msms Cheat Sheet

May 3, 2011

This cheat sheet is for the program outlined in Ewing and Hermisson (2010).

`-help`	Print out options documentation. This is often more up to date than this document. If you are unsure of a option, this provides invaluable "live" documentation.
`-ms` nsamples nrep	The total number of samples and number of replicates. The `-ms` can be omitted if these are the first two arguments.
`-N` N_e	Set N_e, note that event times are in discrete generation times in units of $4N_e$. Not required if there is no selection.
`-t` θ	Set the value of $\theta = 4N_e\mu$
`-s` s	Condition on the number of segregating sites. Just a little slower than using `-t` and uses more memory.
`-T`	Output gene trees.
`-L`	Output tree length statistics.
`-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 are omitted then an infinite sites recombination model is used.
`-G` α	Set growth parameter of all populations to α.
`-I` npop n1 n2 ... [$4N_e m$]	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 are 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. WARNING: This switch behaves differently from *ms* in the strict definition. We consider that most people expect that `-ej` is modeling a split in forward time and hence the deme i is turned off pastward.
`-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 parameter needs to be scaled accordingly.
`-SAA` α_{AA}	Set the selection strength of the homozygote in units of $2N_e s$.
`-SAa` α_{Aa}	Set the selection strength of the heterozygote in units of $2N_e s$.

`-Smu` $4N_e\mu'$ Set the forward mutation rate for the selected allele. That is the mutation from the wild type a to derived type A.

`-Snu` $4N_e\nu'$ Set the backward mutation rate for the selected allele. That is the mutation from the selected type A to the wild type a.

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

`-Sc` t i α_{AA} α_{Aa} α_{aa} Set the selection strength in deme i to the specified values pastward from time t.α is in units of $2N_es$

`-SF` t

`-SF` t f

`-SF` t i f Set the selection simulation stopping condition to fixation at time t pastward from sampling time. t is time into the past, i is the deme and f is the frequency. The first case assumes fixation across all populations, the second case assumes frequency is across all populations. Selection is not used forward in time from this point. 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.
Note the demographic model must be time invariant for this option to work properly.

`-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`.

`-Smark` Include the selected locus in the mutation output.

`-oTPi` w s `[onlySummary]` Output windowed θ estimates (both Wattersons and π based estimators) and Tajima's D with window size w and step size s. If `onlySummary`, then only the averages of all replicates are output. The output format is a table formatted as follows: The first column is the bin position. The second column is the Watterson's θ estimator. The third column is the π[1] estimator and the last column is Tajima's D. The summary also contains the standard deviations for the previous data column. Thus, column 3 is the standard deviation of the Wattersons θ estimators.

`-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.

`-oTrace` Print the frequency trajectory of the forward simulations. The first column is the time in $4N_e$ generations pastward from present. Then each column is the relative frequency of the beneficial allele in each deme. This format is the same as required when specifying a trajectory.

`-Strace` *filename* Rather than simulate the forward trajectory, specify the trajectory in a text file. The format of the `-oTrace` option is valid for input. Note that you must include "unused" demes produced with `-ej` or `-es` options even if the frequency is zero. Also it is not required to specify every generation. Just a time and frequencies in a decreasing order. *msms* will use linear interpolation for generations between specified time points.

`-threads` n Specify the number of threads to use. This permit very easy use of multicore machines. The number of threads should be only as much as you have cores available. This will increase memory by the same factor as threads, so 2 threads will use twice as much memory as one. Also this is not effective if each simulation is very fast, as the cores spend most of their time waiting to output data.

`-seed` v Set the seed. The seed is very different from *ms* since *msms* uses quite a different random number generator. This is a 64 bit number that can be specified either in hex with a `0x` prefix or normal decimal. *msms* goes to some effort to randomize the seed value so you you don't need to set seed values on cluster environment. When using the

[1] average pairwise difference

`-threads` the seed for "iteration n" will be the same and hence give the same result. However the order of reported results will generally be different.

References

Ewing, G. and Hermisson, J. (2010). Msms: a coalescent simulation program including recombination, demographic structure and selection at a single locus. *Bioinformatics*, **26**(16), 2064–2065.