Making a Tree Using RAxML

William Shoemaker and Jay T. Lennon

March 8, 2017

OVERVIEW

Maximum likelihood (ML) is one of the preferred ways of constructing a phylogenetic tree. ML uses the process of finding the parameter value that maximizes the likelihood of the data. In addition, ML has the advantage of being the estimation method that is least affected by sampling error, is fairly robust to the assumptions of a particularly substitution model, allows you to compare different trees, and takes into account nucleotide states (as opposed to just distance). However, ML estimation is a computationally intensive process that can require a lot of time. This is because ML estimation needs to find the tree that gives the data the highest probability, a difficult task that requires the use of optimized algorithms. Because ML estimation can take a long time, you may want to use Randomized Axelerated Maximum Likelihood (RAxML) package for generating maximum likelihood trees with bootstrap support (http://sco.h-its.org/exelixis/web/software/raxml/index.html). Instructions below are for running RAxML on the high speed computer, Mason at IU.

If you are an IU student, postdoc, or faculty member you can register for an account on Mason. See the IU Mason webpage for more information (https://kb.iu.edu/d/bbhh).

Step 1. Create an alligned fasta file

There are differernt ways of doing this that are beyond the scope of this description.

Step 2. Create a shell script that will run RAxML

Below is an example of shell script titled `RAxML_mason.sh` that will make an ML tree file. The # sign in the code below are comments that provide some annotation. In addition, adjust `walltime` for appropriate lenght of time needed. You should also change and include the full directory to where your shell script and fasta file are located. You should change the `-o` option to the name of your appropriate outgroup sequence (if necessary). More description on RAxML code can be found here: http://sco.h-its.org/exelixis/web/software/raxml/hands_on.html

```
#!/bin/bash
#PBS -k o
#PBS -l nodes=2:ppn=8,vmem=100gb,walltime=24:00:00
#PBS -M lennonj@indiana.edu
#PBS -m abe
#PBS -j oe

module load raxml/8.0.26

# cd into the directory with your alignment

cd /N/dc2/projects/Lennon_Sequences/LTDE_Tree

raxmlHPC-PTHREADS -T 4 -f a -m GTRGAMMA -p 12345 -x 12345 -o Methanosarcina -# autoMRE -s ./deathcurves

# -T = number of threads
# -f = specifies bootstrapping algorithm with ML generating tree at same time
```

```
# -m = substitution model, generalized time reversible gamma
# -p = starts tree randomly
# -x = starts tree randomly
# -o = outgroup (name after fasta entry)
# -#  = determines number of bootstrap replicates
# -s = aligned fasta file to be analyzed
# -n = name of output file
```

Step 3. Move files to Mason and execute shell script

To run RAxML, you will need to move files to Mason. This can be done as follows. First, open a terminal window and navigate to the directory on your local computer containing your shell script file and your alligned fasta file. Next you will secure copy the file from your local computer to Mason using commands like this (you will be asked for your password):

```
scp RAxML_mason.sh lennonj@mason.uits.iu.edu:/N/dc2/projects/Lennon_Sequences/LTDE_Tree
```

Now you need to open another terminal window and log into Mason as follows (you will be asked for your password)

```
ssh lennonj@mason.uits.iu.edu
```

Now navigate to your folder containing recently **scp**'d files like this:

```
cd /N/dc2/projects/Lennon_Sequences/LTDE_Tree
```

Now you are ready to submit your job by typing the following at the terminal:

```
qsub RAxML_mason.sh
```

You can check on the status of your job typing the following at the terminal (substitute your username):

```
qstat -ulennon
```

If you need to kill the job, type the following at the terminal after retrieving your job number from qstat

```
qdel 332659
```

To move output files back to your local computer, do the following at the terminal:

```
scp lennonj@mason.uits.iu.edu:/N/dc2/projects/Lennon_Sequences/LTDE_Tree/LTDE.ML /Users/lennonj/Github/
```