Biology 545
Phylogenetics

Laboratory 5: Species-tree Estimation

Since the mid 1980's, we have been aware that coalescent stochasticity, sometimes called incomplete lineage sorting (ILS), can cause the topology of a gene tree to deviate from that of the species tree (early work reviewed by Pamilo and Nei, 1988). For several decades, this ILS was largely ignored by phylogeneticists, but in the last 15 years or so, methods have been developed to account for coalescent stochasticity in estimating the species tree (i.e., the history of divergence by speciation). These approaches use a multispecies coalescent framework to model this process of gene trees evolving within a species tree and are becoming increasingly important as our typical phylogenetic data sets include data from more than a single gene (often many more). We'll discuss these issues in lecture after the second exam.

For gaining practical experience, we'll use two quartets-based approaches to estimate a species tree from multilocus data for six species of chipmunks (Sarver et al. 2021. Syst. Biol., doi.org/10.1093/sysbio/syaa085). The first is called SVDquartets (Chifman & Kubatko 2014) and is the easiest to run. It is perhaps theoretically better justified in that it does not assume all incongruence between the gene trees and species tree is due to ILS and even permits an evaluation of hybridization as a source of incongruence during species-tree estimation. This approach is best implementation in PAUP* and it estimates a species tree directly from the data, that is, with no intermediate step of estimating gene trees first. The other common quartets approach is called ASTRAL and is in its third implementation (ASTRAL-III; Zhang et al., 2018). This approach first requires one to estimate gene trees and then builds a species tree by maximizing the congruence among the quartet trees induced by each gene tree (Miriarab et al., 2014).

Estimating a Species Tree using SVDquartets

First, we'll use PAUP* to estimate a species tree directly from sequence data. The data file is called "NuclearData.nex" and you can download it from the course website: https://www.webpages.uidaho.edu/~jacks/NuclearData.nex.

Execute the nexus file in PAUP using the same procedures you used in Lab 2. Note that this is a concatenated dataset containing 221,556 nucleotides from 1060 loci in 54 chipmunks from 6 ingroup species plus an outgroup. However, SVDquartets treats all sites as unlinked under a multispecies coalescent framework, so this analysis is emphatically not analogous to running an ML (or MP, or MinEvol, or Bayesian) analysis on the (same) concatenated data (where all sites are assumed to have evolved on the same gene tree). Also note that at the end of the nexus file (which can be opened with a text editing program like textWrangler or notepad), there is a PAUP Block that assigns individuals to their species.

Load the PAUP module on the cluster, launch PAUP, and execute the nexus datafile.

Now, you can estimate a species tree for these species using the following command:

```
svdQ evalQ=all taxpartition=munkspecies boot=standard;
```

Since there are ${}_{54}C_4$ (i.e., 54 choose 4) possible quartets (316,251) for 54 taxa, we can evaluate all of them (evalQ=all). With bigger data sets, we could sample, say, 1,000,000 quartets randomly. You'll do 100 bootstrap replicates (the default), and the output will be an estimate of the species tree that includes an assessment of nodal support. This analysis should require <15 minutes of run time. After the run is complete, save the tree with the following command:

```
SaveTrees file = BIOL545_chipmunk_SVDquartet.tre format = Newick
brLens = yes supportValues = Both trees = all;
```

Next, open FigTree and import the BIOL545_chipmunk_SVDquartet.tre file. Under the 'Tree' tab at the top-left, select "midpoint rooting" and then "increasing node order." Then, select the "Node Labels" tab on the left-side column and display the support values (by default, the support values are referred to as 'label'). Lastly, export the tree as a figure (e.g., png or pdf) to hand in.

Estimating a Species Tree using ASTRAL (Accurate Species Tree Algorithm)

For any gene tree, the relationships of each quartet of taxa can be represented by its 4-taxon tree topology, and this can be done for all genes in a multilocus data set. The rationale for the ASTRAL approach is that the species tree that maximizes congruence of quartet trees across all genes and taxa is a consistent estimator of the species tree (Mirarab et al., 2014).

Thus, ASTRAL requires gene trees to be estimated prior to estimating the species tree so that the topologies of the quartets for each gene can be enumerated. This has both advantages and disadvantages. The strength of this approach is that because we estimate individual gene trees, we can identify genes that have highly discordant topologies. These could have biologically meaningful causes and therefore this approach can lead to a better understanding of genome evolution. A disadvantage is that for species-tree estimation to be most accurate, the gene trees must be estimated accurately and be well-resolved. Since genes may be too short and have evolved too slowly for this to be the case, strategies to concatenate genes into bins, or artificial "super-genes," have been developed. Binning strategies include random (naive) binning and statistical binning, where genes for which gene trees do not conflict are concatenated into bins. However, because the multispecies coalescent model assumes no recombination within "genes" and free recombination among "genes" a biologically motivated approach would bin genes into groups based on synteny.

This approach was taken by Sarver et al. (2021) for the dataset that we're using today; genes were assigned to chromosomes and chromosome trees estimated. We'll use these as input trees for ASTRAL, while acknowledging that there likely has been some recombination within each chromosome that we're ignoring.

ASTRAL (Accurate Species TRee Algorithm) is a command-line application that requires Java 1.6 or later. Obtain ASRAL by navigating the following github page: https://github.com/smirarab/ASTRAL and downloading the Astral.5.7.8.zip folder. After downloading the zipped folder, extract its contents to a location you'll remember (e.g., Desktop).

The data file is called 'astral_nuc_trees.tre' and contains all the input gene trees in Newick format. Move this file to the ASTRAL.5.7.8 folder. Next, open a terminal (or Command Prompt if using Windows) and navigate to the Astral.5.7.8 folder (e.g., cd C:\Users\name\Desktop\Astral.5.7.8\Astral). Test if everything is fine with the following command:

```
java -jar astral.5.7.8.jar
```

This should print the help page if your installation was successful. Let's run ASTRAL on the input dataset. Type:

```
java -jar astral.5.7.8.jar -i astral_nuc_trees.tre -a
astral_mapping_file.txt -o astral_nuc_trees_output.tre
```

The output is a tree in Newick format, which can be viewed in FigTree. In addition to topology, ASTRAL also infers branch lengths (reported in coalescent units) and nodal support. Open the .tre file in FigTree and follow the same procedure as above to export a figure (e.g., png or pdf) to hand in.

(Optional)
Additional information from the ASTRAL run can be obtained using the following command:

```
java -jar astral.5.7.7.jar -i astral_nuc_trees.tre -a
astral_mapping_file.txt -o logOutput.tre 2> logOutput.log
```

This exports some important information such as:

- Number of taxa and their names
- Number of loci
- Normalized quartet score (proportion of input gene tree quartet trees satisfied by the species tree). The higher the number, the less discordant your gene trees are.