STELLS: A Program to Infer Species Tree from Gene Tree Topologies under Incomplete Lineage Sorting by Maximum Likelihood

User Manual

Version 2.0.0

October 27, 2015

Yufeng Wu
CSE Department, University of Connecticut Storrs, CT 06269, U.S.A.
Email: ywu@engr.uconn.edu

This software is provided “as is without warranty of any kind. In no event shall the author be held responsible for any damage resulting from the use of this software. The program package, including source codes, executables, and this documentation, is distributed free of charge. If you use this program in a publication, please cite the following reference:

Yufeng Wu, Coalescent-based Species Tree Inference from Gene Tree Topologies Under Incomplete Lineage Sorting by Maximum Likelihood, Evolution, vol. 66, pages 763-775, 2012.

If you use STELLS v2.0.0 to infer the population tree from haplotypes, please cite the following references:

Yufeng Wu, An Algorithm for Computing the Gene Tree Probability under the Multispecies Coalescent and its Application in the Inference of Population Tree, submitted for publication, 2015.
Yufeng Wu, A Coalescent-based Method for Population Tree Inference with Haplotypes, Bioinformatics, 31:691-698, 2015.

1 Getting Started with STELLS

1.1 Program availability

STELLS is written in C++. Executables for popular platforms such as Linux 32 bits or 64 bits and MacOS are downloadable from the author’s personal webpage:

`http://www.engr.uconn.edu/~ywu/STELLS.html`. Files can be downloaded using "Save Link/Target As..." After downloading the softwares, you may need to change file access permissions (e.g. chmod u+x stells-linux). In case that you want to compile the code yourself, source code is also available for download at the above URL. To compile the code, first put the gzip file in the directory youd like and unzip it: use gunzip and tar commands such as:

▷ gunzip ⟨stells-src.tar.gz⟩
▷ tar -xvf ⟨stells-src.tar⟩

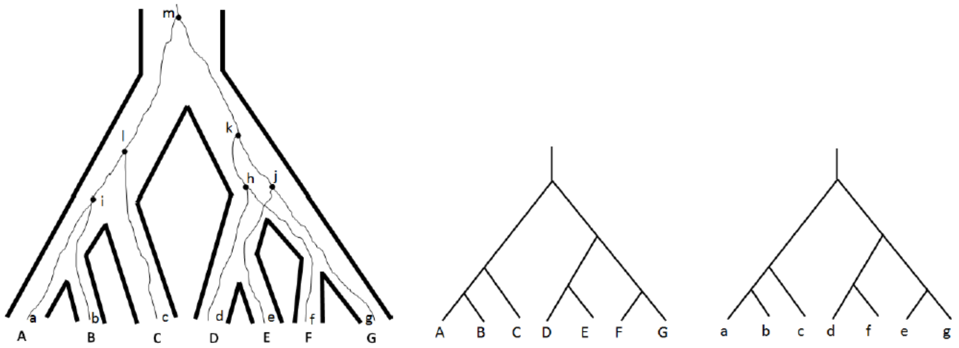
Then type:

▷ make at the prompt. This creates an executable called stells, which can be run by typing
▷ stells at the prompt. You will need to specify some input options - see below.

1.2 What is STELLS?

First, where does the name STELLS come from? It stands for Species Tree infErence by maximum Likelihood under Lineage Sorting.

The objective of STELLS is, given a set of gene tree topologies (i.e. branch lengths are not needed), infer the species tree that best fits the given gene tree topologies under the coalescent theory. The main biological question addressed by STELLS is incomplete lineage sorting (or simply lineage sorting), which is widely known to potentially cause the gene trees to be different from the species tree. The underlying genealogical process is multispecies coalescent. See Figure 2 for an illustration. As shown in Figure 2, multispecies coalescent determines gene genealogies, which may be different from the underlying species (or population) tree.



(a) Gene tree in the species tree (b) The species tree (c) The gene tree

Figure 1:

Figure 2: *A gene tree (thin lines) in the species tree (thick lines). Interior nodes of the gene tree correspond to coalescent events of gene lineages. The species tree is shown separately in part (b), so is the gene tree in part (c).*

The STELLS program was originally developed for inferring species tree from gene tree topologies (see Wu, Evolution 2012). That is, STELLS was targeted for the well-known gene tree and species tree problem. Since the publication of Wu (Evolution, 2012), I have extended the applications of STELLS to the inference of population divergence history (called population tree). In Wu (Bioinformatics, 2015) and Wu (manuscript, 2015), I showed that STELLS can be used to infer the population tree with haplotypes from multiple closely related populations (or species). The current release of STELLS (v2.0.0) is mainly for the population tree inference (although it may also be used for the original species tree inference). Namely, the latest version of STELLS implements new algorithms that run much faster than the original STELLS when there are multiple gene lineages (alleles) from each population (species). See my recent papers (e.g. Wu, Bioinformatics, 2015) for more details.

1.3 How does STELLS work?

STELLS is based on the coalescent theory. STELLS can search for the species tree that gives the maximum likelihood of the given input gene trees. The technical details of the method behind STELLS can be found from my Evolution 2012 paper as listed above.

Briefly, STELLS is based on the computation of the so-called gene tree probability for a given species tree. Gene tree topology is the probability of observing a gene tree topology (yes, topology only; branch length of gene tree is not used) in a given species tree (with branch length) based on incomplete lineage sorting and coalescent theory. STELLS performs tree space search for the best species tree that maximizes the product of gene tree probability of each of the given gene tree topology.

2 Functionalities and Usage of STELLS

STELLS supports two main functionalities. In either case, the user should provide one or more gene tree topologies.

1. A species tree is given. In this case, STELLS first computes the gene tree probability for this given species tree. Then, STELLS finds the optimal branch lengths of the species tree that maximizes the probability of these gene trees.

2. No species tree is given. In this case, STELLS uses maximum likelihood scheme to search for the best species tree (with topology and branch lengths) that best explains the given gene tree topologies.

Something you need to know before using the program: the gene trees are *rooted*. STELLS does not need branch lengths of gene trees. If branch lengths are given, they are ignored. The running time varies. STELLS can finish relatively quickly when the number of taxa is no more than 20 (each with one or a few gene lineages) and the number of gene trees is in the hundreds. If you want to run with larger data, you may need to turn on the coarse mode flags (see below). In addition, STELLS allows non-binary trees, although it is usually faster when binary trees are used.

When you have multiple gene lineages from single populations, there is another option to speedup the species (population) tree inference by using the approach based on neighbor joining from pairwise population distances. See the following for more details.

2.1 Preparing inputs

To run STELLS, you need to prepare a gene tree file. The gene trees file contains multiple lines, one for each tree. The trees should be in the popular Newick format. Again, you do not need to assign branch length for the gene trees. If you do, these branch lengths will be ignored. You can use any string (including integers) to denote the species. You should not include lines other than the tree Newick strings in the gene tree file. A simple example is as follows:

```
(A,(D,(B,C)))
(C,(D,(B,A)))
(D,(C,(A,B)))
```

You may have multiple alleles for some species. STELLS requires these alleles all labeled by the species (i.e. with the same label). Each gene tree can have arbitrary number of alleles for a species. However, it is required the set of species in any two gene trees are the same. That is, there exists no species that appears in one gene tree but does not appear in the other gene tree. For example:

```
(A,(D,(A,(B,C))))
(C,(C,(D,(B,A))))
(D,(C,(A,B)))
```

Note: STELLS treats two alleles of one species as inter-changeable. In principle, one can assign distinct labels to alleles from the same species. The probability computed by STELLS is the probability of an arbitrary way of distinct label assignment (not the sum of all ways of distinct label assignment).

STELLS also allows you to specify a particular species tree. The species tree should be in a separate file and contains a single line: the Newick format of the species tree. A species tree should have exactly the same set of species as the gene trees. A species appears exactly once in the species tree. Moreover, you need to assign branch length to each edge of the species tree. You may assign some initial branch length (e.g. 1.0) in case you do not know it. STELLS may search for optimal branch length for you (see later). For example, here is one example of the species tree file.

```
(A:1.0,(B:0.5,(C:1.0,D:1.0):1.5):0.5)
```

Non-binary trees. In some cases, one may want to use non-binary (i.e. multifurcating) gene trees. This may arise, for example, some branching pattern in a gene tree cannot be fully resolved. STELLS allows non-binary trees as input. You will still use the Newick format for a non-binary tree. For example,

```
(D,(A,B,C))
```

In the case of non-binary tree, the gene tree probability is defined to be the sum of probability of all compatible binary gene tree topologies. We say a binary gene tree topology is compatible with a non-binary tree if we can obtain the binary tree topology by resolving the non-binary nodes of the non-binary tree. In the above example, there are three compatible binary topologies:

```
(D,(A,(B,C)))
```

(D,((A,B),C))
(D,((A,C),B))

The gene tree of the non-binary tree is then the sum of the probabilities of these three binary topologies. One should note that probability of non-binary topologies is usually slower to compute than binary topologies. The more unrefined a gene tree is, the slower STELLS becomes. Note that very unrefined gene trees don't add much more information for ancestral inference. Therefore, it may be a good idea to apply preprocessing to remove some of those very unrefined trees.

2.2 Basic usage

There are two basic operation modes in STELLS. In either case, you must provide a gene tree file using the -g option (with the gene tree name following the -g). First, you can specify -s option (followed by the species tree name) to let STELLS focus on this particular species tree stored in the species tree file. That is,

```
▷ ./stells-linux -g <gene-tree-file-name> -s <species-tree-file-name>
```

The format of these two files are described above. In this mode, STELLS calculates the coalescent likelihood of the gene trees on the given species tree (i.e. the sum of log-likelihood of each gene tree on the species tree). Then, STELLS will try to optimize the branch length of the species tree in order to find higher likelihood for the given species tree, and the species tree with optimal branch length will be output.. Note: optimizing branch length will be slower. If you do not want to optimize branch length, you can use the -B option as follows to omit the branch length optimization:

```
▷ ./stells-linux -B -g <gene-tree-file-name> -s <species-tree-file-name>
```

In the second mode, no species tree is given. That is,

```
▷ ./stells-linux -g <gene-tree-file-name>
```

In this mode, STELLS will search the space of species tree to find the one that leads to the highest coalescent likelihood of the gene trees. By default, STELLS uses the MDC (a parsimony approach) to find the initial species tree topologies. The user may choose to search for species tree by providing a list of initial trees using the -T option:

```
▷ ./stells-linux -T <init-species-tree-file> -g <gene-tree-file-name>
```

STELLS outputs a single species tree with the highest likelihood. The species tree has branch lengths, which are in the standard coalescent units. That is, suppose there are g generations for a branch, then the branch length is g/2Ne, where Ne is the effective population size. *Warning*: search for species trees is usually a computational intensive task. Also, it helps to provide more gene tree topologies for more accurate inference results.

Sometimes, it may be desirable to get a sense of species tree distribution (i.e. likelihood of a number of good candidate species trees). For this purpose, STELLS also supports evaluation of near-parsimony species trees as follows.

```
▷ ./stells-linux -S -d <k> -g <gene-tree-file-name>
```

The -d flag is optional, which controls how many trees to evaluate. Here, k roughly means how far apart STELLS is to search from the most parsimonious trees. The larger k is, the more trees are to be searched (but the slower STELLS will be). By default, level is set to five. For the sake of efficiency, I suggest to begin with level=1. In this mode, STELLS would use the initial trees whose number is determined by the -d option, and then explore neighbor trees STELLS stores a number of trees in order to give a sense on how the likelihood distribution looks like. The log-likelihood along with the alternative species trees are stored in a file called treeprobs.out. Note if -S option is not invoked, treeprobs.out is not updated and only a single species tree is output.

2.3 Handling larger input using coarse mode and multithreading

STELLS becomes slow when the data size grows. My experience shows that on a relatively good computer (3 GHz Linux machine with 3 GB memory), STELLS can find the maximum likelihood estimate of the species tree for around 100-500 gene trees, with the number of species no more than 20. However, when the data size is large (say 500 gene trees, each with 30 taxa), STELLS becomes slower.

In order to allow inference on larger data, the user can run STELLS in the coarse mode. This mode is less accurate but is often faster. Another option is using multi-threading option (see below). To turn on the coarse mode, use the following options:
-x Kx: set Kx to be a small number (say 500) to speedup. By default, Kx=5000. This option limits the search for the near-MDC optimal species trees (i.e. more heuristic search of MDC trees to make it faster). This can be useful when data is very large and even the MDC step becomes slow.
-S : restricted search of tree space. This can significantly reduce the running time, but of course can lead to less accurate inference. When -S is specified, STELLS will only evaluate near-MDC optimal trees (i.e. does not perform NNI local search). How many near-MDC optimal trees depends on the MDC level (set by the -d option).
-d k: set k to a small value (say 1) can make STELLS run faster. Here, k is the near-MDC level. By default, the near-MDC level is 5.
-c Kc: set Kc to a small number (say 10) to speedup. By default, Kc=infinity. This option limits the number of configurations (i.e. data structure needed for probability computation).
-a Ka: set Ka to a small number (say 500) to speedup. By default, Ka=infinity. This option also limits the number of configurations (i.e. data structure needed for probability computation).

A little more on the -S option. We can classify all species tree topologies by their optimal level on the MDC criteria (minimizing deep coalescent, a parsimony score for species tree). Level 1 is the most optimal MDC trees, level 2 is for trees with the next best MDC scores, and so on. If -S is used, then STELLS will evaluate all species tree topologies whose MDC level is within some MDC level (by default 5, but can be set by the -d option). Note that -S 1 is not exactly the same as MDC since there may be multiple MDC trees and STELLS would pick the one with higher likelihood. Unless data size is very large, the -x option is usually not needed since the MDC step is often much faster than the probability computation. Also note: the -d, -x and -S are mainly for exploring species tree space (i.e. cutting corners in species tree search). Sometimes even computing gene tree probability for a given species tree can be slow. In this case, you can use -c and/or -a option. These two options speed up the probability computation and thus apply for the cases with or without given species tree.

For example, if STELLS is too slow for a data (when searching for species tree), use:

```
▷ ./stells-linux -S -d 1 -g <gene-tree-file-name> -c 10 -x 500
```

If you notice that STELLS is too slow even for a given species tree, try this:

```
▷ ./stells-linux -g <gene.trees> -s <species.tree> -c 50 -a 50
```

Note, however there may be trade off between accuracy and efficiency: less accurate results may be found in coarse mode. In general, I would recommend not using these "corner-cutting" options as long as STELLS can finish the computation within some reasonable amount of time. In case that speedup is really needed, I suggest you to try different value combinations to see how to speedup. To obtain more accurate results, you should try to use less restricted values (usually meaning larger values in the options).

Multithreading The version 2.0.0 adds multithreading option. To turn on multithreading, use -t <number of threads>. For example, the following will make STELLS run with five threads. Note that multithreading is only used for the species tree inference (that is, it does not help if you only want to compute the gene tree probability and do not infer species trees).

```
▷ ./stells-linux -t 5 -g <gene.trees>
```

Note that computing the gene tree probability and inferring species tree are usually memory intensive. When you use multiple threads, you may reduce the running time but at the same time you may use more memory. Therefore, I would suggest to monitor the memory usage while using multiple threads.

2.4 Computing the gene tree probability with multiple gene lineages per species and the inference of population trees

In version 2.0,0, a new algorithm for computing the gene tree probability is implemented. The main advantage of this algorithm is that it is faster than the original algorithm in Wu (Evolution 2012) when the number of gene lineages per species is large. For example, suppose you have two species and for each species you have 20 gene lineages. The new algorithm is much faster than the old algorithm in this case. When the number of population is larger (say four or more) and/or the number of gene lineages per population is small (say 1 or 2), the new algorithm appears to be slower than the original algorithm. To use the new algorithm, use the following command:

```
▷ ./stells-linux -f -g <gene.trees>
```

Now what if you have multiple gene lineages per population but there are multiple (say 10) populations? In this case, STELLS has an approach of inferring population tree using neighbor joining based on pairwise population distances. In order to obtain accurate population distances, it is desirable to have multiple gene lineages per species. Also note that the inferred neighbor joining population tree should be viewed as **unrooted**. To use this inference approach:

```
▷ ./stells-linux -p -g <gene.trees>
```

Note that it is usually better to also turn on the -f option:

```
▷ ./stells-linux -f -p -g <gene.trees>
```

Inference of gene trees from haplotypes A common population genetic data comes in the form of genotypes or haplotypes. STELLS assumes haplotypes. If only genotypes are available, you may want to first infer the haplotypes from the genotypes. With the genotypes, at present, one may infer the gene trees from haplotypes. There are several ways.

In the paper Wu (2014), we assume there is no (or fewer) recombination in each locus. Then one can build gene trees directly from haplotypes. See Wu (2014) for more details. In the STELLS_{H} web page (an extension for inferring population tree directly from haplotypes), I have provided some supporting scripts/code to infer gene trees from haplotypes. Then the inferred gene trees can be used for population tree inference. Also see Wu (2015) paper for more information. Note that the more unrefined gene trees are (i.e. with nodes having large degrees), the slower STELLS will become. Thus, you may want to discard very unrefined trees.

2.5 Command line options

For ease of reference, I now provide the list of command line options.

The following list is mandatory.

1. -g <filename>. This species the file that contains one or more gene trees. Each line species a gene tree in the Newick format. Do not include anything else in this file. Note that branch lengths can be included but will be ignored by STELLS. As explained before, the taxon name of gene trees should match the taxon name you have for the species under study. You may have or or more gene alleles for the same species and you should use the same name (i.e. there can be duplicate taxon names in the gene trees). It is important to note that you need to have the same set of taxa in each of the gene trees. That is, each gene tree must have at least one gene allele for each species/population appeared in the gene tree file.

The following lists options that are optional.

1. -s <filename>. This species the species tree to evaluate. If -s is given, STELLS only searches for optimal branch lengths of the given species tree but will not search for optimal species tree topologies.

2. -T <filename>: if you prefer, you can provide STELLS a file containing the initial trees where STELLS is to use when searching for MLE. This may be useful say when you have some ideas what the species tree might be like. The initial tree file should contain one tree (in Newick format) per line, and have no other information.

3. -N <K>: there are often a number of species trees that are just almost as good as the MLE (or approximate MLE). You can invoke this option to make STELLS keep and output the best K (which should be an integer) species trees encountered during MLE search. These near-optimal trees will be output in a file called ¡gene-tree-file¿-nearopt.trees, where ¡gene-tree-file¿ is what you provide as the gene tree file name. In that file, log-likelihood is given followed by a alternative tree. By default, K=10 (i.e. the top 10 species trees are always output by STELLS).

4. -B: Sometimes you may only want to compute probability for a particular species tree and do not want to spend time in searching for the optimal branch length of the provided species tree. In this case, use -B can cut down the computation time. Note: you should not use this if you want to search for optimal species tree (i.e. it should only be used when -s option is used).

5. -v: in case you want to see what ancestral configurations are present (and what are their probabilities), you can add this option. This will activate the verbose mode, and the lists of ancestral configurations will be output at each species tree node.

6. -x Kx: set Kx to be a small number (say 500) to speedup. By default, Kx=5000. This option limits the search for the near-MDC optimal species trees (i.e. more heuristic search of MDC trees to make it faster). This can be useful when data is very large and even the MDC step becomes slow.

7. -S : restricted search of tree space. This can significantly reduce the running time, but of course can lead to less accurate inference. When -S is specified, STELLS will only evaluate near-MDC optimal trees (i.e. does not perform NNI local search). How many near-MDC optimal trees depends on the MDC level (set by the -d option).

8. -d k: set k to a small value (say 1) can make STELLS run faster. Here, k is the near-MDC level. By default, the near-MDC level is 5.

9. -c Kc: set Kc to a small number (say 10) to speedup. By default, Kc=infinity. This option limits the number of configurations (i.e. data structure needed for probability computation).

10. -a Ka: set Ka to a small number (say 500) to speedup. By default, Ka=infinity. This option also limits the number of configurations (i.e. data structure needed for probability computation).

11. -hm $\mathtt{l}_{\mathtt{min}}$: sets the lower bound on the branch lengths considered by STELLS to $\mathtt{l}_{\mathtt{min}}$ during the branch length optimization step. By default, $\mathtt{l}_{\mathtt{min}}$=0.005 (coalescent unit).

12. -hM $\mathtt{l}_{\mathtt{max}}$: sets the upper bound on the branch lengths considered by STELLS to $\mathtt{l}_{\mathtt{max}}$ during the branch length optimization step. By default, $\mathtt{l}_{\mathtt{max}}$=5.0 (coalescent unit).

13. -t <number of threads>: sets the number of threads to use for the species tree inference. The number of threads should be at least two (by default, STELLS uses one thread).

14. -f: use the alternative gene tree probability algorithm, which is faster when the number of gene lineages is large. Unless the number of populations/species is small (say two or three), you may need to also use the -p otion.

15. -p: infer the species tree from pairwise population distance (which is estimated from the gene tree probability). Neighbor joining is used to infer the species tree from the pairwise distance. In order to obtain accurate population distance, it is desirable to have multiple gene lineages from each population/species. Thus, it is also helpful to use the -f option together. Note: the species tree inferred with -p option should be treated as an **unrooted** tree. Without the -p option, the inferred species tree by STELLS should be viewed as a rooted tree.

3 FAQ

STELLS seems to crash on my data

It is possible that STELLS may contain bugs. You may report to me on such cases (with the data that causes the crash).

For large data, STELLS may use large amount of memory. So it is a good practice to monitor how much memory STELLS uses. If STELLS uses too much memory, then you may apply the speedup tricks (see below) which usually also reduces the amount of memory used.

STELLS seems to be too slow...

For large data, STELLS can be slow. The most effective way to speed up is use the $-c$ K_c option. You may set K_c to a small value. For example, let $K_c = 1$ or 5. Usually STELLS is very fast. This works for either species tree search or just gene tree probability computation.

One should note that all these speedup tricks are heuristics. You are likely to get less accurate results with these speedup techniques. So a good strategy is to experiment with the settings. For example, start with small K_c values and gradually increases the value of K_c. A good sign to look for is that the computed likelihood does not increase much with larger K_c values.

How accurate is the species tree inferred by STELLS?

I have performed some simulation in my original paper, which suggests STELLS is reasonably accurate. The following more recent independent studies conducted more simulation. There results put STELLS in an overall favorable position in terms of inference accuracy when comparing with several existing Bayesian or maximum likelihood methods:

R. B. Harris, M. D. Carling and I. J. Lovette, The influence of Sampling Design on Speies Tree Inference: A New Relationship for the New World Chickadees (AVES: POECILE), Evolution, 2013, doi=10.1111/evo.12280.
M. DeGiorgio and J. H. Degnan, Robustness to divergence time underestimation when inferring species trees from estimated gene trees, Systematic Biology, 2013, in press.

How may I interpret the branch lengths?

The inferred branch lengths by STELLS is in the coalescent units. That is, for g generations, the time is measured by $\frac{g}{2N}$ for a diploid organism, where N is the effective population size. Therefore, if you want to convert the time in years to the coalescent units, you need to first have an estimate of (i) generation time t_g, and (ii) N. For example, suppose $t_g = 20$ and $N = 10,000$, then $20,000$ years is equal to $\frac{20000}{20*2*10000} = 0.05$ coalescent units. Also note that STELLS does not assume clock property of the species tree.

The inferred branch lengths do not make sense...

First note that STELLS searches the branch length in certain range. In this version (1.6.1), the range is $[0.005, 5.0]$ (in earlier version, the lower bound was set to 0.1). This may be suitiable for many applications. But in some cases, you may want to use -hm and -hM to change the lower and upper bounds of this range. Moreover, STELLS does not assume clock property. Also note that coalescent units are affected by effective population size and any changes of population size. So you should consider these factors when interpreting the inferred branch lengths.

What if there are missing taxa in some gene tree?

This is a known issue for the current implementation of STELLS. I do plan to address this when I get a chance. For now, you may take subsets of taxa from gene trees so that each gene tree has the same set of taxa. And then you may try to combine the inferred species trees on these subsets. This can be done, for example, with the supertree methods.

4 Revision History

1. 10/27/2015: Release of v.2.0.0. Implement faster gene tree probability computation algorithms that can work with large gene trees when the number of populations (species) is small. Also implement a fast approach based on neighbor joining with pairwise population distance (note: this needs multiple gene lineages from each population). Other changes include adding multithreading option which may speed up the species tree inference.

2. 2/22/2014: Release of v.1.6.1. Fix some issues when dealing with multifurcating gene trees.

3. 05/08/2012: Release of v.1.6. Add -B option to allow users to only compute gene tree probability for a given species tree without searching for optimal species tree branch length. Also fix a bug in gene tree probability computation for large non-binary trees.

4. 04/05/2012: Release of v.1.5. Add -a option to allow faster computation of gene tree probability for some more difficult cases. Also allow the dumping of ancestral configurations for each species tree node (the -v option).

5. 09/15/2011: Release of v.1.4. Allow a user to choose initial trees. Also, output the best K trees in addition to the MLE.

6. 05/15/2011: Release of v.1.3. Now support non-binary trees.

7. 03/14/2011: Release of v.1.2. Support more accurate species tree branch length estimate. This often gives more accurate likelihood computaiton.

8. 11/25/2010: Release of v.1.1. Support heuristic local search of the space of species trees using nearest neighbor interchange (NNI).

9. 09/07/2010: Release of v.1.0. Include basic functionality of computing gene tree probability and search for MLE of the species tree by evaluating only trees within certain distance from parsimony trees.