` pairs, separated by at least one space

→ each pair on a single line.

Parameter specification:

```
replicates 10
generations 100000
```

Comments:

```
# this is a one-line comment
#/ this is the start of a block comment
text within a block comment
text within a block comment
text within a block comment
/# this is the end of a block comment
```

Syntax of the parameter file: matrix

matrix arguments: comma separated elements within curly brackets

```
matrix_param {{row 1}{row 2}{col1, col2, col3}{m41,m42,m43}}

dispersal_matrix {{0.9,0.05,0.05}{0.05,0.9,0.05}{0.05,0.05,0.9}}
```

Syntax of the parameter file: matrix

matrix arguments: comma separated elements within curly brackets

```
matrix_param {{row 1}{row 2}{col1, col2, col3}{m41,m42,m43}}

dispersal_matrix {{0.9,0.05,0.05}{0.05,0.9,0.05}{0.05,0.05,0.9}}
```

matrix arguments can be split across multiple lines:

```
dispersal_matrix {{0.9,0.05,0.05}
                  {0.05,0.9,0.05}
                  {0.05,0.05,0.9}}
```

Syntax of the parameter file: types

Numbers

```
dispersal_rate       .001
delet_mutation_rate  2.35e-7
generations          1e5
```

Syntax of the parameter file: types

Numbers

```
dispersal_rate       .001
delet_mutation_rate  2.35e-7
generations          1e5
```

Booleans

Works on presence/absence basis: presence = TRUE, absence = FALSE. We add a parameter to select (turn-on) a feature.

```
delet_save_genotypes
quanti_freq_output
```

Syntax of the parameter file: types

Numbers

```
dispersal_rate       .001
delet_mutation_rate  2.35e-7
generations          1e5
```

Booleans

Works on presence/absence basis: presence = TRUE, absence = FALSE. We add a parameter to select (turn-on) a feature.

```
delet_save_genotypes
quanti_freq_output
```

Strings (un-quoted)

```
filename     10pop-N1000-disp0.1-mut0002
root_dir     simProjectA/model1
```

Syntax of the parameter file: macros

Macros are small functions that help write parameter arguments:

repeat: `rep(x, n, each=1, sep=",")`

```
quanti_random_genetic_map  {{rep(10,5)}}
quanti_pleio_matrix {rep("{0,1}{1,0}", 3, sep="")}
```

results in:

```
quanti_random_genetic_map  {{10,10,10,10,10}}
quanti_pleio_matrix {{0,1}{1,0}{0,1}{1,0}{0,1}{1,0}}
```

Syntax of the parameter file: macros

Macros are small functions that help write parameter arguments:

repeat: `rep(x, n, each=1, sep=",")`

```
quanti_random_genetic_map  {{rep(10,5)}}
quanti_pleio_matrix {rep("{0,1}{1,0}", 3, sep="")}
```

results in:

```
quanti_random_genetic_map  {{10,10,10,10,10}}
quanti_pleio_matrix {{0,1}{1,0}{0,1}{1,0}{0,1}{1,0}}
```

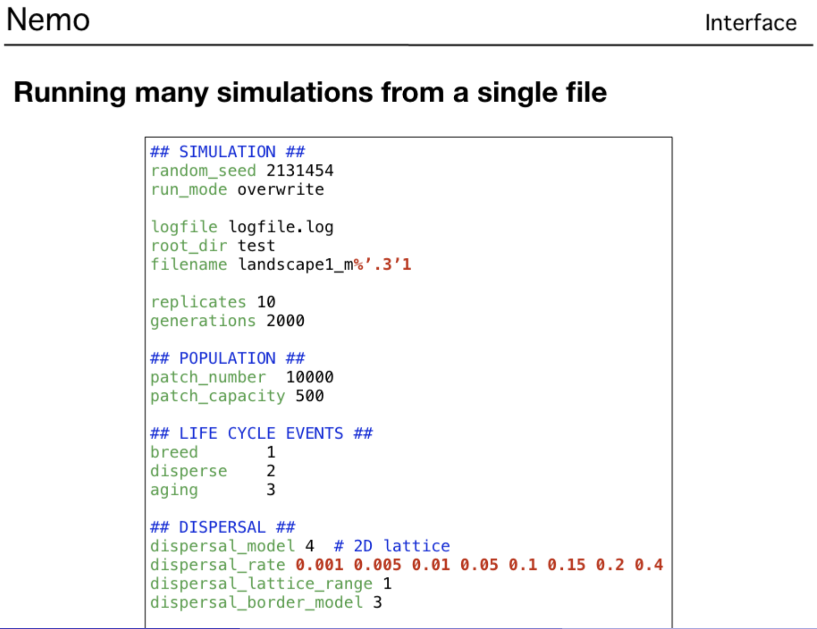
sequence: `seq(from, to, by, sep=",")`

```
quanti_genetic_map  {{seq(1,50,5)}}
dispersal_rate      seq(0.01,0.1,0.01,sep=" ")
```

results in:

```
quanti_genetic_map  {{1,6,11,16,21,26,31,36,41,46}}
dispersal_rate      0.01 0.02 0.03 0.04 0.05 0.06 ... 0.1
```

Sequencial parameters

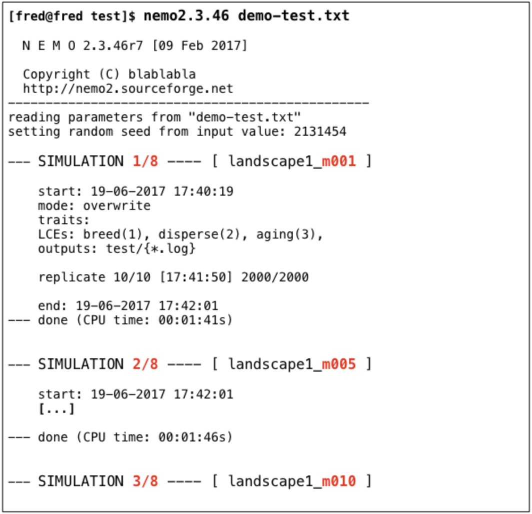


Nemo Interface

Running many simulations from a single file

```
## SIMULATION ##
random_seed 2131454
run_mode overwrite

logfile logfile.log
root_dir test
filename landscape1_m%’.3’1

replicates 10
generations 2000

## POPULATION ##
patch_number  10000
patch_capacity 500

## LIFE CYCLE EVENTS ##
breed       1
disperse    2
aging       3

## DISPERSAL ##
dispersal_model 4  # 2D lattice
dispersal_rate 0.001 0.005 0.01 0.05 0.1 0.15 0.2 0.4
dispersal_lattice_range 1
dispersal_border_model 3
```

Running many simulations from a single file



```
[fred@fred test]$ nemo2.3.46 demo-test.txt

  N E M O 2.3.46r7 [09 Feb 2017]

  Copyright (C) blablabla
  http://nemo2.sourceforge.net
------------------------------------------------
reading parameters from "demo-test.txt"
setting random seed from input value: 2131454

--- SIMULATION 1/8 ---- [ landscape1_m001 ]

    start: 19-06-2017 17:40:19
    mode: overwrite
    traits:
    LCEs: breed(1), disperse(2), aging(3),
    outputs: test/{*.log}

    replicate 10/10 [17:41:50] 2000/2000

    end: 19-06-2017 17:42:01
--- done (CPU time: 00:01:41s)


--- SIMULATION 2/8 ---- [ landscape1_m005 ]

    start: 19-06-2017 17:42:01
    [...]

--- done (CPU time: 00:01:46s)


--- SIMULATION 3/8 ---- [ landscape1_m010 ]
```

Syntax of the parameter file

temporal arguments:

```
(@g0 initial_value, @g123 new_value1, @g456 new_value2)
```

Temporal arguments must be within `'()'`, comma-separated, and start with `@g0` for initialization at the first generation.

external file argument: `&path/to/file.txt`

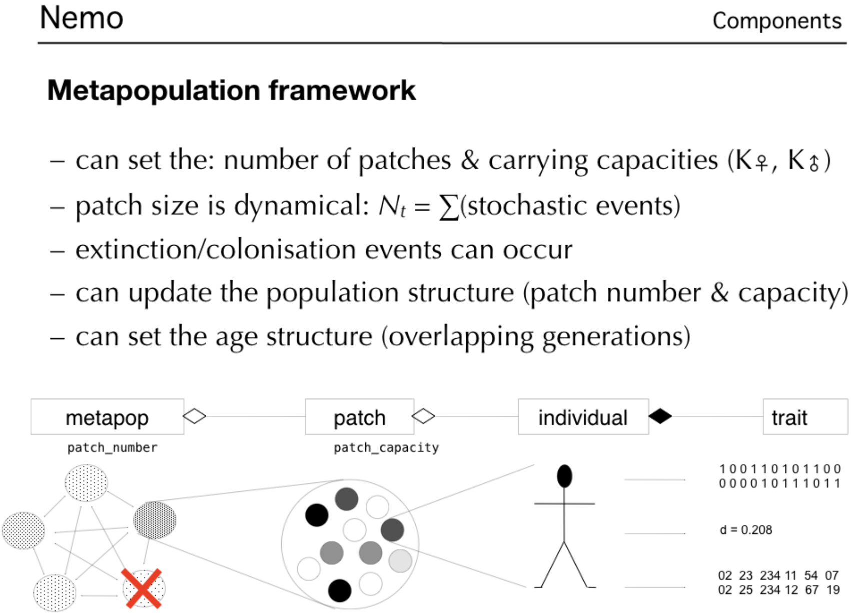
The argument value in the file must be specified with the same format as if it were provided in the parameter file itself.

line continuation: character `\`

Arguments can spread over multiple lines when using the 'escape' character \. Example:

```
(@g0 initial_value, @g123 new_value1, @g456 new_value2, \
@g789 new_value3)
```

Population component

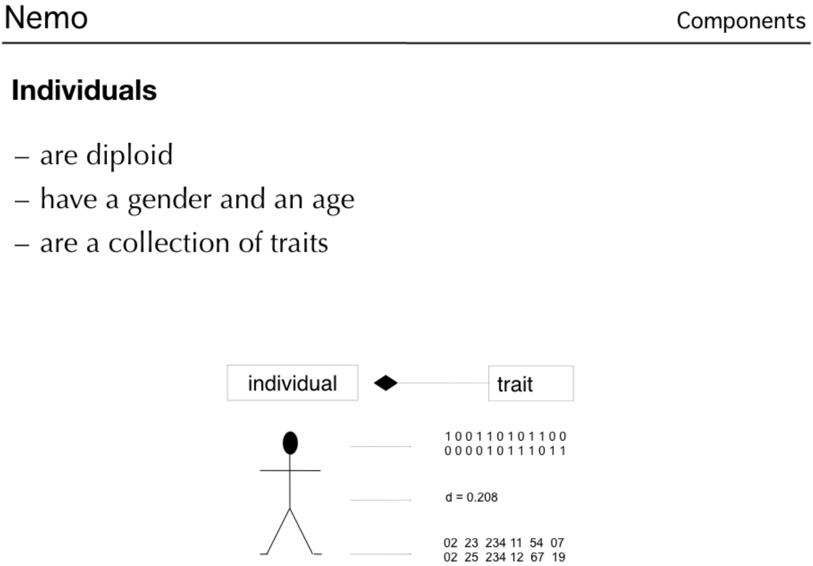


Nemo Components

Metapopulation framework

- can set the: number of patches & carrying capacities (K♀, K♂)
- patch size is dynamical: $N_t = \sum(\text{stochastic events})$
- extinction/colonisation events can occur
- can update the population structure (patch number & capacity)
- can set the age structure (overlapping generations)

Population component



Nemo

Components

Individuals

- are diploid
- have a gender and an age
- are a collection of traits

Population component

Population parameters

```
# setting 1:
patch_number      128
patch_capacity    1000  # will fill patches with 500 females and 500 males

#setting 2:
patch_number      128
patch_nbfem       1000 # only females if we model hermaphrodites (see mating system 4
patch_nbmal       0

#setting 3:
patch_capacity    {{10,0,23,50,1298,5,0,0,0,1000}} #patch_number = array size = 10
```

The parameter `patch_capacity` sets the *maximum patch size*, or its *carrying capacity K*.

Population component

Population parameters

```
# setting 1:
patch_number      128
patch_capacity    1000  # will fill patches with 500 females and 500 males

#setting 2:
patch_number      128
patch_nbfem       1000 # only females if we model hermaphrodites (see mating system 4
patch_nbmal       0

#setting 3:
patch_capacity    {{10,0,23,50,1298,5,0,0,0,1000}} #patch_number = array size = 10
```

The parameter `patch_capacity` sets the *maximum patch size*, or its *carrying capacity* K.

If set to 0, the patch will be empty after *regulation* (see the `aging` LCE).

Population component

Population parameters

```
# setting 1:
patch_number     128
patch_capacity   1000  # will fill patches with 500 females and 500 males

#setting 2:
patch_number     128
patch_nbfem      1000 # only females if we model hermaphrodites (see mating system 4
patch_nbmal      0

#setting 3:
patch_capacity   {{10,0,23,50,1298,5,0,0,0,1000}} #patch_number = array size = 10
```

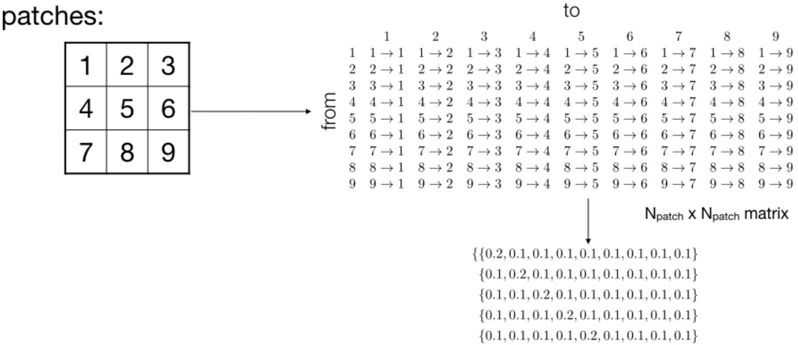
The parameter `patch_capacity` sets the *maximum patch size*, or its *carrying capacity* K.

If set to 0, the patch will be empty after *regulation* (see the `aging` LCE).

Useful when one wants to “open” the patch at a later generation with a temporal arguments.

Dispersal and representation of Space

- spatially explicit, but no explicit x-y coordinates
- space abstracted into a migration matrix
- reduced (connectivity) matrices allow simulation of large landscapes



patches:

1	2	3
4	5	6
7	8	9

to

from

	1	2	3	4	5	6	7	8	9
1	1 → 1	1 → 2	1 → 3	1 → 4	1 → 5	1 → 6	1 → 7	1 → 8	1 → 9
2	2 → 1	2 → 2	2 → 3	2 → 4	2 → 5	2 → 6	2 → 7	2 → 8	2 → 9
3	3 → 1	3 → 2	3 → 3	3 → 4	3 → 5	3 → 6	3 → 7	3 → 8	3 → 9
4	4 → 1	4 → 2	4 → 3	4 → 4	4 → 5	4 → 6	4 → 7	4 → 8	4 → 9
5	5 → 1	5 → 2	5 → 3	5 → 4	5 → 5	5 → 6	5 → 7	5 → 8	5 → 9
6	6 → 1	6 → 2	6 → 3	6 → 4	6 → 5	6 → 6	6 → 7	6 → 8	6 → 9
7	7 → 1	7 → 2	7 → 3	7 → 4	7 → 5	7 → 6	7 → 7	7 → 8	7 → 9
8	8 → 1	8 → 2	8 → 3	8 → 4	8 → 5	8 → 6	8 → 7	8 → 8	8 → 9
9	9 → 1	9 → 2	9 → 3	9 → 4	9 → 5	9 → 6	9 → 7	9 → 8	9 → 9

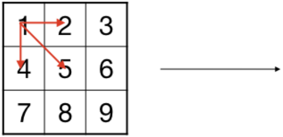
N_{patch} x N_{patch} matrix

$\{\{0.2, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1\}$
$\{0.1, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1\}$
$\{0.1, 0.1, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1\}$
$\{0.1, 0.1, 0.1, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1\}$
$\{0.1, 0.1, 0.1, 0.1, 0.2, 0.1, 0.1, 0.1, 0.1\}$

Dispersal and representation of Space

- spatially explicit, but no explicit x-y coordinates
- space abstracted into a migration matrix
- reduced (connectivity) matrices allow simulation of large landscapes

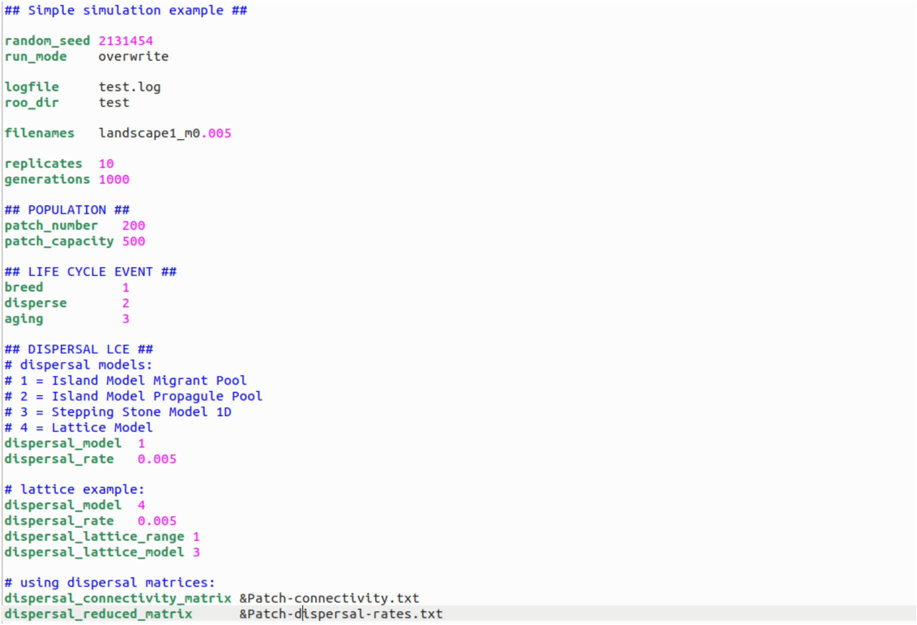
patches:



1	2	3
4	5	6
7	8	9

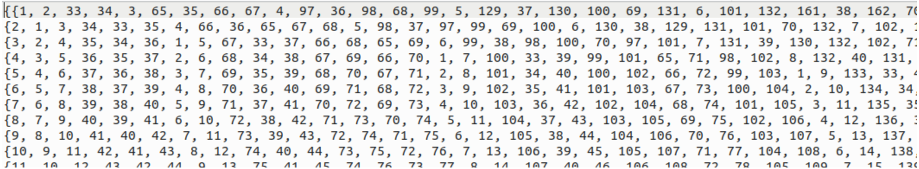
```
connectivity_matrix {
                     {1,2,4,5}
                     {2,1,3,4,5,6}
                     {3,2,5,6}
                     ...
                     {9,5,6,8}
                    }
     reduced_matrix {
                     {0.7,0.1,0.1,0.1}
                     {0.5,0.1,0.1,0.1,0.1,0.1}
                     {0.7,0.1,0.1,0.1}
                     ...
                     {0.7,0.1,0.1,0.1}
                    }
```

N_{patch} rows



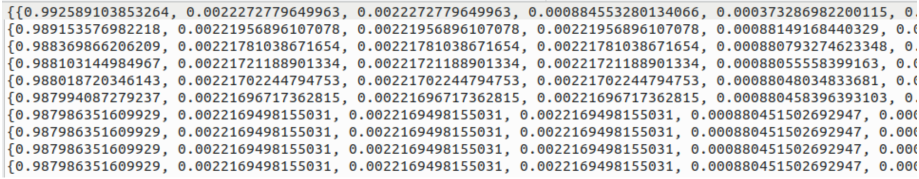
```
## Simple simulation example ##

random_seed 2131454
run_mode    overwrite

logfile     test.log
roo_dir     test

filenames   landscape1_m0.005

replicates  10
generations 1000

## POPULATION ##
patch_number   200
patch_capacity 500

## LIFE CYCLE EVENT ##
breed          1
disperse       2
aging          3

## DISPERSAL LCE ##
# dispersal models:
# 1 = Island Model Migrant Pool
# 2 = Island Model Propagule Pool
# 3 = Stepping Stone Model 1D
# 4 = Lattice Model
dispersal_model  1
dispersal_rate   0.005

# lattice example:
dispersal_model  4
dispersal_rate   0.005
dispersal_lattice_range 1
dispersal_lattice_model 3

# using dispersal matrices:
dispersal_connectivity_matrix &Patch-connectivity.txt
dispersal_reduced_matrix      &Patch-dispersal-rates.txt
```

dispersal connectivity matrix:



```
{{1, 2, 33, 34, 3, 65, 35, 66, 67, 4, 97, 36, 98, 68, 99, 5, 129, 37, 130, 100, 69, 131, 6, 101, 132, 161, 38, 162, 7
{2, 1, 3, 34, 33, 35, 4, 66, 36, 65, 67, 68, 5, 98, 37, 97, 99, 69, 100, 6, 130, 38, 129, 131, 101, 70, 132, 7, 102,
{3, 2, 4, 35, 34, 36, 1, 5, 67, 33, 37, 66, 68, 65, 69, 6, 99, 38, 98, 100, 70, 97, 101, 7, 131, 39, 130, 132, 102, 7
{4, 3, 5, 36, 35, 37, 2, 6, 68, 34, 38, 67, 69, 66, 70, 1, 7, 100, 33, 39, 99, 101, 65, 71, 98, 102, 8, 132, 40, 131,
{5, 4, 6, 37, 36, 38, 3, 7, 69, 35, 39, 68, 70, 67, 71, 2, 8, 101, 34, 40, 100, 102, 66, 72, 99, 103, 1, 9, 133, 33,
{6, 5, 7, 38, 37, 39, 4, 8, 70, 36, 40, 69, 71, 68, 72, 3, 9, 102, 35, 41, 101, 103, 67, 73, 100, 104, 2, 10, 134, 34
{7, 6, 8, 39, 38, 40, 5, 9, 71, 37, 41, 70, 72, 69, 73, 4, 10, 103, 36, 42, 102, 104, 68, 74, 101, 105, 3, 11, 135, 3
{8, 7, 9, 40, 39, 41, 6, 10, 72, 38, 42, 71, 73, 70, 74, 5, 11, 104, 37, 43, 103, 105, 69, 75, 102, 106, 4, 12, 136,
{9, 8, 10, 41, 40, 42, 7, 11, 73, 39, 43, 72, 74, 71, 75, 6, 12, 105, 38, 44, 104, 106, 70, 76, 103, 107, 5, 13, 137,
{10, 9, 11, 42, 41, 43, 8, 12, 74, 40, 44, 73, 75, 72, 76, 7, 13, 106, 39, 45, 105, 107, 71, 77, 104, 108, 6, 14, 138
{11, 10, 12, 43, 42, 44, 9, 13, 75, 41, 45, 74, 76, 73, 77, 8, 14, 107, 40, 46, 106, 108, 72, 78, 105, 109, 7, 15, 13
```

dispersal rate matrix:



```
{{0.992589103853264, 0.0022272779649963, 0.0022272779649963, 0.000884553280134066, 0.000373286982200115, 0.
{0.989153576982218, 0.00221956896107078, 0.00221956896107078, 0.00221956896107078, 0.00088149168440329, 0.0
{0.988369866206209, 0.0022178103867l654, 0.00221781038671654, 0.00221781038671654, 0.000880793274623348, 0.
{0.988103144984967, 0.00221721188901334, 0.00221721188901334, 0.00221721188901334, 0.000880555558399163, 0.0
{0.988018720346143, 0.00221702244794753, 0.00221702244794753, 0.00221702244794753, 0.000880480348333681, 0.0
{0.987994087279237, 0.00221696717362815, 0.00221696717362815, 0.00221696717362815, 0.000880458396393103, 0.
{0.987986351609929, 0.0022169498155031, 0.0022169498155031, 0.0022169498155031, 0.000880451502692947, 0.000
{0.987986351609929, 0.0022169498155031, 0.0022169498155031, 0.0022169498155031, 0.000880451502692947, 0.000
{0.987986351609929, 0.0022169498155031, 0.0022169498155031, 0.0022169498155031, 0.000880451502692947, 0.000
{0.987986351609929, 0.0022169498155031, 0.0022169498155031, 0.0022169498155031, 0.000880451502692947, 0.000
```

Life-cycle events:

aging: increase the age of the individuals, perform patch regulation

breed: mate and breed, create new offspring generation

breed_wolbachia: breed and *Wolbachia* transmission/infection

cross: perform a half-sib, full-sib mating design (NCI)

disperse: offspring dispersal

disperse_evoldisp: offspring dispersal with evolving dispersal rates

extinction: random patch extinction or harvesting

regulation: patch regulation (to carrying capacity)

resize: modify population size (patch number and/or size)

save_files: write output files to disk

save_stats: record statistics

store: save simulation data to binary files

selection: perform viability selection on the offspring generation

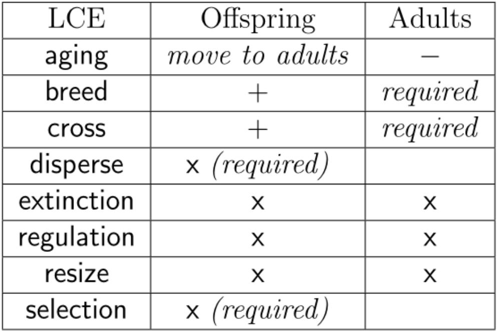
breed_disperse: breed with backward migration (Wright-Fisher model)

breed_selection: breed with selection (faster)

breed_selection_disperse: all in one (Wright-Fisher with selection)

Life-cycle events:

→ effect on population composition:



LCE	Offspring	Adults
aging	*move to adults*	–
breed	+	*required*
cross	+	*required*
disperse	x *(required)*	
extinction	x	x
regulation	x	x
resize	x	x
selection	x *(required)*	

Table 4.1 *in* Nemo-manual p.33

Ordering of the LCEs

The LCEs can be set in *almost* any order in the life cycle. Some dependencies exist because the LCEs change the content of the age classes.

This works

```
breed                1 # adds offspring
viability_selection  2
disperse             3
aging                4 #removes offspring
```

This doesn't works

```
breed                1
aging                2 #removes offspring
viability_selection  3 #no offspring
disperse             4 #no offspring
```

This works too

```
breed                1
disperse             2
viability_selection  3  # in arrival patch
aging                4
```

Ordering of the LCEs

The LCEs can be set in *almost* any order in the life cycle. Some dependencies exist because the LCEs change the content of the age classes.

This works

```
breed                1 # adds offspring
viability_selection  2
disperse             3
aging                4 #removes offspring
```

This doesn't works

```
breed                1
aging                2 #removes offspring
viability_selection  3 #no offspring
disperse             4 #no offspring
```

This works too

```
breed                1
disperse             2
viability_selection  3  # in arrival patch
aging                4
```

The minimum life cycle

```
breed                1
aging                2
```

Ordering of the LCEs

The LCEs can be set in *almost* any order in the life cycle. Some dependencies exist because the LCEs change the content of the age classes.

This works

```
breed                1 # adds offspring
viability_selection  2
disperse             3
aging                4 #removes offspring
```

This works too

```
breed                1
disperse             2
viability_selection  3  # in arrival patch
aging                4
```

This doesn't works

```
breed                1
aging                2 #removes offspring
viability_selection  3 #no offspring
disperse             4 #no offspring
```

The minimum life cycle

```
breed                1
aging                2
```

This one is fine too!

```
patch_number        1
patch_capacity      100
source_pop          path/to/sim-P1-N100.bin
## life cycle :
save_stats          1
save_files          2
```

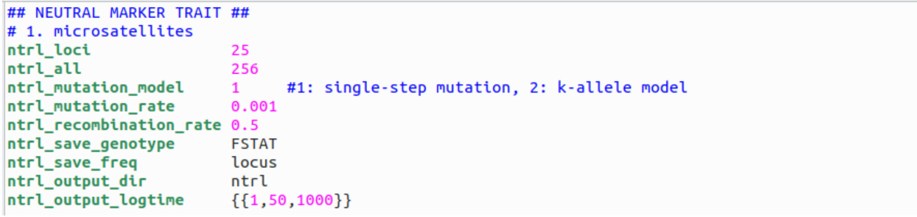
Traits:

sequence-based traits

`ntrl`	:	neutral markers (microsat, SNP)
`delet`	:	deleterious mutations (biallelic loci)
`quanti`	:	quantitative trait loci (continous, biallelic)
`dmi`	:	Dobzhanski-Muller incompatibility loci
`disp`	:	dispersal locus (evolving dispersal rates)
`wolb`	:	*Wolbachia* infection status (endosymbiont with CI)

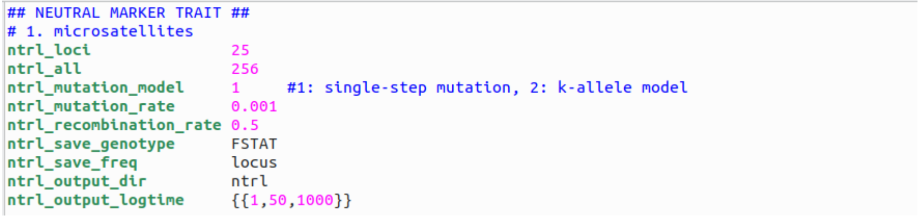
BUT, traits are **optional**, individuals may not carry any trait

Neutral trait `ntrl`

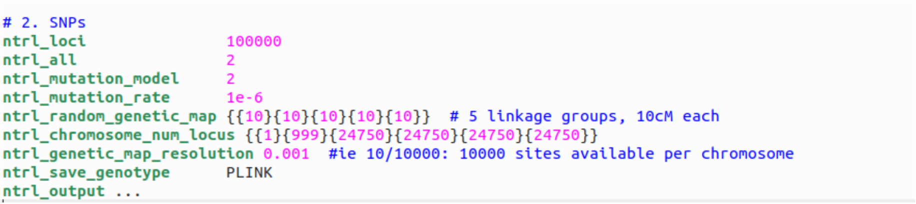


```
## NEUTRAL MARKER TRAIT ##
# 1. microsatellites
ntrl_loci               25
ntrl_all                256
ntrl_mutation_model     1      #1: single-step mutation, 2: k-allele model
ntrl_mutation_rate      0.001
ntrl_recombination_rate 0.5
ntrl_save_genotype      FSTAT
ntrl_save_freq          locus
ntrl_output_dir         ntrl
ntrl_output_logtime     {{1,50,1000}}
```

Neutral trait `ntrl`

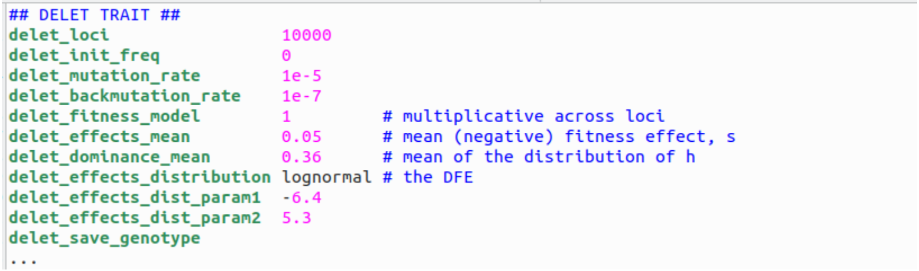


```
## NEUTRAL MARKER TRAIT ##
# 1. microsatellites
ntrl_loci                25
ntrl_all                 256
ntrl_mutation_model      1     #1: single-step mutation, 2: k-allele model
ntrl_mutation_rate       0.001
ntrl_recombination_rate  0.5
ntrl_save_genotype       FSTAT
ntrl_save_freq           locus
ntrl_output_dir          ntrl
ntrl_output_logtime      {{1,50,1000}}
```



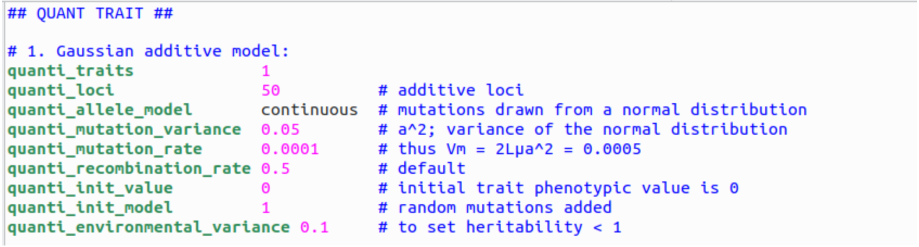
```
# 2. SNPs
ntrl_loci                100000
ntrl_all                 2
ntrl_mutation_model      2
ntrl_mutation_rate       1e-6
ntrl_random_genetic_map {{10}{10}{10}{10}{10}}  # 5 linkage groups, 10cM each
ntrl_chromosome_num_locus {{1}{999}{24750}{24750}{24750}{24750}}
ntrl_genetic_map_resolution 0.001  #ie 10/10000: 10000 sites available per chromosome
ntrl_save_genotype       PLINK
ntrl_output ...
```

Deleterious mutations trait `delet`



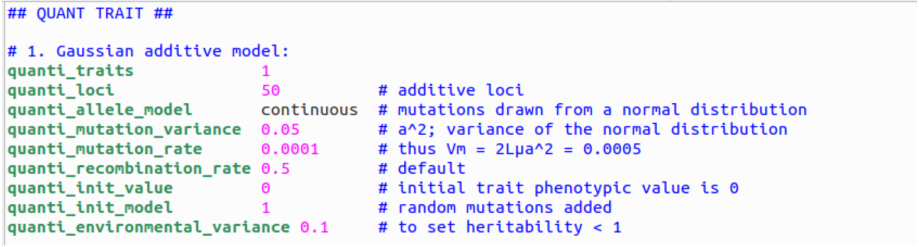
```
## DELET TRAIT ##
delet_loci                 10000
delet_init_freq            0
delet_mutation_rate        1e-5
delet_backmutation_rate    1e-7
delet_fitness_model        1         # multiplicative across loci
delet_effects_mean         0.05      # mean (negative) fitness effect, s
delet_dominance_mean       0.36      # mean of the distribution of h
delet_effects_distribution lognormal # the DFE
delet_effects_dist_param1  -6.4
delet_effects_dist_param2  5.3
delet_save_genotype
...
```

Quantitative traits `quant`

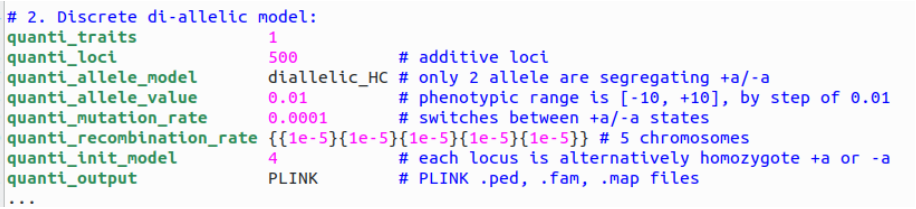


```
## QUANT TRAIT ##

# 1. Gaussian additive model:
quanti_traits             1
quanti_loci               50          # additive loci
quanti_allele_model       continuous  # mutations drawn from a normal distribution
quanti_mutation_variance  0.05        # a^2; variance of the normal distribution
quanti_mutation_rate      0.0001      # thus Vm = 2Lμa^2 = 0.0005
quanti_recombination_rate 0.5         # default
quanti_init_value         0           # initial trait phenotypic value is 0
quanti_init_model         1           # random mutations added
quanti_environmental_variance 0.1     # to set heritability < 1
```

Quantitative traits `quant`

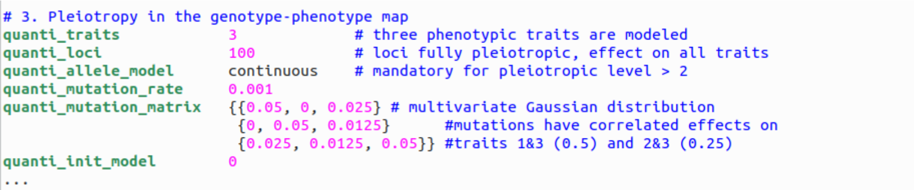


```
## QUANT TRAIT ##

# 1. Gaussian additive model:
quanti_traits                 1
quanti_loci                   50          # additive loci
quanti_allele_model           continuous  # mutations drawn from a normal distribution
quanti_mutation_variance      0.05        # a^2; variance of the normal distribution
quanti_mutation_rate          0.0001      # thus Vm = 2Lµa^2 = 0.0005
quanti_recombination_rate 0.5             # default
quanti_init_value             0           # initial trait phenotypic value is 0
quanti_init_model             1           # random mutations added
quanti_environmental_variance 0.1         # to set heritability < 1
```



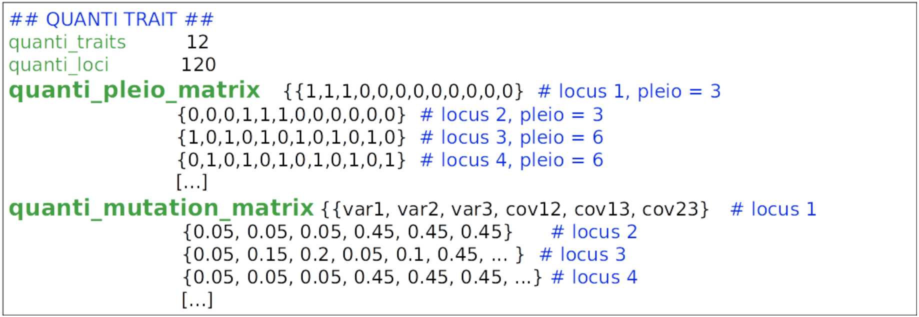
```
# 2. Discrete di-allelic model:
quanti_traits                 1
quanti_loci                   500         # additive loci
quanti_allele_model           diallelic_HC # only 2 allele are segregating +a/-a
quanti_allele_value           0.01        # phenotypic range is [-10, +10], by step of 0.01
quanti_mutation_rate          0.0001      # switches between +a/-a states
quanti_recombination_rate {{1e-5}{1e-5}{1e-5}{1e-5}{1e-5}} # 5 chromosomes
quanti_init_model             4           # each locus is alternatively homozygote +a or -a
quanti_output                 PLINK       # PLINK .ped, .fam, .map files
...
```

Pleiotropic traits with quant



```
# 3. Pleiotropy in the genotype-phenotype map
quanti_traits            3             # three phenotypic traits are modeled
quanti_loci              100           # loci fully pleiotropic, effect on all traits
quanti_allele_model      continuous    # mandatory for pleiotropic level > 2
quanti_mutation_rate     0.001
quanti_mutation_matrix   {{0.05, 0, 0.025} # multivariate Gaussian distribution
                          {0, 0.05, 0.0125}      #mutations have correlated effects on
                          {0.025, 0.0125, 0.05}} #traits 1&3 (0.5) and 2&3 (0.25)
quanti_init_model        0
...
```

Variable pleiotropy in the GP map with `quant`

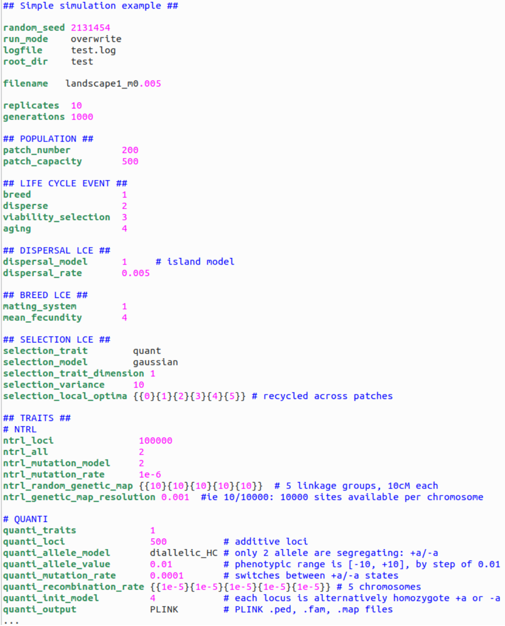


```
## QUANTI TRAIT ##
quanti_traits       12
quanti_loci        120
quanti_pleio_matrix   {{1,1,1,0,0,0,0,0,0,0,0,0} # locus 1, pleio = 3
                  {0,0,0,1,1,1,0,0,0,0,0,0}  # locus 2, pleio = 3
                  {1,0,1,0,1,0,1,0,1,0,1,0}  # locus 3, pleio = 6
                  {0,1,0,1,0,1,0,1,0,1,0,1}  # locus 4, pleio = 6
                  [...]
quanti_mutation_matrix {{var1, var2, var3, cov12, cov13, cov23}   # locus 1
                  {0.05, 0.05, 0.05, 0.45, 0.45, 0.45}      # locus 2
                  {0.05, 0.15, 0.2, 0.05, 0.1, 0.45, ... }  # locus 3
                  {0.05, 0.05, 0.05, 0.45, 0.45, 0.45, ...} # locus 4
                  [...]
```

Jobran Chebib

Availabale in nemo2.4.0!!!

Puting components together



```
## Simple simulation example ##

random_seed 2131454
run_mode    overwrite
logfile     test.log
root_dir    test

filename   landscape1_m0.005

replicates  10
generations 1000

## POPULATION ##
patch_number         200
patch_capacity       500

## LIFE CYCLE EVENT ##
breed                1
disperse             2
viability_selection  3
aging                4

## DISPERSAL LCE ##
dispersal_model      1     # island model
dispersal_rate       0.005

## BREED LCE ##
mating_system        1
mean_fecundity       4

## SELECTION LCE ##
selection_trait        quant
selection_model        gaussian
selection_trait_dimension 1
selection_variance     10
selection_local_optima {{0}{1}{2}{3}{4}{5}} # recycled across patches

## TRAITS ##
# NTRL
ntrl_loci               100000
ntrl_all                2
ntrl_mutation_model     2
ntrl_mutation_rate      1e-6
ntrl_random_genetic_map {{10}{10}{10}{10}{10}}  # 5 linkage groups, 10cM each
ntrl_genetic_map_resolution 0.001  #ie 10/10000: 10000 sites available per chromosome

# QUANTI
quanti_traits             1
quanti_loci               500          # additive loci
quanti_allele_model       diallelic_HC # only 2 allele are segregating: +a/-a
quanti_allele_value       0.01         # phenotypic range is [-10, +10], by step of 0.01
quanti_mutation_rate      0.0001       # switches between +a/-a states
quanti_recombination_rate {{1e-5}{1e-5}{1e-5}{1e-5}{1e-5}} # 5 chromosomes
quanti_init_model         4            # each locus is alternatively homozygote +a or -a
quanti_output             PLINK        # PLINK .ped, .fam, .map files
...
```

Other features

- Store and reload: use *store* LCE to save and load binary files
- Stage-structured populations: NEMO-AGE
- Multi-species simulations: *in development*
- Submit jobs on a cluster from an input file: NEMOSUB
- Parallelization: using MPI version to run replicates in parallel on a cluster

nemosub

nemosub: submitting simulations to a job scheduler

```
## job scheduling parameters ## <-- nemosub
sub_program sbatch
sub_script_directive SBATCH
sub_parameters -mem-per-cpu=16000
sub_script_args nemo3.0.16
sub_jobname camp%’.4'2g%1s%3

## dummy sequence parameters
scenario 0 1 2 3          # %3; env change scenario
grid 1 2 3 4 6 5 6 11     # %1; landscape grid

## NEMO PARAMS ##
filename campanula_mut%'.4'2_g%1_s%3_VS50

## SOURCE POP ##
source_pop campanula_mut%'.4'2_g%1_VS50
source_preserve
[...]

## POPULATION ##
patch_number 1024
patch_nbfem (@g0 &init_K_campanula_%1.txt, @g1 1000)

## RATE ENV CHANGE ##
rate_environmental_change (@g0 &nochange.txt, @g10 &rate_2020_s%3_%1.txt, @g21
&rate_2030_s%3_%1.txt, @g31 &rate_2040_s%3_%1.txt, @g41 [...] )
```

```
[fred@fred test]$ nemosub env-change.txt
```

Nemo: what have we done with it?

- Architecture of adaptive phenotypic divergence

Yeaman & Guillaume 2009 (Evolution); Yeaman & Whitlock 2011 (Evolution); Débarre et al. 2015 (Am Nat); Yeaman 2015 (Am Nat); **with phenotypic plasticity:** Schmidt & Guillaume 2017 (Heredity)

- Multivariate quantitative genetics models with pleiotropy

Guillaume & Whitlock 2007 (Evolution); Guillaume 2011 (Evolution); Guillaume & Otto 2012 (Genetics); Chebib & Guillaume 2017, 2021 (Evolution, Genetics)

- Eco-Evolutionary dynamics in heterogeneous environments

species' range evolution: Gilbert et al. 2017 (Am Nat); Cotto et al. 2017 (Nat Comm); Postuma et al. 2020 (Ecosphere) **with phenotypic plasticity:** Schmidt et al. 2019 (Am Nat)

- Reconstructing patterns of genetic variation from demographic scenarios

Dynamics of deleterious mutations during recolonization: Grossen et al. 2020 (Nat Comm); **Adaptive variation for genomic vulnerability analysis:** Dauphin et al. 2021 (Glob Change Biol)

- Multi-species eco-evo dynamics in changing environments

Species competition in space and time: Gidoin et al. *in prep*

Resources

Nemo:

- homepage: https://nemo2.sourceforge.io
- info: NemoSimul@twitter; https://groups.google.com/g/nemo-announce
- support & discussion: https://groups.google.com/g/nemo-simul
- **Nemo-age** (stage-structured version): https://bitbucket.org/ecoevo/nemo-age-release

Papers:

Guillaume, F. and Rougemont, J. (2006) **Nemo: an evolutionary and population genetics programming framework.** *Bioinformatics* 22:2556-2557

Cotto, O., Schmid, M. and Guillaume, F. (2020) **Nemo-age: spatially explicit simulations of eco-evolutionary dynamics in stage-structured populations under changing environments.** *Methods in Ecology and Evolution* 11:1227-1236

Contributors: Jacques Rougemont, Samuel Neuenschwander, Alistair Blachford, Sam Yeaman, Kim Gilbert, Olivier Cotto, Max Schmid, Jobran Chebib, Champak Beeravolu, Stéphane Chantepie, Cindy Gidoin

Syntax of the parameter file: macros

Dealing with argument strings:

In example `rep("{0,1}{1,0}", 3, sep="")`, we added quotes “ ” to enclose the first argument to *rep()*.

- ⇒ Arguments to a macro must be quoted when they include *delimiters* (commas ‘,’) or *enclosers* (parentheses: ‘()’, ‘{}’).
- ⇒ This is required to not interpret within-string elements as separate arguments to the macro.

Macros can be nested:

Example: `rep(seq(1,5,1), 2)`, but this will not work! Nemo will return:

```
***ERROR*** macro "rep" can have max 4 arguments:  rep(x, n, each=1, sep=",")
```

This error is issued because `rep` was called as: `rep(1,2,3,4,5, 2)`.

- ⇒ Result of `seq(1,5,1)` must be quoted with macro `q()`:

```
rep(q(seq(1,5,1)),2)
```

- ⇒ the call to `rep` becomes: `rep("1,2,3,4,5",2)`