# MetaPopGen 2.0: A multi-locus genetic simulator to model populations of large size

Marco Andrello, Florence Débarre, Stéphanie Manel

## To cite this version:

Marco Andrello, Florence Débarre, Stéphanie Manel. MetaPopGen 2.0: A multi-locus genetic simulator to model populations of large size. Molecular Ecology Resources, 2020, 10.1111/1755-0998.13270 . hal-03002459

## HAL Id: hal-03002459

## https://hal.science/hal-03002459

Submitted on 13 Nov 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DR. MARCO  ANDRELLO (Orcid ID : 0000-0001-7590-2736)

DR. FLORENCE  DEBARRE (Orcid ID : 0000-0003-2497-833X)

PROF. STÉPHANIE  MANEL (Orcid ID : 0000-0001-8902-6052)

Article type      : Resource Article

Title

# MetaPopGen 2.0: a multi-locus genetic simulator to model populations of large size

Authors

Marco Andrello[1*], Christelle Noirot[2], Florence Débarre[3] and Stéphanie Manel[2]

Affiliations

[1] MARBEC, Univ Montpellier, CNRS, Ifremer, IRD, Sète, France.

[2] CEFE, Univ Montpellier, CNRS, EPHE-PSL University, IRD, Univ Paul Valéry Montpellier 3, Montpellier, France

[3] Sorbonne Université, CNRS, INRAE, IRD, UPEC, Univ. de Paris, Institut d'Ecologie et des Sciences de l'Environnement de Paris (iEES-Paris), F75005, Paris, France

* Corresponding author: marco.andrello@gmail.com

Running title
Multi-locus genetic simulator

This article has been accepted for publication and undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the Version of Record. Please cite this article as doi: 10.1111/1755-0998.13270

This article is protected by copyright. All rights reserved

Accepted Article

## Abstract

Multi-locus genetic processes in subdivided populations can be complex and difficult to interpret using theoretical population genetics models. Genetic simulators offer a valid alternative to study multi-locus genetic processes in arbitrarily complex scenarios. However, the use of forward-in-time simulators in realistic scenarios involving high numbers of individuals distributed in multiple local populations is limited by computation time and memory requirements. These limitations increase with the number of simulated individuals. We developed a genetic simulator, METAPOPGEN 2.0, to model multi-locus population genetic processes in subdivided populations of arbitrarily large size. It allows for spatial and temporal variation in demographic parameters, age structure, adult and propagule dispersal, variable mutation rates and selection on survival and fecundity. We developed METAPOPGEN 2.0 in the R environment to facilitate its use by non-modeler ecologists and evolutionary biologists. We illustrate the capabilities of METAPOPGEN 2.0 for studying adaptation to water salinity in the striped red mullet *Mullus surmuletus*.

Keywords

Adaptation. Connectivity. Dispersal. Genetic simulator. Landscape genetics.

Accepted Article

This article is protected by copyright. All rights reserved

## 1 Introduction

The study of the effects of recombination, selection and drift in a single population in a multi-locus context has received considerable attention in the literature, while multi-locus models to study these micro-evolutionary processes in subdivided populations are rarer (see Bürger, 2019 for a recent review). This is partly due to the complexity of general multi-locus models of selection with migration between multiple demes, which make their interpretation prohibitively difficult, especially for population geneticists without a formal mathematical background. The popularization of personal computers during the last three decades and the continuous increase in computing power have facilitated the emergence of simulation approaches to population genetics problems and the development of dozens of simulation models (reviewed in Hoban, Bertorelle, & Gaggiotti, 2012). Genetic simulators are complementary to mathematical models and can help circumvent the difficulties posed by the mathematical complexity of multi-locus genetic models in subdivided populations. They are increasingly used to investigate basic and applied questions in molecular ecology (Hoban 2014).

Many of these simulators can handle multi-locus systems in subdivided populations using backward-in-time (i.e. coalescent) or forward-in-time simulation approaches. Backward-in-time simulators are generally faster, but preclude life history modelling and are therefore limited to situations in which deviations from the reproductive scheme assumed by the Wright–Fisher model are minor (Hoban 2014). Forward-in-time simulators can model more complex situations, making them more suited to predictive studies at a short timescale. EasyPop (Balloux 2001) and SimuPop (Peng & Kimmel 2005; Peng & Amos 2008) were among the first forward-in-time simulators that could handle multi-locus processes in subdivided populations and produce individual multi-locus genotypes distributed in a varying number of populations. Nemo and quantiNemo (Guillaume & Rougemont 2006; Neuenschwander *et al.* 2008, 2019) are very popular forwards-in-time simulators to model spatially heterogeneous selection on Mendelian and quantitative traits in species with complex life-cycles. CDPOP and CDMetaPOP (Landguth & Cushman 2010; Landguth *et al.* 2017) are forward-in-time simulators of gene flow in complex landscapes in which the movement of individuals can be a function of landscape surfaces. These are only a few examples among a high number of genetic simulators differing in many features; a

This article is protected by copyright. All rights reserved

very useful, nearly exhaustive, continuously-updated and searchable database of genetic simulators is maintained at http://popmodels.cancercontrol.cancer.gov/gsr/.

Forward-in-time simulators are slower and generally require more computer memory than backward-in-time simulators because they are usually individual-based. This can hinder their application to realistic populations comprising thousands of individuals. A solution to this problem is the use of a class-based approach, where individuals are categorized according to a feature of interests (usually their genotype) that is sufficient to determine the dynamics of the system under study. This approach reduces computation time and memory requirements for populations of large size. Some years ago, we used such an approach to develop METAPOPGEN (Andrello & Manel 2015), a genetic simulator running in the R environment (R Core Team 2018) to simulate population genetics in subdivided populations in species with complex life-cycles. By using genotype numbers instead of individuals, the simulations done with METAPOPGEN can be considerably faster than those run with individual-based genetic simulators. For example, computation times of METAPOPGEN and Nemo 2.2.0 were 8 seconds and 52 seconds, respectively, for an island model with 20 demes, 3000 individuals per deme, one locus with two alleles and 200 generations (see Andrello & Manel 2015 for further details).

METAPOPGEN was limited to a single locus. Here we present METAPOPGEN 2.0, a new simulator build on METAPOPGEN capable of simulating multiple loci for species with large population size. We first describe the new simulator, then validate it on two theoretical examples, compare it to individual-based simulations, and illustrate its application to study the spatial scale of adaptation to water salinity in a coastal marine fish (the red mullet, *Mullus surmuletus*).

## 2 Description of the simulator

METAPOPGEN 2.0 is an R package to simulate a diploid population structured into *n* demes, with *z* age classes, either monoecious or dioecious, and connected by adult and propagule dispersal. All demographic parameters (survival probabilities, fecundities, dispersal probabilities, carrying capacities; see **Table 1**) can be deme- and time-dependent; in addition, adult dispersal probabilities can be age-dependent, while survival probabilities and fecundities can be age- and genotype-dependent. The number of alleles and the mutation probabilities are locus-specific.

This article is protected by copyright. All rights reserved

Accepted Article

Genetic drift is modelled through random number generators at each of the five phases of the life cycle (survival, adult dispersal, reproduction, propagule dispersal and recruitment; **Figure 1**) and selection can be modelled by setting genotype-specific survival probabilities and fecundities. Since these parameters are also deme- and time-dependent, it is possible to model complex scenarios of selection in space and time.

[Insert Table 1 here]

[Insert Figure 1 here]

The user sets the value of the demographic and genetic parameters (**Table 1**). Then, the simulator iterates the state variable `N[i,j,x,t]` (the number of individuals of genotype `i` and age `x`, in deme `j` at time `t`) through the five phases of the life cycle. In the case of dioecious life cycles (i.e. separate sexes), the number of females and males are tracked using two state variables, `N_F[i,j,x,t]` and `N_M[i,j,x,t]`. Since the functions used to iterate the life cycle are the same in the monoecious and dioecious cases, they are given below using the notation of the monoecious case. The following provides details on how the five life cycle phases are modelled (sections 2.1 to 2.5), the recombination option (section 2.6) and the parametrization of multi-locus vital rates (section 2.7) and a brief presentation of how to initialize and run the simulations, and analyse the results (sections 2.8 to 2.10). Full documentation for all the functions and datasets is available in the R package and tutorials are available on the GitHub repository github.com/MarcoAndrello/MetaPopGen.

## 2.1 Phase 1. Survival

Survival of individuals is modelled through a random draw from a binomial distribution

```
Nprime[i,j,x,t] = rbinom(1, N[i,j,x,t], sigma[i,j,x,t])
```

where `rbinom` is the random number generator function for the binomial distribution in R, `sigma[i,j,x,t]` is the annual survival probability and the `1` indicates that the draw is done once. A function is available to set deme- and genotype-dependent survival probabilities (see section 2.7)

This article is protected by copyright. All rights reserved

## 2.2 Phase 2. Adult dispersal

Adult dispersal is modelled through a random draw from a multinomial distribution:

```
rmultinom(1, Nprime[i,j,x,t], delta.ad[,j2,x,t])
```

where rmultinom is the random number generator function for the multinomial distribution in R. delta.ad[j1,j2,x,t] is the dispersal probability from deme j2 to deme j1 for individuals of age x at time t. The function rmultinom uses the vector of dispersal probabilities from j2 to the other demes (delta.ad[,j2,x,t]) to draw the number of individuals of deme j2 dispersing to the other demes. This is repeated over all demes to obtain the number of individuals after dispersal, Nprimeprime[i,j,x,t].

delta.ad[j1,j2,x,t] can be supplied by the user or computed using built-in functions to create dispersal probabilities under the assumptions of the island model [create.dispersal.IM()] or under the assumption of dispersal probability decreasing exponentially with distance, by supplying the spatial coordinates of demes [create.dispersal.coord()].

## 2.3 Phase 3. Reproduction

Reproduction is made of two subphases, production of gametes and union of gametes.
Production of gametes can be modelled either as a fixed or a random process as a function of the female and male fecundities phi_F[i,j,x,t] and phi_M[i,j,x,t] defined by the user. Under the “fixed” option, each individual of genotype i of age x in deme j at time t produces exactly phi_F[i,j,x,t] and phi_M[i,j,x,t] gametes. Alternatively, the total number of female and male gametes produced by individuals of genotype i in deme j at time t, f_F[i,j,t] and f_M[i,j,t], are modelled as random draws from Poisson distributions with expected fecundities phi_F[i,j,x,t] and phi_M[i,j,x,t], respectively:

```
rpois(Nprimeprime[i,j,x,t], phi_F[i,j,x,t]))
```

```
rpois(Nprimeprime[i,j,x,t], phi_M[i,j,x,t]))
```

and then summed over age classes. Production of gametes is then modelled through a random draw from a multinomial distribution:

Accepted Article

This article is protected by copyright. All rights reserved

Accepted Article

```
rmultinom(1, f_F[i,j,t], meiosis_matrix[,i])
```

```
rmultinom(1, f_M[i,j,t], meiosis_matrix[,i])
```

`meiosis_matrix[u,i]` gives the probability that an individual of genotype `i` produces a gamete of type `u` accounting for recombination and mutation (**Box 1**), and is built by the simulator as a function of the number of alleles per locus `l`, `allele_vec[l]`, mutation probability per locus `mu[l]` and recombination probability `r`. The function `rmultinom` uses the vector of gamete production probabilities from individuals of genotype `i` (`meiosis_matrix[,i]`) to draw the gametes produced from those individuals. The multinomial draw, repeated over genotypes, gives the numbers of female and male gametes of type `u` in deme `j` at time `t`, respectively `G_F[j,u,t]` and `G_M[j,u,t]`.

[Insert Box 1 here]

Union of gametes is modelled through a multivariate hypergeometric distribution using the random number generator function `rMWNCHypergeo` from the R package `BiasedUrn` (Fog 2015):

```
rMWNCHypergeo(1, G_M[j,,t], G_F[j,u,t])
```

The multivariate hypergeometric distribution is the multivariate analog of the univariate hypergeometric distribution and is used here to sample without replacement from a multinomial distribution. The idea is to sample the available male gametes and couple them to the available female gametes. Let `G_F[j,u,t]` be the number of female gametes of type `u` in deme `j` at time `t` to be coupled. The function then samples `G_F[j,u,t]` male gametes from the male gamete pool. The vector `G_M[j,,t]` gives the number of male gametes of all types in deme `j` at time `t`, and is used to define the probability of sampling without replacement. The sampling is repeated over all types of female gametes to obtain the number of propagules of genotype `i` in deme `j` at time t, `L[i,j,t]`.

## 2.4 Phase 4. Propagule dispersal

Propagule dispersal is modelled as adult dispersal (phase 2) but using `delta.prop[,j,t]` as the vector of dispersal probabilities:

```
rmultinom(1, L[i,j,t], delta.prop[,j,t])
```

This article is protected by copyright. All rights reserved

Accepted Article

and gives the number of settling individuals of genotype `j` in deme `i` at time `t`, `S[i,j,t]`. In the example on the red mullet, we show a propagule dispersal probability matrix computed from a biophysical model of larval dispersal.

## 2.5 Phase 5. Recruitment

In the recruitment phase, each age class is shifted to the next and settlers are recruited into the first age class. A random draw of `kappa0[j,t]` individuals (the deme carrying capacity, set by the user) is retained in the deme.

Alternatively, phase 3, 4 and 5 can be merged under the backward migration option, which can be used to reproduce Wright's island model when there is only one age class. With this option, exactly `kappa0[j,t]` new individuals are recruited per deme by randomly taking gametes from the local deme with probability 1 - `migr` and from a different deme with probability `migr`.

## 2.6 Recombination

METAPOPGEN 2.0 can simulate linkage between two loci. The recombination probability `r` can range from 0 (completely linked loci) to 0.5 (completely independent loci) and is taken into account in the calculation of `meiosis_matrix` (**Box 1**). Recombination dynamics can be complex with more than two loci, for example giving rise to crossover interference; this happens when a crossover event causing recombination between two loci affects the recombination rates of other loci on the same chromosome (Hillers 2004). For this reason, `r` must be set to 0.5 when the number of loci is higher than two, i.e. only independent loci can be simulated.

## 2.7 Genotype-dependent vital rates

Survival probabilities and fecundities can be genotype-dependent, allowing for simulating selection. METAPOPGEN 2.0 provides functions to set survival probabilities and fecundities as functions of environmental conditions and the genotype of biallelic loci (e.g. SNPs), `create.multilocus.rate()`. This can be understood as if the genotype *i* determines the optimal environmental condition $\xi_i$ for the organism. It is assumed that, at each locus, one allele reduces the optimal condition (the "-" allele) and the other increases it (the "+" allele). $\xi_i$ is thus a function of the number of "+" alleles forming the multi-locus genotype. The vital rate (survival or fecundity) $w_{ij}$ of genotype *i* in deme *j* is an exponential function of the difference between the

This article is protected by copyright. All rights reserved

Accepted Article

environmental condition of the deme, *x<sub></sub>*$x_j$, and the optimal environmental condition for the genotype, $\xi_i$:

$$w_{ij} = w_{max} \cdot \exp\left[\frac{-(x_j - \xi_i)^2}{2\omega^2}\right] \quad , \qquad \text{(eq. 1)}$$

where $w_{max}$ is the maximal theoretical vital rate. $\omega$ can be interpreted as an inverse of selection strength of the environment on individuals (Schiffers *et al.* 2013). The vital rate is maximized when the optimal condition of the genotype perfectly matches local environmental condition, and decreases exponentially at a rate inversely proportional to $\omega$ as environmental conditions change. In the case study, we give an example of application of equation (1) to parametrize survival probabilities as functions of water salinity.

### 2.8 Simulation initialization

The initial parameters that need to be set by the user are the number of alleles at each locus `allele_vec[l]`, the recombination probability `r`, the mutation probability for each locus `mu[l]`, the number of demes `n`, the number of age-classes `z`, the carrying capacity of each deme `kappa0[j,t]` and the sexuality of the species (either `"monoecious"` or `"dioecious"`). The function `initialize.multilocus()` takes these parameters, sets the genotype indexing and return the list `init.par`, containing the parameters needed to run the multi-locus simulations and the initial composition of the demes `N1[i,j,x]`. Initialization is needed to define the genotype indexing as a function of the number of alleles at each locus and the recombination rate (**Box 2**). The genotype indexing is visible as row names of `N1[i,j,x]` and can then be conveniently used to set the remaining parameters dependent on genotype, namely survival probabilities and fecundities. Step-by-step instructions to initialize and perform the simulations are given in a tutorial.

[Insert Box 2 here]

### 2.9 Simulation run

Simulations are performed using the functions `sim.metapopgen.monoecious.multilocus()` or `sim.metapopgen.dioecious.multilocus()`, depending on the sexuality of the

This article is protected by copyright. All rights reserved

species. The argument of these functions are the `init.par` list, the survival probabilities, the fecundities, the propagule and adult dispersal probabilities, the number of generations of simulations `T_max` and some other parameters controlling the output.

### 2.10 Simulation output

The simulations return the variables `N[i,j,x,t]`, `Nprime[i,j,x,t]`, `Nprimeprime[i,j,x,t]`, `L[i,j,t]` and `S[i,j,t]` at each time step as chosen by the user. Several functions are available for basic analysis, such as calculating single-locus genotype frequencies [`freq_genotypes()`], single-locus allele frequencies [`freq_alleles()`], gamete frequencies [`freq_gametes()`], observed and expected heterozygosities [`het_obs()` and `het_exp`], $F_{ST}$ [`fst_multilocus()`] and linkage disequilibrium [`ld()`].

## 3 Theoretical validation

We tested whether METAPOPGEN 2.0 could reproduce the results of two population genetics model for which theoretical predictions are known. We first consider a single monoecious population with discrete generations and follow the fate of two independent loci with two alleles each in absence of mutation and selection. We set migration = "backward" to simulate random draws from an infinite gamete pool. Under these assumptions, the proportion of heterozygous individuals is expected to decline according to the relationship:

$$H_t = H_0\left(1 - \frac{1}{2N}\right)^t \qquad \text{(eq. 2)}$$

where $H_t$ is the expected proportion of heterozygous individuals at generation $t$, $H_0$ is the initial proportion, $N$ is population size and $t$ is the generation. **Figure 2** shows that the decline in observed heterozygosity averaged over 100 replicate populations follows the expectation given by equation 2. The R code `WrightFisher.R` to perform the simulations is available on Dryad (Andrello *et al.* 2020).

[insert Figure 2 here]

We then tested whether METAPOPGEN 2.0 could reproduce the results of the two-locus spatial population genetic model with dispersal and selection studied by Slatkin (1975). Briefly, Slatkin's

This article is protected by copyright. All rights reserved

Accepted Article

model considers two unlinked loci with two possible alleles at each locus (A1, A2 and B1, B2) and a large number of demes each identified by a spatial coordinate *z* on a unidimensional landscape. The parameters controlling the dynamics of the model are the selection coefficients at the two loci, *s* and *t*, and the dispersal kernel, which is assumed symmetric and described by the standard deviation of dispersal distance $\sigma_{disp}$. Slatkin (1975) considered the case where one allele at each locus (A1 and B1) is favoured on the left side of the landscape, the other allele (A2 and B2) is favoured on the right side of the landscape, and the two selection coefficients are equal ($s = t$). The equilibrium state of the model can be characterized by the difference in gamete frequencies $B$(z) = $f$(A1B1) – $f$(A2B2).

We implemented Slatkin's model in MetaPopGen 2.0 using `n` = 100 demes with `kappa0` = 1000 individuals each (see R code `SlatkinModel.R`). Selection was implemented through differences in fecundities `phi_F` and `phi_M` between the four gametes, calculated using a base fecundity value of 1000 propagules multiplied by the relative fitness values of Table 1 in Slatkin (1975). We considered two scenarios of selection strength (selection coefficient $s$ = 0.02 and 0.05) and three scenarios of dispersal (standard deviation of dispersal distance $\sigma_{disp}$ = 0.5, 1 and 2). We computed gamete frequencies using the function `freq_gametes()` and used them to calculate $B(z)$. We then calculated the expected value of $B$(z) using equation 14 in Slatkin (1975). The results obtained with MetaPopGen 2.0 show that $B$(z) increases from the centre of the landscape to the margin, following closely the theoretical predictions (**Figure 3**).

[Insert Figure 3 here]

## 4 Computation time

We compared the computation time of MetaPopGen 2.0 with that of Nemo 2.2.0 on an island model with a finite number of demes (`n` = 10, 15 or 20), number of individuals per deme `kappa0` = 2000, 4000 or 6000, and number of loci `l` = 2, 4 or 6. We set the other demographic and genetic parameters of MetaPopGen and Nemo to the same values (See files `IslandModelMetaPopGen.R` and `IslandModelNemo.ini`) and we ran simulations for 1000 generations so both programs converged to the same $F_{ST}$.

This article is protected by copyright. All rights reserved

Accepted Article

For two and four loci, MetaPopGen 2.0 was always faster than Nemo 2.2.0 (**Figure 4**). With six loci, MetaPopGen 2.0 was slower than Nemo 2.2.0 across the scenarios analysed. The computation times of Nemo 2.2.0 increased with the number of individuals per demes and the number of deme, indicating a dependence of computation time on the total number of individuals, as expected for individual-based simulations. The computation times of MetaPopGen 2.0 were not affected by the number of individuals, but increased with the number of loci and the number of demes, reflecting a dependence on the size of the genotype and deme dimensions of the R objects.

[Insert  Figure 4 here]

## *5* Example: adaptation to water salinity in the red mullet *Mullus surmuletus*

We illustrate the capabilities of MetaPopGen 2.0 by investigating the multi-locus patterns and processes of adaptation to local water salinity in a coastal marine fish, the striped red mullet *Mullus surmuletus* (Linnaeus, 1758), in the Mediterranean Sea.

### 5.1 Parametrization

We modelled `n` = 100 local demes spaced at 100 km covering the entire Mediterranean coastline, including islands. We used a `dioecious` life-cycle and with `z` = 5 age classes, according to the known life-cycle of the species (Reñones *et al.* 1995; Mehanna 2009). Female fecundity `phi_F` was set to 0, 6852, 11089, 16052 and 20974 eggs per year for the first to the fifth age-class, respectively (Mehanna 2009), for all demes. In absence of data on male fecundity, we set an age-invariant fecundity `phi_M` to $10^6$ sperms per year in all demes for all age classes except the first one, which correspond to newly recruited individuals.

Propagule dispersal probabilities `delta.prop` were taken from the biophysical larval dispersal model used in Boulanger *et al.* (2020) (**Figure 5a**). In absence of knowledge on adult dispersal, we modelled site fidelity by setting adult dispersal probabilities `delta.ad` between different demes to zero. Values of sea surface salinity for the Mediterranean Sea were obtained from the oceanographic model NEMOMED8 (Somot *et al.* 2006) as average over the period 1990–2013 (**Figure 5b**). Carrying capacity `kappa0` was set to 5000 individuals in all demes and years.

This article is protected by copyright. All rights reserved

Accepted Article

[Insert Figure 5 here]

Dalongeville, Benestan, et al. (2018) identified three loci significantly associated with water salinity in *M. surmuletus*, thus potentially implicated in mechanisms of salinity tolerance and adaptation. Accordingly, we simulated four biallelic loci, among which three loci under selection from salinity (adaptive loci) and one neutral locus to track neutral genetic differentiation. Since no explicit test for linkage was conducted in Dalongeville, Benestan, et al. (2018), we assumed that the four simulated loci were unlinked (`r` = 0.5). We further assumed a mutation probability `mu` = 10$^{-6}$ per locus. As each biallelic locus can give rise to three genotypes, with $L$ = 3 unlinked loci under selection, the number of multi-locus adaptive genotypes is $3^L = 3^3 = 27$ (**Box 2**, eq. B4). At the three loci under selection, we assumed that one allele reduced the phenotypic value (the "-" allele) while the other increased it (the "+" allele). This gives rise to seven different combinations of number of "+" alleles per genotype. We assigned to each combination an optimal salinity ranging from 36 to 39 practical salinity units (PSU) at intervals of 0.5, according to the range of water salinity in the Mediterranean Sea.

We assumed that only survival was under selection, whilst fecundity was unaffected by salinity, and we parametrized survival probabilities `sigma_F` and `sigma_M` using the function `create.multilocus.rate()` [equation (1)] for all age-classes. As annual survival probabilities of *M. surmuletus* in natural conditions are unknown, we arbitrarily set $w_{max} = 0.8$ and $\omega$ = 1.

The starting allele frequencies were set to 0.5 for both alleles at all loci. The simulations were run for `T_max` = 100 time steps (years), to illustrate how demes could adapt to their local salinity conditions starting from homogeneous allele frequencies. Each simulation was replicated ten times. See R code `MullusSimulations.R`.

### 5.2 Results

The allele frequencies at the neutral locus remained relatively stable during the simulation, showing small fluctuations due to genetic drift and gene flow between demes (**Figure 5a-d**, "LocusD"). The allele frequencies at the adaptive loci showed monotonic increasing or decreasing trends depending on the value of the selective environmental variable in the deme (salinity). For

This article is protected by copyright. All rights reserved

Accepted Article

example, in deme 1 (high salinity, x = 38.5; **Figure 5b**), the frequency of the salinity-tolerant alleles increased and reached unity at all three adaptive loci in about 50 years (**Figure 6**). In deme 69 (low salinity, x = 32.4), the frequency of the salinity-tolerant alleles decreased to about 0.2 in the same time (**Figure 6**). These dynamics were driven by strong directional selective pressure at the adaptive loci due to extreme salinity values. Different replicates of the simulations produced the same results (not shown).

There were also demes showing differences in allele dynamics between replicates. For example, in deme 48 (low salinity, x = 35.7), replicate #9 and #10 showed marked differences in the frequency of the salinity-adaptive allele at the three adaptive loci (**Figure 6**). However, the mean number of “+” alleles in the deme reached an equilibrium value at about 3.4 in both replicates, as most individuals had either three or four “+” alleles (**Figure 6**).

[Insert  Figure 6 here]

## 6 Discussion

Simulations are necessary to study evolutionary dynamics in complex landscapes in species with complex life cycles with traits under the control of multiple loci. Here, we have presented MetaPopGen 2.0, a genetic simulator to model multi-locus genetic systems in subdivided populations. MetaPopGen 2.0 is versatile regarding the customizable values of demographic parameters. Propagule and adult dispersal can be set using user-defined dispersal matrices. Survival and fecundities can take age-, deme- and genotype-specific values, so that the user can model different selection pressure for different demes and age classes. Survival, fecundities, propagule dispersal and adult dispersal can also be variable in time. In the case of two loci, recombination rates between loci can be set by the user to explore the effects of different recombination schemes. The simulations can therefore incorporate the spatial and temporal heterogeneity in dispersal patterns and selective environmental variables normally observed in natural populations.

There are many other genetic simulators that allow to study multi-locus systems in species with complex life cycles and in complex landscapes. Nemo 2.3.51 and quantiNemo 2.0 can simulate

This article is protected by copyright. All rights reserved

Accepted Article

multiple demes, dioecious and monoecious life-cycles. However, they do not include age structure, demographic parameters (fecundity and dispersal) cannot be time-dependent and loci must have the same number of alleles and mutation rate. CDPOP and CDMetaPOP simulate individuals on a fixed grid with age-structure, selection and time-varying demographic rates, and are especially suitable to model spatially-explicit systems. MetaPopGen 2.0 is not spatially explicit, but, in the application example, we have shown that demographic parameters can be set as a function of environmental variables. While Nemo/quantiNemo and CDPOP/CDMetaPOP are coded respectively in C and in python, MetaPopGen 2.0 is developed within the R environment, which is very popular among ecologists and evolutionary biologists to perform statistical, spatial, and other analyses (Paradis *et al.* 2017). MetaPopGen 2.0 may therefore be easier to approach than other simulators for non-modeler users who want to modify its content to include new capabilities.

The most important new feature of MetaPopGen 2.0 relative to its predecessor MetaPopGen (Andrello & Manel 2015) is the possibility to simulate multiple loci. This addition required the development of a new mode of representation of genetic information and new functions to simulate the production and union of gametes (see section 2.1 to 2.6 and Box 2). MetaPopGen 2.0 also features a new dispersal phase for adults, a backward migration scheme and numerous functions for initialization of simulations and analysis of results.

The recursion equations of MetaPopGen 2.0 are based on the frequency-based approach of MetaPopGen, which greatly reduces computation time and memory needs to simulate populations with large numbers of individuals. This advantage has made MetaPopGen an useful tool to explore the processes shaping the genetic structure of species with abundant populations, such as marine fish and invertebrates (Handal *et al.* Early view; Marandel *et al.* 2018) and terrestrial plants (Smith *et al.* 2020). Memory needs and computation time increase with the number of loci simulated, because the numbers of possible unique gametes and genotypes increase geometrically with the number of loci and alleles per locus (Box 2), and increasing the number of classes entails a parallel increase in computation time. The comparison of computation times between MetaPopGen 2.0 and Nemo 2.2.0 suggests that, in order to simulate adaptive dynamics in small populations with more than a few loci, individual-based

This article is protected by copyright. All rights reserved

simulators like ALADYN (Schiffers & Travis 2014), NEMO and QUANTINEMO (Guillaume & Rougemont 2006; Neuenschwander *et al.* 2008, 2019) or CDPOP and CDMETAPOP (Landguth & Cushman 2010; Landguth *et al.* 2017, 2020) are still the best option. However, numerous species show phenotypic traits under the control of a few loci only (Courtier-Orgogozo *et al.* 2020) and some of these traits are related to fitness and local adaptation. For example, resistance of sugar beet to necrotic yellow vein virus is under the control of three loci (Scholten *et al.* 1999) and bud set in European aspen (*Populus tremula*) is under the control of a single locus (Wang *et al.* 2018). In this cases, METAPOPGEN 2.0 can provide a faster alternative to individual-based simulators to study genetic processes in complex landscapes, for species with overlapping generations and arbitrarily large population sizes (such as fish and plants).

**Acknowledgements**

Part of this work was funded by PSL Environment through the RESMOD project.

**Author contributions**

M.A., F.D. and S.M. designed the initial study; M.A. coded METAPOPGEN 2.0 with inputs from C.N. M.A. performed the theoretical validation and the analysis of computation time. M.A. and C.N. performed the example simulations on *M. surmuletus*. M.A. wrote the manuscript with inputs from F.D. and S.M.

**Data accessibility statement**

METAPOPGEN 2.0 is available as an R package on GitHub at the address: github.com/MarcoAndrello/MetaPopGen. Code and data to perform the simulations are available on Dryad: 10.5061/dryad.jq2bvq87d (Andrello *et al.* 2020).

Accepted Article

This article is protected by copyright. All rights reserved

## BOXES

-----------------------------------------**Box 1. Meiosis matrix**---------------------------------------------------

The meiosis_matrix[u,j] used in the reproduction phase of METAPOPGEN 2.0 corresponds to the probability that an individual of genotype *j* produces a gamete of type *u*, and is calculated as:

$$ME[u,j] = \sum_{k=1}^{U} MU[k,u]RE[k,j]$$

*RE[k,j]* is the probability that genotype *j* produces gamete *k* after segregation and crossover. *MU[k,u]* is the probability that a gamete of type *u* mutates into a gamete of type *k* and is calculated as product of single-locus mutation probabilities mu[l]. For example, with two loci and two alleles, the probability that gamete A1B1 mutates into gamete A1B2 is (1-mu[1])*mu[2]. The summation is done over all types of gametes. Supplementary **Tables S1**, **S2** and **S3** give examples of MU, RE and ME for the case of two loci with two alleles, r = 0.1, mu[1] = 0.1 and mu[2] = 0.2.

-------------------------------------------------**END OF BOX 1**-------------------------------------------------------

Accepted Article

This article is protected by copyright. All rights reserved

Accepted Article

**-------------------Box 2. Gamete-based and locus-based storing methods------------------------------**

MetaPopGen 2.0 uses two mapping methods to link couples of uniting gametes to genotypes. A gamete-based representation is used for linked loci (currently limited to two loci as explained in the main text) and a locus-based representation for unlinked loci to reduce memory usage and computation time.

We illustrate the differences between the two methods using the example of two loci A and B with two alleles each (A1, A2, B1 and B2), which generate four multi-locus gametes (A1B1, A1B2, A2B1 and A2B2). The gamete-based representation, which was also used in the previous versions of MetaPopGen, is a triangular matrix of size equal to the number of possible unique gametes (**Table B1**). It allows calculating the genotype frequencies of newborns from the parental gamete frequencies in a straightforward manner. The locus-based representation, on the other hand, is a multi-dimensional array with number of dimensions equal to the number of loci and cannot be directly used to calculate the genotype frequencies of newborns. However, in the case of unlinked loci (recombination rate *r* = 0.5), the locus-based representation provides a more efficient way of storing data than the gamete-based representation. This is because the two double heterozygote genotypes (A1B1/A2B2 and A1B2/A2B1) are equivalent in terms of multi-locus gamete production, and can be pooled into the same genotype (A1A2/B1B2).

To assess the differences in memory requirement between the two methods, let $n_{a_i}$be the number of alleles at locus *i*, and *L* the number of loci. In the gamete-based representation, the number of possible unique gametes is

$$n_\gamma = \prod_{i=1}^{L} n_{a_i} \qquad \text{(eq. B1)}$$

and the number of possible unique diploid genotypes is

$$n_G = \frac{n_\gamma(n_\gamma + 1)}{2} \qquad \text{(eq. B2)}$$

In the locus-based representation, the number of possible unique single-locus diploid genotypes at locus *i* is

This article is protected by copyright. All rights reserved

$$n_{G_i} = \frac{n_{a_i}(n_{a_i} + 1)}{2} \qquad \text{(eq. B3)}$$

and the number of possible unique genotypes is

$$n_G = \prod_{i=1}^{L} n_{G_i} \qquad \text{(eq. B4)}$$

The gain in efficiency increases with the number of loci and alleles per locus (**Figure B1**). In the case of linked loci ($r < 0.5$), there is no alternative to the gamete-based representation because the double heterozygote genotypes produce gametes in different proportions.

[Insert Table B1 here]

[Insert Figure B1 here]

-------------------------------------------------**END OF BOX 2**-------------------------------------------------------

Accepted Article

This article is protected by copyright. All rights reserved

Accepted Article

## TABLES

**Table 1. Parameters and variables used in MetaPopGen 2.0**

| R object | Definition | Sexuality[1] | Life cycle phase[2] |
|---|---|---|---|
| *Dimensions* | | | |
| i | Genotype | | |
| j | Deme | | |
| l | Locus | | |
| x | Age | | |
| t | Time | | |
| u | Gametotype | | |
| | | | |
| *Parameters* | | | |
| allele_vec[l] | | | |
| delta.ad[j1,j2,x,t] | Adult dispersal probability from deme j2 to deme j1 | | 2 |
| delta.prop[j1,j2,t] | Propagule dispersal probability from deme j2 to deme j1 | | 4 |
| kappa0[j,t] | Deme carrying capacity | | 5 |
| mu[l] | | | |
| n | Number of demes | | |
| N1[i,j,x] | Initial number of individuals | m | |
| N1_F[i,j,x] | Initial number of female individuals | d | |

This article is protected by copyright. All rights reserved

Accepted Article

| | | | |
|---|---|---|---|
| N1_M[i,j,x] | Initial number of male individuals | d | |
| phi_F[i,j,x,t] | Female fecundity | both | 3 |
| phi_M[i,j,x,t] | Male fecundity | both | 3 |
| r | Recombination probability | | |
| sigma[i,j,x,t] | Survival probability | m | 1 |
| sigma_F[i,j,x,t] | Female survival probability | d | 1 |
| sigma_M[i,j,x,t] | Male survival probability | d | 1 |
| T_max | Simulation time | | |
| z | Number of age classes | | |
| *Variables* | | | |
| L[i,j,t] | Number of propagules (e.g. larvae) | m | 4 |
| L_F[i,j,t] | Number of female propagules | d | 4 |
| L_M[i,j,t] | Number of male propagules | d | 4 |
| N[i,j,x,t] | Number of individuals | m | 1 |
| N_F[i,j,x,t] | Number of female individuals | d | 1 |
| N_M[i,j,x,t] | Number of male individuals | d | 1 |
| Nprime[i,j,x,t] | Number of individuals after survival | m | 2 |
| Nprime_F[i,j,x,t] | Number of female individuals after survival | d | 2 |
| Nprime_M[i,j,x,t] | Number of male individuals after survival | d | 2 |

This article is protected by copyright. All rights reserved

Accepted Article

| | | | |
|---|---|---|---|
| `Nprimeprime[i,j,x,t]` | Number of individuals after survival and adult dispersal | m | 3, 5 |
| `Nprimeprime_F[i,j,x,t]` | Number of female individuals after survival and adult dispersal | d | 3, 5 |
| `Nprimeprime_M[i,j,x,t]` | Number of male individuals after survival and adult dispersal | d | 3, 5 |
| `S[i,j,t]` | Number of propagules after dispersal (settlers) | m | 5 |
| `S_F[i,j,t]` | Number of female propagules after dispersal (settlers) | d | 5 |
| `S_M[i,j,t]` | Number of male propagules after dispersal (settlers) | d | 5 |

[1] m, monoecious; d, dioecious

[2] Life cycle phase where the variable is used: 1, survival; 2, adult dispersal; 3, reproduction; 4, propagule dispersal; 5, recruitment

This article is protected by copyright. All rights reserved

Accepted Article

**Table B1. Representation of multi-locus genotypes in MetaPopGen 2.0.** Example with two loci A and B with two alleles each. The double heterozygote genotypes are in *italics*.

Gamete-based

|  |  | Female gamete |  |  |  |
|---|---|---|---|---|---|
|  |  | A1B1 | A1B2 | A2B1 | A2B2 |
| Male gamete | A1B1 | A1B1 / A1B1 |  |  |  |
|  | A1B2 | A1B2 / A1B1 | A1B2 / A1B2 |  |  |
|  | A2B1 | A2B1 / A1B1 | *A2B1 / A1B2* | A2B1 / A2B1 |  |
|  | A2B2 | *A2B2 / A1B1* | A2B2 / A1B2 | A2B2 / A2B1 | A2B2 / A2B2 |

Locus-based

|  |  | Locus B |  |  |
|---|---|---|---|---|
|  |  | B1B1 | B1B2 | B2B2 |
| Locus A | A1A1 | A1A1 / B1B1 | A1A1 / B1B2 | A1A1 / B2B2 |
|  | A1A2 | A1A2 / B1B1 | *A1A2 / B1B2* | A1A2 / B2B2 |
|  | A2A2 | A2A2 / B1B1 | A2A2 / B1B2 | A2A2 / B2B2 |

This article is protected by copyright. All rights reserved

Accepted Article

**Figure captions**

**Figure 1. Life cycle used in MetaPopGen 2.0.** The life cycle includes five phases (survival, adult dispersal, reproduction, propagule dispersal and recruitment) and starts with `N[i,j,x,t]`: number of individuals of genotype `i` in deme `j` of age `x` at time `t` before survival. `Nprime[i,j,x,t]`: number of individuals of genotype `i` in deme `j` of age `x` at time `t` after survival and before adult dispersal. `Nprimeprime[i,j,x,t]`: number of individuals of genotype `i` in deme `j` of age `x` at time `t` after survival and adult dispersal. `L[i,j,t]`: number of propagules of genotype `i` in deme `j` at time `t` before propagule dispersal. `S[i,j,t]`: number of propagules of genotype `i` in deme `j` at time `t` after propagule dispersal (settlers). In the recruitment phase, `N[i,j,x,t+1]` is calculated using `S[i,j,t]` to fill the first age class and `Nprimeprime[i,j,x,t]` to fill the older age classes, closing the life-cycle.

**Figure 2. Simulation of a two-locus system in single populations.** Decline in the proportion of heterozygous individuals ($H_t$) with generations for the first locus (left panel) and the second locus (right panel). Grey lines are $H_t$ in each of 100 replicate populations, the black line is the average $H_t$ over populations and the red dashed line is the expected value of $H_t$ calculated with equation 2.

**Figure 3. Simulation of a landscape with selection and gene flow.** Values of the difference in gamete frequencies $B(z) = f(\text{A1B1}) - f(\text{A2B2})$ per deme along a unidimensional landscape with 100 demes. Only the right-hand portion of the landscape, where alleles A1 and B1 are favoured over A2 and B2, is shown. Dots and bars shown mean and 95% confidence intervals of ten MetaPopGen 2.0 simulation replicates for two selection coefficients ($s$ = 0.02, left panel; $s$ = 0.05, right panel) and three values of standard deviation of dispersal distance ($\sigma_{disp}$ = 0.5, red; $\sigma_{disp}$ = 1, green; $\sigma_{disp}$ = 2, blue). The solid lines are the theoretical predictions given by Slatkin (1975).

**Figure 4. Comparison of computation times of Nemo 2.2.0 and MetaPopGen 2.0**. Computation times (seconds) for running 100 generations are shown an island model with various number of demes, inividuals per deme and loci. Results for Nemo with 6000 individuals and 20 demes are not available because the program crashed. Runs were executed on an Intel i5-8500 CPU, 3.00 GHz with 32 Gb RAM.

This article is protected by copyright. All rights reserved

**Figure 5. Simulation of adaptation to water salinity in the red mullet, input data.** (a) Larval dispersal probabilities between demes obtained through the biophysical model of Boulanger et al (2020). b) Salinity measured in practical salinity units (PSU) in the 100 demes of the Mediterranean Sea and main rivers.

**Figure 6. Simulation of adaptation to water salinity in the red mullet, results.** Top four panels: allele frequencies at the four loci; for the adaptive loci (A, B and C), the curves show the frequency of the "+" allele (increasing survival at higher salinities). The four panels show results for different combinations of demes and replicates. Bottom two panels: mean number of "+" alleles in deme 48 in two replicate simulations.

**Figure B1. Comparison of gamete-based and locus-based representation methods.** Number of possible unique genotypes as a function of the number of loci and the number of alleles per locus, calculated through equations 1 to 4.

Accepted Article

This article is protected by copyright. All rights reserved

Accepted Article

## References

Andrello, M. & Manel, S. (2015). MetaPopGen: an R package to simulate population genetics in large size metapopulations. *Molecular Ecology Resources*, 15, 1153–1162.

Andrello, M., Noirot, C., Débarre, F. & Manel, S. (2020). *Dataset for METAPOPGEN 2.0: a multi-locus genetic simulator to model populations of large size*. Available at: https://doi.org/10.5061/dryad.jq2bvq87d. Last accessed 30 September 2020.

Balloux, F. (2001). EASYPOP (version 1.7): a computer program for population genetics simulations. *J. Hered.*, 92, 301–302.

Boulanger, E., Dalongeville, A., Andrello, M., Mouillot, D. & Manel, S. (2020). Spatial graphs highlight how multi-generational dispersal shapes landscape genetic patterns. *Ecography*, 43, 1167–1179.

Bürger, R. (2019). Multilocus population-genetic theory. *Theoretical Population Biology*.

Courtier-Orgogozo, V., Arnoult, L., Prigent, S.R., Wiltgen, S. & Martin, A. (2020). Gephebase, a database of genotype–phenotype relationships for natural and domesticated variation in Eukaryotes. *Nucleic Acids Res*, 48, D696–D703.

Dalongeville, A., Benestan, L., Mouillot, D., Lobreaux, S. & Manel, S. (2018). Combining six genome scan methods to detect candidate genes to salinity in the Mediterranean striped red mullet (*Mullus surmuletus*). *BMC Genomics*, 19, 217.

Fog, A. (2015). *Biased Urn Model Distributions*. R package. .

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

Handal, W., Szostek, C., Hold, N., Andrello, M., Thiébaut, E., Harney, E., *et al.* (Early view). New insights on the population genetic structure of the great scallop (*Pecten maximus*) in the English Channel, coupling microsatellite data and demogenetic simulations. *Aquatic Conservation: Marine and Freshwater Ecosystems*, n/a.

Hillers, K.J. (2004). Crossover interference. *Curr. Biol.*, 14, R1036-1037.

Hoban, S. (2014). An overview of the utility of population simulation software in molecular ecology. *Molecular Ecology*, 23, 2383–2401.

Hoban, S., Bertorelle, G. & Gaggiotti, O.E. (2012). Computer simulations: tools for population and evolutionary genetics. *Nat. Rev. Genet.*, 13, 110–122.

This article is protected by copyright. All rights reserved

Accepted Article

Landguth, E.L., Bearlin, A., Day, C.C. & Dunham, J. (2017). CDMetaPOP: an individual-based, eco-evolutionary model for spatially explicit simulation of landscape demogenetics. *Methods in Ecology and Evolution*, 8, 4–11.

Landguth, E.L. & Cushman, S.A. (2010). cdpop: A spatially explicit cost distance population genetics program. *Molecular Ecology Resources*, 10, 156–161.

Landguth, E.L., Forester, B.R., Eckert, A.J., Shirk, A.J., Menon, M., Whipple, A., *et al.* (2020). Modelling multilocus selection in an individual-based, spatially-explicit landscape genetics framework. *Molecular Ecology Resources*, 20, 605–615.

Marandel, F., Lorance, P., Andrello, M., Charrier, G., Le Cam, S., Lehuta, S., *et al.* (2018). Insights from genetic and demographic connectivity for the management of rays and skates. *Canadian Journal of Fisheries and Aquatic Sciences*, 75, 1291–1302.

Mehanna, S.F. (2009). Growth, mortality and spawning stock biomass of the striped red mullet *Mullus surmuletus*, in the Egyptian Mediterranean waters. *Mediterranean Marine Science*, 10, 5–18.

Neuenschwander, S., Hospital, F., Guillaume, F. & Goudet, J. (2008). quantiNemo: an individual-based program to simulate quantitative traits with explicit genetic architecture in a dynamic metapopulation. *Bioinformatics*, 24, 1552–1553.

Neuenschwander, S., Michaud, F. & Goudet, J. (2019). QuantiNemo 2: a Swiss knife to simulate complex demographic and genetic scenarios, forward and backward in time. *Bioinformatics*, 35, 886–888.

Paradis, E., Gosselin, T., Grünwald, N.J., Jombart, T., Manel, S. & Lapp, H. (2017). Towards an integrated ecosystem of R packages for the analysis of population genetic data. *Molecular Ecology Resources*, 17, 1–4.

Peng, B. & Amos, C.I. (2008). Forward-time simulations of non-random mating populations using simuPOP. *Bioinformatics*, 24, 1408–1409.

Peng, B. & Kimmel, M. (2005). simuPOP: a forward-time population genetics simulation environment. *Bioinformatics*, 21, 3686–3687.

R Core Team. (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

This article is protected by copyright. All rights reserved

Accepted Article

Reñones, O., Massutí, E. & Morales-Nin, B. (1995). Life history of the red mullet *Mullus surmuletus* from the bottom-trawl fishery off the Island of Majorca (north-west Mediterranean). *Marine Biology*, 123, 411–419.

Schiffers, K., Bourne, E.C., Lavergne, S., Thuiller, W. & Travis, J.M.J. (2013). Limited evolutionary rescue of locally adapted populations facing climate change. *Phil. Trans. R. Soc. B*, 368, 20120083.

Schiffers, K.H. & Travis, J.M.J. (2014). ALADYN – a spatially explicit, allelic model for simulating adaptive dynamics. *Ecography*, 37, 1288–1291.

Scholten, O.E., De Bock, Th.S.M., Klein-Lankhorst, R.M. & Lange, W. (1999). Inheritance of resistance to beet necrotic yellow vein virus in *Beta vulgaris* conferred by a second gene for resistance. *Theor Appl Genet*, 99, 740–746.

Slatkin, M. (1975). Gene Flow and Selection in a Two-Locus System. *Genetics*, 81, 787–802.

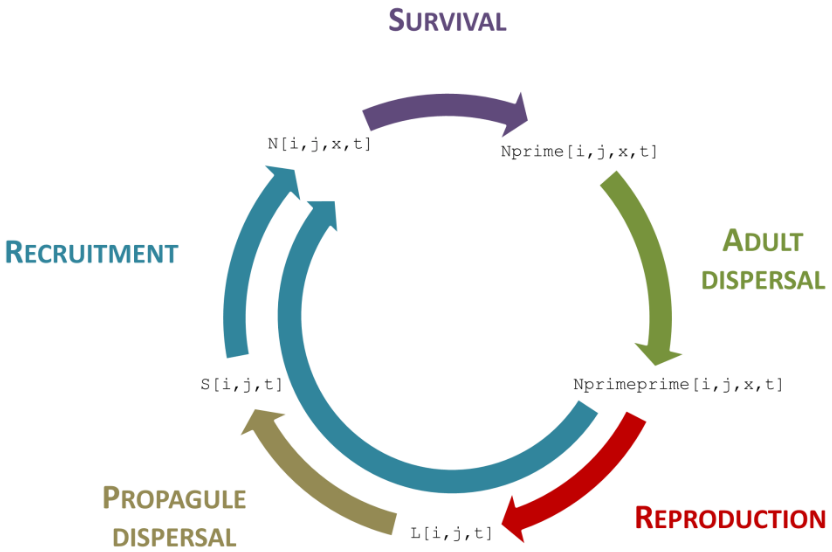
Smith, A.L., Hodkinson, T.R., Villellas, J., Catford, J.A., Csergő, A.M., Blomberg, S.P., *et al.* (2020). Global gene flow releases invasive plants from environmental constraints on genetic diversity. *PNAS*, 117, 4218–4227.

Somot, S., Sevault, F. & Deque, M. (2006). Transient climate change scenario simulation of the Mediterranean Sea for the twenty-first century using a high-resolution ocean circulation model. *Climate Dynamics*, 27, 851–879.

Wang, J., Ding, J., Tan, B., Robinson, K.M., Michelson, I.H., Johansson, A., *et al.* (2018). A major locus controls local adaptation and adaptive life history variation in a perennial plant. *Genome Biology*, 19, 72.

This article is protected by copyright. All rights reserved

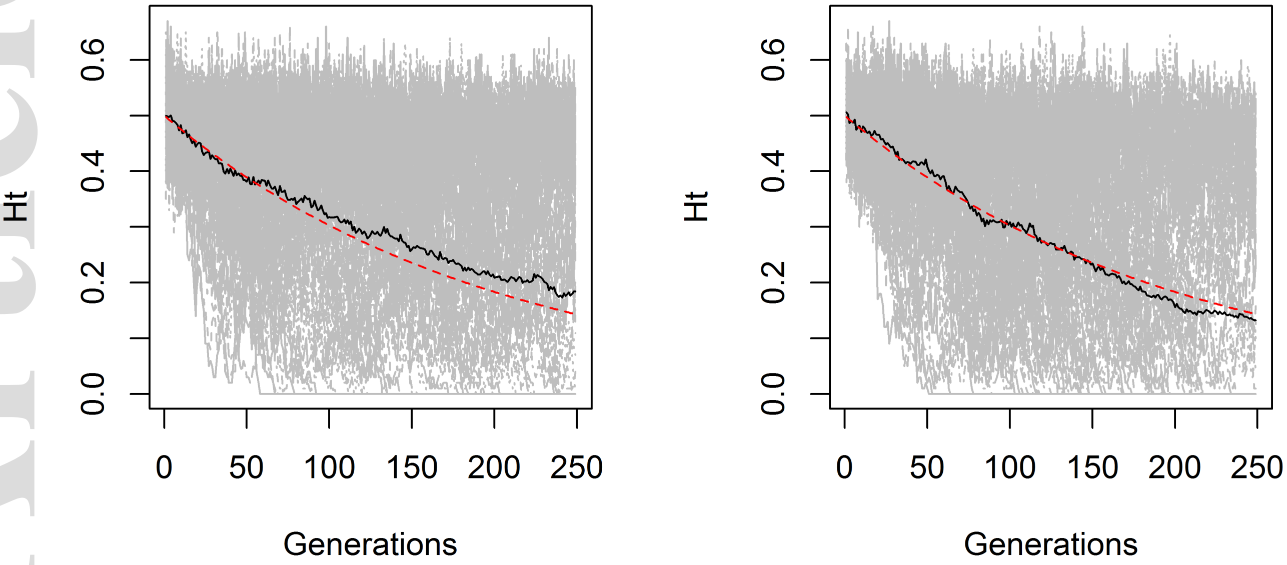
Accepted Article



men_13270_f1.png

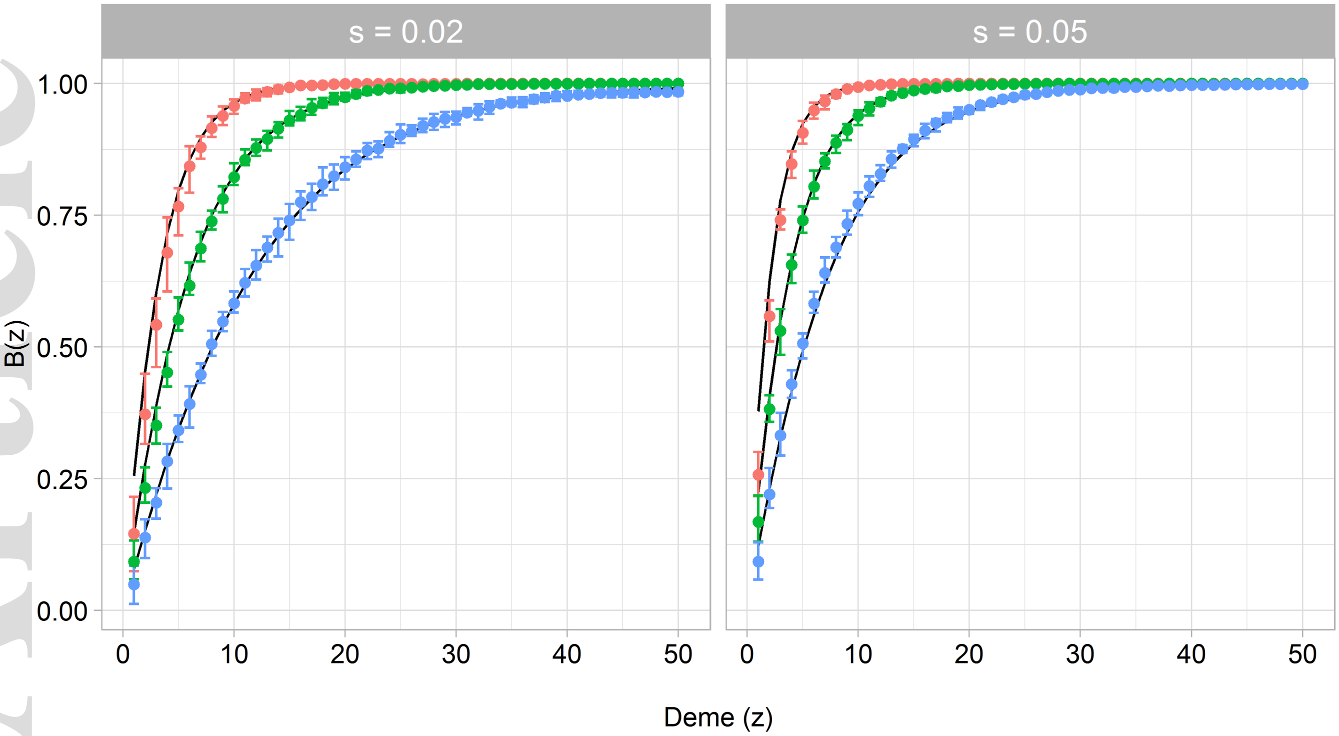
This article is protected by copyright. All rights reserved

Accepted Article



men_13270_f2.png

This article is protected by copyright. All rights reserved

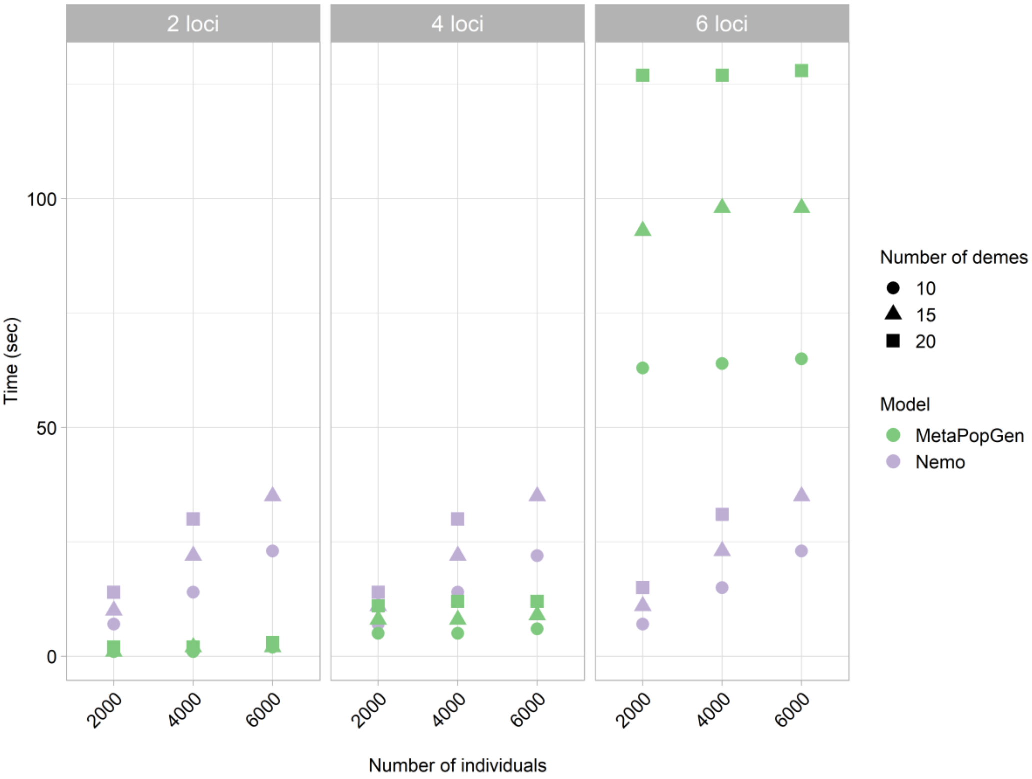


men_13270_f3.png

Accepted Article

This article is protected by copyright. All rights reserved

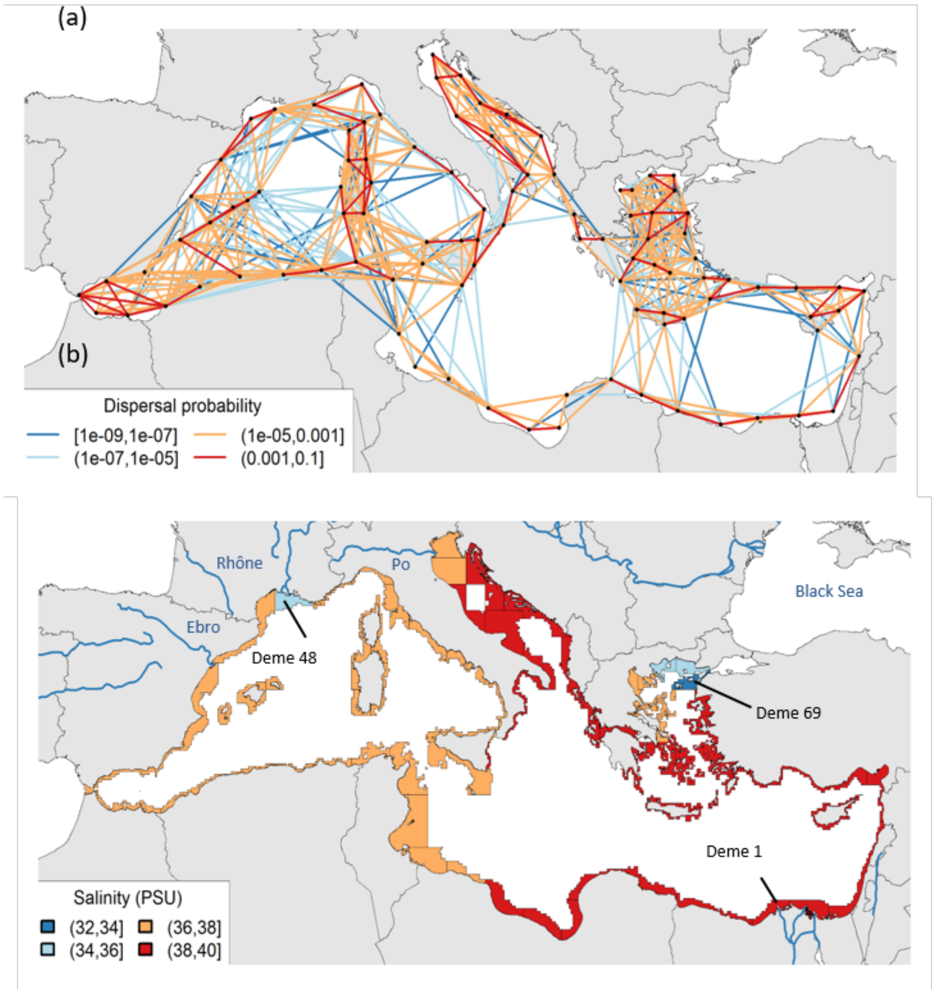
Accepted Article



men_13270_f4.png

This article is protected by copyright. All rights reserved

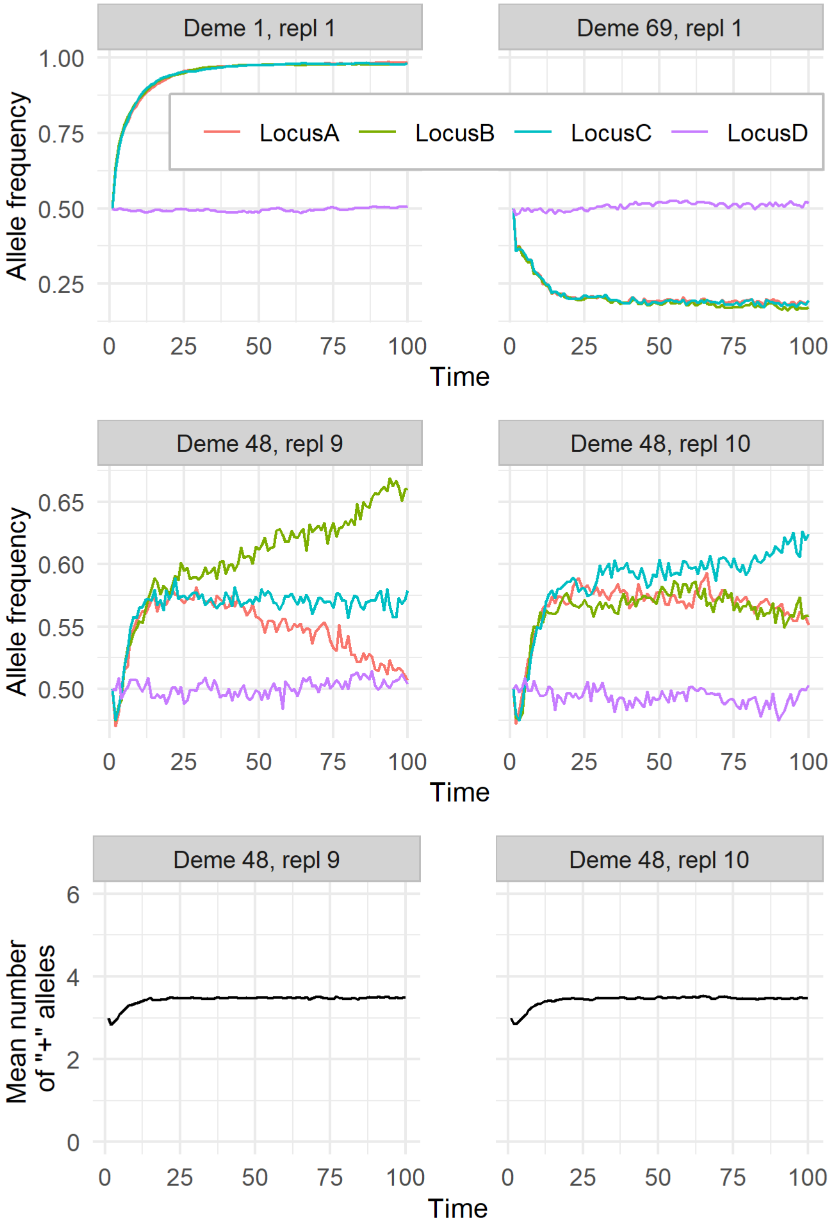
Accepted Article



men_13270_f5.png

This article is protected by copyright. All rights reserved

Accepted Article



men_13270_f6.png

This article is protected by copyright. All rights reserved