Dr. Christopher Blair
BIO 4250
Molecular Evolution & Phylogenetics

Overview and Required Software

Over the course of the semester we will work on multiple computer tutorials to familiarize ourselves with some of the commonly used analytical tools uses in molecular evolution and phylogenetics. Some activities will be based on your textbook, whereas others will be more computational in nature. Thus, **you will likely need to download software to your computer**. All of the programs we will use should work on both a PC and Mac. Installation instructions are provided for each system on the website for each program. **Please make sure that you have all required software downloaded and installed before the day of the activity**. Time is limited, and we do not want to spend class troubleshooting program installation. If you are having difficulty obtaining a program, please email your instructor so he/she can help. Data sets will be provided to you by your instructor. Below is a weekly breakdown of the programs you will need. In addition, each tutorial contains questions that are intended to guide you during your analyses. Please take the time to answer each question. If you find that you are having difficulty answering the questions, you may need to go back to a previous topic and work more slowly.

Week 9

1. A general text editor. I recommend **BBedit for Mac** (https://www.barebones.com/products/bbedit/) and **Notepad++ for Windows** (https://notepad-plus-plus.org/downloads/). We will visualize and manipulate data in a text editor for several tutorials.

2. **Aliview** (https://ormbunkar.se/aliview/). Aliview is a general sequence alignment viewer and editor that is very user friendly.

3. **SeaView** (http://doua.prabi.fr/software/seaview). Similar to Aliview, but provides an easy way to concatenate genes into one long alignment.

Week 10

1. **IQ-TREE** (http://www.iqtree.org/). IQ-TREE is a program used to estimate phylogenies using the maximum likelihood (ML) criterion.

2. **FigTree** (https://github.com/rambaut/figtree/releases). FigTree is a program used to visualize phylogenetic trees (not estimate them).

Week 11

1. **MrBayes** (http://nbisweden.github.io/MrBayes/). MrBayes is a program used to estimate phylogenies using Bayesian inference. It is quite versatile, and continues to be a very popular package for phylogenetic inference.

2. **FigTree** (https://github.com/rambaut/figtree/releases). FigTree is a program used to visualize phylogenetic trees (not estimate them).

Week 12

1. **BEAST** (http://www.beast2.org/). Like MrBayes, BEAST is a program that can be used to estimate phylogenies using Bayesian inference. BEAST is also commonly used to infer divergence times and evolutionary rates.

2. **Tracer** (https://github.com/beast-dev/tracer/releases). Tracer is a program used to monitor the results of Bayesian phylogenetic analyses. It can be used in conjunction with both BEAST and MrBayes.

3. **FigTree** (https://github.com/rambaut/figtree/releases). FigTree is a program used to visualize phylogenetic trees (not estimate them).

Week 13

1. **PAUP*** (http://phylosolutions.com/paup-test/). PAUP* is a popular program that was originally used to estimate phylogenies using the maximum parsimony (MP) criterion. The program can also be used for ML and distance-based phylogenies. It implements the species tree method SVDquartets.

2. **Aliview** (https://ormbunkar.se/aliview/). Aliview is a general sequence alignment viewer and editor that is very user friendly.

Dr. Christopher Blair
BIO 4250
Molecular Evolution & Phylogenetics

Week 9 – Databases and Multiple Sequence Alignment

The primary objectives of this lab are to introduce students to three important topics in phylogenetics including (1) **databases**; (2) **multiple sequence alignment**; (3) **common file formats**. Data that are used for phylogenetic analysis generally come from one or two sources. First, data can be collected *de novo* (e.g. sequencing new genes or collecting a suite of morphological characters) or data can be pulled from online repositories. One of the most popular databases is GenBank from the National Center for Biotechnology Information (NCBI). NCBI also maintains the composite database ENTREZ that allows users to query multiple databases simultaneously. Note that these databases are updated regularly, and the number of hits you see is likely to differ from those in the text.

Databases

For phylogenetics, there are a few particularly important databases to be aware of. Navigate to the following website (ncbi.nlm.nih.gov/search/). One of the most important databases is the **Nucleotide** database under the Genomes heading (Fig. 1).

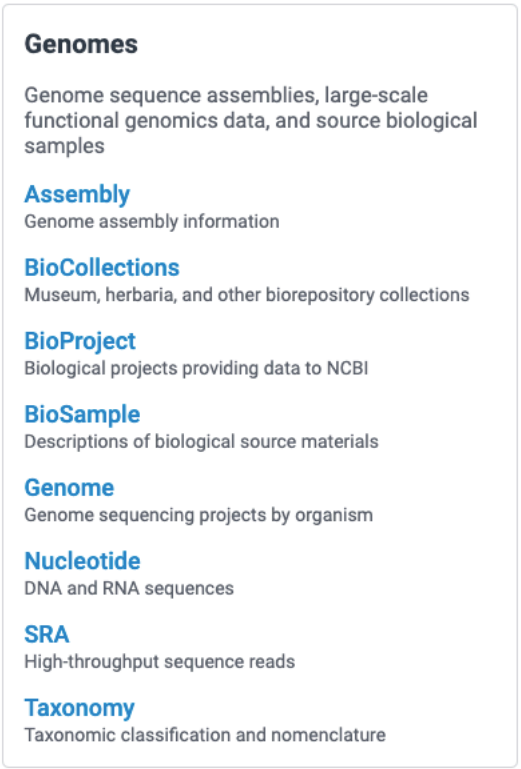

Fig. 1. Where to find the Nucleotide database on NCBI.

Clicking on Nucleotide will open a new link where you can specifically query a nucleotide database only. Two other databases of relevance are the **Protein** database and all of the literature databases that can be used to search for papers relevant to a specific subject or query.

Go ahead and search for *human papillomavirus type 13* in the search bar and click Search. Again, this will search all the listed databases for human papillomavirus type 13 and display the results. Spend some time exploring the different databases to become familiar with them. For example, clicking on the PubMed link will eventually bring you to abstracts of relevant papers. Click on the Nucleotide link to bring up the following page (Fig. 2).

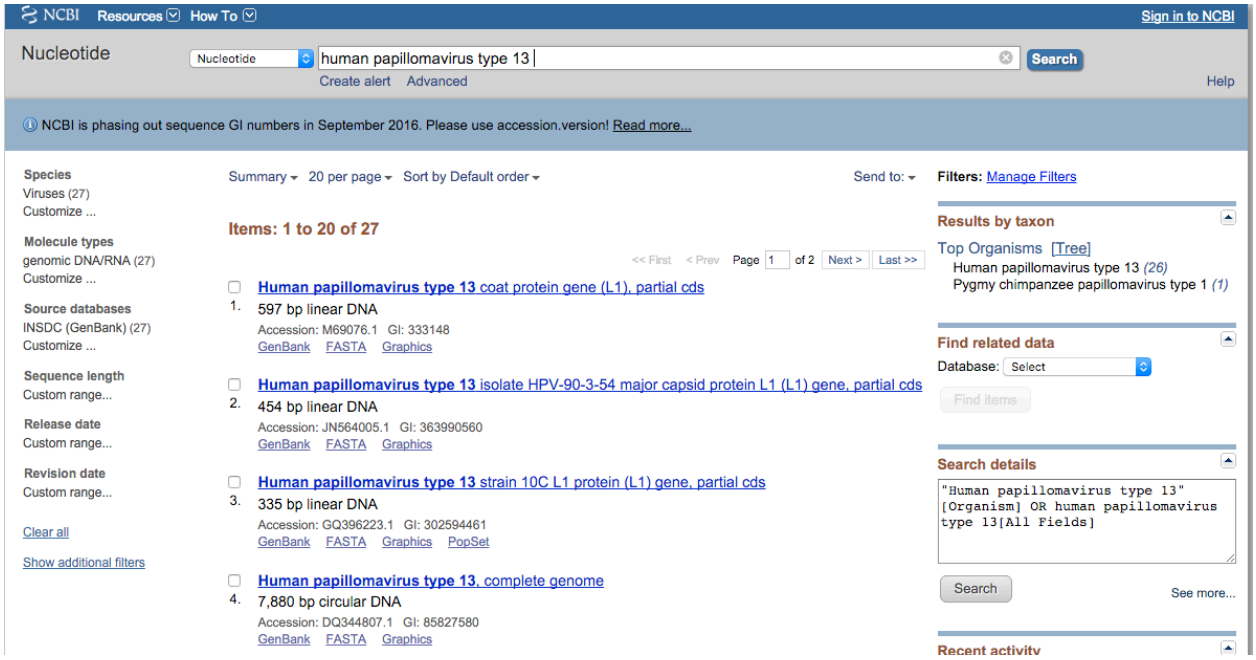

Fig. 2. Screenshot resulting from a search for human papillomavirus type 13 in the Nucleotide (GenBank) database.

This page lists nucleotide information from several genes pertaining to human papillomavirus type 13. Clicking on an item will bring up the full GenBank entry for the sequence (Fig. 3).

Human papillomavirus type 13 coat protein gene (L1), partial cds

GenBank: M69076.1

FASTA Graphics

Go to:

```
LOCUS       PPHL1AA                  597 bp    DNA     linear   VRL 02-AUG-1993
DEFINITION  Human papillomavirus type 13 coat protein gene (L1), partial cds.
ACCESSION   M69076
VERSION     M69076.1  GI:333148
KEYWORDS    coat protein.
SOURCE      Human papillomavirus type 13
  ORGANISM  Human papillomavirus type 13
            Viruses; dsDNA viruses, no RNA stage; Papillomaviridae;
            Alphapapillomavirus.
REFERENCE   1  (bases 1 to 597)
  AUTHORS   Williamson,A.L. and Dennis,S.J.
  TITLE     The use of the polymerase chain reaction for the detection of human
            papillomavirus type 13
  JOURNAL   J. Virol. Methods 31 (1), 57-65 (1991)
   PUBMED   1849917
COMMENT     Original source text: Human papillomavirus type 13 (isolate
            SSV-Natal) viral DNA.
FEATURES             Location/Qualifiers
     source          1..597
                     /organism="Human papillomavirus type 13"
                     /mol_type="genomic DNA"
                     /isolate="SSV-Natal"
                     /db_xref="taxon:10573"
     gene            1..597
                     /gene="L1"
     CDS             <1..>597
                     /gene="L1"
                     /experiment="experimental evidence, no additional details
                     recorded"
                     /codon_start=2
                     /product="coat protein"
                     /protein_id="AAA47016.1"
                     /db_xref="GI:555288"
                     /translation="DPYGDRLFFYLRKEQMFARHFFNRAGSVGEQIPAELYVKGSNTL
                     SNSIYYNTPSGSLVSSEAQLFNKPYWLQKAQGHNNGICWGNHLFVTVVDTTRSTNMTV
                     CAATTSSLSDTYKATEYKQYMRHVEEFDLQFIFQLCTIKLTAEVMSYIHTMNPTILED
                     WNFGLSPPPNGTLEDTYRYVQSQAITCQKPTPDKEKQDP"
ORIGIN
        1 ggatccttat ggagacagat tatttttta tctgcgaaag gaacaaatgt ttgcaaggca
       61 tttctttaac agggcaggct ctgttggtga acaaatccca gcagaattat atgttaaggg
      121 tagtaataca ctttctaata gtatttacta taatactccc agtggctctc ttgtgtcttc
      181 tgaggcccag ttgtttaata aaccttattg gttacaaaag gcccagggac acaataatgg
      241 tatatgttgg ggcaatcact tgtttgttac tgtagttgat actacacgca gtactaacat
      301 gactgtgtgt gcagccacta catcatctct ttcagacaca tataaggcca cagaatataa
      361 acagtacatg cgacatgtag aagaatttga tttacaattt atttttcaat tgtgcactat
      421 taaattaact gcagaggtta tgtcatatat tcatactatg aatcctacaa ttctagaaga
      481 ctggaacttt gggctatctc cccctcctaa tggaacatta gaagacacat atagatatgt
      541 acaatctcag gccataacgt gtcaaaagcc tacacctgat aaagaaaaac aggatcc
//
```

Fig. 3. Full GenBank entry for Accession Number M69076.1.

There is a lot of information presented in a GenBank entry that you should be familiar with. One of the most important items is the Accession Number, in this case M69076.1. This is the number that is used to identify a specific GenBank entry and is assigned by NCBI during sequence submission. You will also find other pieces of relevant information, including sequence length, annotation information, authors, title of article and journal of publication, etc. The term CDS refers to coding sequence and signifies that the sequence belongs to a protein coding gene. As

such, GenBank provides both the full nucleotide sequence and the resulting translation in the correct reading frame.

Before we continue, it is a good idea to become familiar with some of the common file types used when working with sequence data. Probably the most common and one of the simplest is FASTA format. Each entry in a FASTA file begins with a > symbol followed by the taxon identifier and then the data.

Example of FASTA format:

```
>Species1
AATTGTCGAATTTGCTA
>Species2
AATTGCGTAATTGACTG
```

The vast majority of phylogenetic and sequence manipulation packages can work with FASTA files in some capacity. File extensions include .fasta, .fas.

A second file format to be familiar with is the NEXUS format, which is geared specifically towards commonly used phylogenetic packages such as PAUP*, MrBayes, and BEAST. NEXUS files are composed of different 'blocks' that can be added manually by the user or automatically by different software packages. Blocks always use the same syntax:

Example of NEXUS format:

```
Begin XXXX;
.
.
.
.
.
End;
```

Note that the semicolons are important! Open up the file **rattlesnakes-CMOS-Aligned-AllTaxa.nexus** using a text editor on your computer (I recommend BBEdit if available). This file contains an alignment of partial sequences of the nuclear protein coding gene CMOS from several rattlesnakes. You can see that this is a NEXUS file by the block near the top. The 'Begin data' block contains general information about the data including the number of taxa (NTAX) and number of characters (NCHAR). Scroll all the way to the bottom and you will see additional blocks of information, one of which is a codon block. As these sequences are part of an exon, nucleotides will belong to one of three codon positions.

The final file type to be familiar with is the PHYLIP format. Like FASTA, this is a relatively simple format that contains two values on the first line—the first indicating the number of taxa and the second representing the length of the sequences. A PHYLIP file containing three taxa and sequences of 15 characters would look like the following:

Example of PHYLIP format:

3 15

Species1 AATGTACCATGGAAT
Species2 AAGGAACCATGGAAC
Species3 AATTGAACCATGGAT

The PHYLIP format was originally developed by Joe Felsenstein for his PHYLIP phylogenetics package, but is today more commonly used for maximum likelihood phylogenetic inference using the programs RAxML-NG (Kozlov et al. 2019) and IQ-TREE (Minh et al. 2020).

Now that we are comfortable with some of the different file types, let's go back to our GenBank entry for human papillomavirus type 13 coat protein gene (L1), partial cds (Accession M69076.1). Say you are interested in downloading this sequence and using it for phylogenetic analysis. This is easily accomplished by using the 'Send to' button in the upper right corner (Fig. 4)

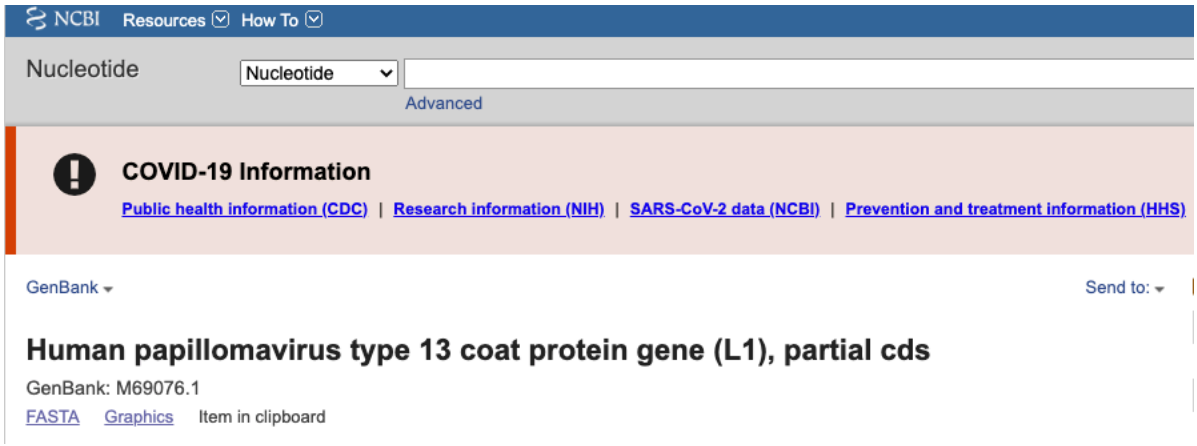

Fig. 4. Sending the data from a GenBank entry to a data file.

Click on Send to > File, and select FASTA Format. This will save a text file containing the sequence. Try opening the file using a text editor (e.g. BBEdit).

This is all fine and dandy, but to perform a phylogenetic analysis we obviously need more than one sequence. Fortunately, GenBank provides an easy way to collect sequences of interest so they do not need to be saved one by one. Go back to the 'Send to' button and instead of choosing File choose Clipboard. This will save the entry to the clipboard to be downloaded later. You will see a Clipboard link on the top right (Fig. 5).

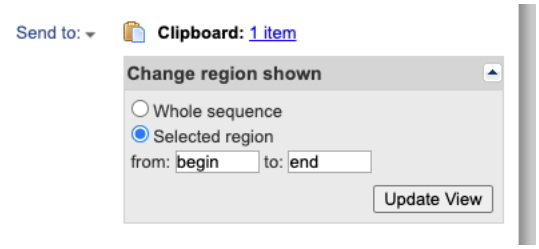

Fig. 5. Saving GenBank entries to a clipboard for subsequent use.

Now, you can search other homologous sequences of interest and add each of them to your clipboard. When you click on the clipboard link you will see all the entries that were saved in your search. You can then export all the sequences in a single text file in FASTA format. This is quite handy and something that you will most likely use at some point.

Say you have a sequence of interest, and are looking for other homologous sequences to compare your sequence of interest to. It may be difficult and time consuming to keep typing things in the search box of GenBank. Fortunately, BLAST does the hard work for us. BLAST contains several different alignment algorithms depending upon the type of query sequence and target database to search (Fig. 6).

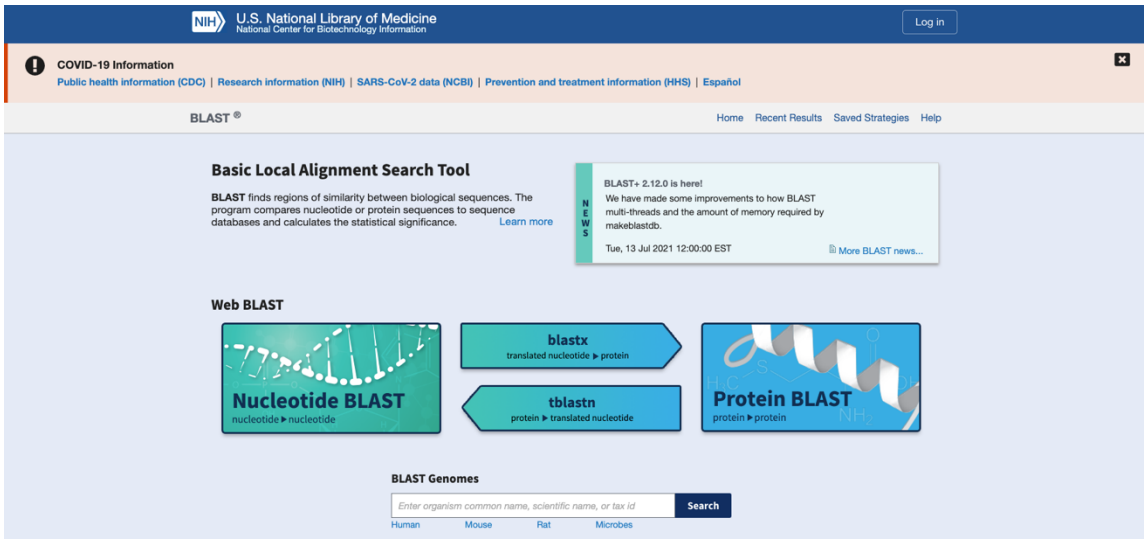

Fig. 6. Alternative versions of BLAST.

Go to the BLAST homepage (https://blast.ncbi.nlm.nih.gov/Blast.cgi) and click on Nucleotide BLAST, and enter the Accession Number **MT456249** in the search box. Make sure the Nucleotide collection database is selected and optimize for somewhat similar sequences (blastn). Go ahead and click the blue BLAST button to start the search. When the search finishes you will see an image resembling the following (Fig. 7). Take some time to go through the results to make sure you understand what is being shown.

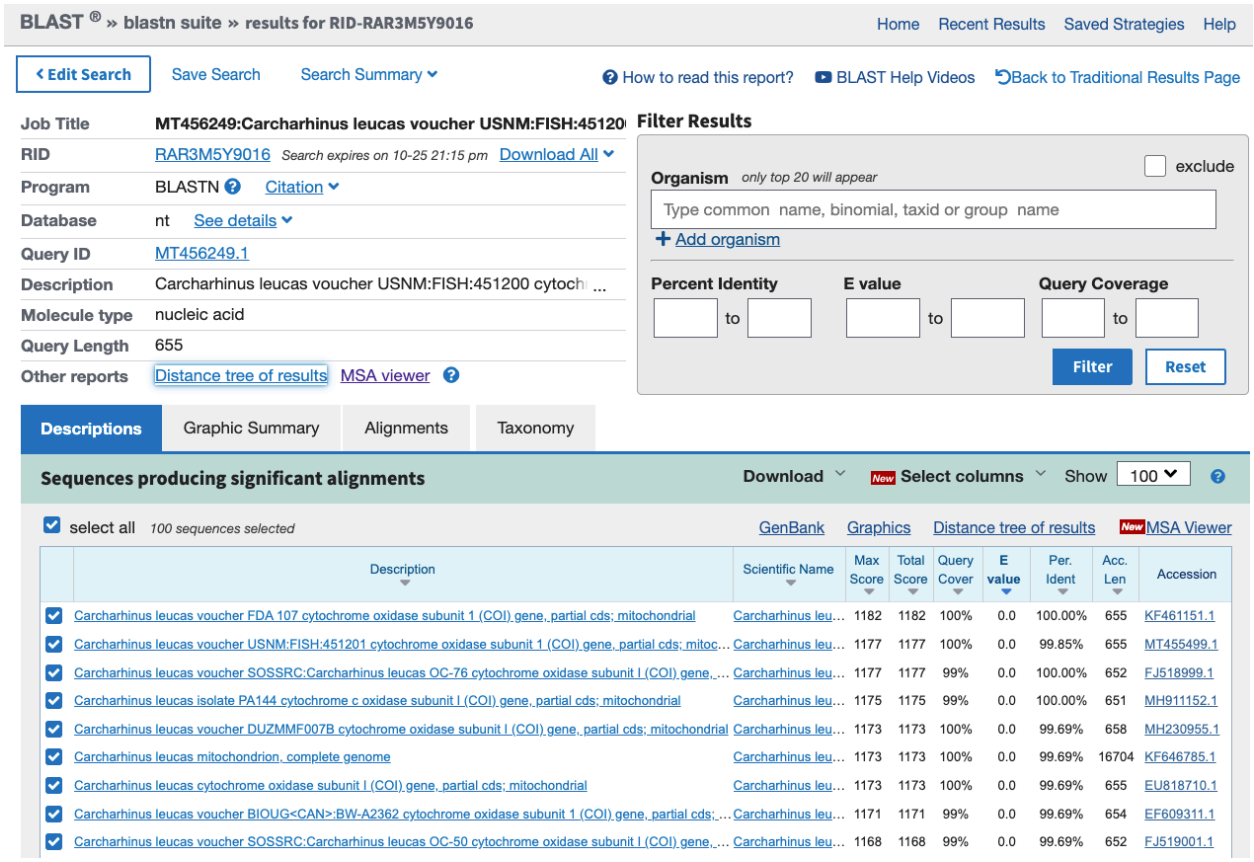

Fig. 7. Results of a BLAST search.

The query sequence is our sequence of interest that we BLASTed (MT456249) and all other rows represent significant hits/alignments. Scroll down and you will see descriptions of the hits, sorted by E() and percent identity. The E-value represents the probability of the match occurring by chance (i.e. not because of true homology). Thus, lower values are more likely to represent similarity in sequences due to shared ancestry. The percent identity column shows how similar the query sequence is to the match. Each entry can be clicked and the data saved as discussed above. If you click on the Alignments tab you will see the actual alignments.

Q: What is the genus and species name for the best match? What is the common name for the organism?

Q: What is the identity of the gene that was queried? Is this a nuclear or mitochondrial gene? Is it a protein coding gene?

Q: What other species are a close match to the query sequence? How did you determine this?

Q: Are all of the best matches from the same genus? If not, what other genera are represented?

Q: What do you think the 'Query Cover' term represents?

Multiple Sequence Alignment

Obtaining an accurate multiple sequence alignment (MSA) is imperative if meaningful phylogenetic analyses are to be conducted. The many different alignment algorithms available attests to the interest in the field of methods development. Some of the more accurate MSA methods include MUSCLE (Edgar 2004), MAFFT (Katoh & Standley 2013), and T-COFFEE (Notredame et al. 2000), although Clustal (Larkin et al. 2007) continues to be one of the most popular methods. The programs come as standalone software for download or web versions where you can upload your data for alignment. Downloaded versions can be run from the command line. A MSA may also be created through manual alignment (i.e. by eye). Manual alignment has several pros and cons. In some cases it is easier for a human to determine homology based on what "looks right". However, when working with highly divergent sequences and/or non protein coding sequences it can be too difficult to align by eye. Aligning by eye is also not feasible when working with genome-scale data sets.

In most cases, we want to actually visualize a MSA and make edits/corrections if necessary. Most MSA algorithms do a decent job providing a good MSA, but sometimes changes are necessary depending on the level of divergence among sequences. Several software packages are available for this purpose. In addition, many of these packages have built-in MSA functions including MUSCLE and/or MAFFT. Two free programs that I like to use are AliView (Larsson 2014) and SeaView (Gouy et al. 2021). These programs also allow you to import/export alignments in different file formats (e.g. FASTA, NEXUS, PHYLIP, etc.).

In this exercise we will work with two data sets, one easy and one difficult. The first data set consists of 18S sequences from several different animal species (**animals.fasta**). These data were used to help determine the phylogenetic affinities of a strange wormlike organism in the genus *Xenoturbella* (Fig. 8).

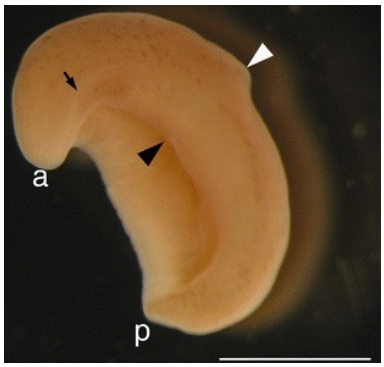

Fig. 8. Image of *Xenoturbella bocki.* Black triangle indicates mouth. Photo credit: Hiroaki Nakano.

Q: What kind of gene is 18S? Is it a protein coding gene? Is it a nuclear or mitochondrial gene? Why do you suppose that researchers chose to sequence the 18S gene in this project?

The second data set consists of RAG1 sequences from multiple species of rattlesnakes (Fig. 9; Blair & Sánchez-Ramírez 2016).

Fig. 9. Eastern Diamondback rattlesnake (*Crotalus adamanteus*). Photo credit: Kevin Enge (flickr.com/photos/myfwc/14996160731).

Q: What kind of gene is RAG1? Is it a protein coding gene? Is it a nuclear or mitochondrial gene? Do you think that RAG1 has a fast rate of evolution?

First, open AliView and import **animals.fasta**.

Q: How many species (sequences) are in the data set? Are the sequences all the same length?

Take some time to try and align some of the sequences manually. Click Edit and turn on edit mode. You can then click on a sequence and move it forward using the spacebar and backward be delete or backspace. Remember that gaps represent insertion/deletion events (indels) and are inserted during a MSA to maintain homology. Try to manually align the two mollusc sequences. Are you able to align these with certainty? Do you know where to insert gaps? It is easy to align the two sequences near the beginning (simply move the *Nucula* sequence two places the right). However, the alignment should get much more difficult in subsequent positions.

Q: Why do you think these sequences are so divergent? After all, these are all 18S sequences.

AliView is pretty versatile and lets you add/remove taxa, edit sequences, translate nucleotides to amino acids specifying different genetic codes, etc.. Because our 18S sequences will be difficult to align by hand, we will perform automatic alignment. Select all sequences (Selection > Select All) and remove all gaps (Edit > Delete All Gaps in all Sequences). You will notice that the sequences no longer have any gaps (-). Click on Align > Change Default Aligner Program > for realigning all. You should see that the MUSCLE algorithm is already built-in. Click Align > Realign everything and you should have a new MUSCLE alignment. Spend a few minutes scrolling through the alignment to see if MUSCLE did a good job.

Q: How long is your MSA? Are there many indels present? Are there any ambiguous regions of the alignment that you may want to remove? Is the level of variability consistent across the gene, or are certain regions more variable than others?

We can save our MUSCLE alignment by clicking File on the menu. You may choose one of several formats including fasta, phylip and nexus. Save the alignment to your desktop.

Now, let's try to align the rattlesnake RAG 1 sequences (**rattlesnakes-RAG1-Unaligned.fasta**). Close the 18S data and open a new AliView window to import the RAG1 data. Similar to the 18S data, take some time to try to manually align the sequences.

Q: Is it easier to manually align the rattlesnake RAG1 data set or the previous 18S data set? Explain your reasoning.

Once you finish manually aligning the rattlesnake data, reopen the unaligned file and perform a MUSCLE automatic alignment.

Q: Is the MUSCLE alignment consistent with your manual alignment? How long is the total alignment? Are there any indels present? Note that gaps and Ns generally represent missing data.

Aliview also allows you to translate nucleotide sequences into amino acid sequences using the **correct reading frame and genetic code**. Make sure that the **Standard Code** is selected under View > Select genetic code for translation. Next, starting from your MUSCLE alignment, try translating the nucleotide sequences in different reading frames. Click View > Show as translation, and you should see an image such as the one below (Fig. 10).

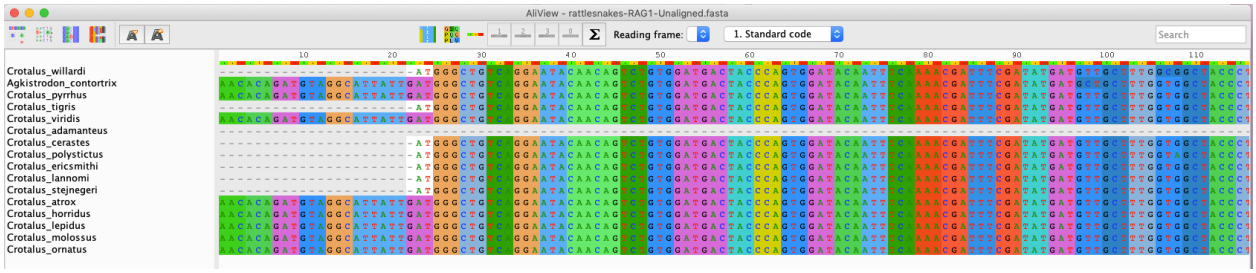

Fig. 10. Rattlesnake RAG1 sequences in translation mode.

Notice that the Standard genetic code is used, which is correct. Our goal is to find the correct reading frame for the translation. For example, should the first nucleotide in the alignment be codon position 1, position 2, or position 3? There are a few ways to do this. In Aliview, click on the sigma icon, which will count the number of stop codons in the alignment. For functional protein coding genes there should be no stop codons present in the middle of the gene.

Q: How many stop codons are present in reading frame 1, reading frame 2, and reading frame 3? Which reading frame do you think is correct?

To make sure we know the correct reading frame, you can BLAST one of the sequences. Click on one of the sequences and then Edit > Copy selection as characters. Paste the sequence into the search box for a nucleotide blast and run the search. Next, click on one of the matches to open up the full NCBI record. Scroll down until you see the translation section that lists the translated amino acid sequence. Does this sequence correspond with your predicted reading frame?

Concatenating several independent MSAs (in class or homework)

AliView is quite useful when working with Sanger sequences and I highly recommend it for your work. Surprisingly, however, AliView is not able to **concatenate sequences**. In concatenation, one MSA is simply added to the end of another MSA and so forth. Concatenation has been a general practice in systematics and phylogenetics for decades, as more characters/data is generally a good thing! However, there are some important limitations to concatenation, which we will discuss later on in the course. An important thing to remember when concatenating is that the **taxon names in each MSA MUST be identical**. If they are not, new rows will be added to the MSA. Thus, it is usually necessary to carefully inspect both your individual and concatenated alignments. Several programs allow you to concatenate sequences, my favorite of which is Geneious. However, Geneious requires a license, which can be pricey. Thus, for this exercise we will use the freely available **SeaView** package (http://doua.prabi.fr/software/seaview).

Go ahead and download SeaView to your system and open it up. For this exercise we will be using two MSAs from rattlesnakes (i.e. **rattlesnakes-CMOS-Aligned-AllTaxa.nexus** and **rattlesnakes-NT3-Aligned-AllTaxa.nexus**). Simply click and drag each alignment into separate SeaView windows. The first step is to make sure that the sequences in both files are actually aligned. In other words, always align each gene/locus individually prior to concatenation (this makes alignment much easier). Next, make sure that the taxon names are identical in both files. If not, you can edit them either directly in SeaView or by using a text editor of your choice. The rest is easy. Click on File > Concatenate in one of your MSA windows and make sure that the 'by name' circle is checked. Hit OK and watch the magic happen! Your second MSA has now been appended to the first.

How does everything look? Were there any errors? Are the data still aligned? We could do this for multiple genes to create a single long MSA for phylogenetic analysis. When concatenating, it is always a good idea to keep track of which sections of the alignment correspond to which gene/locus and codon position. This will be important when we partition the data for phylogenetic analysis. Also, gaps, Ns, or ?s can be used in a MSA to represent missing data.

Q: Which nucleotide positions in the concatenated alignment correspond to CMOS? Which positions correspond to NT3?

References

Blair C, Sánchez-Ramírez S. 2016. Diversity-dependent cladogenesis throughout western Mexico: Evolutionary biogeography of rattlesnakes (Viperidae: Crotalinae: *Crotalus* and *Sistrurus*). *Molecular Phylogenetics and Evolution* 97, 145-154.

Edgar RC. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research* 32, 1792-1797.

Gouy M, Tannier E, Comte N, Parsons DP. 2021. Seaview Version 5: A Multiplatform Software for Multiple Sequence Alignment, Molecular Phylogenetic Analyses, and Tree Reconciliation. *Methods in Molecular Biology* doi: 10.1007/978-1-0716-1036-7_15.

Kattoh K, Standley DM. 2013. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. *Molecular Biology and Evolution* 30, 772-780.

Kozlov AM, Darriba D, Flouri T, Morel B, Stamatakis A. 2019. RAxML-NG: a fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference. *Bioinformatics* 35, 4453-4455.

Larkin MA, Blackshields G, Brown NP, Chenna R, McGettigan PA, McWilliam H, Valentin F, Wallace IM, Wilm A, Lopez R, Thompson JD, Gibson TJ, Higgins DG. Clustal W and Clustal X version 2.0. *Bioinformatics* 23, 2947-2948.

Larsson A. 2014. AliView: a fast and lightweight alignment viewer and editor for large datasets. *Bioinformatics* 30, 3276-3278.

Minh BQ, Schmidt HA, Chernomor O, Schrempf D, Woodhams MD, von Haeseler A, Lanfear R. 2020. IQ-TREE: New models and efficient methods for phylogenetic inference in the genomic era. *Molecular Biology and Evolution* 37, 1530-1534.

Notredame C, Higgins DG, Heringa J. 2000. T-coffee: a novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology* 302, 205-217.

Dr. Christopher Blair
BIO 4250
Molecular Evolution & Phylogenetics

Week 10 – Maximum Likelihood (ML) Analysis in IQ-TREE

Models of molecular evolution

This week we are going to use the program IQ-TREE (Minh et al. 2020) to perform phylogenetic analyses using maximum likelihood (ML). ML and Bayesian methods are presently the most common approaches used to estimate phylogenetic trees. Both techniques require a **model of sequence evolution** that is used to calculate likelihood scores. In general, models have two components: (1) a matrix that includes parameters for the substitution from one nucleotide (or amino acid) to another; (2) base frequencies. Pages 424-427 in your textbook provide a good introduction to substitution models in phylogenetics. The simplest nucleotide model is the **Jukes-Cantor model (JC69)**. This model assumes that the rate of substitution is the same between any pair of nucleotides (i.e. the rate from G to A is the same as C to T, A to C, etc.). The JC69 model also assumes that the bases occur at equal frequencies (25%). In contrast, the **General Time Reversible model (GTR)** assumes different rates of substitution between each pair of nucleotides. The GTR model also assumes different base frequencies. There are many models that fall in between the JC69 and the more parameter rich GTR (refer to the text for additional examples).

All substitution models can also include two additional parameters that account for different evolutionary rates in the multiple sequence alignment. One parameter accommodates the proportion of invariant sites (i.e. sites in which each individual/species share the same base). This parameter is denoted as 'I'. The second parameter accommodates rate variation by assuming a gamma distribution of rates and a shape parameter alpha. This parameter is denoted as 'G'. Thus, we can add +I, +G, or both to any model (e.g. GTR+I+G). In general, we need to balance model complexity and accuracy. We want to choose a model that adequately captures the variation in the data, but we do not want to use an overly complex model that introduces too many parameters. What we need is an objective way to select a substitution model for our analysis. Fortunately, this is easily done in several commonly-used software packages. IQ-TREE has a built-in function called ModelFinder (Kalyaanamoorthy et al. 2017) that we will use. **Remember that a model of evolution is necessary to calculate the overall likelihood of a given phylogenetic tree and branch lengths.**

ML analysis based on a single gene/locus

In this activity we will be performing ML phylogenetic analyses using both single locus and multilocus data sets. When we estimate a phylogeny using a single gene the resulting tree is referred to as a **gene tree**. This tree shows the evolutionary relationships among homologous gene copies sampled from each species. The topology of the gene tree may or may not be the same as the **species tree**, which depicts the evolutionary relationships among the taxa under study. For many questions in molecular evolution, the species tree is of primary interest. Today,

there are many so-called “species tree” methods that seek to estimate the relationships of our species of interest, while accounting for the fact that gene tree topologies may differ. We will explore species trees later on in the semester. One commonly used technique is the **concatenation** method of species tree inference. Here, we simply combine multiple sequence alignments from individual genes into one long supermatrix, which is used as input to our ML analysis.

For our first exercise we will perform ML analysis on two data sets, the **animals.phy** data we used previously and the **example2.phy** data that contains mtDNA sequences from several animals.

*A. ML analysis of **animals.phy***

Recall that these are homologous 18S sequences from several divergent animal phyla. These data were used to help determine the phylogenetic affinities of the strange *Xenoturbella* wormlike organism. I have gone ahead and performed a multiple sequence alignment (MSA) of the sequences using MUSCLE (Edgar 2004) in AliView (Larsson 2014). IQ-TREE should have already been installed on your computer. Note that IQ-TREE is a command line program with no graphical user interface. Thus, we need to open up a terminal window.

1. Click the magnifying glass in the upper right corner and type ‘terminal’ into the search bar. This will open the shell that we will use for our analysis.

2. Next, we need to navigate to the directory where the IQ-TREE executable is located (called **iqtree2**). We do this using the ***cd*** command in Unix. Make sure to place the data in the same directory as IQ-TREE.

3. Typing ***./iqtree2 -h*** will bring up a list of all the available options in IQ-TREE. You will see that there are many options for analysis, but our goal here is to familiarize ourselves with the basics. Thus, we will not fiddle with most of the default values.

4. Run a simple ML analysis using the command ***./iqtree2 -s animals.phy.*** This simple command will first use ModelFinder to determine the best evolutionary model for the data. This is accomplished by determining the likelihood (-LnL) score for each model and using information theoretic criteria (AIC, AICc, BIC) to choose the best model. Models with the lowest AIC, AICc and BIC values are generally the best.

5. You will notice that IQ-TREE deposited several output files in your working directory. The **.log** file is a summary of what is shown in your terminal window. This is a good file to have when you are writing up your methods section.

6. The **.iqtree** file contains the results of the analysis.

Q: How many sites are in your alignment?

How many sites are constant? How many are parsimony informative? Note that parsimony informative sites are sites where there are more than one base present, and each base is represented by at least two sequences.

Q: Which substitution model was chosen by ModelFinder? What was the weighted BIC score of the best model?

Q: What were the relative rates of substitution between bases based on the chosen model?

Q: What were the empirical base frequencies?

If you scroll to the bottom of the file you will see a graphic of the ML tree. **Note that IQ-TREE infers unrooted phylogenies**. However, in most cases we want to root the tree to infer the direction of evolutionary change. This is commonly done by using an **outgroup**. An outgroup is a species or set of species more distantly related to the ingroup, but not too distantly related as to make the comparison uninformative.

7. In your working directory you should see a file ending in **.treefile.** Open this file in the program **FigTree**. FigTree is a general viewer for phylogenetic trees that provides several good options for manipulation. First, make the species names larger and easier to read. Click the dropdown menu for **Tip labels** and increase the font size. Next, root the tree using the cnidarian (*Nematostella vectensis*). Click on the branch leading to this species and then click the **Reroot** button. Your tree should look like the following (Fig. 1).

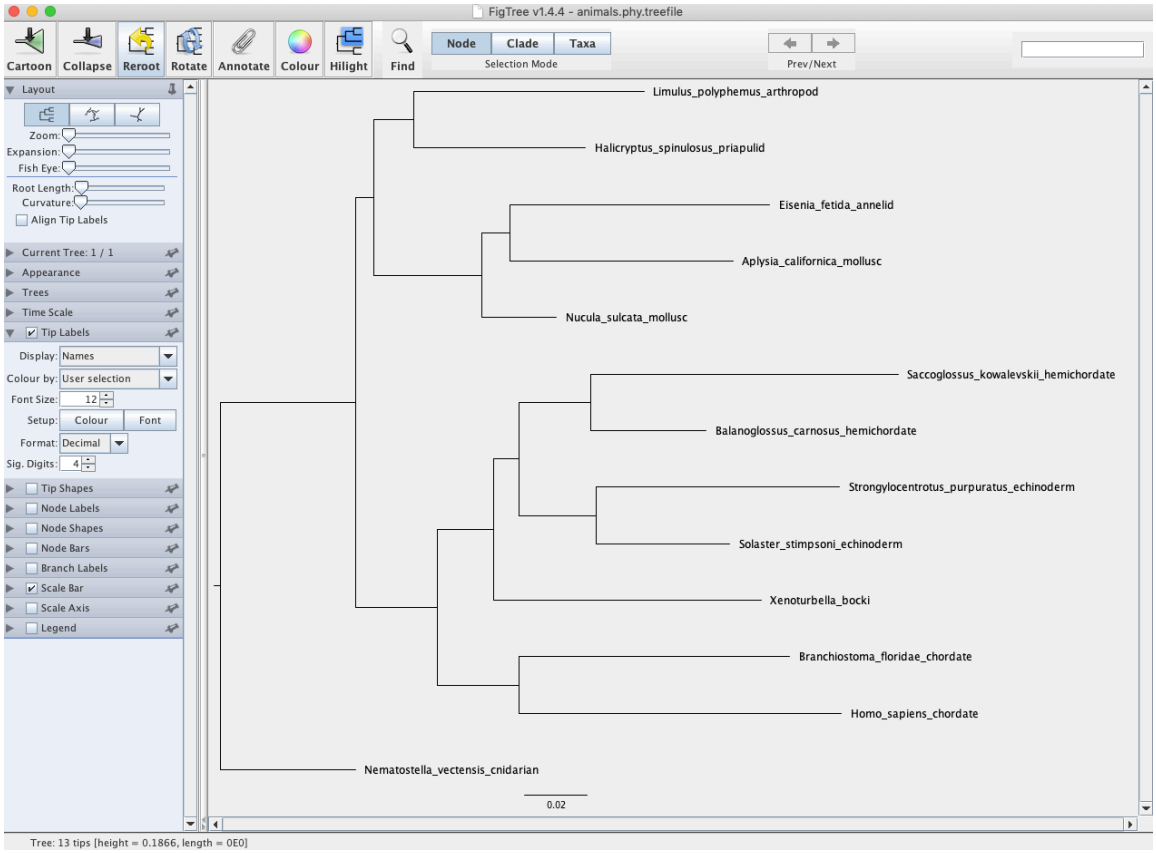

Fig. 1. Phylogenetic trees of animals inferred from maximum likelihood analysis of 18S sequences.

Q: Are all phyla monophyletic? Remember that monophyly is a grouping that includes the common ancestor and all descendants from that ancestor.

Q: What is the sister taxon to humans? What does this organism look like?

Q: Are protostomes and deuterostomes monophyletic?

Q: Does your analysis suggest that *Xenoturbella* is a protostome or deuterostome? Explain your reasoning.

Q: Does your analysis suggest that *Xenoturbella* is a mollusc?

Congratulations on performing your first ML phylogenetic analysis! In addition to the tree topology, we may be interested in determining how confident we are in the relationships based on our data. This can be accomplished by performing a **non-parametric bootstrap analysis** (refer to page 392 in your text). In brief, with bootstrapping we create *x* new pseudoalignments by sampling characters with replacement from the original MSA. A new ML analysis is conducted on each new alignment, and we then determine how often the same relationships are recovered in the bootstrap replicates versus the original alignment. For example, a bootstrap value of 100 means that a clade was found in all of the bootstrap trees. In general, bootstrap values >70 indicate strong support. Let's perform **100 bootstrap replicates** for our animal data.

8. Go back to your terminal window and enter the following command:

   ./iqtree2 -s animals.phy -b 100 -T AUTO

   Here, the *-b* flag says that we want to conduct a standard non-parametric bootstrap analysis with 100 replicates. To utilize multiple computer threads we include the *-T* flag.

9. Go ahead and examine the .**log** and **.iqtree** files as before. This time, the tree image in the .iqtree file has bootstrap support mapped on the nodes. However, let's look at the tree in FigTree.

10. Open the new **.treefile** in FigTree and process it the same as we did above (remember to re-root). To add bootstrap support, click the box for **Branch Labels** and then the drop down arrow. Under **Display** select **label**. You should now see bootstrap support values.

    Q: Are the phylogenetic relationships strongly supported? Is there strong support for a protostome and deuterostome clade? Where is there weak support in your tree?

    Q: Is there strong support for the placement of *Xenoturbella*? Does your analysis suggest that this is a mollusc? What is the sister group to *Xenoturbella*?

B. *ML analysis of **example2.phy***

Next, work through the same pipeline yourself using the **example2.phy** alignment that consists of mtDNA sequences from several vertebrate species. **Use the lungfish as the outgroup**. See if you can ultimately reproduce the figure below (Fig. 2). Note that when performing a bootstrap analysis IQ-TREE will also provide a bootstrap consensus tree (.contree file). You can think of this as an average summary of all the bootstrap replicates. In general, researchers present the ML tree from the original data with the bootstrap values mapped on. This is all stored in the .treefile file.

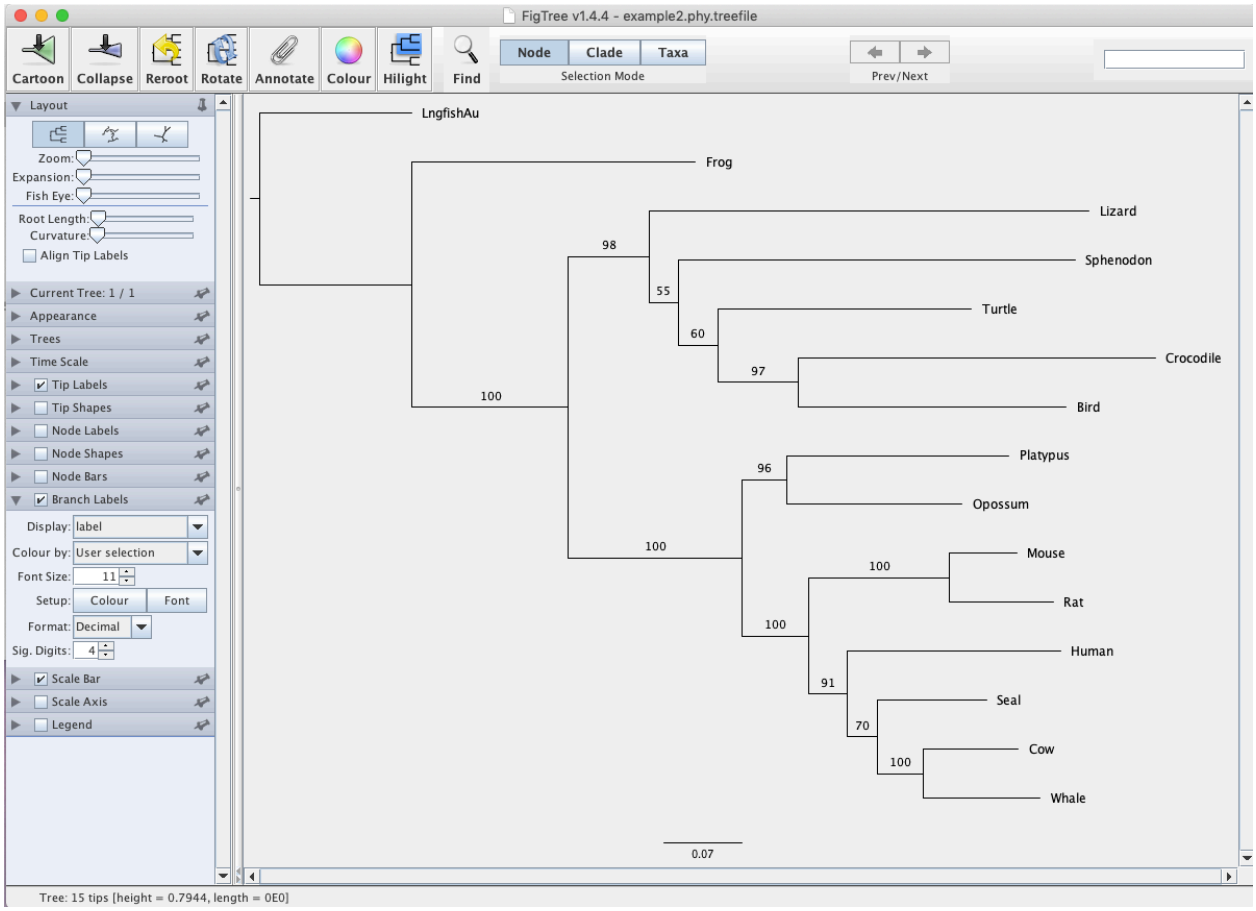

Fig. 2. ML phylogeny of vertebrates based on mtDNA sequences. Values at nodes represent non-parametric bootstrap support from 100 replicates.

Q: How long was your MSA? How many sites were constant? How many were parsimony informative?

Q: Which substitution model was selected for this data set? What was the BIC weight for the best model?

Q: Is your tree strongly supported? Where is there uncertainty?

Q: Is there strong support for the monophyly of mammals? How do you know?

Q: What is *Sphenodon* and where on Earth can we find it?

Q: Which group is the closest living relative of crocodiles? Do your data strongly support this conclusion?

Q: What is the sister group to humans?

ML analysis based on partitioned data

It is now common to base phylogenetic hypotheses off information from multiple genes. One traditional approach for this is to combine (concatenate) multiple genes together to form one long alignment (supermatrix). This supermatrix can then be analyzed using either an **unpartitioned** or a **partitioned** model. When we partition a concatenated alignment, we are allowing for different regions of the alignment to evolve under a different substitution model and set of parameters. Different programs have slightly different requirements for specifying a partition file. In IQ-TREE we can do this in either a NEXUS file or a RAxML-style file. Open the file **example2.nex** in a text editor and you will see the following.

```
#nexus

begin sets;
        charset part1 = 1-999\3 2-999\3;
        charset part2 = 3-999\3;
        charset part3 = 1000-1998;

        charpartition mine = HKY:part1, GTR+G:part2, GTR+G: part3;
end;
```

This is a NEXUS file that divides the alignment into three partitions. We will once again use the vertebrate mtDNA data as an example. If you open up the data (**example2.phy**) you can confirm that the length of the alignment is 1998 bp. Thus, all sites in the alignment have been assigned to a distinct partition. Our first partition consists of first and second codon positions. Remember that codons consist of three bases. Part2 is solely for the third position of codons. Finally, part3 consists of the remainder of the alignment. We can then name our partition scheme ('mine') and assign substitution models to each partition. Here, we are using the HKY model for part1, GTR+G for part2, and GTR+G for part3. Note that in this example we assign the same model to part2 and part3. This is **NOT** the same as lumping the two partitions and assigning a single GTR+G model. When we partition the data, IQ-TREE will provide separate parameter estimates for each partition. Let's now run a partitioned ML analysis!

1. Make sure that the alignment (**example2.phy**), partition file (**example2.nex**) and IQ-TREE executable are all in the same directory.

2. Run the analysis using the following command:

   ./iqtree2 -s example2.phy -p example2.nex -b 100 -T AUTO –prefix example2_partitioned

   which will perform a partitioned ML+bootstrap analysis using the models we provided in example2.nex. We added the prefix flag so we do not get confused with output files generated from the same data set under different analyses.

3. When finished, open up the .log file. You will notice that IQ-TREE loaded the three partitions just as we specified.

4. Open up the .iqtree file and you will also see your partitions and the models we assigned.

   Q: What is the log-likelihood and BIC score for your tree?

5. Open up the tree if FigTree and process it the same as before.

   Q: Is the tree identical to your previously unpartitioned analysis? If not, what are the differences?

In the previous example we *a priori* defined partitions and assigned substitution models to each. This is definitely one common approach to analyze concatenated alignments. A second approach is to statistically test what the best partition scheme is for your data. Once again, we start by dividing the data into biologically meaningful partitions. However, we can use IQ-TREE to test the fit of other partitions. For example, IQ-TREE uses the PartitionFinder algorithm (Lanfear et al. 2012) to merge our partitions to improve model fit. The algorithm stops when no better partition scheme can be found. Once the best partition scheme is detected, ModelFinder will select the best substitution model for each partition. Finally, IQ-TREE will perform a ML search using the best partition scheme and best substitution model(s). **Note that the ModelFinder/PartitionFinder algorithm can only merge partitions, not split them further**. Let's run an analysis.

1. Run the following command:

   ./iqtree2 -s example2.phy -p example2.nex -m MFP+MERGE -b 100 -T AUTO –prefix example2_partitioned_ModelFinder

   Notice that we added -m MFP+MERGE to our command. This is a very useful tool that will perform all of the following steps:

   a. Determine if merging *a priori* defined partitions leads to a better model fit.
   b. Select the best substitution model(s) for the best partition scheme.
   c. Estimate a ML phylogeny using the best partition scheme and substitution model(s).

   Note that we once again run 100 bootstrap replicates to assess confidence in our tree.

2. Carefully examine the .log and .iqtree files.

   Q: What is the BIC score of the full partition model? What is the BIC score of the best partition model? What does this mean? What partition scheme and model were used for your final analysis?

Q: What is the log-likelihood and BIC score for your tree? According to BIC scores, is your first partitioned analysis (where you specified partitions and models) or your second (where you tested alternative partitions) better?

References

Edgar RC. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research* 32, 1792-1797.

Kalyaanamoorthy S, Minh BQ, Wong TKF, von Haeseler A, Jermiin LS. 2017. ModelFinder: Fast model selection for accurate phylogenetic estimates. *Nature Methods* 14, 587-589.

Lanfear R, Calcott B, Ho SYW, Guindon S. 2012. PartitionFinder: Combined Selection of Partitioning Schemes and Substitution Models for Phylogenetic Analyses. *Molecular Biology and Evolution* 29, 1695-1701.

Larsson A. 2014. AliView: a fast and lightweight alignment viewer and editor for large datasets. *Bioinformatics* 30, 3276-3278.

Minh BQ, Schmidt HA, Chernomor O, Schrempf D, Woodhams MD, von Haeseler A, Lanfear R. 2020. IQ-TREE: New models and efficient methods for phylogenetic inference in the genomic era. *Molecular Biology and Evolution* 37, 1530-1534.

Dr. Christopher Blair
BIO 4250
Molecular Evolution & Phylogenetics

Week 11 – Bayesian inference

Today we will begin working with Bayesian methods to estimate phylogeny and associated evolutionary parameters. Bayesian methods have become extremely popular, partly because they are inherently statistical and allow for a fair degree of flexibility with analyses. Bayesian methods have several similarities with maximum likelihood (ML), particularly through the use of the likelihood function. However, unlike ML, a Bayesian phylogenetic analysis returns an entire posterior distribution of trees and associated parameters (instead of a single ML estimate). There are pros and cons to both methods and often heated debates at scientific meetings and other venues as to which technique is superior. Bayesian analyses require the use of **prior distributions** on parameters. As the analysis progresses, the prior is updated via the data to return the posterior distribution. How to specify an appropriate prior is one of the most contentious issues in Bayesian phylogenetics (and Bayesian analysis more generally).

We will be working with one of the most popular programs for Bayesian phylogenetics — MrBayes (Huelsenbeck et al. 2001; Ronquist & Huelsenbeck 2003; Ronquist et al. 2012). MrBayes is strictly a command line program, which is useful to prevent researchers from 'black boxing' their analyses. **Note that MrBayes requires the data to be in NEXUS format**. Similar to the ML lab with IQ-TREE, we will first be working with both the 18S animals data set and the vertebrate mtDNA data set. We will once again be performing unpartitioned and partitioned analyses. Make sure that all the input files are in the same directory as the MrBayes executable (*mb*).

A. Bayesian analysis of the animals 18S data set

Our first objective is to estimate a Bayesian phylogeny of the 18S animals data set from your textbook. As MrBayes requires a nexus file as input, I have already converted animals.phy to **animals.nexus**.

1. MrBayes should already be installed on your computer. Your instructor will help you locate the executable. Once MrBayes is loaded, import your data file using the command

   exe animals.nexus

2. Typing ***help*** at the command prompt will bring up a large number of options for analysis. Pay particular attention to the ***lset*** and ***prset*** commands that control the substitution model and priors respectively. Remember that a major difference between Bayesian and maximum likelihood is that Bayesian methods require priors for all parameters. Typing ***help lset*** will show you the current settings and options. For nucleotide data we can set the number of substitutions to 1, 2, or 6. These correspond to some of the more common models we discussed in class. For example, *nst* =1 would be used under the JC or F81 model that assume a single substitution rate between nucleotides. Conversely, the

parameter rich GTR model assumes different substitution rates for each pair of nucleotides, thus we would set *nst*=6. We can also specify gamma distributed rate heterogeneity and a proportion of invariable sites using the ***lset*** command.

Q: How would you specify the K2P substitution model in MrBayes?

After any changes to the model are made, typing

help lset

a second time should illustrate the changes.

A useful new feature of MrBayes is that there is an option that obviates the need to specify a substitution model *a priori*. Instead, the analysis will sample all of the possible substitution models. In addition, because our sequences are so divergent, we will want to account for different substitution rates across sites. We can do this by adding a parameter for the proportion of invariant sites (I) and gamma (G). Thus, we will specify

lset nst=mixed rates=invgamma

Typing ***help lset*** again will allow you to make sure that the changes have been implemented (Fig. 1).

```
Model settings for partition 1:

Parameter    Options                                 Current Setting
---------------------------------------------------------------------
Nucmodel     4by4/Doublet/Codon/Protein                4by4
Nst          1/2/6/Mixed                               Mixed
Code         Universal/Vertmt/Invermt/Yeast/Mycoplasma/
             Ciliate/Echinoderm/Euplotid/Metmt         Universal
Ploidy       Haploid/Diploid/Zlinked                   Diploid
Rates        Equal/Gamma/LNorm/Propinv/
             Invgamma/Adgamma/Kmixture                 Invgamma
Ngammacat    <number>                                  4
Nlnormcat    <number>                                  4
Nmixtcat     <number>                                  4
Nbetacat     <number>                                  5
Omegavar     Equal/Ny98/M3                             Equal
Covarion     No/Yes                                    No
Coding       All/Variable/Informative/Nosingletons
             Noabsencesites/Nopresencesites/
             Nosingletonabsence/Nosingletonpresence    All
Parsmodel    No/Yes                                    No

---------------------------------------------------------------------
```

Fig. 1. Model settings for MrBayes. Note that the settings for *Nst* and *Rates* have been changed from the default values.

3. It's always a good idea to check the **priors** before running a Bayesian analysis. In MrBayes, this can be done by typing ***help prset***. As you will see, there are many many options, but the default values here are sensible. Thus, we will not make any changes.

4. Before running the analysis, it is good practice to check the model and the parameters to be estimated (Fig. 2).

   showmodel

```
Model settings:

   Data not partitioned --
      Datatype  = DNA
      Nucmodel  = 4by4
      Nst       = Mixed
                  Substitution rates, expressed as proportions
                  of the rate sum, have a Dirichlet prior
                  (1.00,1.00,1.00,1.00,1.00,1.00)
      Covarion  = No
      # States  = 4
                  State frequencies have a Dirichlet prior
                  (1.00,1.00,1.00,1.00)
      Rates     = Invgamma
                  The distribution is approximated using 4 categories.
                  Shape parameter is exponentially
                  distributed with parameter (1.00).
                  Proportion of invariable sites is uniformly dist-
                  ributed on the interval (0.00,1.00).

Active parameters:

   Parameters
   ----------------------
   Revmat              1
   Statefreq           2
   Shape               3
   Pinvar              4
   Ratemultiplier      5
   Topology            6
   Brlens              7
   ----------------------
```

Fig. 2. Current model and active parameters.

Q: How many parameters are in your model? What are they, and what do they mean?

Q: Are we favoring a particular tree topology *a priori*? How do you know?

5. Finally, we need to specify how long we wish to run our analysis. Typing ***help mcmc*** will bring up a list of options. The three most important options to be aware of are the following:

   Ngen

   Nruns

   Nchains

   Ngen specifies how long to run the analysis. We will use the default of 1 million generations. **Nruns** specifies how many independent runs we wish to implement. In general, it is good practice to execute multiple independent runs from different starting trees. If the analysis is working well, both independent runs should give us the same answer. If they do not, we may have to spend time troubleshooting. We will use the default of 2 independent runs. **Nchains** is related to a process called Metropolis Coupling. A value of 4 indicates that each independent run will consist of 4 chains that are sampling parameter values. One chain is what we call the 'cold chain' and the remaining chains are 'heated'. Heated chains can often sample parameter space more efficiently and can help make sure that the cold chain does not get 'trapped' in a suboptimal region. The **SwapFreq** setting specifies how often two chains try to swap states. We will keep the default value of 1. **Samplefreq** specifies how often the chain is sampled. Let's change the default value of 500 to 100.

   mcmcp samplefreq=100

   This means that the analysis will return 1,000,000/100 = 10,000 samples of the posterior which we can summarize.

   In any type of Bayesian analysis we want to make sure that we discard samples early on in the analysis. This is because these samples have relatively low likelihood values and we do not want the parameter estimates to bias our analysis. The default settings in MrBayes will discard the first 25% of samples for the cold chain (**relburnin=yes, burninfrac=0.25**). This is what we will use for our analysis.

6. We are now ready to run our first Bayesian phylogenetic analysis! Simply type

   mcmc

   at the command prompt and the analysis should start. You should see lots of negative numbers on each line (Fig. 3). These represent log likelihood values from each chain. Remember that we implemented 2 runs, each with 4 chains (2 x 4 = 8 total chains). The two runs are separated by the * symbol and the cold chain is indicated by brackets [].

```
Average standard deviation of split frequencies: 0.005118

306000 -- [-10027.199] (-10030.902) (-10037.089) (-10033.205) * [-10027.144] (-10028.752) (-10037.620) (-10030.791) (...0 remote chains...) -- 0:08:00
307000 -- [-10036.319] (-10037.676) (-10043.984) (-10042.374) * [-10032.630] (-10034.824) (-10029.500) (-10039.365) (...0 remote chains...) -- 0:07:58
308000 -- (-10034.079) (-10038.846) (-10030.170) [-10033.838] * (-10026.879) [-10031.281] (-10029.959) (-10035.586) (...0 remote chains...) -- 0:07:58
309000 -- (-10031.938) (-10032.895) [-10033.160] (-10046.366) * (-10034.989) [-10034.864] (-10038.368) (-10034.038) (...0 remote chains...) -- 0:07:58
310000 -- (-10032.412) (-10038.313) [-10034.291] (-10042.722) * (-10035.484) [-10030.497] (-10029.504) (-10037.693) (...0 remote chains...) -- 0:07:56

Average standard deviation of split frequencies: 0.002102
```

Fig. 3. Example of a MrBayes analysis in progress. This analysis used two independent runs, each with four chains (one cold and three heated).

The analysis should take approximately 10 min to complete. When finished, MrBayes will ask if you wish to continue with the analysis. Before you answer, look closely at the **Average standard deviation of split frequencies**. This value is an indication of how different your two independent runs are (the smaller the better). To make sure that we have confidence in our results, make sure that this number is <0.01. In my analysis the value is 0.002789. Thus, I will type ***no*** into the prompt.

7. If you look inside your working directory, you will see that MrBayes output two .p files and two .t files. The .p files are parameter values and the .t files store the phylogenetic trees sampled. We will first summarize the parameters. Type

   sump

   at the prompt and MrBayes will summarize the results in both .p files. Make sure that 25% of samples are discarded as burnin. If you scroll down, you will see a figure showing generation versus log likelihood (Fig. 4). There should be no obvious trends/patterns in the plot. If there are, you likely need to run the analysis longer.

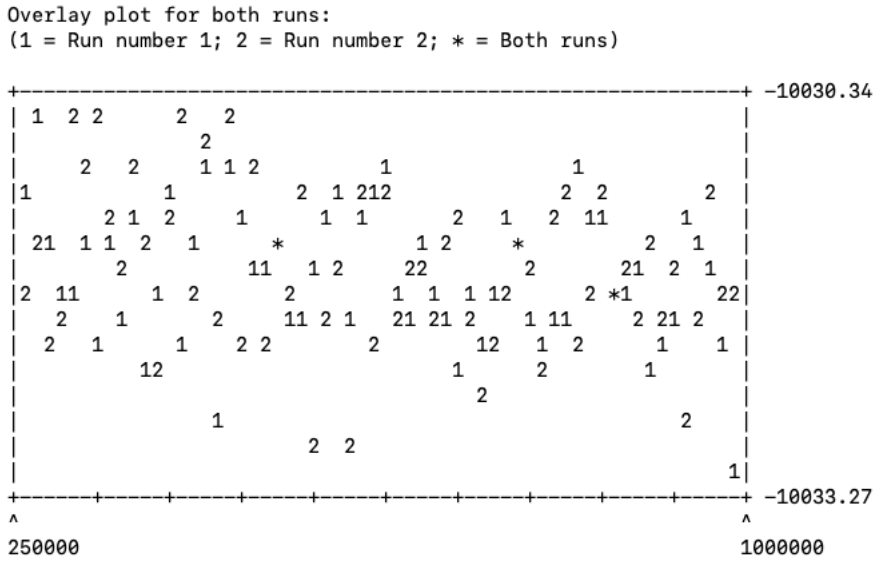

Fig. 4. Diagnosing stationarity in MrBayes. The x-axis represents generation and the y-axis indicates log likelihood. When stationarity is reached there should be no obvious trends in likelihood values.

If you continue to scroll down you will see a table with parameter estimates (Fig. 5).

			95% HPD Interval					
Parameter	Mean	Variance	Lower	Upper	Median	min ESS*	avg ESS	PSRF+
TL	1.170621	0.002632	1.072903	1.274263	1.169414	5831.91	6065.15	1.000
r(A<->C)	0.100476	0.000071	0.084739	0.118693	0.099971	2449.78	2694.85	1.000
r(A<->G)	0.213854	0.000248	0.183037	0.244840	0.213495	1749.10	1858.76	1.000
r(A<->T)	0.096185	0.000069	0.078000	0.110842	0.097176	2442.80	2504.12	1.000
r(C<->G)	0.109263	0.000132	0.093146	0.134669	0.106173	2120.96	2178.75	1.000
r(C<->T)	0.384775	0.000381	0.348287	0.425048	0.384404	1788.83	1828.00	1.000
r(G<->T)	0.095447	0.000068	0.077375	0.109688	0.096532	2603.46	2695.00	1.000
k_revmat	3.890215	0.430517	3.000000	5.000000	4.000000	2755.53	2870.86	1.000
pi(A)	0.222392	0.000067	0.206996	0.238822	0.222316	1886.96	2037.24	1.000
pi(C)	0.257066	0.000074	0.241111	0.274499	0.256819	1556.67	1618.07	1.001
pi(G)	0.293147	0.000083	0.275259	0.310633	0.293289	1612.88	1725.92	1.001
pi(T)	0.227395	0.000065	0.211099	0.242474	0.227376	2037.18	2063.90	1.000
alpha	0.507536	0.010044	0.346288	0.718548	0.493353	708.24	787.97	1.000
pinvar	0.189131	0.005158	0.041467	0.321252	0.190651	797.46	829.08	1.000

* Convergence diagnostic (ESS = Estimated Sample Size); min and avg values correspond to minimal and average ESS among runs.
ESS value below 100 may indicate that the parameter is undersampled.
+ Convergence diagnostic (PSRF = Potential Scale Reduction Factor; Gelman and Rubin, 1992) should approach 1.0 as runs converge.

Fig. 5. Parameter estimates from a MrBayes analysis.

Pay attention to ESS and PSRF values. Ideally. We want ESS values > 200 and PSRF values near 1. If this is not the case. We may need to run the analysis for longer. Here, it looks like we are ok.

Q: Look at the list of parameters and their values. Do you know what each parameter represents?

8. The final step is to summarize the trees in the .t files. Remember that unlike ML, a Bayesian analysis returns a posterior distribution of trees. In many cases this consists of thousands of alternative trees. Thus, we need some way of summarizing these into one useful summary tree. There are a few ways we could do this. MrBayes generates a **majority rule consensus tree**. This is a tree where clades occur in >50% of sampled trees (post burnin). Type

   sumt

   and MrBayes will summarize your trees.

Q: How many total trees are in your .t files?

Q: How many total trees are retained after burnin?

As you scroll down you will see additional information about the analysis. Near the bottom you will see two phylogenies. The first is the majority rule consensus tree with **posterior probabilities** on nodes. Posterior probability values are a measure of statistical support. In general, values > 95 indicate strong support. Finally, you will see a phylogram with branch lengths measured in expected substitutions per site.

9. If you look inside your working directory you should see a file ending in **.con.tre**. This is your consensus tree from MrBayes. Open this file in FigTree and process the tree similar to what we did for our IQ-TREE analyses.

   a. Increase the font size for taxon names.
   b. Root the tree
   c. Add posterior probability values (Node Labels > Display > prob(percent)

   Your phylogeny should look similar to the one below (Fig. 6).

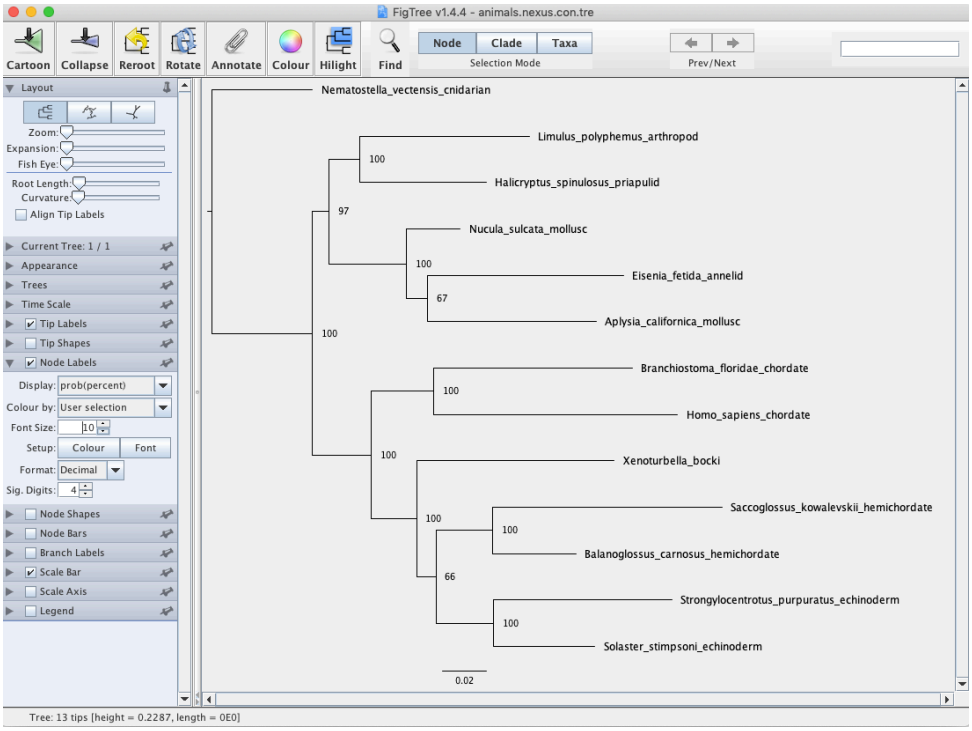

Fig. 6. Bayesian majority rule consensus tree of animal phyla inferred from 18S sequences. Values at nodes represent posterior probability values.

Q: Is your Bayesian phylogeny the same as your ML phylogeny from the previous week?

Q: Does your Bayesian analysis suggest that *Xenoturbella* is a mollusc?

B. Bayesian analysis of the vertebrate mtDNA data set

See if you can replicate the steps above to perform a Bayesian phylogenetic analysis of the vertebrate mtDNA data (**example2.nexus**). Note that we will change some of the analysis settings here. By following the directions above, **change Ngen to 500,000, Samplefreq to 50, and Diagnfreq to 1000.** Raise your hand if you need help with this. Remember to root the tree using the lungfish. Recall that we also used this data set to estimate a ML tree in IQ-TREE. You should obtain a tree similar to Fig. 7.

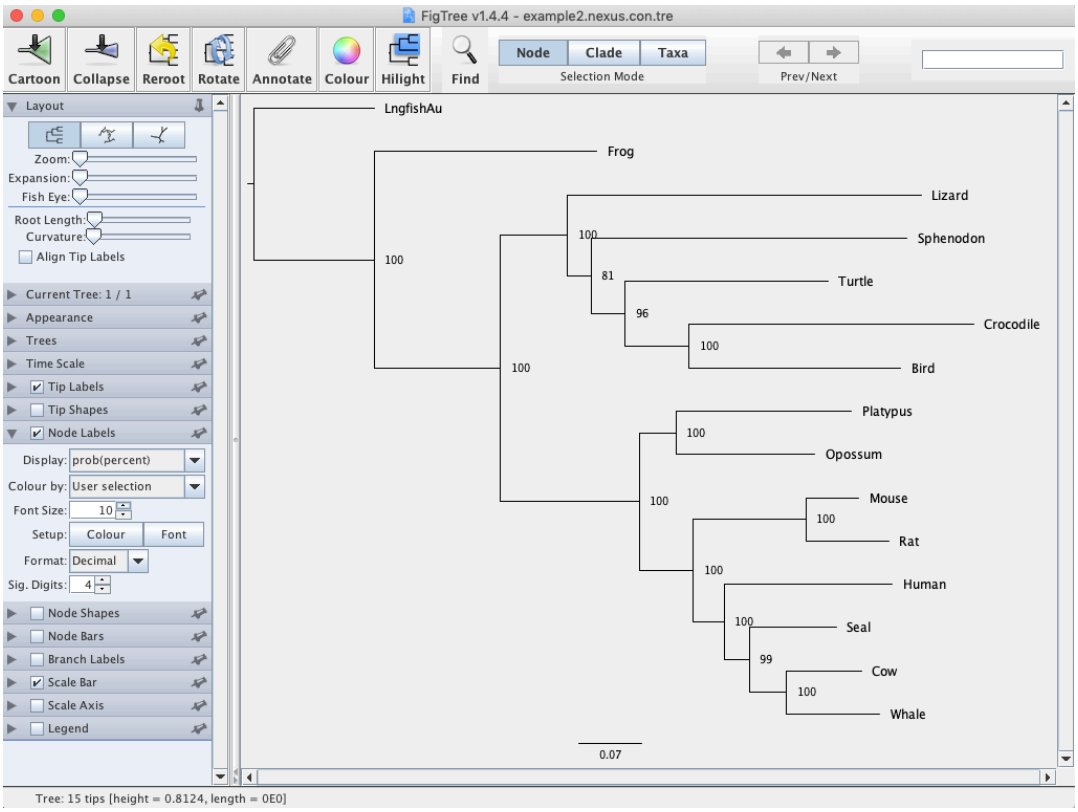

Fig. 7. Bayesian majority rule consensus tree for the vertebrate mtDNA data set. Values on branches represent posterior probability values.

Q: Describe the phylogeny in your own words. What are the major evolutionary findings? Try to use the following words in your description: *clade, monophyletic, sister, posterior probability*. What relationships are not supported?

Q: How does your Bayesian tree compare to your ML tree you estimated last week?

C. Partitioned Bayesian analysis of the vertebrate mtDNA data set

Similar to our ML phylogenetic analysis using IQ-TREE, we can perform a partitioned Bayesian analysis in MrBayes. Again, this is useful when analyzing data from multiple genes, combined nucleotide and amino acid data, or combined molecular and morphological data. Note that this can be done by creating a separate text file, by appending a **mrbayes block** to the data file (common), or by directly typing commands at the MrBayes prompt.

1. Open up the file **example2-Partitioned.nexus** in a text editor and scroll to the bottom. This file is identical to the one we just analyzed, but I have added a mrbayes block that defines partitions. Note that like IQ-TREE, we are specifying three different data partitions and naming the partition scheme 'mine'. Also note that we are not specifying substitution models for our partitions here. Instead, we will do this in MrBayes.

2. Execute example2-Partitioned.nexus

   exe example2-Partitioned.nexus

   If all goes well, MrBayes should tell you that it successfully read the mrbayes block and recognizes the three partitions (Fig. 8).

```
Reading mrbayes block
   Defining charset called 'part1'
   Defining charset called 'part2'
   Defining charset called 'part3'
   Defining partition called 'mine'
   Setting mine as the partition, dividing characters into 3 parts.
   Setting model defaults
   Seed (for generating default start values) = 767135436
Exiting mrbayes block
Reached end of file
```

Fig. 8. Text illustrating that MrBayes recognizes the data partitions specified in the input nexus file.

3. Typing

   showmodel

   will show the current analysis settings. **Make sure you see information here for all three partitions**.

4. Our next objective is to specify models for each partitions. We can do this easily using the ***applyto*** command. Assume that we want to assign the same model as we did for our unpartitioned analysis.

   lset applyto=(1,2,3) nst=mixed rates=invgamma

   Typing ***showmodel*** again should show the changes.

5. Scroll down until you see a table that shows the current parameters to be estimated for each partition (Fig. 9). The same number across partitions indicates that only a single parameter estimate will be made across partitions. For example, in the current settings MrBayes will estimate a single set of state frequencies for Partitions 1-3.

```
Active parameters:

                    Partition(s)
   Parameters       1  2  3
   ------------------------
   Revmat           1  1  1
   Statefreq        2  2  2
   Shape            3  3  3
   Pinvar           4  4  4
   Ratemultiplier   5  5  5
   Topology         6  6  6
   Brlens           7  7  7
   ------------------------
```

Fig. 9. Current active parameters in a partitioned MrBayes analysis. Note that all parameters are linked across partitions.

   Note that we generally want the topology and branch lengths **linked** among partitions (i.e. we want one topology and set of branch lengths). Conversely, we probably want different model parameters (e.g. alpha, nucleotide substitution rates) for each partition. Thus, we will need to **unlink parameters**.

unlink statefreq=(all) revmat=(all) shape=(all) pinvar=(all)

If we then type ***showmodel*** again we will now see that these parameters are unlinked and MrBayes will estimate each of their values (Fig. 10).

```
Active parameters:

                     Partition(s)
   Parameters         1  2  3
   ---------------------------
   Revmat             1  2  3
   Statefreq          4  5  6
   Shape              7  8  9
   Pinvar            10 11 12
   Ratemultiplier    13 13 13
   Topology          14 14 14
   Brlens            15 15 15
   ---------------------------
```

Fig. 10. Current active parameters in a partitioned MrBayes analysis. Note that the parameters *Revmat*, *Statefreq*, *Shape*, and *Pinv* are now unlinked across partitions.

6. Also make sure that you set the evolutionary rates among partitions to variable.

   prset applyto = (all) ratepr=variable

   as it is highly unlikely that each gene/codon position is evolving at the same rate.

7. We should now be ready to run our partitioned Bayesian analysis.

   mcmc

8. When the analysis finishes summarize the parameters and trees using the ***sump*** and ***sumt*** commands, respectively.

   Q: Did we run the analysis long enough? Why or why not?

9. Visualize the tree with posterior probability values (.con file) in FigTree.

   Q: Are there any differences between your unpartitioned and partitioned analyses? Explain.

References

Huelsenbeck JP, Ronquist F. 2001. MrBayes: Bayesian inference of phylogeny. *Bioinformatics* 17, 754-755.

Ronquist F, Huelsenbeck JP. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics* 19, 1572-1574.

Ronquist F, Teslenko M, van der Mark P, Ayres DL, Darling A, Höhna S, Larget B, Liu L, Suchard MA, Huelsenbeck JP. 2012. MrBayes 3.2: Efficient Bayesian Phylogenetic Inference and Model Choice Across a Large Model Space. *Systematic Biology* 61, 539-542.

Dr. Christopher Blair
BIO 4250
Molecular Evolution & Phylogenetics

Week 12 – Divergence time estimation

The primary goal of this lab is to use molecular data to estimate evolutionary rates and divergence times. Divergence time estimation is of significant importance to multiple fields of biology ranging from conservation to epidemiology. Unfortunately, some of the theory and methods to estimate rates and divergence times can be a bit abstract and confusing. Several different **clock models** can be used to estimate divergence times, ranging from a **strict clock** to an **unrooted model** where each branch has its own independent evolutionary rate. A strict clock model assumes that each branch on the phylogeny has the same evolutionary rate. This assumption generally holds for closely related species, but is violated when estimating divergence times among distantly related taxa. In such cases, a **relaxed clock** model (Drummond et al., 2006) may be more appropriate.

Today, we will be working with the software BEAST (Bouckaert et al., 2019), and we will be **simultaneously estimating phylogenetic relationships, evolutionary rates, and divergence times**. Note that there are multiple ways to incorporate a temporal aspect into phylogenetic analyses to separate rate from time. These include the following:

1. Utilize a previously published substitution rate for the gene and group of interest (e.g. 0.02 substitutions per site per million years for avian mtDNA). This is generally the approach when no other information is available. Bayesian methods (like BEAST) allow for uncertainty in the rate estimate by the specification of a parametric **prior distribution**.

2. Use fossil information to constrain times of ancestral nodes. Because of the uncertainty in fossil age and fossil placement on phylogenies, Bayesian methods are once again a good choice. Like the rate, uncertainty in fossil age can be incorporated into the analysis through priors. Often, fossils are used to constrain the **minimum** age of nodes. In other words, the divergence time between two lineages is at least as old as the fossil, but possibly older. A **lognormal prior distribution** is commonly used.

3. Use biogeographic information to calibrate nodes. For example, if one fish species occurs on one side of the Isthmus of Panama and its sister species occurs on the other, one can calibrate the ancestral node with the timing of the closure of the Isthmus. This method inherently assumes that divergence or speciation was directly caused by the biogeographic feature, which may often not be the case.

4. Tip dating can be used for temporal calibration if samples are serially sampled at certain time points. This is a common method to calibrate divergence times in quickly evolving organisms such as viruses.

5. Secondary calibrations. Nodes can be temporally calibrated based the results from a previous analysis that utilized some of the same taxa. This method is generally discouraged, though can be useful in cases where other techniques are not available.

In this lab we will be working with **21 primate TRIM5α sequences** (nDNA) from The Phylogenetic Handbook (Lemey et al., 2009). For simplicity we will focus on divergence time estimation using a single gene/locus. However, you should note that in this example we are inferring **gene divergence times**, which may or may not correspond to **species divergence times** (the parameter we are usually interested in). Download the NEXUS file (**primates.nexus**) and open the file in Aliview. Note that the data have already been aligned.

A. Strict Clock Analysis

It is generally a good rule of thumb to always analyze your data under a strict clock model (and possibly a simple nucleotide substitution model) first to gauge whether you get good mixing and decent effective sample size (ESS) values under the chosen model. Mixing refers to how well the chain samples the posterior distribution, whereas ESS values relate to how many independent samples are used to estimate parameters.

1. Open up BEAUti and import your alignment. You can simply click and drag the alignment file into the BEAUti window. Make sure to click the dropdown arrow and select “Import Alignment” (Fig. 1).

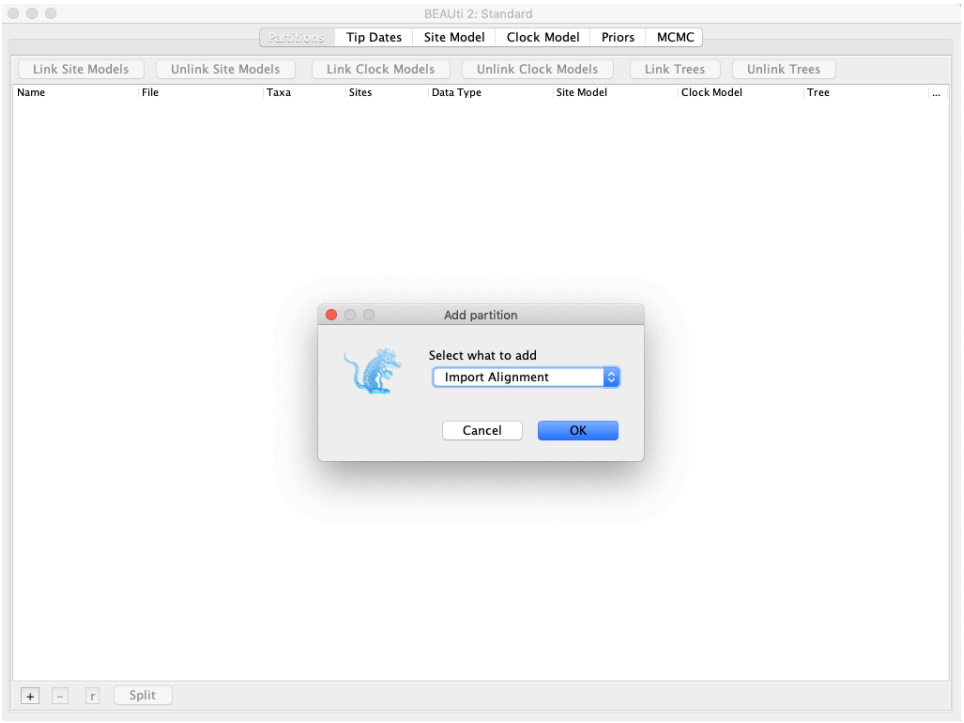

Fig. 1. Importing a multiple sequence alignment into BEAUti.

2. Click on the **Site Model** tab and specify the substitution model. For this exercise we will use the GTR+I+G model. Select GTR from the dropdown menu and make sure that

base frequencies are estimated. Enter a value of 4 for Gamma Category Count and 0.5 for Proportion Invariant. Finally, make sure the estimate box is checked for both the Shape and Proportion Invariant parameters (Fig. 2).

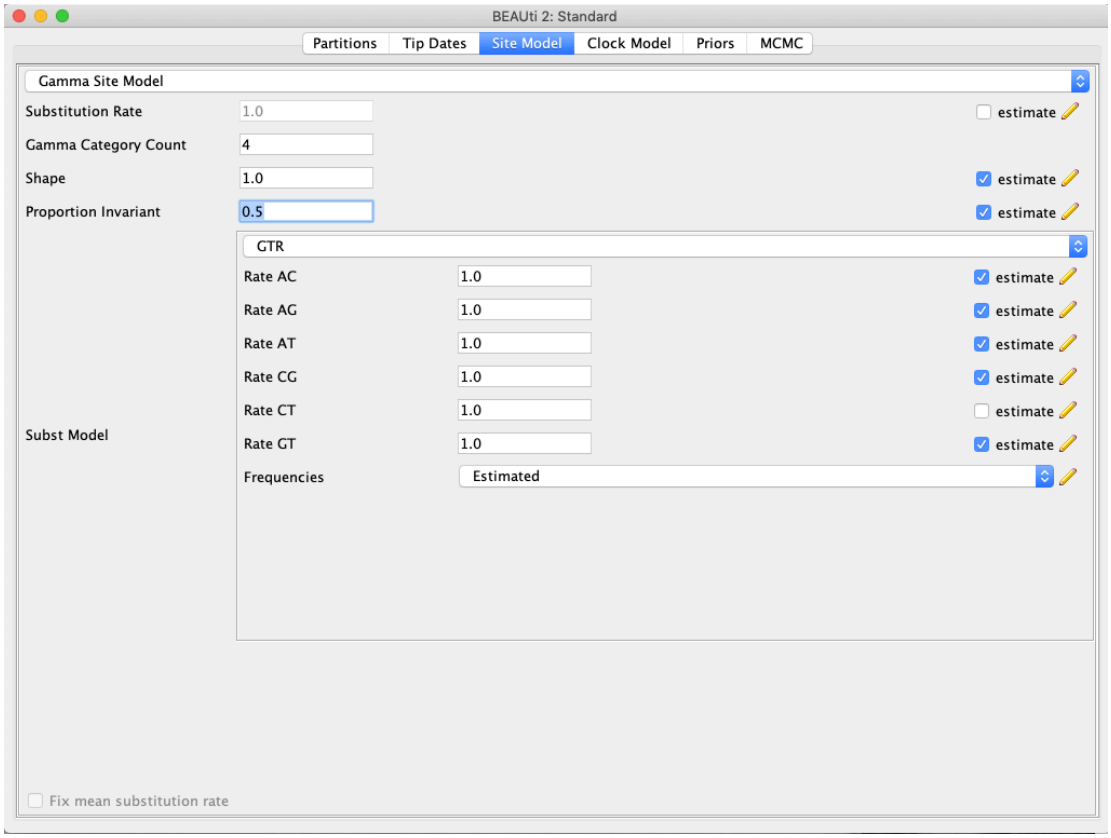

Fig. 2. Selection of a nucleotide substitution model in BEAUti. Here, we are specifying the GTR+I+G model.

3. Click the Clock Model tab and you will see that a Strict Clock model is set. We will leave this as is for our first analysis.

4. Now click on the **Priors tab** where we will setup a temporal calibration. Remember that BEAST is a Bayesian program, and thus all parameters we wish to estimate need priors. There are several published data sets that indicate that humans and chimpanzees diverged approximately 6 million years ago. We will use this information to calibrate the most recent common ancestor (MRCA) of humans and chimpanzees.

5. Click on the little "+ Add Prior" button to specify a new prior.

   Name the taxon set anything you wish, and then choose the human and chimp sequences and click the ">>" button.

6. You should now see a new prior created where we can specify our temporal information. The first thing to do is to **click the monophyletic box**, so that all sampled trees will group humans and chimps together (Fig. 3).

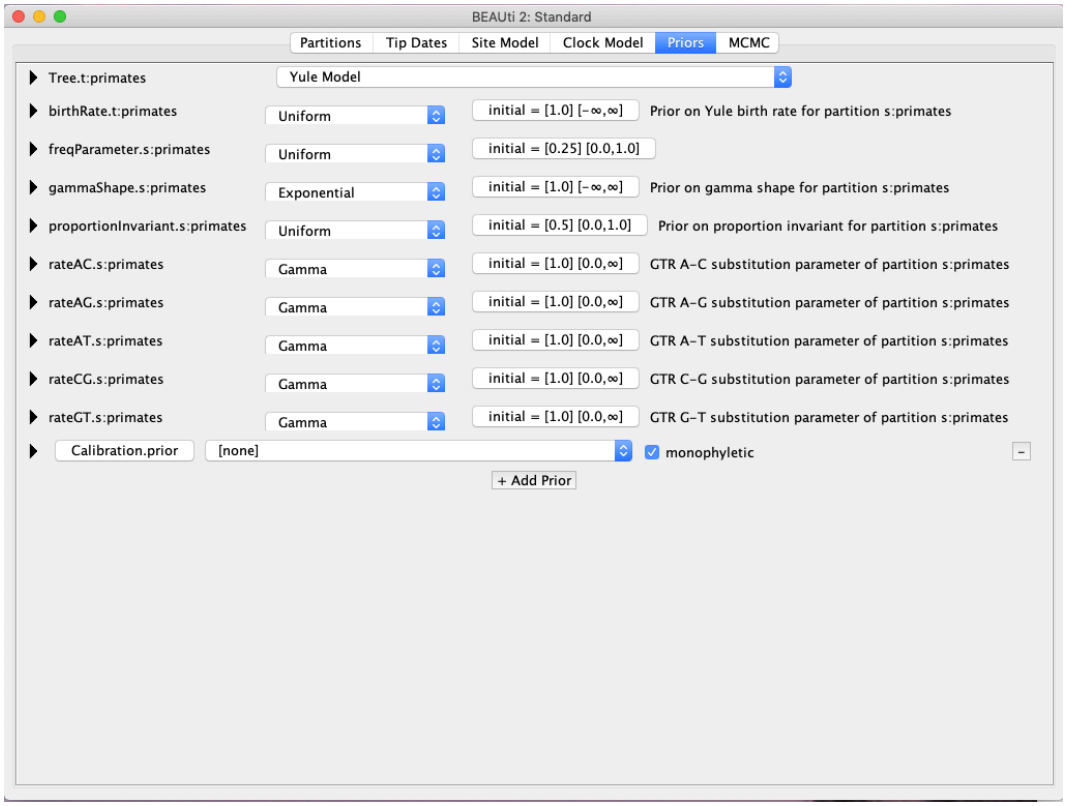

Fig. 3. Setting up priors for Bayesian divergence time estimation in BEAST.

7. For our temporal calibration, click the center box and select **Normal** so we can specify a normal distribution for the prior of divergence time between human and chimp. Now click the little black arrow next to our new prior where we can enter a mean and sigma for the normal distribution. We will use a **mean of 6 and a sigma of 1.0**, which will give us a 95% prior range between 4-8 Mya. Do not change anything else. Go ahead and close this black arrow. If you click on the **Clock Model** tab you will see that a strict clock is selected and that the estimate box is checked. Because we calibrated the MRCA of chimp and human with a known time, BEAST will be able to provide an estimated substitution rate.

8. There are a couple more priors to change before we are ready to start the analysis. First, we will **change the Tree Prior from a Yule model to a Calibrated Yule Model**. Next, **change the priors on birthrate and clockRate to a Gamma distribution with an Alpha of 0.001 and Beta of 1000 (Fig. 4)**.

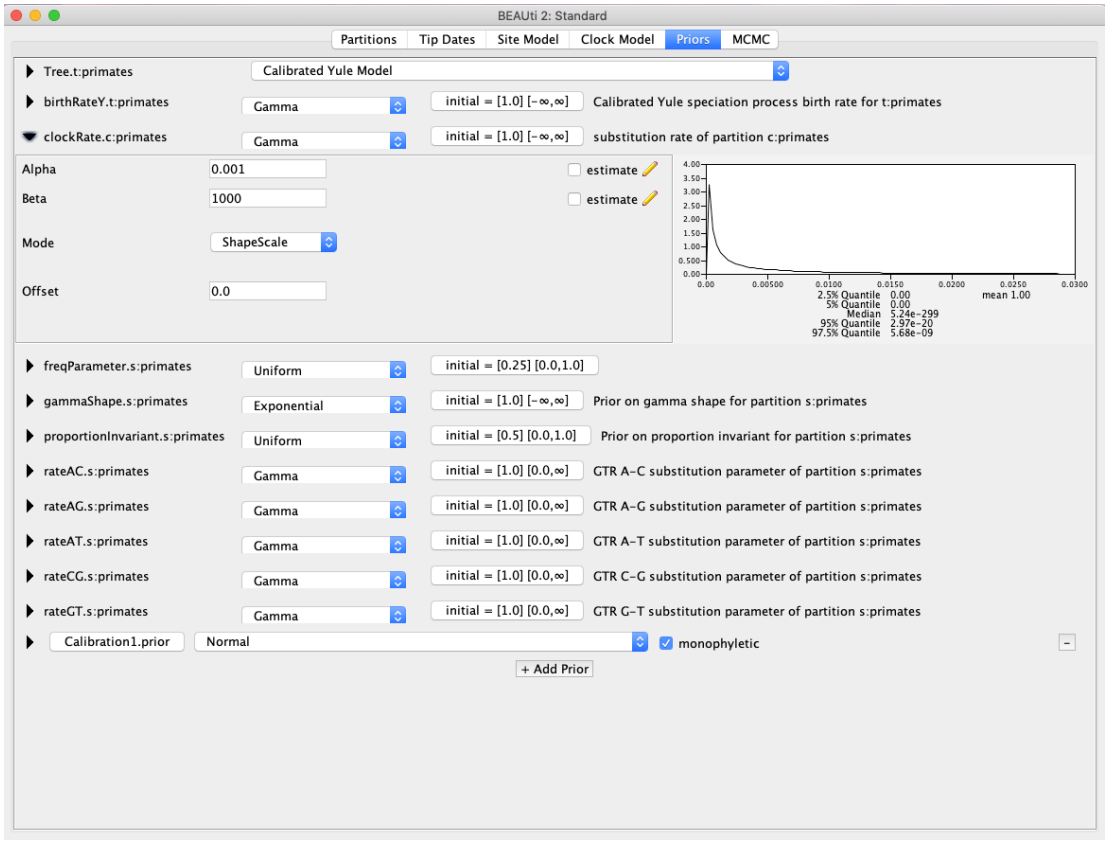

Fig. 4. Changing the prior for the clockRate parameter.

9. Finally, click the MCMC tab where we can specify the chain length and sampling frequency. For this analysis we will use the default chain length of 10 million generations and sampling frequency of 1000 generations. Click **File >Save as** and save the xml file. You may need to add the .xml extension. Open BEAST and load your xml file. The analysis should begin and finish in under 10 minutes or so. When the analysis finishes, open the **log file** in **Tracer** (Rambaut et al., 2018) to check for adequate mixing and ESS values. Your results should look something like Fig. 5. Click the 'Trace' button on the upper right and then click through each parameter. Each trace should look like a 'hairy caterpillar' if the chain was mixing well.

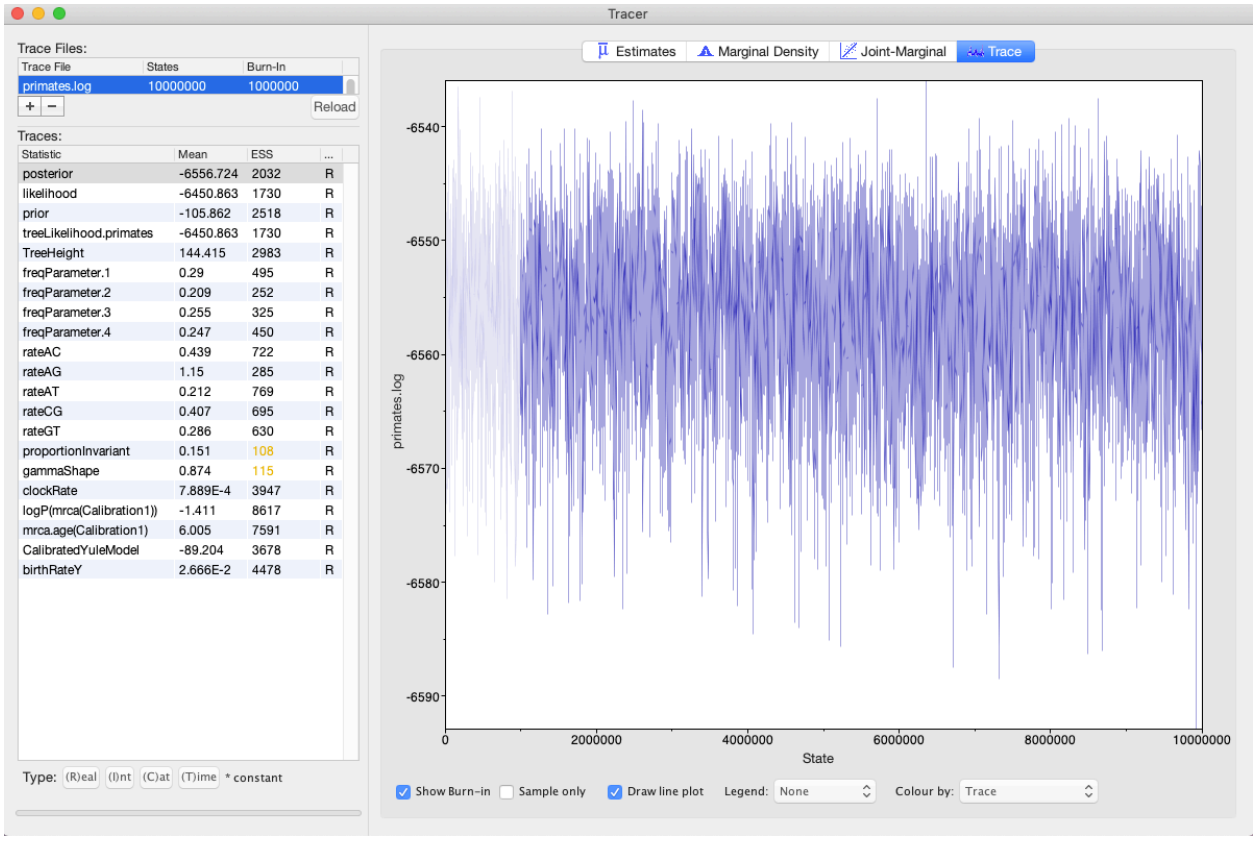

Fig. 5. Tracer results from a Bayesian phylogenetic/divergence time analysis in BEAST. Listed are all parameters estimated by the model. ESS values for two parameters (proportionInvariant, gammaShape) are flagged in yellow because they are <200. Thus, the analysis should be run for a longer duration. Users can also execute two independent BEAST runs and subsequently combine them to increase ESS values.

Q: What was the mean rate of evolution of the TRIM5α sequences? What are the units?

Q: What was the mean TreeHeight? This is the time of the original divergence at the root node of the tree. What are the units?

10. Because Tracer suggests that our analysis looks satisfactory, we can now obtain an estimate of phylogeny and associated divergence times. Navigate to your BEAST directory and open **TreeAnnotator**. Specify a **burnin percentage of 10%, maximum clade credibility tree, and mean node heights**. Select the **primates.trees file** from the analysis and name your output file.

Q: What was burnin, and why do we need to use it in Bayesian phylogenetic analysis?

11. Open up the tree in **FigTree** where we will add posterior probabilities and 95% highest posterior densities for divergence times. Click on **Branch Labels and Display posterior** to show posterior probabilities on branches. Now click on **Node Bars and Display height 95% HPD**. This will show the uncertainty in divergence times for nodes. This obviously is not informative unless we include some type of scale on the tree. Click the box next to **Scale Axis** and then **Reverse axis**. Hopefully you have something resembling Fig. 6.

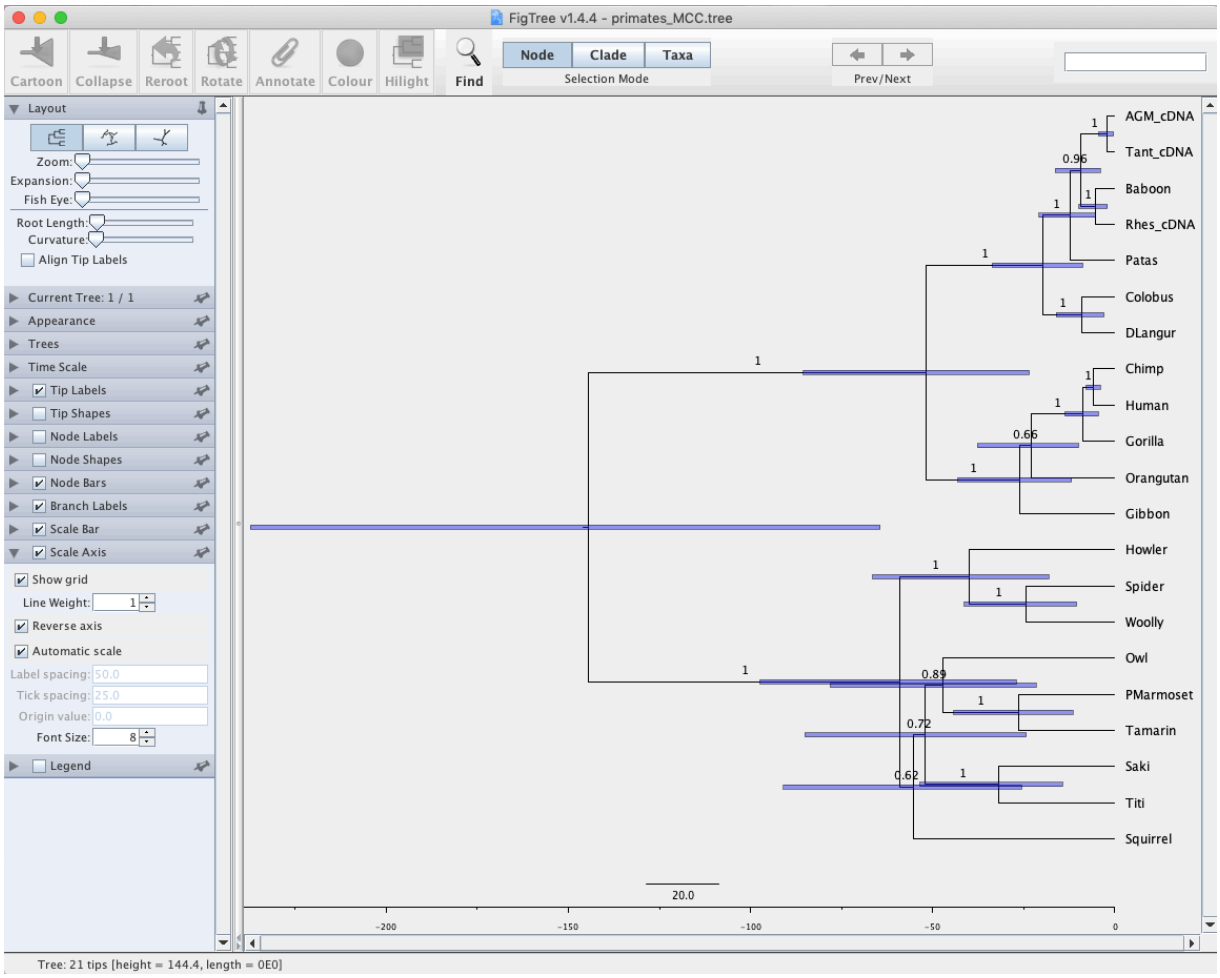

Fig. 6. Maximum clade credibility tree from a Bayesian phylogenetic analysis of primate sequences. Values at nodes represent support values (posterior probabilities). Horizontal bars illustrate 95% HPDs for divergence times.

Q: Approximately how long ago did the Howler Monkey diverge from its sister group?

Q: Is the major clade that contains human strongly supported? Explain your reasoning.

B. Relaxed Clock Analysis

In the previous analysis we used a strict clock model, which assumes that the rate of evolution is the same across all branches of the phylogeny. A strict clock model is often inappropriate when working with divergent sequences from different species. To determine if a relaxed clock model would fit the primate data better, we will run another BEAST analysis, this time specifying a **relaxed clock – lognormal model**. All other settings will be identical to our strict clock analysis.

1. Go back into BEAUti and click on the **Clock Models** tab. Change the model from **Strict Clock to Relaxed Clock Log Normal**. Make sure that the appropriate site model is still specified.

2. Under the **Priors** tab set the prior for the **ucldMean parameter** to a Gamma distribution with alpha = 0.001 and beta = 1000 (Fig. 7). The ucldMean parameter represents the mean evolutionary rate across the tree, explicitly accounting for rate variation among branches/lineages.

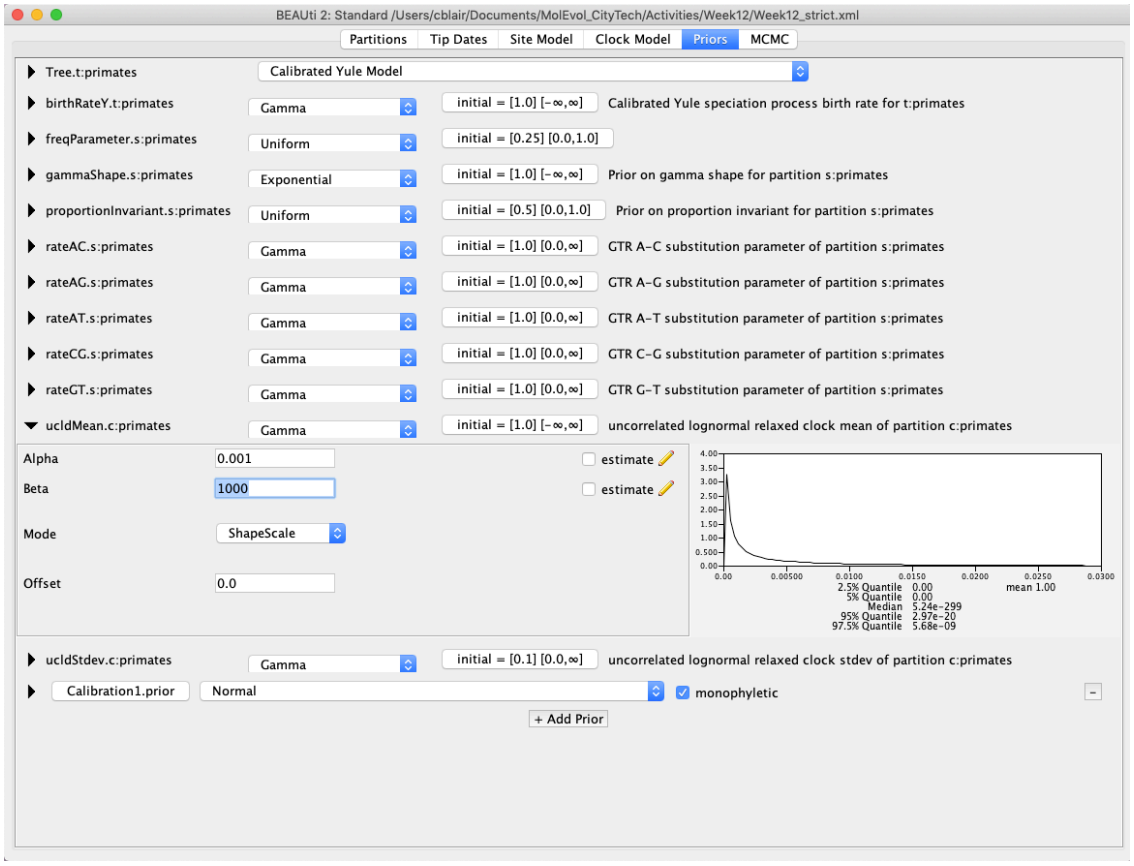

Fig. 7. Changing the prior for the ucldMean parameter in BEAUti.

3. In the MCMC tab we will once again run the chain for 10 million generations, sampling every 1000. However, change the filename of the log and tree file so that they do not overwrite your previous strict clock analysis. Perhaps something like '**primates-RelaxedClock.log**.' Save the new xml file and run BEAST again. The analysis should take about 15 min to complete.

4. When finished, open the log file in Tracer. It may be useful to import the log file for both the strict clock and relaxed clock analyses in the same window to compare parameter estimates (Fig. 8). The first thing to do is make sure all the traces look good and all ESS values are >200. If so, then we know that the analysis has been run for long enough. In this case, ESS vales for two parameters are low. Thus, the best practice would be to run the analysis for longer. However, we will work with these results for the remainder of the tutorial. Take some time to compare parameter estimates between the two analyses. Is there any major conflict that jumps out at you?

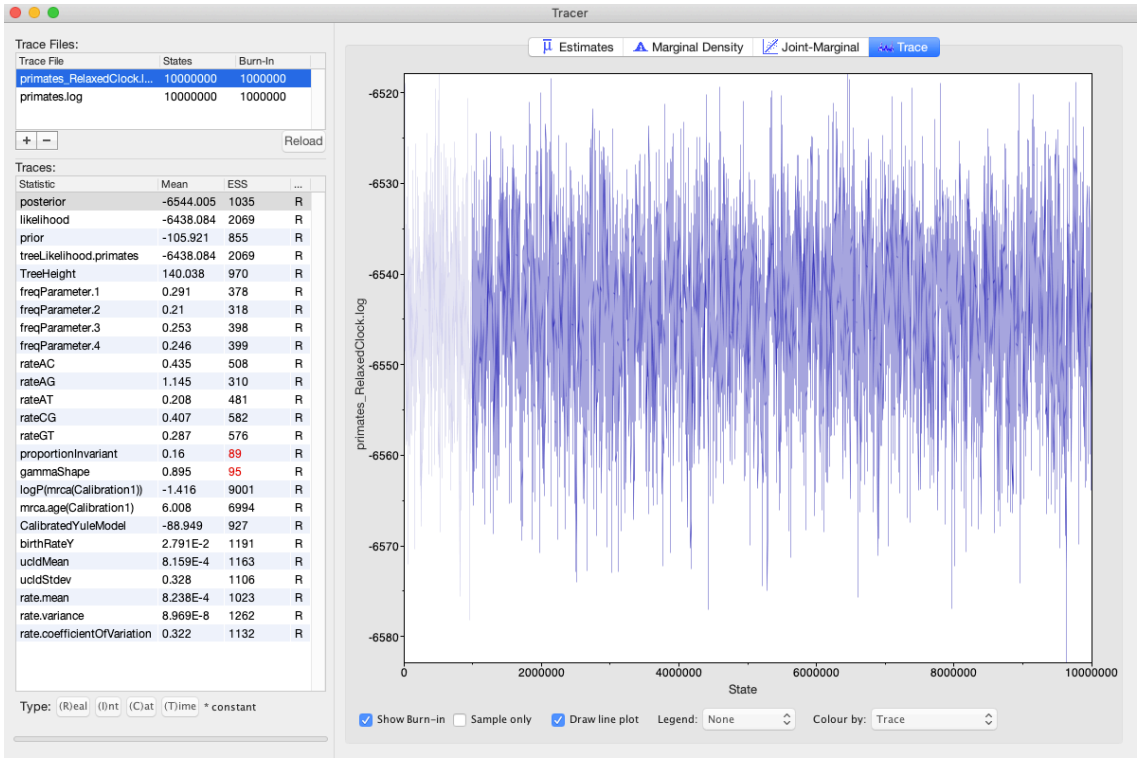

Fig. 8. Trace file for a relaxed clock analysis of the primate sequences in BEAST. Note that ESS values for two parameters are too low, indicating that the analysis should be run longer.

Q: How does the ucldMean parameter in the relaxed clock model compare to the clockRate parameter in the strict clock analysis? What does this mean?

5. Click on the ***rate.coefficientOfVariation*** parameter in the relaxed clock analysis. This tells you how clock-like the data are. Values closer to zero provide evidence for a strict clock, whereas higher values signify large branch rate heterogeneity. In general, if 0 is included in the 95% HPD for this parameter we should not reject a strict clock. In other words, we don't want to overparameterize our analysis.

   Q: What are the parameter estimates (mean and 95% HPD) for the rate.coefficientOfVariation parameter? Can we/should we reject a strict clock?

6. As before, use TreeAnnotator to create a maximum clade credibility (MCC) tree using a burnin value of 10% and mean node heights. Open up this new tree in a new FigTree window to compare to your strict clock tree.

   Q: Are the two tree topologies identical? If not, are the differences strongly supported? Remember that Bayesian posterior probability values >.95 indicate strong support.

   Q: How do the estimated divergence times compare?

   Q: What are some possible explanations for why the two trees might be so similar?

References

Bouckaert R, Vaughan TG, Barido-Sottani J, et al. 2019. BEAST 2.5: An advanced software platform for Bayesian evolutionary analysis. *PLoS Computational Biology* 15, e1006650.

Drummond AJ, Ho SYW, Phillips MJ, Rambaut A. 2006. Relaxed phylogenetics and dating with confidence. *PLoS Biology* 4, e88.

Lemey P, Salemi M, Vandamme AM. 2009. The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing, 2nd edition. Cambridge University Press, Cambridge, UK.

Rambaut A, Drummond AJ, Xie D, Baele G, Suchard MA. 2018. Posterior summarization in Bayesian phylogenetics using Tracer 1.7. *Systematic Biology* 67, 901-904.

Week 13 – Species Tree Analysis Using NGS Data

In this tutorial you will become familiar with one method used to estimate **species trees** using multilocus genomic data from next-generation sequencing (NGS) data. The simplest method to estimate a species tree from these data is the traditional concatenation approach (e.g. ML in IQ-TREE), where different gene alignments are combined into one **supermatrix**. However, concatenation assumes that individual genes share the same history, and thus, the supermatrix approach is not statistically consistent under the multispecies coalescent model. Broadly speaking, we can classify species tree methods into the following categories:

a) Fully Bayesian (or likelihood) methods. These methods estimate species trees directly from the sequence data and include **StarBEAST2** (Ogilvie et al. 2017), **BPP** (Flouri et al. 2018), **SNAPP** (Bryant et al. 2012). Most of these methods can also estimate other parameters such as population sizes and divergence times. Although these methods show relatively decent accuracy, they are extremely computationally demanding and may be of limited use with NGS data sets. However, researchers have been able to run BPP successfully with hundreds to thousands of loci, depending on the analysis.

b) Summary statistic (two-step) methods. Summary statistic methods estimate a species tree from estimated gene trees (not the sequence data). Thus, these methods will be sensitive to the accuracy of the reconstructed gene trees, which can be a factor of the data, substitution model, and gene tree reconstruction method. Many of these methods show good accuracy when there is a high degree of incomplete lineage sorting (ILS) in the data, which may be the case with large population sizes and/or short internal branches (i.e. rapid speciation). One of the most popular programs is **ASTRAL** (Zhang et al. 2018). As the accuracy of these methods depends on the accuracy of gene trees, longer loci are preferred that contain enough phylogenetically informative characters.

c) Non-Bayesian methods that estimate species trees from unlinked SNP or multilocus data. The program **SVDquartets** (Chifman & Kubatko 2014, 2015), estimates a species tree by utilizing observed site pattern frequencies in SNP or multilocus sequence data. As the program does not use Bayesian inference it is quite fast. Benefits over summary statistic methods include the estimation of a species tree directly from the data (versus gene trees). The method does assume a strict clock, but it appears to show decent accuracy even when this assumption is violated. The model is also statistically consistent under the multispecies coalescent model, so is a good choice (versus concatenation) in data sets with a large degree of ILS. SVDquartets is currently one of the best methods (non-likelihood) for estimating species trees with RADseq data.

   Link to video tutorial = https://paup.phylosolutions.com/tutorials/video-tutorials/.

d) Concatenation. Combining multiple gene alignments into one large supermatrix and estimating a species tree from this alignment. Although concatenation assumes that all genes share the same history (i.e. does not account for ILS), it does show good

accuracy in many cases. Thus, it is generally recommended to calculate species trees using both concatenation and coalescent-based approaches.

Today we will be estimating species trees using **SVDquartets implemented in PAUP*** (Swofford 2003).

Species tree analysis in SVDquartets

PAUP* can be run either through a graphical user interface (GUI) or the command line. In this tutorial I will provide instructions for both approaches. The data we will use come from a phylogenomic study on alligator lizards (Blair et al. 2021; Fig. 1). The authors collected thousands of ultraconserved element (UCE) loci (Faircloth et al. 2012) through a technique called target sequence capture. These data were then used to estimate phylogenetic relationships and divergence times. The data also supported the recognition of a new genus (*Desertum*).

Fig. 1. Pygmy Alligator Lizard (*Gerrhonotus parvus*). Image credit: Michael Price (https://commons.wikimedia.org/wiki/File:Gerrhonotus_parvus_5880611.jpg)

1. Open the data file **anguid80_SVDq.nexus** in Aliview to get a sense of the sequence data.

Q: How many sequences/individuals are in the data set?

Q: How long is the concatenated alignment? Remember that although SVDquartets takes a concatenated alignment as input, it is considered a coalescent method.

You will likely notice that the taxon names are not very informative. This data set contains multiple individuals for most species. SVDquartets requires that we specify which sequence corresponds with each species through a **taxon block**.

2. Open up the data file in any text editor and scroll to the bottom. You should see a block that contains species names on the left, and sequence ID on the right (Fig. 2). In this example we are calling our taxon partition 'species'.

```
BEGIN SETS;
    taxpartition species =
        Elgaria_kingii: MXH226Elki,
        Barisia_levicollis: MX339BleCHIH,
        Barisia_rudicollis: MXH232BruMEX,
        Mesaspis_gadovii: MXH211Mega,
        Mesaspis_viridaflava: MXH250Mevi,
        Abronia_taeniata: MXH210Abta,
        Abronia_graminea: MXH225Abgr,
        G_lugoi: MXH132GlugCOAH MXH221GlugCOAH,
        G_parvus: MXH230GparNL MXH129GparNL MXH184GparNL,
        G_sp: MXH131GlioMICH MXH213GlioCOL,
        G_rhombifer: Coloptychon_rhombifer_MX246,
        G_liocephalus: MXH212GlioOAX MXH214GlioCHIA MXH220GlioOAX,
        G_ophiurus: MXH138GophVER MXH127GophSLP MXH217GophTAM,
        G_mccoyi: MXH130GinfCOAH,
        G_infernalis: MXH126GinfNL MXH140GinfNL MXH216GinfTX Gerrhonotus_infernalis_MX134

END;
```

Fig. 2. Assigning individuals to species for species tree analysis using SVDquartets. You can also enter this information directly in PAUP*.

3. Open up PAUP* and execute the nexus file

   Command line: **exe anguid80_SVDq.nexus*;***

GUI: File > Open

The data set consists of 33 taxa and 1,904,599 characters. A useful feature of SVDquartets is that it allows you to estimate both a **lineage tree** (using individuals as terminals) and a **species tree** where individuals are allocated to species. Typing

svdq ?

into the terminal will bring up a list of options. SVDquartets estimates sets of quartet trees (species trees of four taxa) and then combines quartets to estimate the full tree. You can tell PAUP* to evaluate all quartets or a specific number (e.g. 100,000). Obviously evaluating all is more comprehensive, though it may be prohibitive on some larger data sets.

4. The first thing we should do is define the **outgroup** to root our species tree. For this analysis we will use the Madrean alligator lizard, *Elgaria kingii,* as our outgroup.

   Command line: **outgroup MXH226Elki;**

   GUI: Data > Define Outgroup and select sample MXH226Elki

5. Let's try running a species tree analysis using the following command:

   Command line: **svdq evalQuartets=all nthreads=auto taxpartition=species bootstrap=no;**

   GUI = Analysis > SVDQuartets and enter the settings shown in Fig. 3.

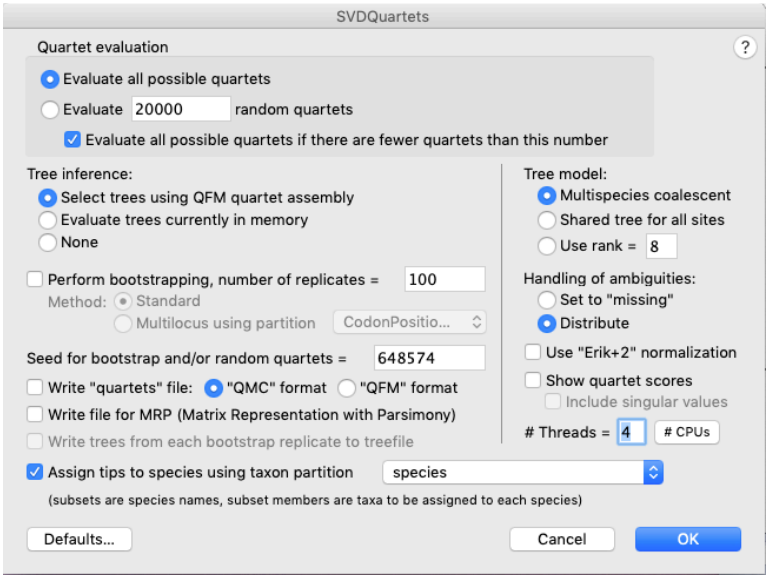

Fig. 3. Setting up a SVDquartets analysis using the graphical user interface (GUI).

When the analysis finishes you should see the SVDquartets tree (Fig. 4).

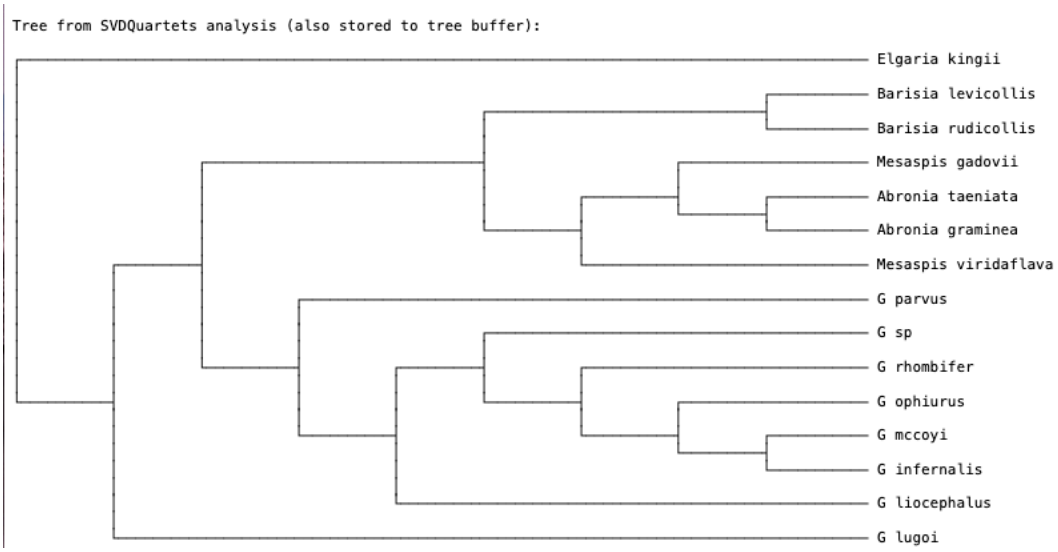

Fig. 4. SVDquartets species tree for anguid lizards.

Q: How many quartets were evaluated?

Q: Are all genera monophyletic? Explain your reasoning. Note that G = *Gerrhonotus*.

6. You can save your species tree using the following:

   Command line: **savetrees root=yes;**
   GUI: Trees > Save Trees to File

7. The **showScores** option displays the SVD score for all three unrooted trees for each quartet. Use the command below to display these scores and examine the differences among quartet trees. Remember that the topology with the lowest score is best. To assemble the species tree containing all species, PAUP* uses a quartet assembly algorithm to assemble the species tree based on the quartet trees with the lowest SVD score. See if you can find the best tree for several quartets.

   Command line: **svdq evalQuartets=all nthreads=auto taxpartition=species bootstrap=no showScores=yes;**

   GUI: Same as before, but check the box for Show quartet scores.

8. Like any type of phylogenetic analysis, we will want to get a sense of uncertainty in the relationships recovered. SVDquartets can use either standard nonparametric bootstrapping or multilocus bootstrapping. For our analysis we will perform standard bootstrapping with 100 replicates.

   Command line: **svdq evalQuartets=all nthreads=auto taxpartition=species bootstrap=yes showScores=no;**

   GUI: Uncheck the box for Show quartet scores, and check the box for Perform bootstrapping (make sure 100 is entered).

   Note that bootstrapping will take a few minutes. When the analysis finishes you should see two trees (Fig. 5). The first tree represents the SVDquartets tree based on the original data. The second tree is a bootstrap consensus tree. You can think of this as a summary tree of the 100 bootstrap replicates. Researchers have the option to either present the bootstrap consensus tree with support values, or show the SVDquartets tree with bootstrap support mapped on. Note that in many cases the SVDquartets tree is identical to the bootstrap consensus tree.

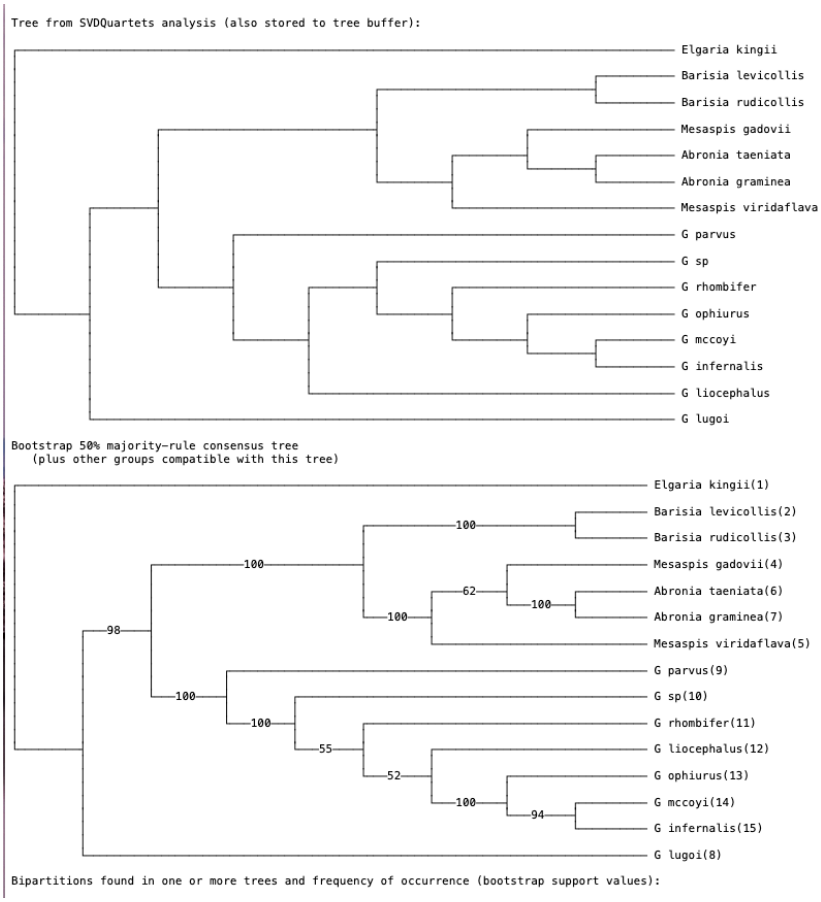

Fig. 5. SVDquartets species tree and bootstrap consensus tree inferred for anguid lizards. Values on branches represent support. In general, bootstrap values >70 indicate moderate-strong support for relationships.

Q: Describe how bootstrapping works in phylogenetic analyses. In our analysis we are using 100 replicates. What does this mean?

Q: Is the SVDquartets tree identical to the bootstrap consensus tree? If not, explain where the differences are.

Q: Does the analysis suggest that a new genus should be erected for *G. lugoi*? Why or why not?

The GUI version of PAUP* has a cool feature that lets you view and print pdfs of your trees.

Click **Trees > Print/View SVDQuartets Trees**

Here, you can make edits and export the results for viewing in other programs.

9. The previous analysis used a taxpartition block ("species") to allocate taxa (individuals) to species for estimation of a species tree. Another cool feature of SVDquartets is that it allows you to also estimate a 'lineage tree' that is statistically consistent with the multispecies coalescent model. This analysis estimates the phylogenetic relationships of all individual taxa in your matrix. Let's perform an exhaustive search of all quartets followed by a bootstrap analysis with 100 replicates.

   Command line: **svdq evalQuartets=all nthreads=auto taxpartition=none bootstrap=yes showScores=no;**

   GUI: Make sure to uncheck the box for assigning individuals to species.

This analysis will take a bit longer, but should take no more than about 15 minutes. Are bootstrap values high? One benefit of performing a 'lineage-based' analysis is that it allows you to test for species monophyly. Note that in many real world studies it can be difficult/challenging to assign individuals to species. Your new SVDquartets tree should resemble Fig. 6.

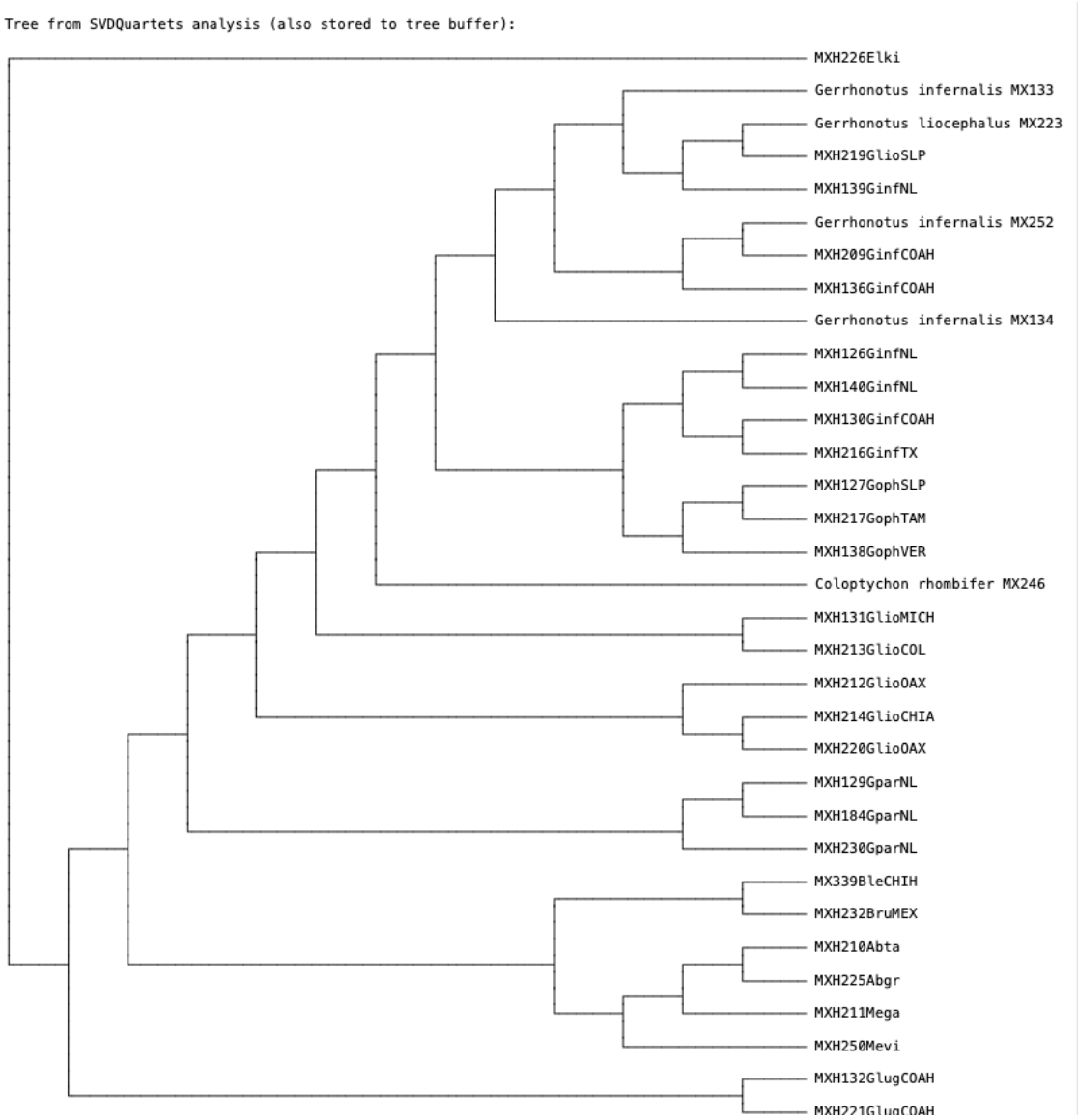

Fig. 6. Lineage-based SVDquartets tree of anguid lizards. Note that in this analysis we are not assigning individual sequences to species. This type of analysis is often helpful for getting a sense of genetic differences between individuals and putative species.

References

Blair C, Bryson RW, García-Vázquez UO, Nieto-Montes de Oca A, Lazcano D, McCormack JE, Klicka J. 2021. Phylogenomics of alligator lizards elucidate diversification patterns across the Mexican Transition Zone and support the recognition of a new genus. *Biological Journal of the Linnean Society,* blab139.

Bryant D, Bouckaert R, Felsenstein J, Rosenberg NA, RoyChoudhury A. 2012. Inferring species trees directly from biallelic genetic markers: bypassing gene trees in a full coalescent analysis. *Molecular Biology and Evolution* 29, 1917-1932.

Chifman J, Kubatko L. 2014. Quartet inference from SNP data under the coalescent. *Bioinformatics* 30, 3317-3324.

Chifman J, Kubatko L. 2015. Identifiability of the unrooted species tree topology under the coalescent model with time-reversible substitution processes, site-specific rate variation, and invariable sites. *Journal of Theoretical Biology* 374, 35–47.

Faiircloth BC, McCormack JE, Crawford NG, Harvey MG, Brumfield RT, Glenn TC. 2012. Ultraconserved elements anchor thousands of genetic markers spanning multiple evolutionary timescales. *Systematic Biology* 61, 717-726.

Flouri T, Jiao X, Rannala B, Yang Z. 2018. Species tree inference with BPP using genomic sequences and the multispecies coalescent. *Molecular Biology and Evolution* 35, 2585-2593.

Ogilvie HA, Bouckaert RR, Drummond AJ. 2017. StarBEAST2 brings faster species tree inference and accurate estimates of substitution rates. *Molecular Biology and Evolution* 34, 2101-2114.

Swofford DL. 2003. PAUP*: phylogenetic analysis using parsimony (*and other methods). Version 4. Sinauer Associates, Sunderland, MA.

Zhang C, Rabiee M, Sayyari E, Mirarab S. 2018. ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees. *BMC Bioinformatics*, 19, 153.