STEM-hy Tutorial Workshop on Molecular Evolution 2013

Getting started: To run the examples in this tutorial, you should copy the file STEM-hy_tutorial_2013.zip from the `/class/shared/` directory and unzip it.

This document will give a quick overview of how to use the program STEM-hy for species tree inference. There are two steps needed to prepare to run the program, and each is described below.

Step 1. Prepare your gene tree file.
To prepare your gene tree file, you must obtain estimates of your gene trees that satisfy the molecular clock. This can be done in several ways, for example, using PAUP*. The gene trees are assumed to be in "mutation" units (rather than in coalescent units) and are each placed on a separate line in Newick format. The gene trees cannot contain polytomies, but branch lengths of 0 are allowed. An example of a gene trees file is included in the file `genetrees_ex_run1.tre` :

```
[0.37137]((Hheurippa:0.005989,(Hcydno:0.001322,Hmelpomene:0.001322):0.004667):0.022778, ...
[1.17059]((Hmelpomene:0.049843,(Hcydno:0.000001,Hheurippa:0.000001):0.049843):0.001, ...
[0.11434](((Hcydno:0.021024,Hheurippa:0.021024):0.020051,Hmelpomene:0.041076):0.002610, ...
[1.35454](((Hheurippa:0.010740,Hcydno:0.010740):0.003498,Hmelpomene:0.014238):0.037654, ...
[0.39096](((Hheurippa:0.008764,Hmelpomene:0.008764):0.001686,Hcydno:0.010450):0.003969, ...
[1.22683](((Hheurippa:0.002431,Hcydno:0.002431):0.062919,Hmelpomene:0.065350):0.0000001, ...
```

The number in brackets in front of each Newick string is the rate multiplier for that locus. This is the method STEM-hy uses to adjust for rate variation across loci. The user must supply this number, but here are some guidelines. First, this is the manner in which ploidy adjustments can be made. For example, mtDNA loci should be given a rate multiplier of 0.5. It is also important to provide an idea of which loci evolve more rapidly than others. I have found that a method suggested by Yang (2002) works well:
(1) Compute average pairwise sequence divergence of each sequence to the outgroup.
(2) Divide all of these values by their overall mean, and assign that number as the rate multiplier for each gene.
After this procedure, further adjustments for ploidy can be applied (e.g., the rate obtained by the above procedure for and mtDNA tree can be multiplied by 0.5).

One this file has been prepared, it should be placed in a file called `genetrees.tre` within

the directory from which you would like to run STEM-hy.

Step 2. Prepare your settings file.
Options are provided to STEM-hy through the `settings.yaml` file. A typical `settings.yaml` file that corresponds to the `genetrees.tre` file above looks like this:

```
properties:
        run: 1
        theta: 0.001
        beta: 0.0005
        burnin: 100
        seed: 3435893
        bound_total_iter: 20
        num_saved_trees: 10
species:
        H._melpomene: Hmelpomene
        H._hecale: Hhecale
        H._cydno: Hcydno
        H._heurippa: Hheurippa
```

There are two sections in the `settings.yaml` file. In the "properties" section, the user can select the type of analysis that should be carried out through the `run` option. There are 5 options:

Five main functions of STEM-hy version 1.0

- Run=1 :::: Estimate a species tree given a set of gene trees using maximum likelihood
- Run=2 :::: Search species tree space for trees of high likelihood
- Run=0 :::: Compute the likelihood of a user-specified tree
- Run=4 :::: Carry out a bootstrap analysis (bootstrapping is on sites within genes)
- Run=3 :::: Assess fit of trees subject to hybridization in the presence of lineage sorting

The other section of the `settings.yaml` gives the information about which samples belong to which species. There is one line for each species, and on these lines, the species names is given first, followed by a colon. The list of samples belonging to that species is listed

next, in a comma-separated list (with a space following each comma). In the example, we have a species called H._melpomene, with one lineage sampled, called Hmelpomene. If three individuals from the species H._melpomene had been sampled, then that line might be replaced with something like:

H._melpomene: Hmelpomene1, Hmelpomene2, Hmelpomene3

One important note: the lineage names listed here MUST match those used in the genetrees.tre file EXACTLY! You may (or may not) get an informative warning message if this is not true.

Running the program

You should now be ready to run this example data set. The `genetrees.tre` and `settings.yaml` files you want to use are in the files `genetrees_ex_run1.tre` and `settings_ex_run1.yaml`, so you will need to copy those to the proper location. After that, you can run the STEM-hy software using the command: java -jar STEM-hy.jar. To summarize, after copying and unzipping the STEM-hy_tutorial_2013.zip file, you should give the following commands:

> cp genetrees_ex_run1.tre genetrees.tre
> cp settings_ex_run1.yaml settings.yaml
> java -jar stem-hy.jar

After issuing these commands, the output will be written to the screen. It will include the maximum likelihood species tree in Newick format (with branch lengths in coalescent units) and the value of the log likelihood. The maximum likelihood tree will also be written to a file called `mle.tre` in the directory from which STEM-hy was called. In order to estimate support values for the nodes of the estimated species tree, the bootstrap can be used.

Incorporating hybridization

STEM-hy can be used to test for hybridization in the presence of incomplete lineage sorting. There are two changes to the above procedure that are needed to do this. First, information about which species is the possible hybrid must be provided in the `settings.yaml` file. Second, a constraint species tree must be provided in Newick format. This is placed in a file whose name is provided in the `settings.yaml` file. For our data set, we have

```
properties:
                run: 3
                theta: 0.001
                beta: 0.0005
                burnin: 100
                seed: 3435893
                bound_total_iter: 20
                num_saved_trees: 10
                hybrid_species: H._heurippa
                hybrid_tree: user-heliconius.tre
species:
                H._melpomene: Hmelpomene
                H._hecale: Hhecale
                H._cydno: Hcydno
                H._heurippa: Hheurippa
```

To run this example, make sure you have the file `user-heliconius.tre` and then issue the following commands:

> cp settings_ex_run3.yaml settings.yaml
> java -jar stem-hy.jar

Again, the output of the program will be written to the screen, as well as to the file `hybrid-results.txt`. It will include each of the log likelihood and AIC values for each of the two non-hybrid species trees and for the hybrid species tree. In this case, the lowest AIC is for one of the non-hybrid trees, and so it appears that the incongruence in gene trees can be explained solely by the coalescent model (no hybridization) in this case.

On your own

I've created four datasets under varying conditions:

M1 No hybridization, long intervals between speciation events.

M2 No hybridization, short intervals between speciation events.

M3 Low-levels of hybridization - B is a hybrid of A and C (species tree as in M1 and M2).

M4 Extensive hybridization - B is a hybrid of A and C (species tree as in M1 and M2).

All data sets have 6 species, 2 individuals/species, and 10 loci. The species are denoted with letters a - f and the individuals with numbers 1 and 2 in the files denoted `genetrees_W.tre`, `genetrees_X.tre`, `genetrees_Y.tre`, and `genetrees_Z.tre` which are in the file `STEM-hy_challenge.zip` in the `/class/shared/` directory.

GOAL: match the data set to the condition listed above. Solutions are linked to on my course wiki page.