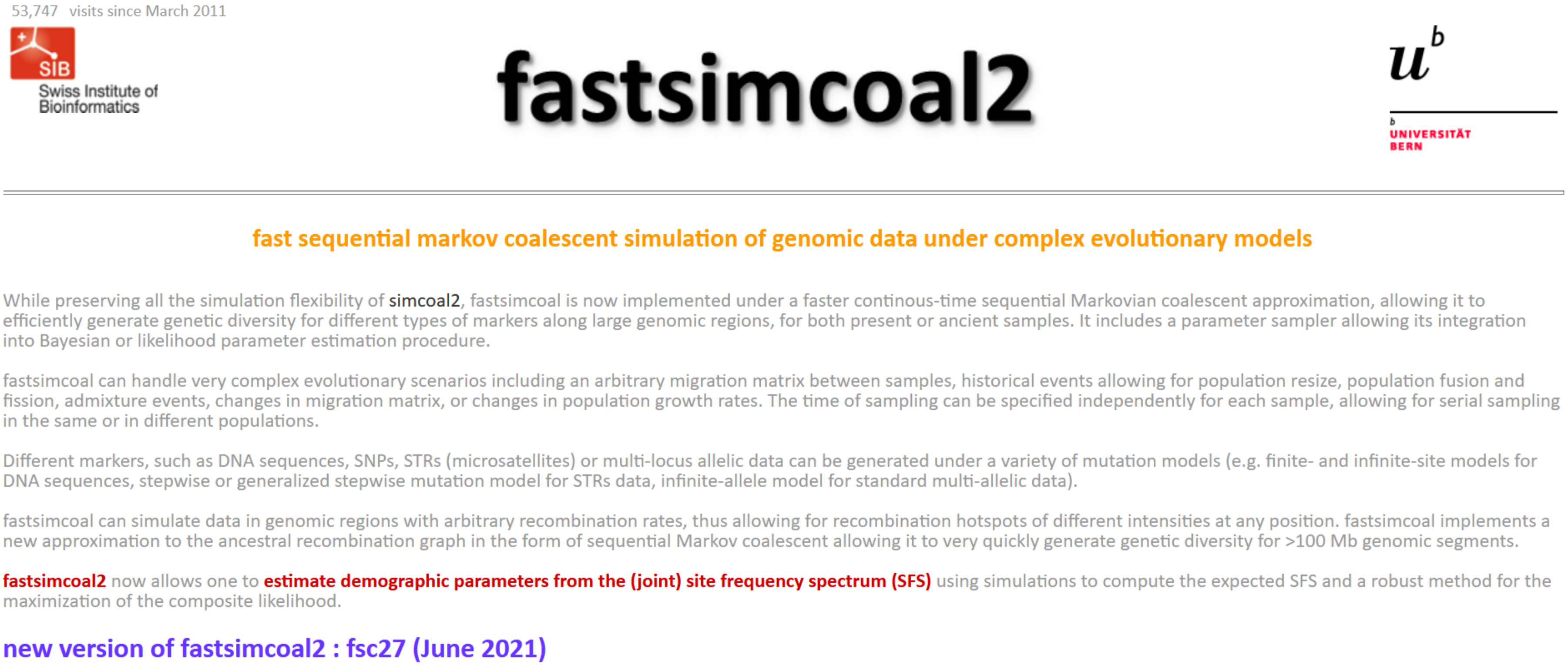
FASTSIMCOAL2 INPUT FILES

Vitor C. Sousa
cE3c, Department of Animal Biology,
Faculty of Sciences of the University of Lisbon
vmsousa@fc.ul.pt

Adapted and based on slides from Laurent Excoffier and fastsimcoal2 manual

https://github.com/vsousa/EG_cE3c



53,747 visits since March 2011

Swiss Institute of Bioinformatics

fastsimcoal2

UNIVERSITÄT BERN

fast sequential markov coalescent simulation of genomic data under complex evolutionary models

While preserving all the simulation flexibility of simcoal2, fastsimcoal is now implemented under a faster continous-time sequential Markovian coalescent approximation, allowing it to efficiently generate genetic diversity for different types of markers along large genomic regions, for both present or ancient samples. It includes a parameter sampler allowing its integration into Bayesian or likelihood parameter estimation procedure.

fastsimcoal can handle very complex evolutionary scenarios including an arbitrary migration matrix between samples, historical events allowing for population resize, population fusion and fission, admixture events, changes in migration matrix, or changes in population growth rates. The time of sampling can be specified independently for each sample, allowing for serial sampling in the same or in different populations.

Different markers, such as DNA sequences, SNPs, STRs (microsatellites) or multi-locus allelic data can be generated under a variety of mutation models (e.g. finite- and infinite-site models for DNA sequences, stepwise or generalized stepwise mutation model for STRs data, infinite-allele model for standard multi-allelic data).

fastsimcoal can simulate data in genomic regions with arbitrary recombination rates, thus allowing for recombination hotspots of different intensities at any position. fastsimcoal implements a new approximation to the ancestral recombination graph in the form of sequential Markov coalescent allowing it to very quickly generate genetic diversity for >100 Mb genomic segments.

fastsimcoal2 now allows one to **estimate demographic parameters from the (joint) site frequency spectrum (SFS)** using simulations to compute the expected SFS and a robust method for the maximization of the composite likelihood.

new version of fastsimcoal2 : fsc27 (June 2021)

http://cmpg.unibe.ch/software/fastsimcoal27/

Bioinformatics, 2021, 1–4
doi: 10.1093/bioinformatics/btab468
Advance Access Publication Date: 23 June 2021
Applications Note

Genetics and population analysis

fastsimcoal2: demographic inference under complex evolutionary scenarios

Laurent Excoffier [1,2,*], Nina Marchi [1,2], David Alexander Marques [3,4,5], Remi Matthey-Doret [1,2], Alexandre Gouy [1,6] and Vitor C. Sousa [1,7]

Examples of observed SFS

1PopExpInst20Mb_DAFpop0.obs

```
1 observations
d0_0     d0_1     d0_2  d0_3  d0_4  d0_5  d0_6  d0_7  d0_8  d0_9  d0_10
19973842 24630    810   173   145   111   88    84    61    56    0
```

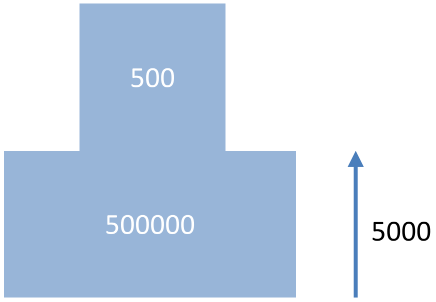
2PopDivMigr20Mb_jointDAFpop1_0.obs

```
1 observations
         d0_0     d0_1  d0_2  d0_3  d0_4  d0_5
d1_0     19985747 8350  1628  360   62    8
d1_1     9660     0     0     0     0
d1_2     4790     0     0     0     0
d1_3     3280     0     0     0     0
d1_4     2490     0     0     0     0
d1_5     1760     13    18    13    19    0
```

2PopDiv20Mb_jointDAFpop1_0.obs

```
1 observations
         d0_0     d0_1  d0_2  d0_3  d0_4  d0_5
d1_0     19985547 8211  1415  316   55    10
d1_1     1266     101   37    16    5     1
d1_2     61142    20    8     2     0
d1_3     48631    12    5     0     0
d1_4     47915    9     2     3     1
d1_5     1189     46    22    19    18    0
```

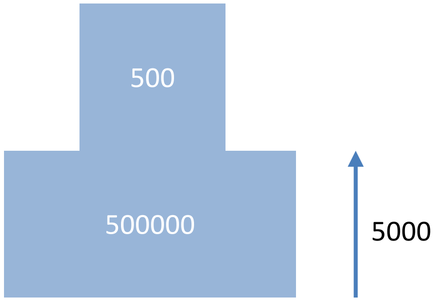
Parameter estimation settings files



1PopExpInst20Mb

Additional files necessary to estimate parameters:
- Template file (TPL) defining the model
- Estimation file (EST) with search range for parameters

Parameter estimation settings files



1PopExpInst20Mb

Additional files necessary to estimate parameters:

- Template file (TPL) defining the model

- Estimation file (EST) with search range for parameters

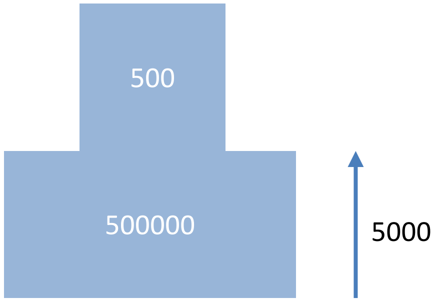
Tags for parameter we want to estimate:
$NPOP$, $TEXP$, $RESIZE$

Template file (filename.tpl)

1PopExpInst20Mb/1PopExpInst20Mb.tpl

```
//Parameters for the coalescence simulation program : fsimcoal2.exe
1 samples to simulate :
//Population effective sizes (number of genes)
$NPOP$
//Samples sizes and samples age
10
//Growth rates: negative growth implies population expansion
0
//Number of migration matrices : 0 implies no migration between demes
0
//historical event: time, source, sink, migrants, new deme size, new growth rate,
migration matrix index
1 historical event
$TEXP$ 0 0 0 $RESIZE$ 0 0
//Number of independent loci [chromosome]
1 0
//Per chromosome: Number of contiguous linkage Block: a block is a set of contiguous loci
1
//per Block:data type, number of loci, per generation recombination and mutation rates
and optional parameters
FREQ  1   0   2.5e-8 OUTEXP
```

Parameter estimation settings files



1PopExpInst20Mb

Additional files necessary to estimate parameters:
- Template file (TPL) defining the model
- **Estimation file (EST) with search range for parameters**

Tags for parameter we want to estimate:
$NPOP$, $TEXP$, $RESIZE$

Estimation file (filename.est)

1PopExpInst20Mb/1PopExpInst20Mb.est

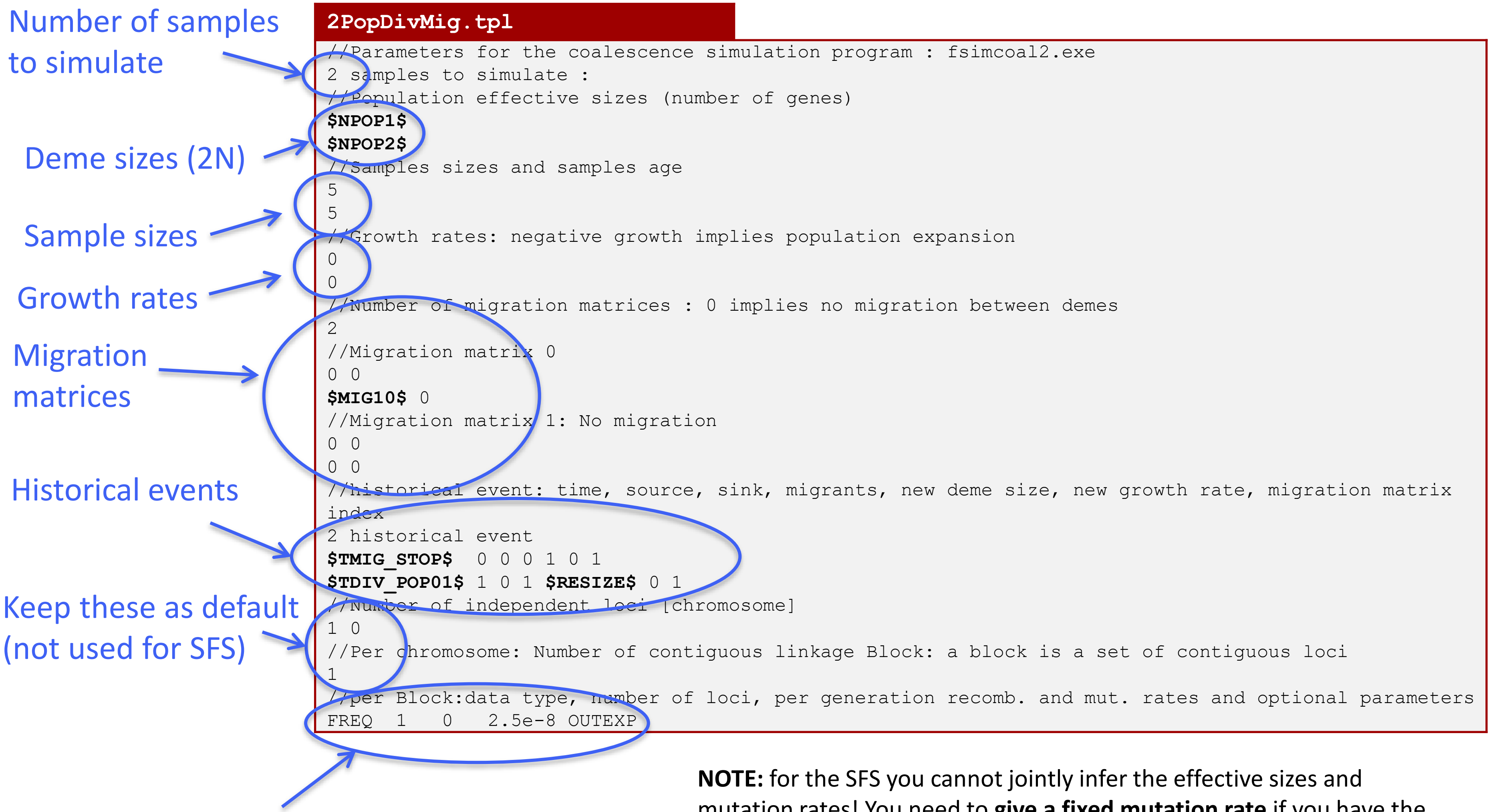
```
// Search ranges and rules file
// ****************************

[PARAMETERS]
//#isInt? #name   #search #min  #max
//all Ns are in number of haploid individuals
1  $NPOP$       logunif  1000   1e7   output
1  $NANC$       logunif  10     1e5   output
1  $TEXP$       unif     10     1e5   output

[RULES]

[COMPLEX PARAMETERS]

0  $RESIZE$   = NANC/NPOP      hide
```

INPUT files for fastsimcoal2:
Defining an evolutionary model with TPL file

2PopDivMig.tpl

```
//Parameters for the coalescence simulation program : fsimcoal2.exe
3 samples to simulate :
//Population effective sizes (number of genes)
$NPOP1_PRESENT$ 0
$NPOP1_ANCIENT$ 1
$NPOP2$         2
//Samples sizes and samples age
8 0
2 2000
4 0
//Growth rates: negative growth implies population expansion
0
0
//Number of migration matrices : 0 implies no migration between demes
2
//Migration matrix 0
0 0
$MIG10$ 0
//Migration matrix 1: No migration
0 0
0 0
//historical event: time, source, sink, migrants, new deme size, new growth rate, migration matrix index
3 historical event
2001 1 0 1 0 0 0
$TMIG_STOP$  0 0 0 1 0 1
$TDIV_POP01$ 1 0 1 $RESIZE$ 0 1
//Number of independent loci [chromosome]
1 0
//Per chromosome: Number of contiguous linkage Block: a block is a set of contiguous loci
1
//per Block:data type, number of loci, per generation recomb. and mut. rates and optional parameters
FREQ  1   0   2.5e-8 OUTEXP
```

INPUT files for fastsimcoal2:
Defining an evolutionary model with TPL file



Number of samples to simulate

Deme sizes (2N)

Sample sizes

Growth rates

Migration matrices

Historical events

Keep these as default (not used for SFS)

2PopDivMig.tpl

```
//Parameters for the coalescence simulation program : fsimcoal2.exe
2 samples to simulate :
//Population effective sizes (number of genes)
$NPOP1$
$NPOP2$
//Samples sizes and samples age
5
5
//Growth rates: negative growth implies population expansion
0
0
//Number of migration matrices : 0 implies no migration between demes
2
//Migration matrix 0
0 0
$MIG10$ 0
//Migration matrix 1: No migration
0 0
0 0
//historical event: time, source, sink, migrants, new deme size, new growth rate, migration matrix
index
2 historical event
$TMIG_STOP$  0 0 0 1 0 1
$TDIV_POP01$ 1 0 1 $RESIZE$ 0 1
//Number of independent loci [chromosome]
1 0
//Per chromosome: Number of contiguous linkage Block: a block is a set of contiguous loci
1
//per Block:data type, number of loci, per generation recomb. and mut. rates and optional parameters
FREQ  1   0   2.5e-8 OUTEXP
```

Definition of genetic data type to simulate.
For SFS inference use:
FREQ 1 0 fixedMutationRate OUTFREQ

NOTE: for the SFS you cannot jointly infer the effective sizes and mutation rates! You need to **give a fixed mutation rate** if you have the number of monomorphic sites.
Otherwise, with "-0" option the mutation rate is ignored.
FREQ indicates you will use the SFS
OUTFREQ means the expected SFS will be output

TPL files

These files are very important! Check carefully all the definitions. Errors in the TPL file are difficult to detect and imply the model specification is incorrect! This means that all inferences will be wrong, and also that all parameter estimates will be incorrect!

Defining population sizes and sample sizes

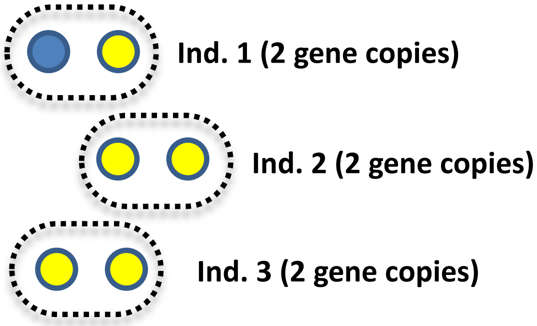
2PopDivMigr10Loci.par

```
//Parameters for the coalescence simulation program : fsimcoal2.exe
2 samples to simulate :
//Population effective sizes (number of genes)
$NPOP1$
$NPOP2$
//Samples sizes and samples age
6
6
//Growth rates: negative growth implies population expansion
0
0
```

Parameter tags

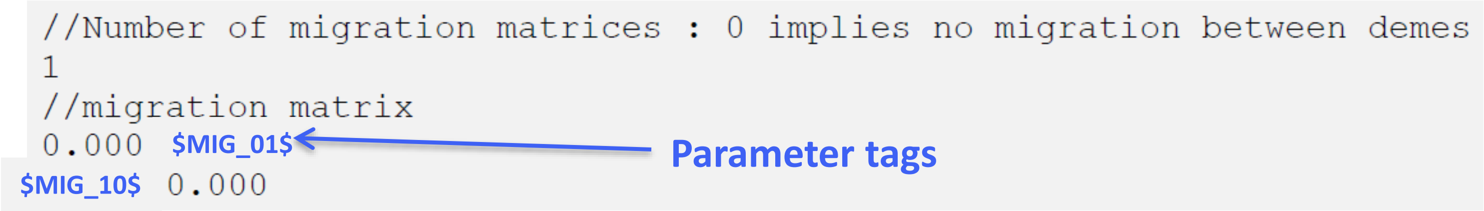
Population effective sizes are given in number of gene copies. For a diploid species with N=500 individuals, this corresponds to a 2N=1000 gene copies, as each individual carries two gene copies at any given site.

The sample size is also given in gene copies. The value of 6 means that we sampled 3 diploid individuals.



TPL files

MIGRATION



```
//Number of migration matrices : 0 implies no migration between demes
1
//migration matrix
0.000  $MIG_01$
$MIG_10$  0.000
```

Parameter tags

The migration matrix can be asymmetric, and in the case the entry m_{ij} list the **migration rates backward in time** from population in row i to population in column j. The above-mentioned matrix states that, for each generation (backward in time), any gene from population 0 has probability MIG_01 to be sent to population 1, and that a gene from population 1 has a probability MIG_10 to move to population 0.

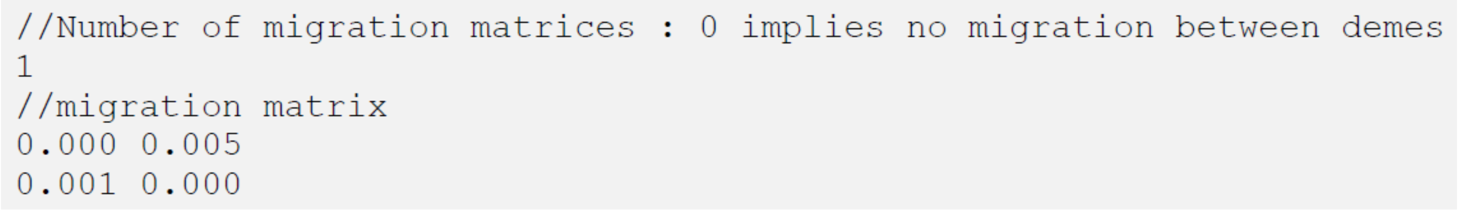
If no migration matrix is defined, no migration is assumed between populations.

1PopStationary10Loci.par

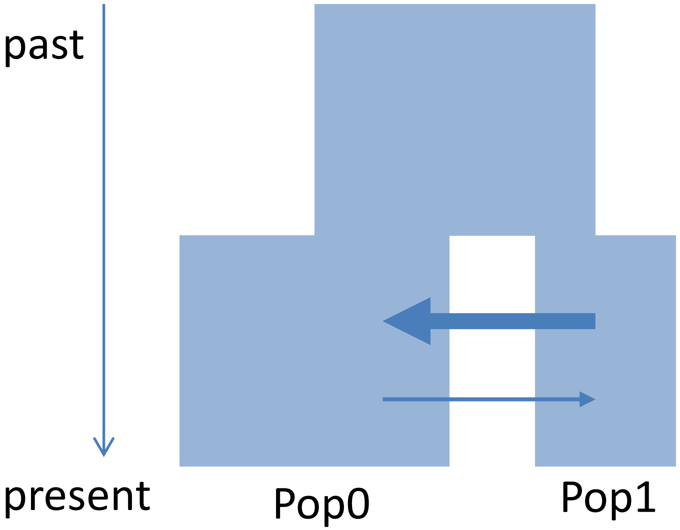
```
//Number of migration matrices : 0 implies no migration between demes
0
```

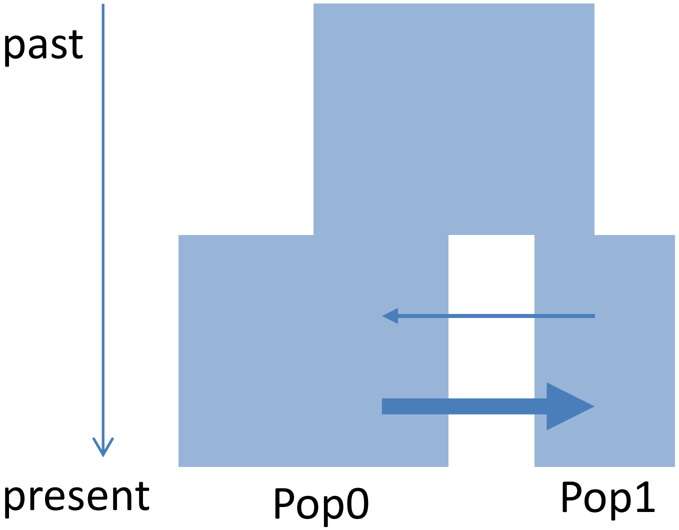
A note on looking backward in time

Assuming that we look **forward in time** and that the size of the arrows are proportional to the migration rate, to what model does the following migration matrix corresponds to?



```
//Number of migration matrices : 0 implies no migration between demes
1
//migration matrix
0.000 0.005
0.001 0.000
```





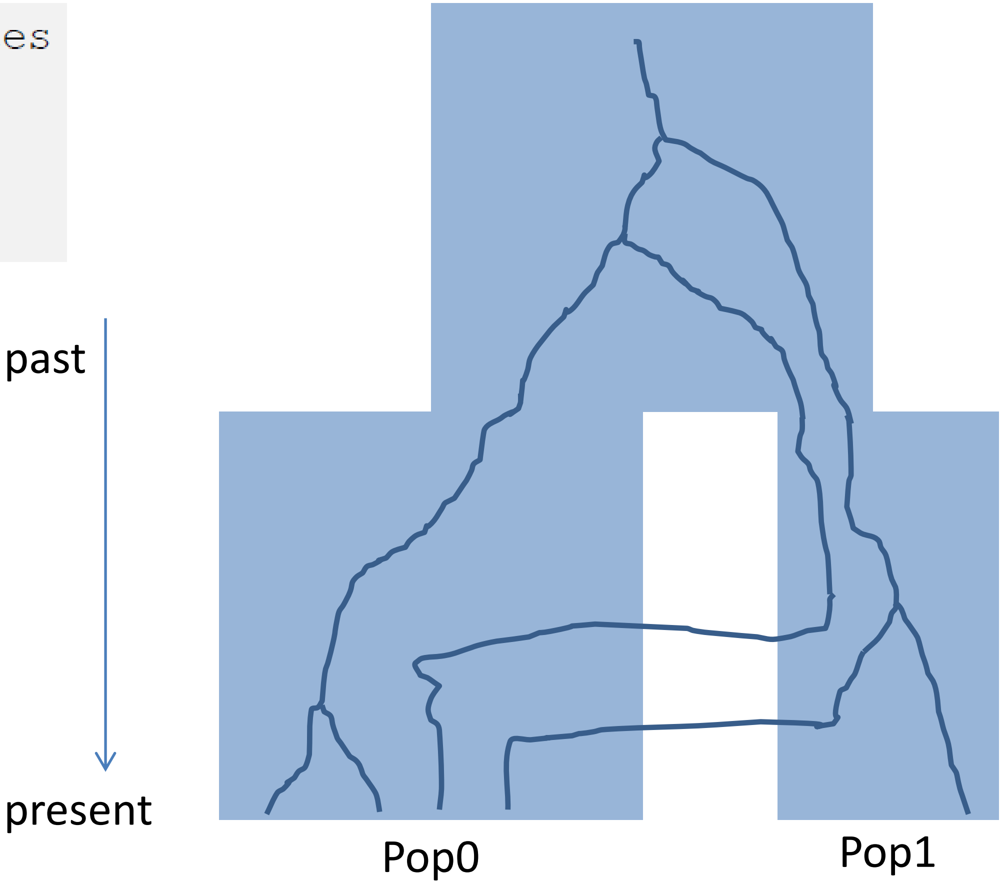
A note on looking backward in time

Assuming that we look **forward in time** and that the size of the arrows are proportional to the migration rate, to what model does the following migration matrix corresponds to?

```
//Number of migration matrices
1
//migration matrix
0.000 0.005
0.001 0.000
```

This means that there are more lineages migrating ("jumping") from pop0 to pop1 backward in time.

Thus, in Pop0 there are many individuals whose ancestors where migrants from Pop1 into Pop0.

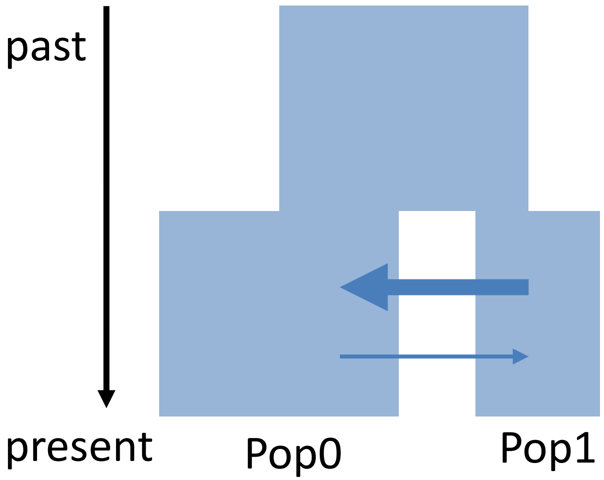


A note on looking backward in time

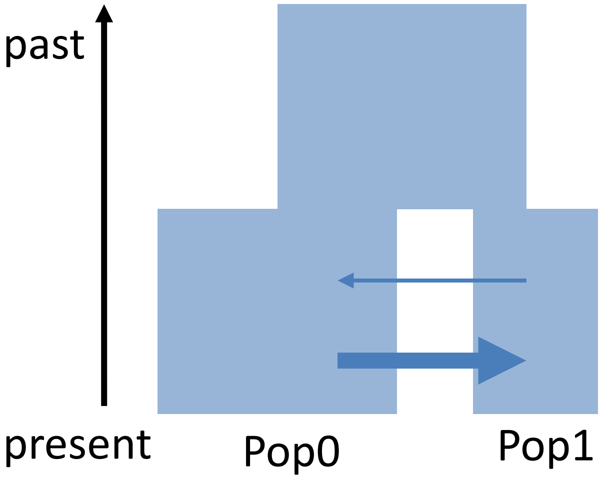
Assuming that we look forward in time and that the size of the arrows are proportional to the migration rate, to what model does the following migration matrix corresponds to?

```
//Number of migration matrices : 0 implies no migration between demes
1
//migration matrix
0.000 0.005
0.001 0.000
```

Note that in the PAR and TPL files everything is backward in time!!



This is the correct model forward in time, meaning there are more migrants moving from pop1 to pop0 each generation.



Backward in time this is the model. Lineages are more likely to move from pop0 to pop1.

Historical events in fastsimcoal2

Historical events can be used to:

- Change the size of a given population
- Change the growth rate of a given population
- Change the migration matrix to be used between populations
- Move a fraction of the genes of a given population to another population. This amounts to implementing a (stochastic) admixture or introgression event.
- Move all genes from a population to another population. This amounts to fusing two populations into one looking backward in time.
- One or more of these events at the same time

Defining the historical events is crucial to have a correct model!

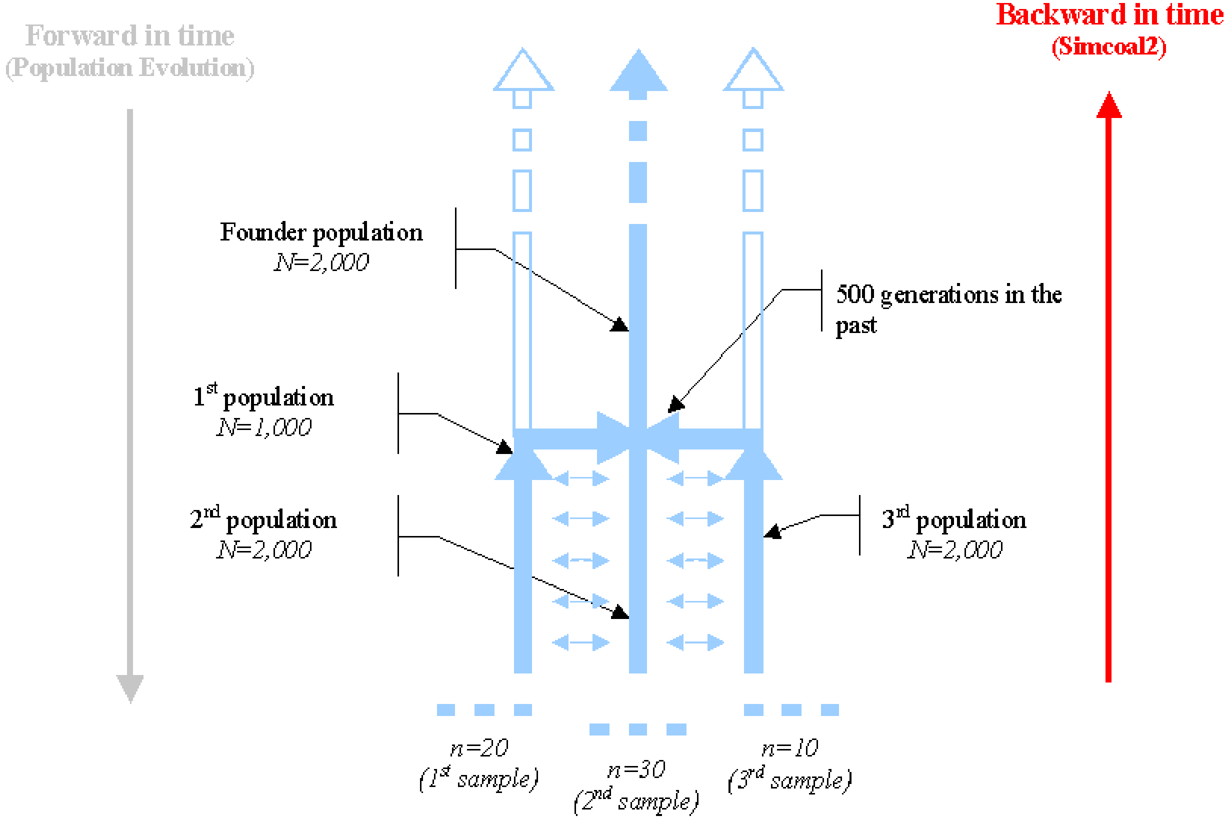
Historical events (backward in time)

Each historical event is coded with a line with the following arguments

time, source, sink, prop. migrants, new deme size, new growth rate, migration matrix index

500 0 1 1 1 0 1

500 2 1 1 1 0 1



500 generations ago, 100% (**proportion migrants=1.0**) of lineages in **pop0** (**source =0**) migrated to **pop1** (**sink=1**). The size of the sink (pop1) remained the same (**new deme size=1.0**, i.e. N2=2000). The new growth rate is zero. The migration rate that is active after the event is given in the migration matrix 1.

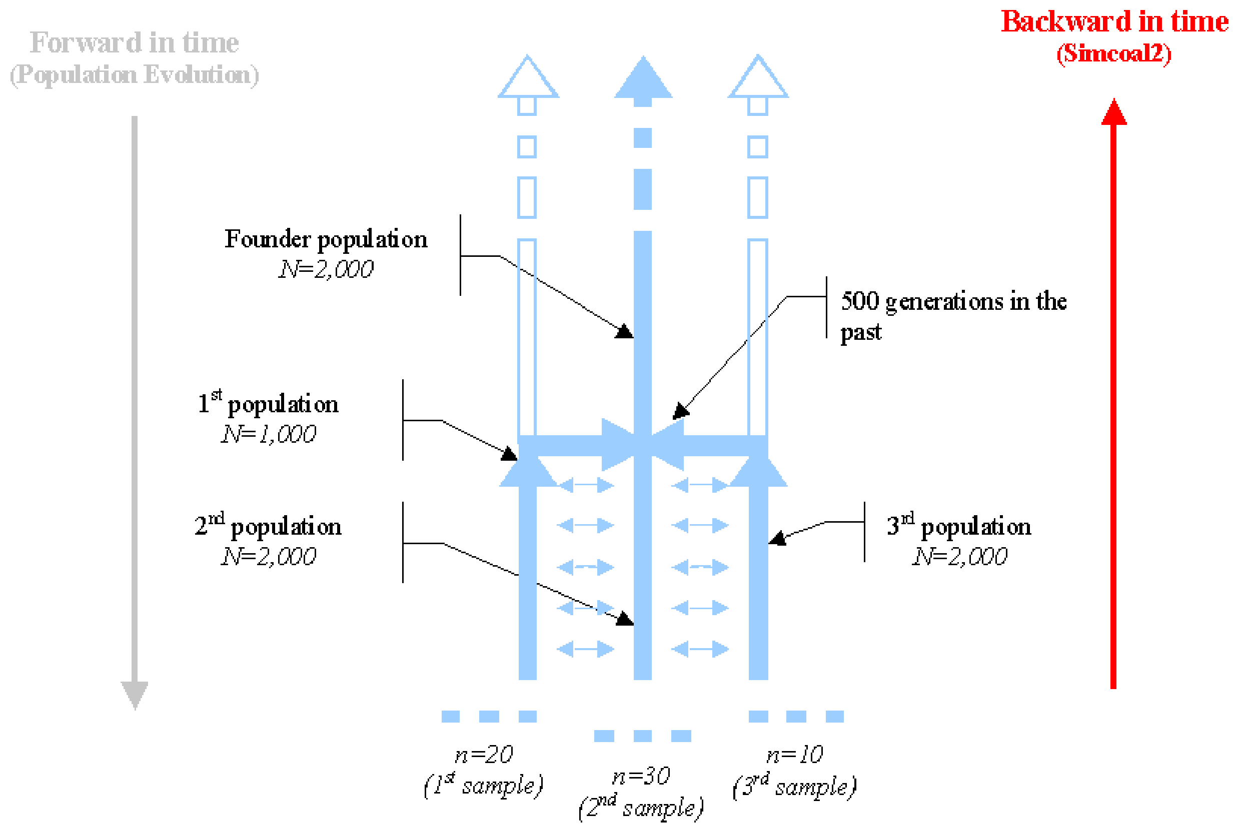
Historical events (backward in time)

Each historical event is coded with a line with the following arguments

time, source, sink, prop. migrants, new deme size, new growth rate, migration matrix index

500 0 1 1 1 0 1

500 2 1 1 1 0 1



500 generations ago, **100%** of lineages (**proportion migrants=1.0**) in **pop2** (**source =2**) migrated to **pop1** (**sink=1**). The size of the sink (pop1) remained the same (**new deme size=1.0**, i.e. N2=2000). The new growth rate is zero. The migration rate that is active after the event is given in the migration matrix 1.

Historical events in fastsimcoal2

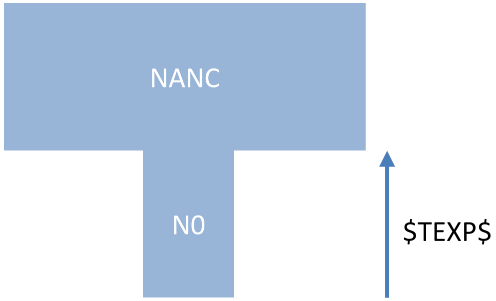
Change the size of a given population

1PopContrInst10Loci.par

```
//Parameters for the coalescence simulation program : fsimcoal2.exe
1 samples to simulate :
//Population effective sizes (number of genes)
1000
//Samples sizes and samples age
10
//Growth rates: negative growth implies population expansion
0
//Number of migration matrices : 0 implies no migration between demes
0
//historical event: time, source, sink, migrants, new deme size, new growth rate, migration matrix index
1 historical event
$TEXP$ 0 0 0 $RESIZE$ 0 0
```

- **$TEXP$** generations ago, 0% (migrants=0) of lineages in pop0 (source) migrated to pop1 (sink). This means that 100% of lineages remained in pop0.
- The sink population (pop0) has a size **$RESIZE$** times larger after the event (**$RESIZE$=$NANC$/$N0$**). Given N0 diploids at time zero, it implies that NANC=N0*RESIZE diploids.
- The migration matrix valid after the event is the migration rate 0. Since it is not defined it imples no migration.

Recent instantaneous demographic contraction



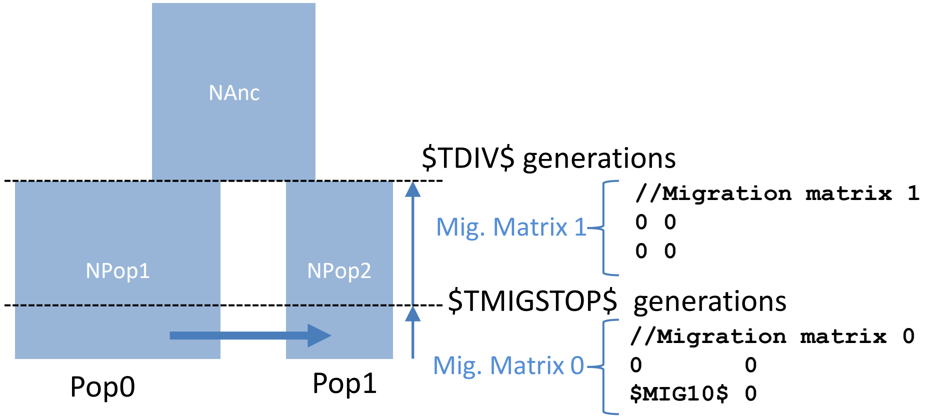
1PopContrInst10loci.par

Historical events in fastsimcoal2

Models with changes in the migration matrix to be used between populations

2PopDivMigr10Loci.par

```
//Number of migration matrices : 0 implies no migration between demes
2
//Migration matrix 0
0        0
$MIG10$ 0
//Migration matrix 1: No migration
0  0
0  0
//historical event: time, source, sink, migrants, new deme size, new growth rate, migration matrix
index
2 historical event
$TMIGSTOP$ 0 0 0 1 0 1
$TDIV$ 1 0 1 $RESIZE$ 0 1
```



```
//Migration matrix 1
0  0
0  0
```

```
//Migration matrix 0
0         0
$MIG10$  0
```

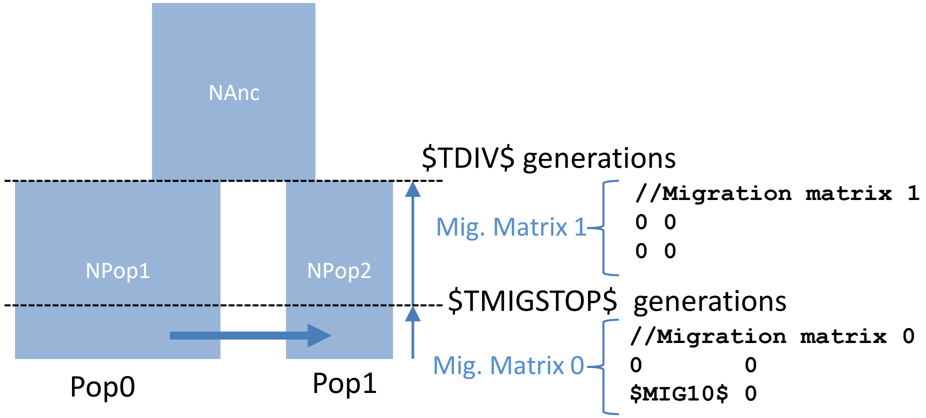
Historical events in fastsimcoal2

Migration matrix can change through time

2PopDivMigr10Loci.par

```
//Number of migration matrices : 0 implies no migration between demes
2
//Migration matrix 0
0        0
$MIG10$ 0
//Migration matrix 1: No migration
0  0
0  0
//historical event: time, source, sink, migrants, new deme size, new growth rate, migration matrix
index
2 historical event
$TMIGSTOP$ 0 0 0 1 0 1
$TDIV$ 1 0 1 $RESIZE$ 0 1
```

- At generation $TMIGSTOP$ in the past, 0% (migrants=0) of lineages migrated from pop0 (source=0) to pop1 (sink=0).
- After the historical event, the deme size of the sink population (pop1) remained the same (new deme size=1).
- After the historical event the growth rate was set to zero.
- After the historical event the migration rate matrix was set to matrix 1, i.e. no migration between populations.



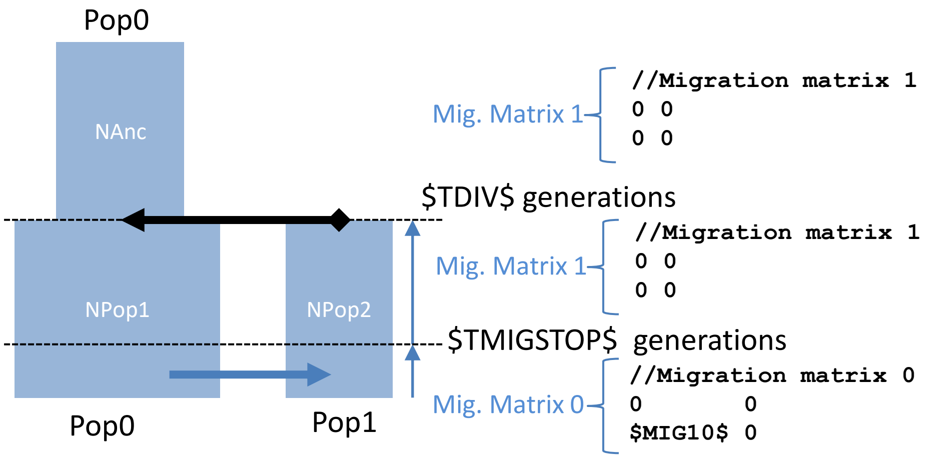
Historical events in fastsimcoal2

Population split (merge populations going backwards in time)

2PopDivMigr10Loci.par

```
//Number of migration matrices : 0 implies no migration between demes
2
//Migration matrix 0
0        0
$MIG10$ 0
//Migration matrix 1: No migration
0  0
0  0
//historical event: time, source, sink, migrants, new deme size, new growth rate, migration matrix
index
2 historical event
$TMIGSTOP$ 0 0 0 1 0 1
$TDIV$ 1 0 1 $RESIZE$ 0 1
```

- At generation $TDIV$ in the past, 100% (migrants=1) of lineages migrated from pop1 (source=1) to pop0 (sink=0).
- After the population split, the deme size of the sink population (pop0) is $NANC$, and hence $RESIZE$=$NANC$/$NPOP0$).
- After the historical event the growth rate of the sink population pop0 is zero.
- After the historical event the migration rate matrix was set to matrix 1, i.e. no migration between populations.



```
//Migration matrix 1
0  0
0  0
```

```
//Migration matrix 1
0  0
0  0
```

```
//Migration matrix 0
0        0
$MIG10$  0
```

Estimation file (.est)

Estimation file ("NoMigSan_Maya.est")

NoMigSan_Maya.est

```
// Search ranges and rules file
// ***************************

[PARAMETERS]
//#isInt? #name   #search.#min  #max
//all Ns are in number of haploid individuals
1  $NPOP1$       unif  10   1e5   output
1  $NPOP2$       unif  10   1e5   output
1  $NANC$        unif  10   1e5   output
1  $NBOTP1$      unif  1    1e3   output   bounded

1  $TDIV$        unif  100  1e4   output
0  $RELTBOT$     unif  1e-5 1     hide     bounded

[RULES]

[COMPLEX PARAMETERS]

0  $RES_BOT_START$ = $NBOTP1$/$NPOP1$      hide
0  $RES_BOT_END$   = $NPOP1$/$NBOTP1$      hide

1  $TBOT_START$ = $TDIV$ * $RELTBOT$    output
1  $TBOT_END$   = $TBOT_START$ + 10     hide

0  $RESIZE0$    = $NANC$/$NPOP1$        hide
```

Each line of [PARAMETERS] section must contain the following:

#isInt? 0 for continuous, 1 for integers

#name Parameter tag name

#search "unif" for uniform scale

"logunif" for log10 scale

#min minimum search range (lower bound)

#max maximum search range. If the keyword bounded is not used, then if likelihood is higher near maximum value, fastsimcoal2 will keep increasing the maximum value. The **bounded** keyword prevents this.

Complex parameters depend on the values of other parameters. Only one operatation per line can be done. Thus, you cannot have something with many operations in a single line:

BLA=(BL * A)+(BLA/LA) - WRONG!

Estimation file (.est)

Estimation file ("NoMigSan_Maya.est")

NoMigSan_Maya.est

```
// Search ranges and rules file
// ****************************

[PARAMETERS]
//#isInt? #name   #search.#min  #max
//all Ns are in number of haploid individuals
1  $NPOP1$       unif  10   1e5   output
1  $NPOP2$       unif  10   1e5   output
1  $NANC$        unif  10   1e5   output
1  $NBOTP1$      unif  1    1e3   output   bounded

1  $TDIV$        unif  100  1e4   output
0  $RELTBOT$     unif  1e-5 1     hide     bounded

[RULES]

[COMPLEX PARAMETERS]

0  $RES_BOT_START$ = $NBOTP1$/$NPOP1$       hide
0  $RES_BOT_END$   = $NPOP1$/$NBOTP1$       hide

1  $TBOT_START$ = $TDIV$ * $RELTBOT$    output
1  $TBOT_END$   = $TBOT_START$ + 10     hide

0  $RESIZE0$    = $NANC$/$NPOP1$        hide
```

Note that complex parameters can be used to define the order of events.

By using a $RELTBOT$ between 1e-5 and 1, and then specifying that

$TBOT_START$ = $TDIV$ * $RELTBOT$

we define that the TBOT_START is always more recent than the time of divergence.

If this is not well specified you can get errors, because events need to happen in a specific order.

Estimation file (.est)

Estimation file ("NoMigSan_Maya.est")

NoMigSan_Maya.est

```
// Search ranges and rules file
// ****************************

[PARAMETERS]
//#isInt? #name   #search.#min  #max
//all Ns are in number of haploid individuals
1  $NPOP1$       unif  10   1e5   output
1  $NPOP2$       unif  10   1e5   output
1  $NANC$        unif  10   1e5   output
1  $NBOTP1$      unif  1    1e3   output  bounded

1  $TBOT_END$      unif  100  1e4   output
0  $TDIV_TBOT_INT$ unif  10   1e3   hide

[RULES]

[COMPLEX PARAMETERS]

0  $RES_BOT_START$ = $NBOTP1$/$NPOP1$      hide
0  $RES_BOT_END$   = $NPOP1$/$NBOTP1$      hide

1  $TBOT_START$  = $TBOT_END$ - 10        output
1  $TDIV$ = $TBOT_END$ + $TDIV_TBOT_INT$ output


0  $RESIZE0$     = $NANC$/$NPOP1$       hide
```

Another solution is to actually estimate the time between time events, as shown on the left.

In this case, we would estimate the parameter $TDIV_TBOT_INT$

And then in complex parameters:

$TDIV$ = $TBOT_END$ + $TDIV_TBOT_INT$

Estimation file (.est)

Estimation file ("NoMigSan_Maya.est")

NoMigSan_Maya.est

```
// Search ranges and rules file
// ****************************

[PARAMETERS]
//#isInt? #name   #search.#min  #max
//all Ns are in number of haploid individuals
1  $NPOP1$       unif  10   1e5   output
1  $NPOP2$       unif  10   1e5   output
1  $NANC$        unif  10   1e5   output
1  $NBOTP1$      unif  1    1e3   output  bounded

1  $TBOT_END$      unif  100  1e4   output
0  $TDIV_TBOT_INT$ unif  10   1e3   hide

[RULES]

[COMPLEX PARAMETERS]

0  $RES_BOT_START$ = $NBOTP1$/$NPOP1$      hide
0  $RES_BOT_END$   = $NPOP1$/$NBOTP1$      hide

1  $TBOT_START$ = $TBOT_END$ - 10         output
1  $TDIV$ = $TBOT_END$ + $TDIV_TBOT_INT$ output


0  $RESIZE0$    = $NANC$/$NPOP1$       hide
```

Finally, a note about inferring bottlenecks associated with founder events.

It is difficult to jointly infer the duration and Effective population size of a bottleneck.

Instead, we can infer the bottleneck intensity, which is given by

$$I_B = \frac{\underline{\text{Time Duration Bottlenek (generations)}}}{(2*\text{Effective size during bottleneck})}$$

Thus, we usually fix the duration of the bottleneck and infer the effective size.

In this case, we fix the duration of the bottleneck to be 10 generations.

If $NBOTP1$ is larger than 500, then actually we would infer no bottleneck, as I_B<0.01 (10/(2*500)).

Balick et al (2015) <u>PLoS Genet</u>. 11(8): e1005436.

EST file simplifications implemented in fastsimcoal2.7

keywords *paramInRange* and *bounded*

- ParamInRange: used when you have parameters that depend on others (e.g. a time of bottleneck younger than divergence time)
- Bounded: used when parameter values cannot be outside the defined limits
- No need to use complex parameters

NoMigSan_Maya.est

```
// Search ranges and rules file
// ****************************

[PARAMETERS]
//#isInt? #name   #search.#min  #max
//all Ns are in number of haploid individuals
1  $NPOP1$       unif  10   1e5   output
1  $NPOP2$       unif  10   1e5   output
1  $NANC$        unif  10   1e5   output
1  $NBOTP1$      unif  1    1e3   output  bounded

1  $TDIV$        unif  100  1e4     output
1  $TBOT_END$    unif  11   $TDIV$  output bounded paramInRange

[COMPLEX PARAMETERS]

1  $TBOT_START$ = $TBOT_END$ - 10    hide
```

TPL file simplifications implemented in fastsimcoal2.7

absoluteResize

Avoid one of the most common errors: In historical events add absoluteResize when using the new sink effective size in absolute rather than relative

With absoluteResize:

NoMigSan_Maya.tpl

```
//historical event:
3 historical event
$TBOT_START$ 0 0 0 $NBOTP1$ 0 0  absoluteResize
$TBOT_END$   0 0 0 $NPOP1$  0 0  absoluteResize
$TDIV$       1 0 1 $NANC$   0 0  absoluteResize
```

NoMigSan_Maya.est

```
1  $NPOP1$        unif  10   1e5   output
1  $NPOP2$        unif  10   1e5   output
1  $NANC$         unif  10   1e5   output
1  $NBOTP1$       unif  1    1e3   output
```

Previous to fsc2.7 without absoluteResize:

NoMigSan_Maya.tpl

```
//historical event:
3 historical event
$TBOT_START$ 0 0 0 $RELBOT1$ 0 0
$TBOT_END$   0 0 0 $RELBOT2$ 0 0
$TDIV$       1 0 1 $RELANC$  0 0
```

NoMigSan_Maya.est

```
1  $NPOP1$        unif  10   1e5   output
1  $NPOP2$        unif  10   1e5   output
1  $NANC$         unif  10   1e5   output
1  $NBOTP1$       unif  1    1e3   output

[COMPLEX PARAMETERS]
0  $RELBOT1$  = $NBOTP1$/$NPOP1$  hide
0  $RELBOT2$  = $NPOP1$/$NBOTP1$  hide
0  $RELANC$   = $NANC$/$NPOP1$    hide
```

EST file functions that you can use

%min%, %max%, if else (condition ? ifTRUE : ifFALSE)

For complex models, you might want to limit some parameters depending on the minimum and/or maximum of other values.

Also, you might want the parameters to take different values conditional on whether a given condition is true or false.

For conditions you can use > (larger than),< (smaller than), == (equal to),!= (not equal to), <= (smaller or equal), >= (larger or equal)

Subsection of an EST file

```
...

[COMPLEX PARAMETERS]
...

1  $maxTsplit2$ = $TSplitAsi$ %max% $TSplitKo$   hide

0  $ResCont$     = $N_Cont$/$N_aKar$                              hide
0  $ResSplitAs$  = $TSplitAsi$ != $maxTsplit2$ ? 1.0 : $ResCont$  hide
0  $ResSplitKo$  = $TSplitKo$ != $maxTsplit2$  ? 1.0 : $ResCont$  hide
```

TPL file simplifications implemented in fastsimcoal2.7

keywords *keep* and *nomig*

When you have changes in migration rates that depend on the order of events, it can become complex to decide what migration matrix to use.

nomig: if this keyword is added at the end of a historical event definition, migrations between demes are suppressed until the end of the current coalescent simulation. If next in line historical events were to specify the use of some new migration matrix, this would be ignored by fastsimcoal.

keep: if instead of a given value for a growth rate or a migration matrix one uses the keyword keep, then the former values of these parameters will be used

Example of TPL with nomig

```
//historical event:
2 historical event
1000 0 0 0 1 0 0 nomig // suppresses migrations between demes for older than 1000 gen
10000 1 0 1 10 0 1     // no migration, even if migration in matrix 1
```

Example of TPL with keep

```
//historical event:
3 historical event
200   0 0 0 1 -0.001 0
1000  0 0 0 1 keep   1    //Keep current growth rate (-0.001) (and change migration matrix
10000 1 0 1 1 0      keep //Keep migration matrix
```

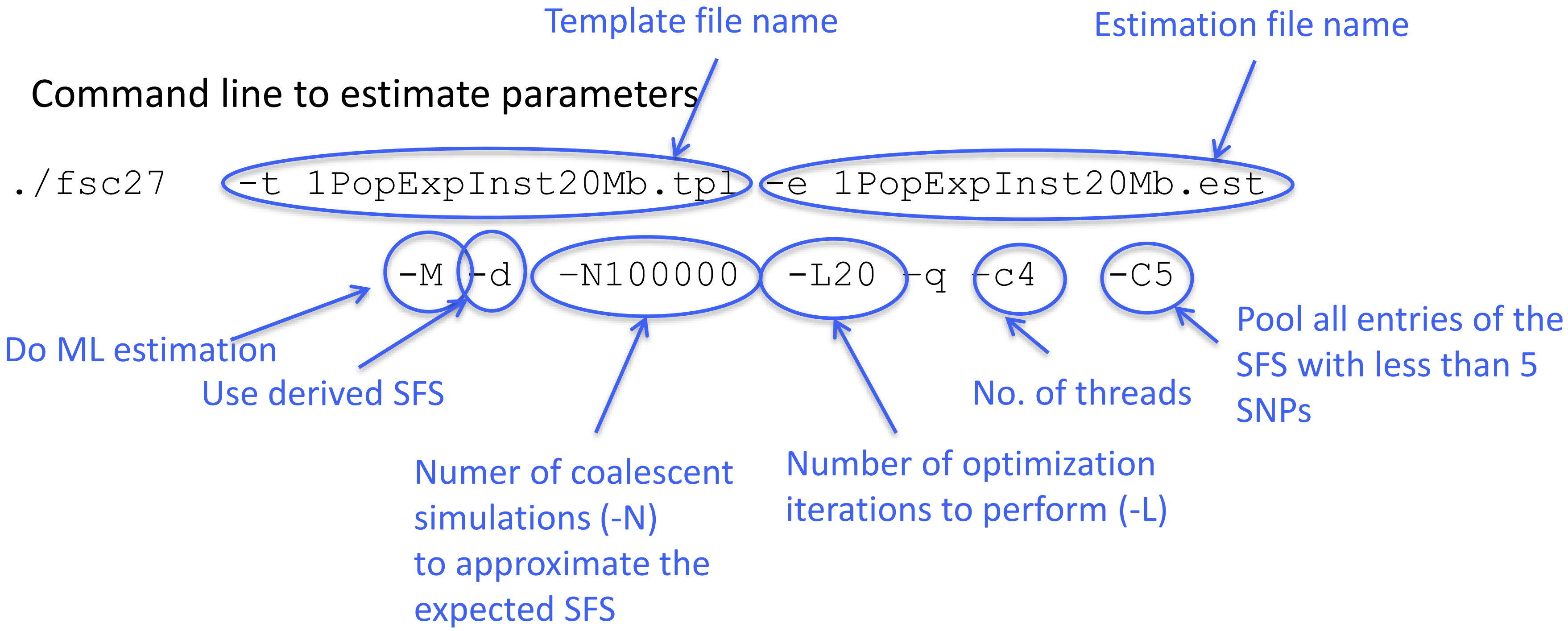
Launching parameter estimations

Command line to estimate parameters

```
./fsc27   -t 1PopExpInst20Mb.tpl -e 1PopExpInst20Mb.est

               -M -d  -N100000  -L20 -q -c4   -C5
```

Observed SFS file must have the same name as template file and extension _DAFpop0.obs. e.g. `1PopExpInst20Mb_DAFpop0.obs`

Launching parameter estimations



Command line to estimate parameters

```
./fsc27   -t 1PopExpInst20Mb.tpl -e 1PopExpInst20Mb.est
          -M -d -N100000 -L20 -q -c4  -C5
```

Observed SFS file must have the same name as template file and extension _DAFpop0.obs. e.g. `1PopExpInst20Mb_DAFpop0.obs`

EST file simplifications implemented since fastsimcoal2.6

keyword *reference* and parameter rescaling

Reference keyword will be used in conjunction of the **–l xx** command line option, which specifies that during xx initial ECM cycles, both monomorphic and polymorphic sites will be used to compute the likelihood. After these xx initial cycles, the likelihood will only be computed (and maximized) based on the polymorphic sites, using the currently estimated reference parameter as fixed, to scale all other parameters. The choice of this (unique) reference parameter is thus important and should be one that affects the total number of polymorphic sites, like the size of an ancestral population, or a divergence time.

Example of est file with reference

```
// Search ranges and rules file
// ****************************

[PARAMETERS]
//#isInt? #name   #search.#min  #max
//all Ns are in number of haploid individuals
1  $NANC$        unif  10   1e5   output reference
1  $NPOP1$       unif  10   1e5   output
1  $NPOP2$       unif  10   1e5   output
1  $NBOTP1$      unif  1    1e3   output  bounded
```