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

Version 2.0

Laura Salter Kubatko[1,2] and Travis Treseder[1]

[1]Departments of Statistics and
[2]Evolution, Ecology, and Organismal Biology
The Ohio State University
Columbus, OH 43210
lkubatko@stat.osu.edu, treseder@stat.osu.edu

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

Kubatko, L.S., B. C. Carstens, and L. L. Knowles. 2009. STEM: Species tree estimation using maximum likelihood for gene trees under coalescence. *Bioinformatics* 25(7): 971-973.

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 (2009; see also Roch and Mossel, 2010), 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 2.0 is written in Clojure and is distributed as a JAR file. It is run as a JAVA application.

STEM 1.1a and lower versions were written in ANSI C. These version have been completely superseded by version 2.0 and their use is not recommended.

New Features in Version 2.0

STEM has been completely re-written in Clojure. This was done with the goal of improving the user interface (input and output files) and the basic program functionality, as well as to allow extensions to the methodology to incorporate horizontal events (this functionality will be added in the next release). The primary differences from Version 1.1 are the following:

- A new option for supplying the set of input gene trees has been included. It is now possible to supply a list of file names that contain the gene trees.
- The format of the settings file has been standardized to YAML format.
- STEM 1.1 could not handle some patterns of missing data. STEM 2.0 has been more extensively tested with a variety of missing data patterns and we believe it now always handles missing data properly.
- STEM 2.0 will compute the likelihood for a user-specified tree with user-specified branch lengths, as well as providing maximum likelihood branch lengths for a user-specified tree.

Downloading the Program

A zip file containing the JAR file, example input files, and documentation can be downloaded from www.stat.ohio-state.edu/∼lkubatko/software/stem/stem.html.

Running the Program

Place the zip file in the directory you'd like and unzip it. STEM 2.0 can be run by issuing the following command

```
> java -jar stem.jar
```

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

In addition, it is possible to give STEM 2.0 a command line argument to indicate the location of the "settings" file, e.g.,

```
>java -jar stem.jar mysettings.yaml
```

2 Using the Program

Input Files

1. File(s) containing the gene trees must be supplied. There are two options for how to give the gene tree information to STEM:

   (a) A file that contains all of the gene trees, where each tree conforms to the Newick format, each tree is separated by a newline character, and each tree is preceded by its rate multiplier (in brackets; this allows each gene to evolve at its own rate - see the paper above for details) may be supplied. This file should be called genetrees.tre. This is unchanged from STEM 1.1., and is probably how you're doing it now. One important thing to note is that **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).

   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);
   ```

   (b) A new option is to declare in the settings file (outlined below) the filename(s) of the tree file(s), each containing any number of Newick-formatted trees. No tree rate multiplier is needed to precede each Newick string. Instead it is set in the settings file. This means that STEM 2.0 assumes that all trees in a particular file have the same rate multiplier. There is no limit to the number of input files.

2. Settings for the program are specified in the "settings" file. The settings file is based on the YAML markup language. The spec can be found here: http://www.yaml.org/

   It is a small, simple, readable format, and for our purposes STEM 2.0 only uses a small subset of its features. The settings file must be named 'settings', 'settings.txt', or 'settings.yaml'. STEM 2.0 looks for the file in that order, in the directory where the STEM jar is executed.

   The settings file itself is broken up into three small sections. Here is an example file:

```
properties:
  run: 2           #0=user-tree, 1=MLE, 2=search
  theta: 0.001
  num_saved_trees: 15
  beta: 0.0005
species:
  Species1: Name1, Name2, Name3
  Species2: Name4, Name5
  Species3: Name6, MyName7
  Species4: Name8
files:
  trees1.tre: 1.0       # notice the space after each ':'
  trees2.txt: 1.23
```

The properties section is where various STEM parameters are set. The species section is similar to STEM 1.1: each species identifier is followed by a comma-separated list of its associated lineages. The files section is last (although, these sections can be in any order in the file). If you are using parsing method (a) above, then there will be no files section, and STEM 2.0 will look for a file named 'genetrees.tre' in the current working directory. If you are going to use method (b), then this is the section where each file is declared, followed by its tree multiplier.

A couple of notes: indentation matters, i.e. it's how YAML delineates sections. Each child of one of the sections is indented (uniformly) more than its parent. And lastly, for now, there must be a space after each ':'.

Most of the properties have reasonable defaults:

`num-saved-trees`: 10 # how many trees to save during simulated annealing search
`burnin-default`: 100
`bound-total-iter`: 200000 # how many iterations for search
`beta`: 0.0005
`mle-filename`: mle.tre # name of file to save likelihood results
`search-filename`: search.tre # name of file to save search results

but any of these can be added to the properties in the settings file if the user wishes to modify them. Descriptions of all properties are given in the Appendix. For more detail, see Salter and Pearl (2001). We make just a few comments about the settings users are most likely to modify here.

The parameter `theta` is the value of $\theta = 4N_e\mu$ to be used for making 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 parameter `seed` can be set for use in the simulated annealing search. Using a user-specified seed allows one to replicate an analysis later (e.g., using the same seed will always produce the same annealing search).

In the "species" 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.

When entering information about species memberships, it is important to list 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. If not, an error message will be printed. 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 `run` is set to 0, the program will read user-specified species trees from the file user.tre and return the likelihood for these tree (branch lengths in coalescent units should be included in the trees and tree should be given 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 `run` = 1, the optimized tree (with branch lengths) and its likelihood is written to standard output (e.g., the screen) as well as to the file "mle.tre". If `run` = 2, the program writes all output to a file called `search.tre`. This file will list each of the `num_saved_trees` trees found during the search, along with information concerning their maximized likelihoods.

3 Running the Program

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

1. Prepare the required input files. You must at least prepare one or more files containing gene trees and the "settings" file. If you wish to evaluate a user-specified tree, then

the file "user.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 `run` to the desired level, and modify the value of `theta`.

3. Run the program by typing `java -jar stem.jar` at the prompt.

4. Upon termination of the program, examine the contents of the screen output and the "mle.tre" or "search.tre" files to check that the program has completed successfully.

5. You should always run the program at least twice: once with `run`=1 and once with `run`=2. For many datasets, there will be several trees that are tied for the maximum likelihood score. Running the simulated annealing search and examining the set of trees and corresponding likelihoods found in the search will allow you to examine these trees.

4 Details of the Implementation

The details of the search algorithm implemented in STEM 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. See Salter and Pearl (2001) for details. These parameters should not need to be modified by most users.

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 `run`=0 and once with `run=2`. Ideally, the program would be run several additional times with `run` = 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, DMS 0505265/0702277 and DEB 0842219. 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. 2009. Maximum tree: a consistent estimator of the species tree. J Math Biol 60:95-106.

Mossel, E. and S. Roch. 2010. Incomplete lineage sorting: Consistent phylogeny estimation from multiple loci. To appear in IEEE/ACM Transactions on Computational Biology and Bioinformatics, available at http://arxiv.org/abs/0710.0262.

6 Frequently Asked Questions – STEM 1.1

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.

Note on version 2.0: The input format for gene trees is now more flexible in terms of branch lengths, so this should no longer be an issue. Please send me an e-mail (lkubatko@stat.osu.edu) if you encounter problems.

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.

Note on version 2.0: This, too, should be improved in the new version, although it still assumes that gene trees are rooted and satisfy the molecular clock. It may behave unpredictably if this is not the case.

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 that 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).

In addition, we have found that occasionally the probability of a coalescent event occurring in a branch is zero, due to overflow. If that occurs, STEM replaces zero with the minimum constant holding the smallest positive nonzero value of 2^{-1074}.

7 Appendix

Short description of selected parameters in properties section of the settings file:

`run`: This option specifies what function the program will perform. If run = 1, the program will return the maximum likelihood species tree and branch lengths, without implementing the search. If run = 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 run = 2, the program will use the simulated annealing method to search for the set of the `num_saved_trees` trees with the highest likelihoods.

`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.

`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.

`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.