STEM: **S**pecies **T**ree **E**stimation using **M**aximum Likelihood

Version 1.1

Laura Salter Kubatko
Departments of Statistics and
Evolution, Ecology, and Organismal Biology
The Ohio State University
Columbus, OH 43210
lkubatko@stat.ohio-state.edu

If you use this program in a publication, please cite the following reference:

Kubatko, L.S., B. C. Carstens, and L. L. Knowles. 2008. STEM: Species tree estimation using maximum likelihood for gene trees under coalescence. Submitted.

About the Program

STEM is a program for inferring maximum likelihood species trees from a collection of estimated gene trees under the coalescent model. The program will either return the exact ML tree computed using the methodology of Liu and Pearl (2008; see also Roch and Mossel, 2008), compute the likelihood for a user-specified tree, or search for a set of highest likelihood trees. The method for searching the space of trees is a simulated annealing algorithm and is described in the reference above (with more detail given in Salter and Pearl, 2001).

1 Program Availability

STEM is written in ANSI C and should compile on most Unix systems. Source code and executables for STEM are available for free at www.stat.ohio-state.edu/∼lkubatko/software/stem/stem.html. The programs have been successfully compiled using the Gnu C Compiler in Unix (Sun/Sparc), Linux, and Mac OSX.

Downloading the Program

A zip file containing source code, an example, and documentation can be downloaded from www.stat.ohio-state.edu/∼lkubatko/software/stem/stem.html.

Compiling the Program

Place the zip file in the directory you'd like and unzip it. Type

```
> make
```

at the prompt. This creates an executable called STEM, which can be run by typing

```
> STEM
```

at the prompt. You will need to have placed several input files in this directory - see below.

2 Using the Program

Gene Tree File

STEM requires several input files. The first of these is the file which gives the information for the gene trees, which must be called genetrees.tre. This file contains the gene trees with branch lengths in Newick format. **Gene trees must be rooted and must satisfy the molecular clock,** although the program won't check for this (you will likely get an error or it will run indefinitely if this it not the case). Directly in front of each gene tree is a number within square brackets that gives the rate multiplier for each gene (this allows each gene to evolve at its own rate - see the paper above for details). Below is an example gene tree file for 8 taxa and 2 gene trees, evolving at the same rates.

```
[1.0](((Name1:0.00123,Name2:0.00123):0.00123,(Name3:0.00121,Name4:0.00121)
:0.00125):0.0010,((Name5:0.0010,Name6:0.0010):0.0014,(MyName7:0.0012,Name8:0.0012)
:0.0012):0.00106);
[1.0]((((Name1:0.00123,Name2:0.00123):0.00133,(Name3:0.0012,Name4:0.0012)
:0.00134):0.0003,(Name5:0.0010,Name6:0.0010):0.00186):0.00064,(MyName7:0.0011,
Name8:0.0011):0.0024);
```

Please note that there are some limitations on the branch lengths specified in the genetrees.tre file: (1) branch lengths cannot use scientific notation (e.g., 2.56e-03 is not allowed); and (2) branch lengths must be in decimal format (e.g., 0.0000000 instead of 0). It is not recommended that branch lengths that are exactly 0 are used (though this may not result in an error); instead a small number (e.g., 0.0000001) should be used. Finally, branch lengths must be reasonable values (in terms of coalescent units) once scaled by θ.

Settings File

Settings for the program are specified in the “settings” file. This file is divided into three parts. An example is shown below.

```
##Parameters to be modified by the user
search:                 2          ##0=user-tree, 1=MLE, 2=search
theta:                  0.001
num_saved_trees:        15
seed1:                  0
seed2:                  0
verbose:                0

##Simulated annealing parameters (shouldn’t need to be changed)
beta:                   0.0005
burnin:                 100
bound_total_iter:       300000
pbound_un:              0.001
pburnin:                0.05

##Species membership information
nspecies:               4
ntaxa:                  8
ngenetrees:             8
Species1 Name1 Name2 Name3 ;  ##Do not remove space between name and semi-colon
Species2 Name4 Name5 ;
Species3 Name6 MyName7 ;
Species4 Name8 ;
```

In the first section, a series of parameters that the user should consider modifying for his particular run are given. The first option (`search`) allows the user to select one of the three analyses that STEM can carry out. If `search` is set to 0, then a species tree will be read from the file “speciestree.tre”, and the likelihood will be computed for that tree. If `search` is set to 1, then the maximum likelihood estimate of the species tree will be computed using the method of Liu and Pearl (2008; see also Roch and Mossel, 2008). If `search` is set to 2, then the simulated annealing algorithm will be used to search species tree space for the set of `num_saved_tree` trees that have the highest likelihood.

The parameter `theta` is the value of $\theta = 4N_e\mu$ to be used for make the correspondence between gene trees branch lengths and species tree branch lengths. All gene tree branch lengths are scaled by dividing by `theta` prior to the analysis. The parameters `seed1` and `seed2` allow the user to set random number seeds. If these are set to 0, seeds are obtained from the system clock. Random number seeds are only needed when `search` is set to 2.

In the second section, parameters to control the simulated annealing algorithm can be set. These will only be used when `search` is set to 2. In most cases, the user will not need to modify these. These parameters are briefly defined in the Appendix. For more detail, see Salter and Pearl (2001).

In the third section, information about the relationships among sampled lineages are given. STEM requires that each sampled lineage be assigned to one species. However, the number of lineages sampled per species can vary both between species and across genes. In addition, STEM allows missing data. Gene trees can contain different taxon samples, and it is allowable to have incomplete samples for some genes for both lineages within a species and for species. Please use caution and common-sense here, however. The performance of STEM has not been thoroughly investigated when there is a large percentage of missing data.

The `nspecies` option in the beginning of the third section refers to the number of species being considered. The `ntaxa` option specified the total number of lineages sampled (even if no gene tree ever contains this many taxa because of incomplete sampling). This name should be equal to the number of lineage names entered at the bottom of this section (described in the next paragraph). The `ngenetrees` option refers to the number of gene trees contained in the file “genetrees.tre”.

When entering information about species memberships, it is important to list the each lineage that is sampled for each species, even if it doesn’t appear in every gene tree. Also, taxon names must be IDENTICAL in all gene trees. Species tree names can be arbitrarily chosen – these are what will be printed in the species tree estimates reported by STEM.

Species Tree File

If `search` is set to 0, the program will read a user-specified species tree from the file speciestree.tre and return the maximum likelihood for this tree (optimal branch lengths are computed). The file must contain a single tree with branch lengths (although these are not used) in Newick format. Below is an example corresponding to the example files above:

```
(1:1.0,(4:1.0,(2:1.0,3:1.0):1.0):1.0);
```

Output Files

if `search` = 0 or 1, the optimized tree (with branch lengths) and its likelihood is written to standard output (e.g., the screen). If `search` = 2, the program writes all output to a file called `results`. This file will list each of the `num_saved_trees` trees found during the search, along with information concerning their maximized likelihoods and when they were encountered in the search. In addition, the program will report the highest likelihood tree found in the search. When `search` is 1 or 2, information about trees is written to the file "treefile.phy" (this is useful for conducting simulation studies).

3 Running the Program

There are several steps involved in running the program, which are outlined below.

1. Prepare the required input files. At least two files are needed: "genetrees.tre" and "settings". If you wish to evaluate a user-specified tree, then the file "speciestree.tre" needs to be prepared as well. All of these files must be placed in the directory from which the program will be run.
2. Modify the settings file as needed. In particular, set the value of `search` to the desired level.
3. Run the program.
4. Upon termination of the program, examine the contents of the screen output and the "results" and "treefile.phy" files to check that the program has completed successfully.

4 Details of the Implementation

The details of the STEM algorithm are fully described in Salter and Pearl (2001) and the reference above and will only be briefly reviewed here so that implementation issues may be discussed. The algorithm is based on a simulated annealing algorithm that has been modified for use with phylogenies. The algorithm works by considering at each iteration a current phylogeny from the set of all possible phylogenies for n species. From each current phylogeny, a new phylogeny is proposed, by modifying the topology (branching pattern) of the tree and finding optimal branch lengths within that tree. Following generation of the proposal tree, the log likelihoods of the two trees (the current tree and the proposal tree) are compared. If the proposal tree has a higher log likelihood, then it will become the current tree for the next iteration. If the proposal tree has a lower log likelihood, then it becomes the current tree with probability proportional to the difference in log likelihood between the two trees. The idea behind the algorithm is that since a tree with a lower log likelihood always has some probability of being accepted by the algorithm, the search should be less likely to become trapped in locally optimal portions of the tree space than an uphill search strategy.

The probability of accepting a tree of lower log likelihood than the current tree is decreased as the algorithm proceeds, with the idea that eventually the search should settle on the optimal tree as moves to trees of lower log likelihood become less likely to be accepted. The manner in which this probability is lower is called the *cooling schedule* in the simulated annealing literature. The parameter controlling the rate of cooling is `beta`, which the user may specify in the settings file. `beta` must be a number between 0 and 1, where values closer to 0 represent slower cooling (increased search time but less chance of returning a locally optimal solution) and values closer to 1 result in more rapid cooling (shorter search time but trees found are more likely to be only locally optimal). The default value given in the settings file should be adequate for most problems, and thus the user will not generally need to change this setting.

The algorithm terminates when one of two conditions are satisfied: (1) a sufficient number of trees have been proposed from the current tree without any of them resulting in acceptance, or (2) the search is alternating between a collection of high-likelihood trees that are separated from one another by a single rearrangement, and a sufficient number of iterations have passed since any alternative trees have been accepted. The parameter which specifies the exact conditions for termination of the algorithm is `pbound_un`. See Salter and Pearl (2001) for details. These parameters should not need to be modified by most users.

The code used to implement the algorithm is elementary, and is not optimized for either speed or memory efficiency. However, performance appears to be quite good in problems of moderate size. Random number generation is accomplished by inclusion of the ”randlib” package.

It is the author's philosophy that it is important to gain information not only about the maximum likelihood tree, but about other trees of high likelihood, since it is often the case that there will be many trees with likelihoods nearly as high as the ML tree. For this reason, it is not recommended to use `num_saved_trees` less than 5. Because any particular run of the simulated annealing portion of the algorithm is not guaranteed to find the ML tree, it recommended that the program be run at least twice, once with `search`=0 and once with `search=2`. Ideally, the program would be run several additional times with `search` = 2 to ensure that the `num_saved_trees` trees of highest likelihood are those that are reported.

I would greatly appreciate hearing about any successes and/or bugs associated with use of the program. I can't promise that I will be able to respond quickly to reported bugs. However, please e-mail me the input and output files, as well as any error messages you get from the program, for the run in which the problem occurred, and I will try to respond quickly to your query. Please e-mail all comments to lkubatko@stat.ohio-state.edu.

5 Acknowledgments/References

Development of this program was funded in part by the National Science Foundation through grants DMS 0104290 and DMS 0505265/0702277. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Salter, L. A. and D. K. Pearl. 2001. Stochastic search strategy for estimation of maximum likelihood phylogenetic trees. *Systematic Biology* 50(1): 7-17.

Liu, L., L. Yu, and D. K. Pearl. 2008. Maximum tree – A consistent estimator of the species tree. Under review.

Mossel, E. and S. Roch. 2008. Incomplete lineage sorting: Consistent phylogeny estimation from multiple loci. Available at http://arxiv.org/abs/0710.0262.

6 Frequently Asked Questions

I get a strange number for the log likelihood. / I get an error when I run my data, even though the example data worked. / The species tree STEM reports has branch lengths that are all zero.

These issues commonly occur when there are branches of length 0 in the gene trees. This will not be a big problem when the zero branch lengths occur for lineages sampled from within the same species, but will be a problem when lineages sampled from different species are connected by a branch of length 0. The reason is that the maximum likelihood estimate of the speciation time must pre-date ALL gene divergence times of the affected lineages. When zero branch lengths are observed between lineages from different species for one or more genes, this forces the MLE of the speciation time to be zero. When this occurs for many pairs of species, a star tree is often the ML species tree. Sometimes STEM may report a strange likelihood in this case, and occasionally it may crash (but hopefully most of the time it will correctly report the likelihood of the star tree).

This occurs fairly commonly within the particular groups for which multilocus species tree inference is most desirable, as such groups are often characterized by recent, rapid radiations. In these cases, investigators must decide whether it is reasonable to exclude taxa/genes from the analysis. Note also that setting all zero branch lengths to something small (e.g., 0.000001) will allow STEM to run, but you will essentially still obtain a star phylogeny for the ML tree.

STEM starts running, but I get an error (or it hangs).

STEM is very picky about the format of the settings file. In particular, there are many places where the spacing matters. A good step to diagnose this problem is to check that the settings file you are using looks as similar as possible to the version of this file distributed with the program (called “settings_template”).

Another issue could be the gene trees in the genetrees.tre file. These must all be rooted (e.g., have a bifurcation and not a trifurcation at the root) and satisfy the molecular clock. The program does not check this carefully, and may react strangely if this is not true.

STEM runs, but the likelihood reported is "nan".

This most commonly indicates a numerical issue with computing the likelihood. The first step is to check the genetrees.tre files to make sure it satisfies the following: (1) branch lengths do not use scientific notation (e.g., $2.56e - 03$ is not allowed); (2) branch lengths are in decimal format (e.g., use 0.00000001 instead of 0); and (3) the scaling you introduce with the setting of θ in the settings file doesn't result in enormous branch lengths (e.g., if a branch length is 0.5 and θ is set to 0.001, then the branch length specified will be $\frac{0.5}{0.001} = 500$ coalescent units).

7 Appendix

Short description of selected parameters in settings file:

`ntaxa`: Gives the total number of lineages included in the gene trees.

`nspecies`: The number of species for which a species tree is to be estimated/evaluated.

`search`: This option specifies what function the program will perform. If search = 1, the program will return the maximum likelihood species tree and branch lengths, without implementing the search. If search = 0, the program will find maximum likelihood branch lengths and return the likelihood for a user-specified tree, placed in the file speciestree.tre (see below). If search = 2, the program will use the simulated annealing method to search for the set of the `num_saved_trees` trees with the highest likelihoods.

`verbose`: This determines the amount of information that is printed to the screen. Virtually all users will leave this setting at 0 (to see more output, use verbose = 1).

`beta`: This parameter determines the rate of cooling in the simulated annealing method. The default value will not need to be modified by most users. See the reference above for details.

`burnin`: Number of iterations in the burn-in period, which is used to estimate some parameters used in the search procedure. For most problems, the default value of 100 iterations should be adequate.

`bound_total_iter`: Bound on the total number of iterations that the algorithm will perform. This option simply prevents the algorithm from running for an indefinite period of time, or may be used to terminate a search after a specified period of time. This bound should be set to some large number, generally several hundred thousand. In practice, the algorithm will generally terminate well before this bound is reached.

`pbound_un`: This probability is used in setting the stopping rule for termination of the algorithm. It is used to determine the number of iterations needed to guarantee that the probability of not selecting a node for rearrangement is at most `pbound_un`.

`pburnin`: To ensure adequate time during the burn-in period, this parameter is used to determine the number of iterations needed to guarantee that the probability of not selecting a node for rearrangement is at most `pburnin`. The total number of iterations used in the burn-in period is this number plus `burnin`.

`num_saved_trees`: Number of trees retained by the algorithm. For example, if `num_saved_trees` is set to 10 (the default) then the 10 trees of highest likelihood encountered during the search procedure will be written to the output files.

`seedj`: One of two random number seeds that can be set by the user. If set to 0, then a random number seed will be set automatically.

`seedk`: One of two random number seeds that can be set by the user. If set to 0, then a random number seed will be set automatically.

`theta`: The value of $\theta = 4N_e\mu$ to be used for make the correspondence between gene trees branch lengths and species tree branch lengths.