msms User Manual

March 9, 2010

1 Introduction

This document describes how to use *msms*, a tool to generate sequence samples under both neutral models and a single locus selection model. *msms* permits the full range of demographics models provided by ms, which is multiple demes, population growth and decay, as well as a changing number of demes, and other demographic parameters over time. The program is designed to be command line compatible to ms, however no prior knowledge of ms is assumed for this document.

Applications of this program include power studies, analytical comparisons, approximated Bayesian computation among many others. Because most applications require the generation of a large number of independent replicates, the code is designed to be efficient and fast. For the neutral case, it is comparable to ms and even faster for large recombination rates. For selection, the performance is only slightly slower, making this one of the fastest tools for simulation with selection.

The program has been developed with a wide number of possible operating systems and hardware in mind. For this reason, the code has been developed in JAVA and can run on any hardware that supports java 1.6. This includes Mac OS X, all current versions of MS Windows, and most Unix flavors (Linux, Sun, BSD). The JAVA programing language is also popular and widely known which should facilitate the writing of extensions for the program.

1.1 Conventions

msms is a command line program and as such must be run from the shell in Unix and Mac OS X or from a command prompt in windows. Generally this document uses the convection that text entered into a command line will be formatted as follows.

```
>java -jar msms.jar
```

Here the `>` denotes the command line prompt, and you do not type this, also note that some command line prompts will be different depending on the system[1].

[1] `$` are common

The "-" with the following text is referred too as a switch. So in the above example we call the `java` command with the `-jar` switch, with the argument `msms.jar`.

Time is measured from the present into the past, and we use the term *pastward*. This is common for coalescent simulations and is the convention in ms. Time units are always in $4N_e$ generations. The present is defined as the time of sampling. That is when sequencing was done[2]

2 Installation

All relevant files are located at `http://www.mabs.at/ewing/msms/` .[3]

2.1 Recommended

You must have java 6 installed. Note that this is in fact java 1.6 so don't worry about the different names, they are the same thing. This can be downloaded from the sun web site from `http://www.java.com/en/download/` . On OSX machine you just need to ensure you have the latest updates from apple. However by default it will still use java 5 even though you have java 6 installed. To change this go to

```
Applications->Utilities->Java Preferences
```

and ensure that java 6 or better is ticked, and dragged to the top of the lists.

For normal installation, download the zip file and unpack to a directory of your choice. This creates a directory called msms with some subdirectories. In particular you will have a bin directory that holds the binaries, or more to the point the program launchers. You will probably want this directory added to the path. Under Unix and OSX you can use simlinks for the `msms` launching scripts. Note that `msms.exe` is for windows machines only.

The rest of this document assumes that the bin directory is in the path. Thus, you only need to use `msms` at the command line to invoke the program. If this is not the case, the command line may need to be prefixed with more options.

2.2 Pure jar

We also make the program available as a jar file with the correctly configured manifest file. To invoke the program no installation is required other than downloading the `msms.jar` file, and then use java with the `-jar` switch:

```
>java -jar msms.jar
```

Note that this is long hand for the normal command,

[2] At this stage we do not consider temporal sampling of populations. However if there was demand for such a feature, this could be easily added.

[3] The period is not part of the web address.

```
>msms
```

that we use throughout this document. Also recall that by default java will assume a maximum memory size of just 64Mb, so for some simulations the use of the java `-Xmx` switch will be required. If you downloaded the normal package, the `msms.jar` file can be found in the lib subdirectory.

2.3 From Source

The source is also provided as a downloaded zip file. This is not the recommended option unless you wish to modify the source code. We use `ant http://ant.apache.org/` as the build tool. The `build.xml` file is in the subdirectory `ant`. Please check the `readme.txt` included in the source download.

2.4 Git

Please check the web site for instructions on the details of the git repository.

3 Simple Usage and Output Format

The basic command line options are:

```
>msms -N popSize -ms sampleCount reps -t theta
```

Here we generate `sampleCount` samples per simulation with a population size of N_e =`popSize` and a $\theta = 4N_e\mu$ of `theta`. Note that because we are simulating under a neutral model and that all parameters are scaled relative to N_e, changing N_e will have no effect on the outcome. However, this is not the case with selection.

3.1 Output

After running the program the following output is generated:

```
>ms -N 1000 -ms 5 2 -t 1
msms 5 2 -t 1 -N 1000
rnd numbers

//
segsites: 2
positions: 0.50061 0.70488
10
00
00
00
01
```

```
//
segsites: 3
positions: 0.11559 0.32324 0.46842
100
000
011
011
011
```

In this example, we have $N_e = 1000, \theta = 1$ with 5 samples and 2 replicates. The output is the same as ms. However, there needs to be some subtle differences. The first line is simply the command line with the `-ms` switch omitted and the `-N` at the end in order to stay compatible with ms parsing tools. Note that we keep the arguments to the `-ms` switch as per ms. So the first line is the command msms followed by the number of samples and the number of replicates. After this comes the rest of the command line followed by the `-N` switch.

The next line is simply the text `rnd numbers`. This will be changed to permit rerunning the simulation with exactly the same seed values. However the random number generators for *msms* are different from ms and hence we do not output misleading numbers.

After this we have a blank line followed by a line with `//` denoting the start of a sample output. The next 2 lines give us the number of segregation sites followed by their position with the neutral locus in increasing order. By default the neutral locus is on the interval between 0 and 1. However, as we see later the user may specify multiple neutral selected loci over different intervals.

Finally we have 5 lines, one line for each sample, with the haplotype information. The derived allele is denoted with a 1 and the ancestral type with a 0. These are in the same order as the positions list. This data is generated under the infinite sites model. We currently do not support other neutral mutation models.

3.2 Recombination

Recombination can be included in the simulation with the `-r` switch, as in the following example:

```
>msms -N 1000 -ms 5 2 -t 1 -r 1
```

Here we have a recombination rate $\rho = 4N_e r$ where r is the probability of recombination per generation between the ends of a unit length locus. Thus the recombination rate of a locus that is 2 units long will have an effective recombination rate twice as large as a single unit locus.

4 Introducing Selection

4.1 With fixation

We now consider the case of selection on a single loci that we assume goes to fixation. The command line is as follows:

```
>msms -N 1000 -ms 5 2 -t 1 -r 1 -SAA sAA -SaA saA -SF time
```

Here, `sAA` and `saA` are the selection strengths for the homozygote AA and hetrozygote aA genotypes respectively. Selection strength is specified in units of $2N_e s$ and $s = w - 1$ with w as the selection coefficient. We assume diploid populations. Finally we specify the fixation time with the `-SF` switch with time specified pastward and in units of $4N_e$ generations. In this case we assume a single founder (a single beneficial mutation) has gone to selection. Or, in other words we condition on a single beneficial mutation going to fixation. In order for the simulations that use `-SF` switch to work, the demographic history, indeed the full model, must be time invariant. That is, all parameters of the model cannot change over time. We will see shortly the options that permit time variant models.

When we have selection, despite the fact that all parameters are scaled to N_e, the actual value becomes important. The forward simulation uses discrete generations and a discrete population size. Thus, the run time is influenced by how large the `-N` switch is set to. Larger is generally slower. Furthermore the variance of the binomial sampling is dependent on N_e and hence the variance between simulation runs will also depend on N_e. Generally, the performance is good enough to use a realistic value for N_e.

Another consideration when simulating with discrete generation is accuracy compared to continuous approximations. Generally, a very small N_e is undesirable because the probability of a single event in a generation becomes large. Thus, the simulations will tend to diverge from the coalescent that requires that the probability of an event in a generation is low. This can occur with high levels of selection or fast exponential growth[4].

Finally, we must ensure that the parameters will permit the beneficial allele to go to fixation. For example if we set the hetrozygote to have higher fitness than the homozygote, then we never reach fixation, and the simulation will run until the computer runs out of memory.

4.1.1 Example

We permit the setting of the wild type homozygote We have a single diploid population with a constant population size of $N_e = 100000$ and a $\theta = 5$ that is experiencing weak selection $s_{AA} = 200, s_{aA} = 100$ that went to fixation 4000 generations ago. There are 10 samples and we want 1000 replicates. I would use the following command line:

[4]Population size increasing into the present.

```
>msms -N 100000 -ms 10 1000 -t 5 -SAA 200 -SaA 100 -SF 1e-2
```

The first set of output looks like:

```
//
segsites: 3
positions: 0.05509 0.21466 0.70900
000
110
100
100
100
100
100
100
001
001
```

4.2 With recurrent mutation

We can also simulate recurrent mutation with some inherent limitations. The `-Smu` switch is used to specify the forward mutation rate. That is the mutation rate from the wild type to the beneficial allele. We do not consider mutation in the other direction. Mutation rate is again $4N_e\mu$ as per θ but we are only considering a single allele. So the command line for the same example above but including mutation in the selected allele is:

```
>msms -N 100000 -ms 10 1000 -t 5 -SAA 200 -SaA 100 -SF 1e-2 -Smu 1
```

Note with this high mutation rate at the selected loci we will get a high proportion of soft sweeps, in contrast to the case with the previous example that results only in hard sweeps. Also consider that there is a probability of both hard and soft sweeps in cases where the mutation rate for the selected loci is non zero.

We can get the number of origins in the *sample* with the `-oOC` switch. The counts of zero and 1 denote a hard sweep, while a count of more than one denote a soft sweep. It may seem like zero origins does not make sense, but this is the case where the Most Recent Common Ancestor (MRCA) of the sample is within the section of the population that carries the beneficial mutation. Therefore, we do not observe a mutation event on the entire coalescent tree. We must also emphasize that the sample origin count is not the same as the population origin count as the former is a "sampling" of the latter.

4.2.1 Example

We consider the case above with the addition of the `-oOC` switch.

```
>msms -N 100000 -ms 10 1000 -t 5 -SAA 200 -SaA 100 -SF 1e-2 -Smu 1
-oOC
```

With an example output:

```
//
segsites: 7
positions: 0.09309 0.14162 0.14192 0.22869 0.31235 0.78424 0.99885
0100000
0100000
0011100
0011100
0010111
0011100
0010100
1100000
0010111
0010100
OriginCount:4
```

And in this case we have a soft sweep. In the above examples if we wanted to include recombination, we can do so by simply specifying the `-r` switch.

4.2.2 Unix tools example

We can use the piping features of the Unix command line to summarize the results easily, however this will not work on windows. One example is the proportion of soft sample sweeps versus hard sample sweeps for a given set of parameters. Note this should be typed as a single line.

```
>msms -N 100000 -ms 10 1000 -t 5 -SAA 200 -SaA 100 -SF 1e-2 -Smu 1
 -oOC |grep -c "O.*[01]$"
```

This will output the number of hard sweeps out of 1000 (since we did 1000 replicates) which in this case is about 160. For more details use the command `man grep`.

4.3 Without fixation

The other way to include selection is to specify a time when selection starts together with the initial frequencies at that time in the different demes. The usage of the switch `-SI` is as follows:

```
>msms -N 1000 -ms 10 1000 -t 5
-SI time npop freq1 freq2 ... -SAA sAA ...
```

Time is pastward as normal and in units of $4N_e$ generations, then we need the number of subpopulations that exist at that time. This is needed so the next arguments can be read correctly. Then we have a relative frequency as a number between 0 and 1 of the A allele for each subpopulation. Note that the beneficial allele may not go to fixation, and may not even be present at sampling time

depending on the selection strength and population sizes. Options to condition on a proportion of beneficial allele at sampling time are being added and should be available shortly.

4.4 Selection locus position

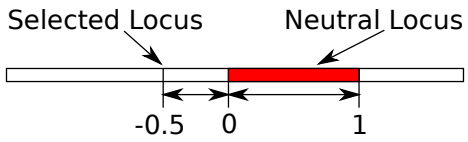

Figure 1: Figure of the locus model. The neutral loci can be anywhere on the line, and there can be more than one. However by default, the neutral locus is 0 to 1 and the selected locus is at 0. The recombination rate is per unit of the "locus line". In this figure, the selected locus is at -0.5. The recombination probability between 0 and 1 is twice the recombination probability between -0.5 and 0.

The position of the selected locus is important relative to the neutral loci, and this is controlled with the `-Sp` switch. The number is the position on the sequence line, while the default neutral locus starts at zero and extends to 1. The default position is zero. Figure 1 shows the relationship between the position and the default neutral loci. We can adjust the recombination between the neutral locus and selected locus by positioning the selected locus further away from the neutral locus. Using this we can position the neutral loci relative to the selected locus to get the desired in locus recombination, and between locus recombination respectively.

5 Structured population models

We now consider the case where we have more than a single deme, that is we have population structure. This is the `-I` switch and its usage is the same as per ms and is as follows:

```
-I npop sample1 sample2 ... sampleN mrate
```

The first argument `npop` is the number of subpopulations. Each subpopulation has population size N_e by default. The next arguments are the number of samples from that deme, there must be the same number of samples as there are subpopulations, and they must add up to the correct total number of samples. Finally, we have the global migration rate `mrate` in units of $4N_em$, where all non diagonal entries in the migration matrix are set to $4N_em/(\texttt{npop}-1)$. The haplotype output is ordered so that the first `sample1` entries are from deme 1, the next `sample2` entries are from the second deme and so forth. Demes are

labeled 1 to `npop` while both θ and the recombination rate are always scaled to the `-N` switch or N_e, not the total population size. In fact in general, all parameters are scaled to N_e as specified by the `-N` switch.

5.1 Subpopulation sizes

We can also specify the population of any subpopulation individually with the `-n` switch. This switch must come after the -I switch. The arguments to the switch are as follows:

```
-n pop scale
```

Where `pop` is the sub population deme label or index and scale is the new size relative to N_e. Any number of `-n` switches can be used, and if the same population label is used in more than one, the last specified value is the one used.

5.2 Migration rates

Finally, one may need more control over the migration matrix. There are two ways to do this. The first method is with the `-m` switch where we can specify a single entry in the migration matrix and has the following syntax:

```
-m i j 4Nm
```

Where `i` and `j` are subpopulation labels and `4Nm` is the new migration rate. We define the migration matrix as follows: $M = m_{ij}\ i \neq j, i, j \in \{1, \ldots, \texttt{npop}\}$ where m_{ij} is the fraction of subpopulation i that is made up of migrants from subpopulation j.

We can specify a complete migration matrix at once with the `-ma` switch. The arguments are:

```
-ma x m12 m13 m21 x m23 m31 m32 x
```

And `m12` is the m_{12} entry of the migration matrix. The diagonal entries are labeled with an `x` but anything that aids readability can be used, and they must be present.

5.2.1 Example

In our first example, we have 2 demes where the second deme is half the size of the first deme. Migration is twice as high in the direction of deme 1 to deme 2. There are 6 samples from the first deme and only 4 from the last deme.

```
>ms -N 10000 -ms 10 100 -t 1 -I 2 6 4 1.0 -n 2 .5 -m 1 2 2.0
```

Here the migration rates are all set to 1.0, and then we set the m_{21} entry to 2.0. Likewise the population size of both demes is set to 10000, and then for deme 2 it is multiplied by .5 with the `-n` switch. We can write the same model using the `-ma` switch as follows:

```
>ms -N 1000 -ms 10 100 -t 1 -I 2 6 4 1.0 -n 2 .5 -ma x 1.0 2.0 x
```

5.3 Including selection

We can include selection globally with the same options that were used in section 4. However, we can control selection strength in each subpopulation separately. This is interesting when considering deme specific selection effects such as habitat variation. The switch is `-Sc` and has the following syntax:

```
-Sc time deme SAA SaA Saa
```

Time is pastward and specifies the time that this switch takes effect. The effect extends pastward indefinitely. If we use a time other than zero (sampling time) the `-SF` option must also have the same time and there must be no changes to any parameters pastward from that point. The deme is the deme label and `SAA SaA Saa` are the selection strengths for the allele in homozygote and hetrozygote configurations. We must still specify the `-SI` or `-SF` switches to turn selection on.

5.3.1 Example

We use the same parameters as the previous example. Only we assume that selection does not affect the hetrozygotes in just one deme[5] and selection acts at a lower level on the homozygotes while the other deme has almost equal selection strength for both the homozygotes and hetrozygotes. The command line is as follows:

```
>msms -N 10000 -ms 10 100 -t 5 -I 2 6 4 1 -n 2 .5 -m 1 2 2
-SAA 1000  -SaA 900 -Sc 0 2 500 0 0 -SF 0
```

We set the selection strength to 1000 and 900 for homozygotes and hetrozygotes respectively globally. We then change the selection parameters for deme 2 with the `-Sc` switch to 500 and 0 respectively and condition on fixation at sampling time.

6 Summary of options

`-ms` nsamples nrep	The total number of samples and number of replicates.
`-N` N_e	Set N_e, note that event times are in discrete generation times in units of $4N_e$, it is safe to set this very high when selection is not considered.
`-t` θ	Set the value of $\theta = 4N_e\mu$
`-T`	Output gene trees
`-r` ρ [nsites]	Set recombination rate $\rho = 4N_e r$ where r is the recombination rate between the ends of a unit length sequence. If nsites is omitted then a infinite sites recombination model is used.
`-G` α	Set growth parameter of all populations to α.

[5] yes this is quite arbitrary

`-I` npop n1 n2 ... [$4N_em$] Set up a structured population model. The sample configuration must add up to the same total number of samples as specified by `-ms`.

`-n` i x Set the size of subpopulation i to xN_e.

`-g` i α_i Set the growth rate of subpopulation i to α_i.

`-m` i j M_{ij} Set the (i, j) element of the migration matrix to M_{ij}.

`-ma` M_{11} ... Set the entire migration matrix.

`-eM` t x Set all elements of the migration matrix at time t to $x/(\text{npop}-1)$

`-es` t i p Split subpopulation i into subpopulation i and npop+1 pastward. Each lineage currently in subpopulation i is retained with probability p, otherwise it is moved to the new population. The migration rates to the new subpopulation is zero and its population size is set to N_e.

`-ej` t i j Join subpopulation i to subpopulation j. All migration matrix entries with subpopulation i are set to zero. The population size of i is also set to zero. With selection this population is ignored pastward from this time.

`-e[X]` t ... Set some parameter pastward from time t. Here `[X]` can be any of `G g n m ma` and the meaning is defined as for the normal command, for example `-en` t i x sets the population size of deme i to xN_e pastward from time t.

-l n a_1 $a_1^{'}$ ... a_n $a_n^{'}$ Set the neutral loci starting and stopping positions for n loci. Note that must be $a_i^{'} < a_{i+1}$ for all i and that there must be $2n$ values. All parameters assume a sequence length of 1. This other parameters need to be scaled accordingly.

`-SAA` $4N_es_{AA}$ Set the selection strength of the homozygote.

`-SAa` $4N_es_{Aa}$ Set the selection strength of the hetrozygote.

`-Smu` $\theta^{'}$ Set the forward mutation rate for the selected allele in units of $4N_e\mu^{'}$. That is the mutation from the wild type a to derived type A.

`-Sp` x Set the position x in the sequence of the selected allele.

`-Sc` t i $4N_es_{AA}$ $4N_es_{Aa}$ $4N_es_{aa}$ Set the selection strength in deme i to the specified values pastward from time t.

`-SF` t Set the selection simulation stopping condition to fixation at time t pastward from sampling time. Note the demographic model must be time invariant for this option not to raise an error. It is up to the user to ensure that the parameters permit the model to always go to fixation, otherwise it will keep simulating till it runs out of memory.

`-SI` t npop x_1 x_2 ... Set the start of selection to time t *forward* in time from this point. The initial frequencies of the beneficial allele are $x_1, x_2, \ldots$. Note that this option is not compatible with `-SF`.

`-oTW` w s `[onlySummary]` Output windowed Watterson's θ estimates with window size w and step size s. If `onlySummary` then only the averages of all replicates is output.

`-oOC` Output the number of Origins of the beneficial allele in the sample. A count of 0 or 1 means a hard sweep if conditioned on fixation.

-tt -oAFS `[jAFS] [onlySummary]` Output allele frequency spectra. If the `jAFS` option is specified, all pairwise deme joint frequency spectra are output.

`-oLD D|D'|r2` Output LD matrix. This is the LD value for each pairwise segregating site as either a measure of D, $D^{'}$ or r^2.

7 Human population example

We now give a example of how to build arbitrary models from the ground up. We first consider the case with no selection, and then add selection as the last part of the exercise.

The model is shown in Figure 2. There are 4 populations with admixture, exponentially growing populations, a bottleneck and migration. We will not concern ourselves too much with specific values for different parameters, but rather keep them simple values to make the example easier to understand.

Events at time Zero

We start with 4 sampled populations with 20 samples from each population and some reasonable initial N_e. In this case, we consider high mutation rates with moderate recombination. We have:

```
>msms -N 10000 -ms 80 -I 4 20 20 20 20 0 -t 100 -r 100 1000
```

But now we must consider admixture. The `CEU` population is mixed with the `MXL` population. If we use the `-es` split switch it creates a new deme 5 rather than join some of the samples from deme 3 (CEU) to 4 (MXL). But we can join deme 4 to the new deme at the same time.

```
-es 0 3 .5 -ej 0 4 5
```

So at time zero, samples from deme 3 stay in deme 3 with probability 0.5. Otherwise the samples or lineages are moved to the newly created deme 5. Since deme 5 is really the MXL that we have sampled, we join deme 4 to deme 5 as well. Note that deme 5 will have no migration parameters and currently nothing has any migration set.

Next we consider growth and population sizes. CHB, CEU and MXL are growing exponentially. We set them to 10, 100 and 200 respectively as follows

```
-g 2 10 -g 3 100 -g 5 200
```

Note that we don't set the 4th deme since we joined it to deme 5. Now we set the initial population sizes relative to N_e. Since YRI is the largest population we assume that's our nominal N_e value. Again we assume the population sizes are .9, 2, and 11. Note that these populations are growing rapidly.

```
-n 2 .0 -n 3 2 -n 5 11
```

Finally we need to set the migration rates mc and mb. We set these to 5 and 2 respectively with the following.

```
-m 1 3 5 -m 3 1 5 -m 1 2 2 -m 2 1 2
```

Note we assume symmetric migration rates so we need to use two `-m` stitches per deme pair.

First event pastward.

The first pastward event is the MXL population joining the CEU population. We assume that this happens 1600 generations into the past. The t_1 time is therefore $1600/(4N_e) = 1600/40000 = 0.04$. This is a deme joining event pastward so we add the following to our command line.

```
-ej 0.04 5 3
```

Nothing else changes so that's all that's required.

Second event pastward

The second event is the joining of the CHB deme with the CEU demes. We set this to be 2000 generations into the past so $t_2 = 0.05$. However this time migration changes as does population size. We also note that there is no longer any exponential growth. We set the B population to be have the size of YRI, and the migration rate ma is 12. Thus we add

```
-ej 0.05 3 2 -en 0.05 2 .5 -em 0.05 1 2 12 -em 0.05 2 1 12
```

The `-en` switch sets the growth rate to zero so we do not need to use any `-eg` switch.

Third event pastward.

We have come to the last population join. This happens 6000 generations into the past. There is nothing else to set in this case so we have.

```
-ej 0.15 2 1
```

Last event

Finally we have a bottleneck 8000 generations ago where the population was reduced to half its nominal value. The last option to add is.

```
-en 0.2 .5
```

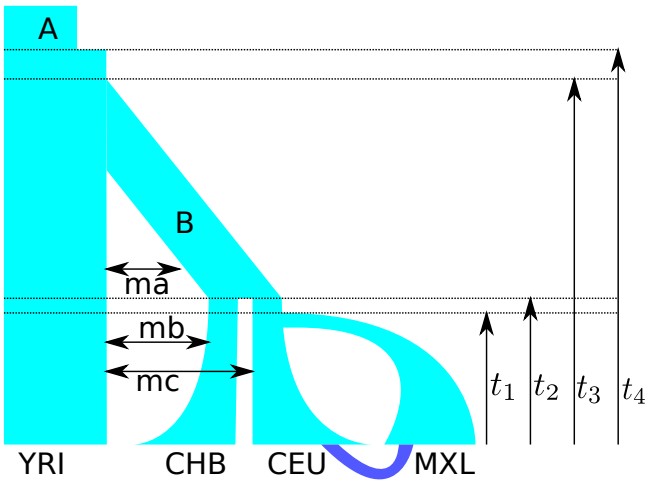

Figure 2: A model of human demographics.

7.1 Complete Command Line & Selection

The complete command line is therefore

```
>msms -N 10000 -ms 80 -I 4 20 20 20 20 0 -t 100 -r 100 1000
-es 0 3 .5 -ej 0 4 5 -g 2 10 -g 3 100 -g 5 200 -n 2 .0 -n 3 2
-n 5 11 -m 1 3 5 -m 3 1 5 -m 1 2 2 -m 2 1 2 -ej 0.04 5 3
-ej 0.05 3 2 -en 0.05 2 .5 -em 0.05 1 2 12 -em 0.05 2 1 12
-ej 0.15 2 1 -en 0.2 .5
```

We claim there is selection in the CEU deme only and that standing variation was initially zero with a medium forward mutation rate at the beneficial locus. We only have to add the following.

```
-SI 0.05 5 0 0 0 0 0 -Sc 0 3 100 50 0 -Smu 0.1
```

First the `-SI` option took the number of demes to be 5 despite the fact that we "joined" one. This is because it still exists and we could set its population size to a non zero value. It is also important to note that the `-SI` option is the only option to work in forward time. That is, selection starts at time 0.05 till the present. While the `-Sc` option works pastward, in this case from sampling time. Finally we set the mutation rate to 0.1.

Thus we can see that to build complicated models and adding selection is straightforward.