MIMARsim - a program to simulate input data sets under the isolation-migration model for MIMAR

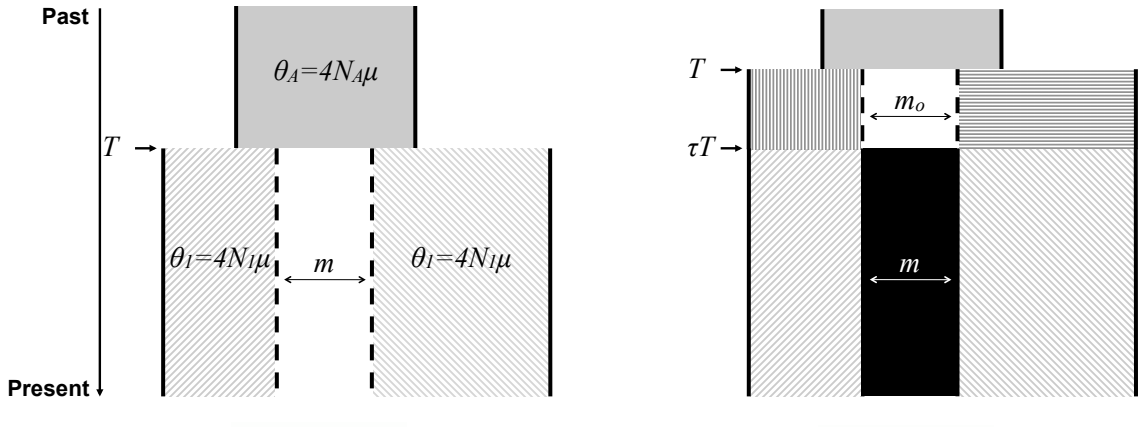
C. Becquet

March 19, 2010

This document describes how to use MIMARsim, a program that simulates a data set under the "isolation-migration" model (see Figure 1a and section "List of parameters and symbols") and creates an input file that can be analyzed by MIMAR (see "MIMARdoc".pdf for details about the model and its parameters). MIMARsim uses an input file that contains information about Y randomly-chosen, independently evolving loci to generate samples under the isolation-migration model with the parameters provided by the user. MIMARsim then computes for each locus the four summaries of the polymorphism required by MIMAR, S_1, S_2, S_s and S_f (see section "The summary statistics" in "MIMARdoc.pdf" for details).

You can use this program if you want to test the performance of MIMAR. For example, you can generate a series of data sets under a model with the same fixed set of parameters, run MIMAR on those data sets to estimate the parameters and assess the bias and precision of the estimations provided by MIMAR for the different data sets.

The program is intended to run on Unix, or Unix-like operating systems, such as Linux or MacOsX. The next section describes how to download and compile the program. The subsequent sections described how to run the program.



(a) Isolation-migration (b) Ancestral structure

Figure 1: **The "isolation-migration" model** (a), in which two populations diverged T generations ago from a common ancestral population. The parameters θ_1, θ_2 and θ_A, are the population mutation rates for populations 1, 2 and the ancestral population, respectively. μ is the mutation rate per bp and N_1, N_2 and N_A are the diploid effective sizes of the first, second and ancestral populations, respectively. The split time in generations is T, m is the symmetrical migration rate between populations per generation such that, $M = 4N_1m$ is the expected number of individuals in population 2 replaced by migrants from population 1 each generation. b) A model of isolation from a structured ancestral population. In this case, T is the time at which the ancestral structure starts while τT is the population split time in generations. m and $m_\circ$ are the symmetrical migration rates between populations per generation after and before the split, respectively (i.e., $M_\circ = 4N_1m_\circ$ is the expected number of individuals in the ancestral population 2 replaced by migrants from the ancestral population 1 each generation). Note that the population structure for $\tau T < t < T$ differs from that for $t < \tau T$.

Contents

What changed since the last version?

- March 18, 2010: I added objects and functions in order to specify the random seeds from the command line. The files "mimarsim.c", "mimarsim.h", "params.c", "rand1.c", "rand2.c", "rand1t.c" and "rand2t.c" were changed.

- February11, 2010: I added the file "make_gametes_noanc.c". The program MIMARsim_noanc simulates polymorphism data under the isolation-migration model with parameters specified by the user and creates an input file that can be analyzed by MIMAR_noanc, i.e., with the summary statistics of the polymorphism calculated assuming unknown ancestral states. A section about this option was added to "MIMARsimdoc.pdf".

- December 28, 2009: The file "params.c" of MIMARsim was changed:

  - The locus-specific recombination rates are now calculated by $\rho_y = w_y\rho(Z_y - 1)$ instead of $w_y\rho(Z_y)$.
  - I added several options to obtain the locus-specific population recombination rates.
  - I changed the instances of "calloc(1..." by "malloc(...".

  I updated "MIMARsimdoc.pdf" accordingly and tried to clarify many points in the documentation. Note that the scaling of the recombination rate can be confusing (at least for me) so be careful when setting recombination information. Thanks to Peter Andolfatto who helped me through my confusions on getting the recombination rates right.

- November 27, 2009:

  - The file "mimarsim.c" of MIMARsim was changed.: I fixed potential memory leaks.
  - The file "streec.c" of MIMARsim was changed: I fixed a bug on memory allocation that occurred when $n_{1y} + n_{2y} \gg n_{11} + n_{21}$ for any $y \in [2, Y]$. Thanks to Yongshuai Sun who mentioned the bug to me.

- March 3, 2009:

  - The file "params.c" of MIMARsim was changed: I corrected a bug on memory allocation. Thanks Susan J. Miller for mentioning this bug to me.
  - The file "mimarsim.c" of MIMARsim was changed so that now when the prior on gene flow is "l *a b*", *a* and *b* can both be negative and the summary output file show the estimate of the gene flow rate. Thanks to Camille Roux for highlighting this problem to me.
  - The documentation was changed. I added a figure to help calculate the summary statistics.

- November 5, 2008: The file "params.c" of MIMARsim was changed: I added the information on the switches allowing the user to generate ancestral structure. Thanks Martin Lascoux for telling me of this lack.

- May 29, 2008: The file "params.c" of MIMARsim was changed: I removed several check flags to allow greater freedom to the user. Thanks Armando Geraldes for mentioning the problem to me.

- September 18, 2007: The files "params.c" and "mimarsim.h" of MIMARsim were changed. In the previous version, the locus names could not start with a number. Now the locus names can be any string of up to 50 characters.

- October 9, 2007: the files "makefile" and "mimarsim.c" of MIMARsim were changed. In the previous versions, an unused function referred to the class "tajd.c". Using the compilation line reported in the section "Downloading and compiling" lead to errors. I commented any references to the class "tajd.c" and the related unused functions so that this compilation error does not occur any longer.

Downloading and compiling

All relevant files are included in the tar file "mimar.tar" available at http://mplab.bsd.uchicago.edu/dataNprograms.htm. Download this tar file to your machine then extract the files from the archive with: tar -xvf mimar.tar. After extracting, type cd mimarsimdir/ and compile the program by typing:

gcc -o mimarsim mimarsim.c params.c make_gametes.c streec.c rand$X$.c -lm

(X is either 1, 2, 1t or 2t) or alternatively, by typing make, which contains this compilation line with optimization and rand1.c.

The choice of compilation depends on which pseudo-random number generator the user has available. "rand1.c" and "rand2.c" call drand48() and drand(), respectively. With "rand1.c" and "rand2.c", MIMARsim first looks for the file "seedmimar" to find the seed values for initializing the random number generator. If no "seedmimar" file is found, the generator is seeded with a default value. In either case, the seed is printed on the second line of the standard output, so that the exact same data set can be generated again if desired. When the simulation is finished, the state of the random number generator is output to "seedmimar". In this way, each time MIMARsim is invoked, a new data set is produced. If you want to produce the same data set, "seedmimar" can be edited to contain the value(s) indicated on the second line of the standard output. (The program can also be compiled with "rand1t.c" and "rand2t.c", which use the system clock for seeding the generators and does not use the file "seedmimar" at all.)

The input file

The user needs to provide the following information for each locus. For locus y:

- The locus name, $\texttt{Name}_y$. Any string of up to 50 characters.
- The length of the locus in base pairs, Z_y.
- The inheritance scalar, x_y (i.e., 1 for autosomal loci, 0.75 for X- and 0.25 for Y- and mtDNA-linked loci).
- The mutation rate variation scalar, v_y (which can be estimated e.g., from divergence data: i.e., if the mutation rates across loci are not homogeneous, set v_y to the ratio of observed to expected divergence at each locus).
- The recombination inheritance and rate variation scalar, w_y.
  - w_y is usually set to ω_y, the ratio of the locus-specific population recombination rate per bp over ρ (i.e., $\omega_y = 0$ for the mtDNA and Y, 0.5 for X and 1 (0.5) for autosomes in mammals (in *Drosophila*)).
  - If you assume that an estimate of the locus-specific population recombination rate per bp is available for each locus from linkage disequilibrium analysis, $\widehat{\rho_{\circ_y}} = \widehat{4N_1 c_y}$, the scalar w_y can be set to $\omega_y \widehat{\rho_{\circ_y}}$ to incorporate this knowledge in the estimation (with the switch "-r 1", see section "Crossing over" for details). Note that the product $\omega_y \widehat{\rho_{\circ_y}}$ is the scaled sex-averaged locus-specific population recombination rate per bp, i.e., for an X-linked locus c is the female recombination rate and $\omega_y = \frac{1}{2}$ so that $\omega_y \widehat{\rho_{\circ_y}} = \widehat{2N_1 c}$.

  - If you assume that an estimate of the locus-specific recombination rate per bp is available for each locus from pedigree analysis, $\widehat{c_y}$, the scalar w_y can be set to $\omega_y \widehat{c}$ to incorporate this knowledge in the estimation (with the switch "`-r 2`", see section "Crossing over" for details). Note that the product $\omega_y \widehat{c_y}$ is the scaled sex-averaged locus-specific recombination rate per bp, i.e., for an X-linked locus, $\widehat{c_y}$ is the estimated female recombination rate so the scaled sex-averaged recombination rate is $\frac{1}{2}\widehat{c_y} = \omega_y \widehat{c_y}$.

- The sample size for the locus in population 1 and 2, n_{1_y} and n_{2_y}, respectively.

To provide this information, use the switch "`-lf input`", to specify the input file name containing the information as described below for Y loci:

$\texttt{Name}_1$	Z_1	x_1	v_1	w_1	n_{1_1}	n_{2_1}
$\texttt{Name}_2$	Z_2	x_2	v_2	w_2	n_{1_2}	n_{2_2}
...						
$\texttt{Name}_Y$	Z_Y	x_Y	v_Y	w_Y	n_{1_Y}	n_{2_Y}

Information for a locus can be separated by a tab or space. Information for each locus needs to be in a single line. Before the loci information, any other information can be included, but it must end by "`//`" to specify that the program should ignore this text.
Below is an example for a four locus data set, in which `locus1_autosom` is autosomal and `locus2_Xlinked`, `locus3_Ylinked` and `locus4_mtDNA` are X- Y- or mtDNA-linked, respectively. The recombination scalar was set accordingly and I assumed that the mutation rate on the mtDNA was 2×10^{-7} (see the file "`inputsim`"):

```
Name           length x_y  v_y w_y n_1 n_2 //
locus1_autosom 1000   1    1   1   10  10
locus2_Xlinked 1000   0.75 1   0.5 10  10
locus3_Ylinked 1000   0.25 1   0   10  10
locus4_mtDNA   1000   0.25 10  0   10  10
```

The basic command line

`mimarsim` ***Y*** `-lf` *input* `-u` $\boldsymbol{\mu}$ `-t` $\boldsymbol{\theta_1}$ `-ej` ***T*** `[options]`

This line shows the simplest usage of `MIMARsim`. There is one argument followed by the parameters (introduced by switches, such as "`-t`"). The argument Y, must appear first, while the switches can appear in any order.

Y	The number of loci considered.
μ	The generational mutation rate per bp.
$\theta_1 = 4N_1\mu$	The population mutation rate per bp for the first population. When the other population mutation rates are not specified, they are equal to θ_1.
T	The split time in generations, at which, backward in time, all lineages in population 2 are moved to population 1.
`input`	The input file name containing the information on the loci with their S statistics (see section "The input file").
`[options]`	A list of any options/switches described in the next sections.

The migration rate is zero by default. If the user provides a value for a parameter, it is fixed; otherwise, if a prior range is provided, the parameter is estimated (see section "Setting the range of the prior distributions"). The user needs to specify θ_1 and T because these two parameters are the minimum information required to built the simplest isolation-migration model, in which the two populations split T generations

ago without subsequent gene flow, and in which the ancestral and descendant populations have the same population mutation rate, θ_1.

In the following basic command line `MIMARsim` will simulate data for the loci defined in the file `"inputsim"` (see example in section "The input file") for a model with θ_1 and T are fixed,: $\theta_1 = \theta_2 = \theta_A = 0.005$, $T = 10,000$ generations and $M = 0$. It will print the information for each loci given `"inputsim"` followed by the four summary statistics of the simulated polymorphism in the standard output file named `"outputsim"` (see section "The standard output").

```
mimarsim 4 -lf inputsim -u 2e-8 -t .005 -ej 1e4 >outputsim
```

Providing the other parameters of the model

To provide information about the `other parameters` of the isolation-migration model, the user needs to use either of the following switches:

-n θ_2 -N θ_A -M M

$\theta_2 = 4N_2\mu$	The population mutation rate per bp for the second population.
$\theta_A = 4N_A\mu$	The ancestral population mutation rate per bp.
$M = 4N_1m$	The expected number of migrants between the two populations each generation, where m is the symmetrical migration rate between the two populations. Note that M is defined in term of N_1 (see section "Spatial structure and migration:" in `"msdoc.pdf"` for further details).

In the following example, `MIMARsim` will proceed as before, but with the parameter values: $\theta_1 = 0.005$, $\theta_2 = 0.003$, $\theta_A = 0.005$, $T = 10,000$ generations and $M = 0.7$.

```
mimarsim 4 -lf inputsim -u 2e-8 -t .005 -n .003 -ej 1e4 -N .005 -M .7
```

Setting the range of the prior distributions

Any or all of the parameters θ_1, θ_2, θ_A, T, M can be sampled from prior distributions instead of fixed. For each parameter to be drawn from prior distributions, the user can provide the bounded support of the uniform prior distribution (for M, the uniform prior is on $\ln(M)$). The ranges are given as follows:

`-t u` a b : draws θ_1 from Uniform$[a, b]$.

`-n u` a b : draws θ_2 from Uniform$[a, b]$.

`-ej u` a b : draws T from Uniform$[a, b]$.

`-N u` a b : draws θ_A from Uniform$[a, b]$.

`-M l` a b : draws $\ln(M)$ from Uniform$[a, b]$.

In the following example, the parameters θ_1, T and M will be drawn from prior distributions. The prior range for θ_1 is Uniform$[0.001, 0.01]$, the prior range for T is Uniform$[0, 100000]$ generations, the expected number of migrants, M, will range between $[0.135, 7.389]$. Since θ_2 is not provided it will be equal to θ_1 at every step, while θ_A is fixed to 0.005.

```
mimarsim 4 -lf inputsim -u 2e-8 -t u .001 .01 -ej u 0 1e5 -N .005 -M l -2 2
```

The standard output

An example of a standard output from `MIMARsim` is reported below (see the file `"outputsim"`):

```
./mimarsim 4 -lf inputsim -u 2e-8 -t .005 -ej 1e4
3579 27011 59243
Parameter values 0.005 0.005 0.04 10000 0.005 0 0
Name           length x_y  v_y w_y n_1 n_2 S_1 S_2 S_s S_f //
locus1_autosom 1000   1    1   1   10  10  14  1   11  0
locus2_Xlinked 1000   0.75 1   0.5 10  10  2   5   2   0
locus3_Ylinked 1000   0.25 1   0   10  10  2   2   0   0
locus4_mtDNA   1000   0.25 10  0   10  10  9   20  14  0
```

The first line is the command line. The second line lists the random number seeds (see section "Downloading and compiling"). The third line lists the values of the isolation-migration model parameters. In order:

1. θ_1
2. θ_2
3. $t = \frac{T}{4N_1}$ in coalescent unit
4. T in generations
5. θ_A
6. $M_{12} = 4N_1 m_{12}$
7. $M_{21} = 4N_1 m_{21}$

When the migration rate is symmetrical as here, the values for M_{12} and M_{21} are both equal to $m = 4N_1 m$. The next line is the header of the information given for each locus ending with `"//"` (required to specify `MIMAR` that the information on the loci start here when the standard output file is used as input to `MIMAR`):

$\texttt{Name}_1$	Z_1	x_1	v_1	w_1	n_{1_1}	n_{2_1}	S_{1_1}	S_{2_1}	S_{s_1}	S_{f_1}
$\texttt{Name}_2$	Z_2	x_2	v_2	w_2	n_{1_2}	n_{2_2}	S_{1_2}	S_{2_2}	S_{s_2}	S_{f_2}
...										
$\texttt{Name}_Y$	Z_Y	x_Y	v_Y	w_Y	n_{1_Y}	n_{2_Y}	S_{1_Y}	S_{2_Y}	S_{s_Y}	S_{f_Y}

The first seven values are the information from the input file followed by the four summaries of the simulated polymorphism, in order:

1. S_1 The number of derived polymorphisms unique to the samples from population 1.
2. S_2 The number of derived polymorphisms unique to the samples from population 2.
3. S_s The number of polymorphisms with shared derived alleles between the two samples.
4. S_f The number of polymorphisms with fixed alleles in either sample.

More complex models

Crossing over

Fixed ρ across loci.

`mimarsim` Y `-lf input -u` μ `-t` θ_1 `-ej` T `-r` ρ

To include crossing-over in the model, use the switch "`-r` ρ " to specify the population cross-over rate parameter, $\rho = 4N_1 c$, where c is the probability of cross-over per generation per bp. For each locus y with recombination scalar $w_y = \omega_y > 0$ (specified in the input file, see section "The input file"), `MIMARsim` will generate genealogies under a coalescent model with recombination (Hudson, 1983). Z_y is the length for

locus y and ω_y is the ratio of the locus-specific population recombination rate per bp over ρ (i.e., $\omega_y = 0$ for the mtDNA and Y, $\frac{1}{2}$ for X and 1 ($\frac{1}{2}$) for autosomes in mammals (in *Drosophila*)). `MIMARsim` obtains the locus-specific population recombination rate $\rho_y = \omega_y 4N_1 c(Z_y - 1)$ in all the options with the switch "`-r`". For example, for the following command line, for `locus1_autosom` and `locus2_Xlinked` in `"inputsim"` (see section "The input file" and the file `"inputsim"`), `MIMARsim` will calculate the locus-specific population recombination rates with $1 \times 999 \times 0.005 \sim 5$ and $0.5 \times 999 \times 0.005 \sim 2.5$, respectively:

```
mimarsim 4 -lf inputsim -u 2e-8 -t .005 -ej 1e4 -r .005
```

Fixed locus-specific ρ.

The user can fix the locus-specific recombination rates assuming that estimates are available as follows:

- "`-r 1`": If one assumes that an estimate of the locus-specific population recombination rate per bp, $\widehat{\rho_{\circ_y}} = \widehat{4N_1 c_y}$, is available for locus y from linkage disequilibrium analysis, set the recombination scalar in the input file to $w_y = \omega_y \widehat{\rho_{\circ_y}}$. To specify that estimates of the population recombination rates are known, use the switch "`-r 1`". In this case, `MIMARsim` obtains the population recombination rate at a locus with $w_y(Z_y - 1)$. For example, if you know that the population recombination rate per bp for `locus1_autosom` is 0.005 and for `locus2_Xlinked` is 0.008, then the recombination scalars will be $w_1 = 1 \times 0.005$ and $w_2 = \frac{1}{2} \times 0.008 = 0.004$ (see section "The input file" and the file `"inputsim_4Nc"`). For the following command line, `MIMARsim` will calculate the locus-specific population recombination rates with $999 \times 0.005 \sim 5$ and $999 \times 0.004 \sim 4$ for the first two loci of the input file `"inputsim_4Nc"`, respectively:

```
mimarsim 4 -lf inputsim_4Nc -u 2e-8 -t .005 -ej 1e4 -r 1
```

  ATTENTION: Make sure that the product $\omega_y \widehat{\rho_{\circ_y}}$ is the scaled sex-averaged locus-specific population recombination rate per bp, i.e., for an X-linked locus, c is the female recombination rate and $\omega_y = \frac{1}{2}$ so that $\omega_y \widehat{\rho_{\circ_y}} = 2\widehat{N_1 c_y}$.

- "`-r 2`": If one assumes that an estimate of the locus-specific recombination rate per bp, $\widehat{c_y}$, is available for locus y from pedigree analysis, set the recombination scalar in the input file to $w_y = \omega_y \widehat{c_y}$. To specify that estimates of the recombination rates are known, use the switch "`-r 2`". In this case, `MIMARsim` obtains the population recombination rate at a locus with $w_y \frac{\theta_1}{\mu}(Z_y - 1)$. For example, if you know that recombination rate per bp for `locus1_autosom` is 1×10^{-8} and for `locus2_Xlinked` is 4×10^{-8}, then the recombination scalars will be $w_1 = 1 \times 10^{-8}$ and $w_2 = \frac{1}{2} 4 \times 10^{-8} = 2 \times 10^{-8}$ (see section "The input file" and the file `"inputsim_c"`). For the following command line, `MIMARsim` will calculate the locus-specific population recombination rates with $999 \frac{\theta_1}{\mu} \times 10^{-8} \sim 2.5$ and $999 \frac{\theta_1}{\mu} 2 \times 10^{-8} \sim 5$ for the first two loci of the input file `"inputsim_4Nc"`, respectively:

```
mimarsim 4 -lf inputsim_c -u 2e-8 -t .005 -ej 1e4 -r 2
```

  ATTENTION: Make sure that the product $\omega_y \widehat{c_y}$ is the scaled sex-averaged locus-specific recombination rate per bp, i.e., for an X-linked locus, $\widehat{c_y}$ is the estimate of the female recombination rate so the scaled sex-averaged rate is $\frac{1}{2}\widehat{c_y} = \omega_y \widehat{c_y}$.

Variable locus-specific ρ.

The recombination rate can be allowed to vary across loci and across steps. In this case, the user needs to use the switches

- "`-r e` $\boldsymbol{\lambda}$": the ratio $r = \frac{c}{\mu}$ is drawn from an exponential distribution prior with mean $\frac{1}{\lambda}$ for each recombining locus. In the following example, $\mathrm{E}(r) = 0.6$:

```
mimarsim 4 -lf inputsim -u 2e-8 -t .005 -ej 1e4 -r e 1.667
```

- "`-r n` $\boldsymbol{\nu}$ $\boldsymbol{\sigma}$": the ratio $r = \frac{c}{\mu}$ is drawn from a normal distribution prior with mean υ and standard deviation σ for each recombining locus. In the following example, r is chosen from Normal$(1, 0.1)$:

```
mimarsim 4 -lf inputsim -u 2e-8 -t .005 -ej 1e4 -r n 1 .01
```

Note that in these cases, the recombination rates for the loci are nuisance parameters, and are chosen independently across recombining loci. `MIMARsim` obtains the population recombination rate at a locus with $r_y w_y \theta_1 (Z_y - 1)$.

Asymmetrical migration rates

`mimarsim` Y `-lf input -u` μ `-t` θ_1 `-ej` T `-m` $\boldsymbol{i}$ $\boldsymbol{j}$ $\boldsymbol{M_{ij}}$

In all the simulation studies we published to date, we only considered symmetrical migration rates (Becquet and Przeworski, 2007, 2009). However, there is also an option to specify asymmetrical migration rates. Thinking forward in time, the expected number of individuals that migrate from population i into population j each generation is $M_{ij} = 4N_1 m_{ij}$, i and $j \in [1, 2]$, $i \neq j$, where m_{ij}, is the fraction of population j that is made up of migrant from population i every generation. Note that M_{ij} is defined in term of N_1 (see section "Spatial structure and migration:" in `"msdoc.pdf"` for further details) .

To fix M_{ij}, simply add "`-m` i j M_{ij}" to the command line. Alternatively, you can estimate M_{ij} by writing "`-m` i j `l` a b", in which case $\ln(M_{ij})$ is drawn from Uniform$[a, b]$.

The following command line will generate genealogies for the isolation-migration model with randomly sampled number of migrants per generation from population 2 into population 1 and from population 1 into population 2.

```
mimarsim 4 -lf inputsim -u 2e-8 -t .005 -ej 1e4 -m 1 2 l -2 2 -m 2 1 l -2 2
```

Ancestral structure. Use at your own peril!

`mimarsim` Y `-lf inputsim -u` μ `-t` θ_1 `-ej` T `-M` $\boldsymbol{M}$ `-eh` τ `-es` τ $\boldsymbol{p}$ `-eM` τ $\boldsymbol{M_o}$

The switches "`-eh`" ("`-ej`" in `ms`) and "`-es`" have been adapted from `ms`. I used similar command lines to generate data from models with structure in the ancestral population before the split (Becquet and Przeworski, 2009). Be extremely careful when using those switches. "`-eh`" and "`-es`" work for generating simulated data sets with `MIMARsim` but SHOULD NOT be used with `MIMAR` or `MIMARgof`. It is the users responsibility to provide sample configurations, migration rates and past demographic events for which the sampled chromosomes will eventually coalesce.

Below is more information about these switches to generate splits and admixtures (see also `msdoc.pdf`). The parameters of the other switches are also different (see Figure 1b):

`-M` M Set the expected number of migrants between the two populations each generations from time $0 < t < \tau T$ generations to $4N_1 m$.

`-eh` τ Backward in time, all lineages in population 2 are moved to population 1 at time τT (switch "`-eh`").

`-es` τ p At time τT, population 1 is split into subpopulation 1 and a new subpopulation 2. Each ancestral lineage in subpopulation 1 is randomly assigned to subpopulation 1 with probability p and subpopulation 2 with probability $1 - p$ (switch "`-es`"). The size of subpopulation 2 is set to N_1. Migration rates to and from the new subpopulation are assumed to be zero and the growth rate of the new subpopulation is set to zero. Subpopulation 1 retains the same growth rate and migration rates as before the event. In the forward direction this corresponds to population admixture. The size, growth rates and migration parameters for the new subpopulation can be immediately modified by following the `-es` command with appropriate additional `-e` commands. Remember, that if changed population size and growth rates are desired at the same time point, that one must put the size change command first followed by the growth rate change command. This is because the size change command changes the growth rate to zero.

`-M` τ $M_\circ$ Set the expected number of migrants between the two populations each generations from time $\tau T < t < T$ generations (i.e., the gene flow rate between the ancestral populations) to $4N_1 m_\circ$.

`-ej` T The time T specified the oldest time at which a split occurred in the model, in this case the split that lead to structure in the ancestral population.

In the following example, the following sequence of events occurs backward in time: Populations 1 and 2 are diverging in total isolation ($M = 0$). At $\tau T = 750,000$ generations ago, all lineages in population 2 are moved to population 1 (switch "`-eh`"). At the same time, the ancestral population (population 1, the only population left), is evenly and randomly split (switch "`-es`") into subpopulation 1 and a new subpopulation 2 (different from the population 2 that existed before 750,000 generations ago). Simultaneously, the gene flow rate between the two ancestral populations is set to $M_\circ = 1$. Then one million generations ago (switch "`-ej` T"), the ancestral structure stopped: all lineages in subpopulation 2 are moved to subpopulation 1.

```
mimarsim 4 -lf inputsim -u 2e-8 -t u 0 .01 -ej 1e6 -M 0 -eh .75 -es .75 .5 -eM 1
```

Other options conserved from ms. Use at your own peril!

I have not tested these additional options so do not guarantee that they will work properly.

Crossing-over and gene conversion:

`mimarsim` Y `-lf input u` μ `-t` θ_1 `-ej` T `-r` $\boldsymbol{\rho}$ `-c` f $\boldsymbol{\lambda}$

See "`msdoc.pdf`" for details.

Exponentially growing or shrinking population size

`mimarsim` Y `-lf input -u` μ `-t` θ_1 `-ej` T `-G` $\boldsymbol{\alpha}$

See "`msdoc.pdf`" for details.

To set individual populations to have different growth rates, the "`-g` i α_i" command is used to set the growth rate of population i to α_i. See "`msdoc.pdf`" for details.

Past demographic events

It is the users responsibility to provide sample configurations, migration rates and past demographic events for which the sampled chromosomes will eventually coalesce. Note also that the program, as is, can not analyze data sets for more than two populations at a time.

To specify that demographic parameters change at specific times in the past, the "`-e`" switches are used. These switches are: "`-eG`", "`-eg`", "`-eb`" (was initially "`-eN`" in `ms`), "`-en`", "`-eM`", "`-em`". In each case, the first parameter following the switch is τ, the ratio of the time of the event divided by the split time of the isolation-migration model. Thus, the time of the event is τT in generations. The arguments following the time parameter specify populations and other relevant parameters, as indicated in the following list:

`-eG` τ α	Set all growth rates to α at time τT generations.
`-eg` τ i α_i	Set growth rate of population i to α_i at time τT generations.
`-eb` τ x	Set all population mutation rates to $x\theta_1$ at time τT generations.
`-en` τ i x	Set population i mutation rate to $x\theta_1$ at time τT generations.
`-eM` τ x	Set the symmetrical migration rate to x at time τT generations.
`-em` τ i j x	Set $4N_1 m_{ij}$ to x at time τT generations.

The following example specifies the first population mutation rate to one-tenth of its current value (in this case θ_1) between $8000 < t < 10000$ generations.

```
mimarsim 4 -lf inputsim -u 2e-8 -t .005 -ej 1e4 -en 1 1 .1 -en .8 1 1
```

If the ancestral alleles are unknown use `MIMARsim_noanc`

I provide the program `MIMARsim_noanc` which generates the input files for `MIMAR_noanc` with the summary statistics calculated assuming *unknown ancestral alleles*. `MIMARsim` and `MIMARsim_noanc` are identical in all other aspects.

Compiling `MIMARgof_noanc`

To compile the program type:

```
gcc -o mimargof_noanc mimargof.c params.c make_gametes_noanc.c streec.c randX.c tajd.c
                                                                    -lm
```

See section "Downloading and compiling" for more details.

Summary of command line options

The following options are required

`-t` θ_1	Set the population mutation rate per bp to $4N_1\mu$ for population 1 (the default population).
`-t u` a b	Set the prior distribution for θ_1 to Uniform$[a, b]$.
`-u` μ	Set the mutation rate per bp to μ.
`-lf input`	Set the input file name.
`-ej` T	Set the time of split to T generations ago. Backward in time, all lineages in population 2 are moved to population 1 at time
`-ej u` a b	Set the prior distribution of T to Uniform$[a, b]$.

The following options are not required but are useful for the use of `MIMARsim` since they define more complex models.

`-n` θ_2	Set the population 2 mutation rate per bp to $4N_2\mu$.
`-n u` a b	Set the prior distribution of θ_2 to Uniform$[a, b]$.
`-N` θ_A	Set the ancestral population mutation rate to $4N_A\mu$.
`-N u` a b	Set the prior distribution of θ_A to Uniform$[a, b]$.
`-M` M	Set the expected number of migrants between the two populations each generations to $4N_1m$.
`M l` a b	Set the prior distribution of $\ln(M)$ to Uniform$[a, b]$.
`-m` i j M_{ij}	Set the expected number of migrants from population i into population j each generation, i and $j \in \{1, 2\}$, $i \neq j$, to $4N_1m_{ij}$.
`-m` i j `l` a b	Set the prior distribution of $\ln(M_{ij})$ to Uniform$[a, b]$.
`-r` ρ	Set the population recombination rate per bp to $4N_1c$.
`-r e` λ	Set the prior distribution of $r = \frac{c}{\mu}$ to Exponential with mean $\frac{1}{\lambda}$.
`-r n` ν σ	Set the prior distribution of $r = \frac{c}{\mu}$ to Normal(ν, σ).
`-r 1`	Set the locus-specific population recombination rates per bp to the value specified with w in the input file.
`-r 2`	Set the locus-specific recombination rates per bp to the value specified w in the input file.

Switches for ancestral structure. Be extremely careful.

`-eh` τ	Backward in time, all lineages in population 2 are moved to population 1 at time τT generations ago. (The time T specified by "`-ej`" must be the oldest such event in the model).
`-es` τ p	Set the time at which population 1 is split into subpopulation 1 and a new subpopulation 2 at the time τT.

The following options are conserved from `ms`. Use at your own peril!

`-se seed1 seed2 seed3`	Specify the random seeds from the command line.
`-f filename`	Read command line arguments from file `filename`.
`-c` f λ	Set ratio of gene conversion to recombination to f and the track length to λ.
`-G` α	Set growth parameter of all populations to α.
`-g` i α_i	Set growth rate of population i to α_i.

The following options specify events occurring at time τT generations. Up to 10 such switches can be used. It is the user's responsibility to specify times that are compatible with the isolation-migration model. Note that the switch "`-ej`" can be used only once.

`-eG` τ α	Set all growth rates to α at time τT generations.
`-eg` τ i α_i	Set growth rate of population i to α_i at time τT generations.
`-eb` τ x	Set all population mutation rates to $x\theta_1$ at time τT generations.
`-en` τ i x	Set population i mutation rate to $x\theta_1$ at time τT generations.
`-eM` τ x	Set the symmetrical migration rate to x at time τT generations.
`-em` τ i j x	Set $4N_1m_{ij}$ to x at time τT generations.

List of parameters and symbols

$\theta_i = 4N_i\mu$	The population mutation rate per bp per generation for population $i \in \{1, 2, A\}$.
μ	The generational mutation rate per bp.
N_i	The diploid effective size for population $i \in \{1, 2, A\}$.
T	The split time in generations, at which, backward in time, all lineages in population 2 are moved to population 1.
$t = \frac{T}{4N_1}$	The split time in coalescent unit.
τ	The ratio of the time of the event divided by the split time of the isolation-migration mode
$M = 4N_1m$	The expected number of individuals in population 2 replaced by migrants from population 1 each generation (in forward direction).
m	The symmetrical fraction of a population that is made up of migrant from the other population each generation.
$M_{ij} = 4N_1m_{ij}$	The expected number of individuals in population j replaced by migrants from population i (in forward direction), i and $j \in \{1, 2\}$, $i \neq j$.
m_{ij}	The fraction of population j that is made up of migrant from population i each generation.
$\rho = 4N_1c$	The population recombination rate per bp per generation.
c	The generational recombination rate per bp.
$r = \frac{c}{\mu}$	
Y	The number of loci considered.
Z	The locus length in base pairs.
n_i	The sample size for the locus in population $i \in \{1, 2\}$.
x	The inheritance scalar reflecting copy number differences for a locus.
v	The mutation rate variation scalar for a locus.
w	The inheritance and rate variation scalar for recombination rate for a locus.
ω	The ratio of the locus-specific population recombination rate per bp over ρ.
S_i	The number of derived polymorphisms unique to the samples from populations $i \in \{1, 2\}$.
S_s	The number of polymorphisms with shared derived alleles between the two samples.
S_f	The number of polymorphisms with fixed alleles in either sample.

List of files in the directory `"mimarsimdir"`

Program files for `MIMAR`	**Examples of input files**	**Examples of standard output files**	**Other or documentation files**
`make_gametes.c`	`inputsim`	`outputsim`	`makefile`
`mimarsim.c`	`inputsim_4Nc`		`make_gametes_noanc.c`
`mimarsim.h`	`inputsim_c`		`MIMARsimdoc.pdf`
`params.c`			`seedmimar`
`rand1.c`			
`rand1t.c`			
`rand2.c`			
`rand2t.c`			

Downloading other programs and documentations

`MIMAR` and `"MIMARdoc.pdf"` are found in `"mimar.tar"` available at `http://przeworski.uchicago.edu/cbecquet/download.html`.

`ms` and `"msdoc.pdf"` are available at `http://home.uchicago.edu/~rhudson1/source/mksamples.html`.

References

Becquet, C. and Przeworski, M., 2007. A new approach to estimate parameters of speciation models with application to apes. *Genome Res.*, **17**:1505–1519.

Becquet, C. and Przeworski, M., 2009. Learning about modes of speciation by computational approaches. *Evolution*, **63**:2547–2562.

Hudson, R. R., 1983. Properties of a neutral allele model with intragenic recombination. *Theor. Popul. Biol.*, **23**:183–201.