

Phylogenetics & Bioinformatics

Colby College BI164

18 January 2024

I. Introduction to Phylogenetics

Objectives

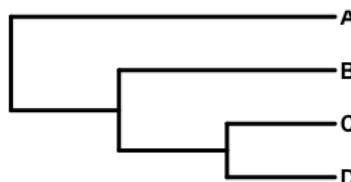
1. Interpret phylogenetic trees based on morphological and molecular characteristics.
2. Connect the information in phylogenetic trees to the evolutionary and natural history of organisms.

Introduction

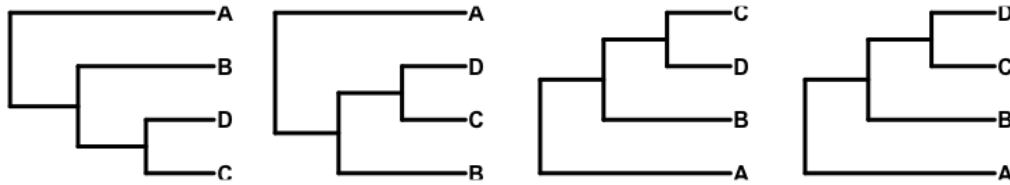
Biological **systematics** is the study of the relationships among living things. Systematists often seek to produce a classification or **taxonomy**, which reflects the evolutionary history of organisms. Taxonomic relationships are often visualized as phylogenetic trees in which organisms are grouped together based on relatedness to their common ancestors.

Understanding a Tree

Let's consider the simple tree to the right, where the letters at the tips represent different species.



- Time in a phylogeny is represented as moving from the base to the tips of the tree (left to right in this example). This tree tells us that the first event was the evolutionary split into two different lineages, one leading to species A and one to the other branches of the tree. The next event was the splitting of B from the line that leads to C and D. The final and most recent event is the splitting of C and D into two separate species.
- You can identify three different monophyletic groups (ABCD, BCD, and CD). A monophyletic group is a clade that consists of a species and all its descendants from a common ancestor. (So, as a counter-example, species A and B are *not* a monophyletic group.)
- Species B is more closely related Species C and D than to Species A. Even though the number of ancestral nodes separating B and the other species is the same (one), you should always look forward in time in the tree to identify closest relatives.
- Species B is no more closely related to C than it is to D. All nodes are able to be rotated 180° without fundamentally changing the tree, because the order of branching is preserved. All of the trees below show the *same* relationships as the figure above.



What do you need to develop a phylogeny?

Characters and character states: To build a phylogeny a researcher must identify traits or characters that are relevant for distinguishing taxa within the group. Characters can include morphological, life history or molecular information. The more characters that are used, the more accurate the tree may be. Once the characters have been identified, each organism is scored or graded on which character state it possesses. A character state is the value of a character. For example if the number of cervical vertebrae is a character in the analysis, then the character state of different taxa might be 7 (for humans) or 14 (for chickens). If the dataset includes nucleotide sequences, then each position in the sequence is a character and the character state for a taxon can be one of the four nucleotides (A, C, G or T).

Outgroups: We cannot know the real character states for ancestral species at the base of our tree. Instead, we must choose an existing species that can stand in for an ancestor. We use these taxa as the outgroup. - The outgroup should represent a lineage that diverges before the group under consideration (the "ingroup"). - Therefore, the outgroup species cannot be a member of the group for which you are creating a tree. The outgroup must be *out of the group*! - With that said, the outgroup should have as many similarities as possible while not being considered a member of the ingroup. Importantly the outgroup must have all the characters being used to create the tree. So, if you were building a phylogeny of fish based on their number of cervical vertebrae, you could not use invertebrates as your outgroup, because they do not have vertebrae! There would be no meaningful value for their character state.

Shared Derived Characters: Not all characters are equally helpful in creating a phylogeny. A character state that is shared by two or more taxa may suggest that those organisms are closely related. From there, you have to distinguish between two types of character states: ancestral and derived. - Ancestral (or "primitive") character states are those that have been inherited and not significantly changed from a distant ancestor. - Derived character states are those that have undergone evolutionary change to a different value within the group. - Characters for which some taxa share derived character states are the most helpful in creating a tree, because it suggests that those taxa share a recent common ancestor since the base of the tree. In contrast shared ancestral character states don't distinguish organisms across evolutionary time. So they are not particularly useful in creating a tree.

Parsimony: A tree should be constructed so that it reflects the most parsimonious (simplest and most straight-forward) evolutionary history. The total number of evolutionary changes is often referred to as the tree length. A tree that reflects a single evolutionary change for a trait is more parsimonious than one with an alternate arrangement that reflects multiple evolutionary changes. Why do you think we can often assume that a more parsimonious evolutionary history is likely to have occurred?

Convergence and Reversal: Sometimes, the same character state can evolve separately in distantly related organisms. In such cases, we say the character state shows evolutionary convergence. Additionally, a derived character state can revert back to the ancestral condition. When this happens it is known as an evolutionary reversal. Either of these situations will cause a character to change more than once.

Molecular and morphological characters

The characters or traits used to generate a phylogeny can come from multiple sources, including morphological characters and biomolecular sequences, as well as information on ecology, behavior or life cycles of organisms. In the early days of systematics, taxonomy was based solely on morphological traits. However choosing characters based on morphology can be subjective. Molecular sequence data is often less prone to bias, and it has become the dominant basis for systematics in recent decades. However it's important to remember that molecular sequences can be influenced by convergent evolution. In the best circumstances, a combination of molecular and morphological data are used to produce the most accurate evolutionary trees.






II. Introduction to Bioinformatics

Bioinformatics is a field of science that develops and applies methods to analyze and interpret biological data, especially when the data sets are large and complex. Bioinformatics and computational biology have become critical in all areas of biology. Therefore developing familiarity with common data types and computational tools of bioinformatics is essential to a modern education in biology.

Case Study: The Origin of Tetrapods

The superclass Tetrapoda comprises the four-limbed vertebrates, including amphibians, reptiles, birds, and mammals. The earliest tetrapods evolved approximately 390 million years ago from an ancient group of fish. However many details of their adaptation to land remain unclear, and this subject is an active area of research among paleontologists.

Today you will explore which type of fish may have given rise to the first tetrapods. Starting with those that have a true jaw, fish are either classified as having cartilaginous (Chondrichthyes) or bony (Osteichthyes) skeletons. Within Osteichthyes, fish are further classified based on the structure of their fins, either ray-finned (Actinopterygii) or lobe-finned (Sarcopterygii). Using living examples of each of these four types of fish and two tetrapods, you will create a phylogeny using amino acid sequences for a mitochondrial protein called cytochrome c oxidase subunit II (abbreviated as COX2 or COII, pronounced "C O 2").

superclass	class	example species	common name	image
	Chondrichthyes	<i>Squalus acanthias</i>	spiny dogfish	
Osteichthyes	Sarcopterygii	<i>Latimeria chalumnae</i>	coelacanth	
Osteichthyes	Sarcopterygii	<i>Protopterus dolloi</i>	lungfish	
Osteichthyes	Actinopterygii	<i>Danio rerio</i>	zebrafish	
Tetrapoda	Amphibia	<i>Bombina variegata</i>	yellow-bellied toad	

Methods

First you need to obtain the COII amino acid sequences from the fish and tetrapods. As a starting point, these sequences have been provided for you as an object accessible from the R package `sativum`. you can preview these sequences by entering `fish.COII` in the R console. You should see the output shown below.

```
AAStringSet object of length 5:
      width seq                                     names
[1]   230 MAHPSQLGFQDAASPVMEELLHF...PIVVEAVPLEHFESWSSLMLEEA shark NP_008526.1...
[2]   230 MAHPSQLGLQDAASPVMEELLHF...PIVLEAIPLDPFEDWSSSMLEEA coelacanth NP_008...
[3]   230 MAHPSQLGLQDAASPVMEELIHF...PIVVEAAPLQHFENWSSLMLEKA lungfish NP_00826...
[4]   230 MAHPAQLGFQDAASPVMEELLCF...PIVVEAVPLEFFENWSSAMLEDA zebrafish NP_0593...
[5]   229 MAHPAQLGFQDAASPIMEELLHF...MPIVVEAVPLKTFENWSSSMLET toad YP_001122775...
```

Your console output may be color-coded to highlight specific amino acids.

In addition to these COII sequences, you will add the amino acid sequence for an additional amphibian, the African clawed frog (*Xenopus laevis*).

The National Center for Biotechnology Information (NCBI (<https://www.ncbi.nlm.nih.gov>)) hosts a series of databases important to bioinformatics, including [GenBank](https://www.ncbi.nlm.nih.gov/genbank/) (<https://www.ncbi.nlm.nih.gov/genbank/>), one of the oldest, most extensive, and most well-annotated databases of biomolecular sequences. Follow the directions below to download protein sequence data for *Xenopus laevis* COII.

1. **Go to NCBI** at <http://www.ncbi.nlm.nih.gov> (<http://www.ncbi.nlm.nih.gov>).
2. **Search** “All Databases” for *Xenopus laevis*. Since we are in search of COII, which is a mitochondrial protein, find the “Proteins” panel and click on “Protein”.

This search returns more than 200,000 results! So you will need to narrow the search.

3. **Refine the search** text to read *Xenopus laevis* [orgn] cytochrome c oxidase ii. It's often safer to perform searches using the full name of a gene or protein as opposed to an abbreviation. This text specifies that the search for COII should be limited to the organism *Xenopus laevis*.
4. The results should now include 10 hits, a manageable number. **Choose one** sequence whose title includes “cytochrome c oxidase subunit II” (or “subunit 2”) but not the word “partial”. (We want to the complete protein sequence.) Make your selection by checking the box to the left of the entry. In the “Summary” drop-down menu, select “FASTA (text)”. This will bring you to the amino acid sequence of the *Xenopus laevis* cytochrome c oxidase subunit II protein in FASTA format.

```
>sp|P00407.1|COX2_XENLA RecName: Full=Cytochrome c oxidase subunit 2; AltName: Full=Cyto
MAHPSQLGFQDAASPIMEELLHFHDHLMVFLISTLVLYIITIMTTKLTNTNLMDAQEIEMVWTIMPA
ISLIMIALPSLRILYLMDVNDPHLTIKAIGHQWYWSYEYTNIEDLSFDSYMIPTNDLTPGQFRLLVDN
RMVVPMEPTRLLVTAEDVLHSAVPSLGVKTDIAIPGRLHQTSFIATRPGVFYGCSEICGANHSFMPIV
VEAVPLTDFENWSSSMLEA
```

The [FASTA format](https://en.wikipedia.org/wiki/FASTA_format) (https://en.wikipedia.org/wiki/FASTA_format) is the most common way DNA and protein sequences are stored and used by bioinformatics software. The first line contains information about the sequence (the metadata), including the NCBI accession number (“P00407.1”), sequence name (“COX2_XENLA”) and description (“RecName: Full=Cytochrome c oxidase...”). The following lines represent the amino acid sequence of the protein using the one-letter [amino acid abbreviations](https://en.wikipedia.org/wiki/Proteinogenic_amino_acid#General_chemical_properties) (https://en.wikipedia.org/wiki/Proteinogenic_amino_acid#General_chemical_properties). This is the sequence that comprises the COII protein in this species.

5. **Download the new sequence** for *Xenopus laevis* COII. There are several ways to do this, but the R package `sativum` provides convenient tools. We will need the sequence's GenBank ID or "GI" number. (Sorry, this is different from the accession number!)

- Option 1. To get the GI number, search the NCBI protein database using the accession number with the function `search.for.ncbi.ids`, following the example sequence below. (If you chose a different *X. laevis* COII sequence, then simply use that sequence's accession number.) That function will return a GI number, "117049" in this example. Then use that GI number in the function `fetch.sequence`

```
frog.id <- search.for.ncbi.ids("P00407.1", database = "protein")
frog.sequence <- fetch.sequence(ids = frog.id, database = "protein", format =
  "AA")
AAStringSet object of length 1:
      width seq                                     names
[1]    229 MAHPSQLGFQDAASPIMEELLHF...MPIVVEAVPLTDFENWSSSMLEA Xenopus_laevis_P0...
```

- Option 2. If you still have the NCBI website up, go back to the page with your search results. Notice that below the description of each sequence, in smaller print, are two numbers labeled "Accession" and "GI". The first is the NCBI accession number and the second is the GenBank GI number we need. You can copy the GI number and paste it into your R command.

```
frog.sequence <- fetch.sequence(ids = "117049", database = "protein", format =
  "AA")
AAStringSet object of length 1:
      width seq                                     names
[1]    229 MAHPSQLGFQDAASPIMEELLHF...MPIVVEAVPLTDFENWSSSMLEA Xenopus_laevis_P0...
```

Either way the result is the same.

Optional: If you're not comfortable with scientific names, like *Xenopus laevis* or if you simply want this sequence's name to match the format of the others, you can change it as shown below.

```
names(frog.sequence) <- "frog P00407.1 COII [Xenopus laevis]"
```

6. **Combine** the old and new COII sequences into one dataset using the `combine.sequences` function.

```
combined.coii <- combine.sequences(fish.COII, frog.sequence)
Combined 5 and 1 sequences: 6
```

7. **Save** this dataset to a file in your R work space.

```
write.fasta(combined.coii, filename = "combined.coii.fasta")
Wrote 6 sequences to file: combined.coii.fasta
```

```
AAStringSet object of length 6:
      width seq                                     names
[1]    230 MAHPSQLGFQDAASPVMEELLHF...PIVVEAVPLEHFESWSSLMLEEA shark NP_008526.1...
[2]    230 MAHPSQLGLQDAASPVMEELLHF...PIVLEAIPLDPFEDWSSSMLEEA coelacanth NP_008...
[3]    230 MAHPSQLGLQDAASPVMEELIHF...PIVVEAAPLQHFENWSSLMLEKA lungfish NP_00826...
[4]    230 MAHPAQLGFQDAASPVMEELLCF...PIVVEAVPLEFFENWSSAMLEDA zebrafish NP_0593...
[5]    229 MAHPAQLGFQDAASPIMEELLHF...MPIVVEAVPLKTFENWSSSMLET toad YP_001122775...
[6]    229 MAHPSQLGFQDAASPIMEELLHF...MPIVVEAVPLTDFENWSSSMLEA frog P00407.1 COI...
```

8. **Align the sequences.** If you run `combined.coii` notice that the amino acids line up pretty well, but not perfectly. The amphibian sequences are shifted on the right. This is a common problem in phylogenetics. Solving it requires aligning the sequences. While you can do this manually, it's more reproducible (and faster!) to use software. Several algorithms have been developed for alignment of multiple sequences of amino acids or nucleotides. We will use a common algorithm for multiple

sequence alignment (MSA) called ClustalOmega (<http://www.clustal.org/omega/>). It can be implemented in R using the `align.sequences` function.

```
coii.alignment <- align.sequences(combined.coii, method = "ClustalOmega", order =
  "input")
using Gonnet
Aligned 6 sequences using method ClustalOmega
print(coii.alignment, show="complete")
```

MsaAMultipleAlignment with 6 rows and 230 columns

```
aln (1..54) names
[1] MAHPSQLGFQDAASPVMEEELLHFHDHLMIVFLISTLVLYIIMAMVSTKLTNKY shark NP_008526.1...
[2] MAHPSQLGLQDAASPVMEEELLHFHDHLMIVFLISTLVFYIILAMTTKMTDKY coelacanth NP_008...
[3] MAHPSQLGLQDAASPVMEEELLHFHDHLMIVFLISTLVLYIIVAMVSTKFTNKF lungfish NP_00826...
[4] MAHPAQLGFQDAASPVMEEELLCFHDHLMIVFLISTLVLYIIAMVSTKLTNKF zebrafish NP_0593...
[5] MAHPAQLGFQDAASPIIMEELLHFHDHLMIVFLISTLVLYIITMTTKLTNTN toad YP_001122775...
[6] MAHPSQLGFQDAASPIIMEELLHFHDHLMIVFLISTLVLYIITIMTTKLTNTN frog P00407.1 COI...
Con MAHPSQLGFQDAASPVMEEELLHFHDHLMIVFLISTLVLYII?AM??TKLTNK? Consensus
```

```
aln (55..108) names
[1] ILDSQEIEIWTILPAVILIMIALPSLRILYLMDEINDPHLTIKAMGHQWYWSY shark NP_008526.1...
[2] ILDAQEIEIWTLLPAIVLILVALPSLRILYLIIDEVENPHLTIKAMGHQWYWSY coelacanth NP_008...
[3] ILDSQEIEIWTILPAVILIMIALPSLRILYLMDEINDPHLTVKAVGHQWYWSY lungfish NP_00826...
[4] ILDSQEIEIWTVLPAILILIALPSLRILYLMDEINDPHVTIKAVGHQWYWSY zebrafish NP_0593...
[5] AMDAQEIEIEMVTIMPAIILIVIALPSLRILYLMDEISDPHLTVKAIGHQWYWSY toad YP_001122775...
[6] LMDAQEIEIEMVTIMPAISLIMIALPSLRILYLMDEVNDPHLTIKAIGHQWYWSY frog P00407.1 COI...
Con ILD?QEIEIWTILPAIILIMIALPSLRILYLMDEINDPHLTIKA?GHQWYWSY Consensus
```

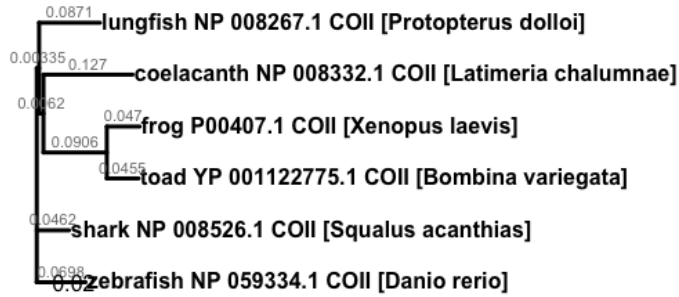
```
aln (109..162) names
[1] EYTDYEDLGFDSYMIOQTQDLTPGQFRLLETDRMVVPMESPIRVLSAEDVLHS shark NP_008526.1...
[2] EYTDYEELSFDSYMTPLQDLNPGQFRLLETDRMVVPMESLIRVLISAEDVLHS coelacanth NP_008...
[3] EYSDYETLNFDSYMTPTQDLTPGQFRLLETDRMVVPMESPIRVLTADDVIHS lungfish NP_00826...
[4] EYTDYENLEFDSYMVPTQDLTPGGFRLLETDRMVVPMESPIRILVSAEDVLHS zebrafish NP_0593...
[5] EFTNYEDLAFDSYMIPTQDLSPGQFRLLEVDRMVVPMESPTRMLITAEDVLHS toad YP_001122775...
[6] EYTDYEDLSFDSYMIPTNDLTPGQFRLLEVDRMVVPMESPTRLLVTAEDVLHS frog P00407.1 COI...
Con EYTDYEDL?FDSYMIPTQDLTPGQFRLLETDRMVVPMESPIRVL??AEDVLHS Consensus
```

```
aln (163..216) names
[1] WTPVPAVGKMDAVPGRNLQTAFIISRPGVYGGQCSEICGANHSFMPIVVEAVPL shark NP_008526.1...
[2] WAVPALGVKMDAVPGRNLQITFMISRPGLYGGQCSEICGANHSFMPIVLEAIPL coelacanth NP_008...
[3] WAVPALGIKMDAVPGRNLQASFITARPGMFYGGQCSEIWGANHSFMPIVVEAAPL lungfish NP_00826...
[4] WAVPSLGIKMDAVPGRNLQTAFIGVSRPGVFYGGQCSEICGANHSFMPIVVEAVPL zebrafish NP_0593...
[5] WAVPALGIKTDIAIPGRNLQTSFIATRPGVFYGGQCSEICGANHSFMPIVVEAVPL toad YP_001122775...
[6] WAVPSLGVKTDIAIPGRNLQTSFIATRPGVFYGGQCSEICGANHSFMPIVVEAVPL frog P00407.1 COI...
Con WAVPALG?KMDAVPGRNLQTSFI?SRPGVFYGGQCSEICGANHSFMPIVVEAVPL Consensus
```

```
aln (217..230) names
[1] EHFESWSSLMLEEA shark NP_008526.1...
[2] DPFEDWSSSMLEEA coelacanth NP_008...
[3] QHFENWSSLMLEKA lungfish NP_00826...
[4] EFFENWSSAMLEDA zebrafish NP_0593...
[5] KTFENWSSSMLET- toad YP_001122775...
[6] TDFENWSSSMLEA- frog P00407.1 COI...
Con ??FENWSSSMLE?A Consensus
```

9. **Infer a phylogeny.** There are many methods to infer phylogenetic relationships from aligned biological sequence data. The neighbor-joining (NJ) method of creating a tree is based on the similarities and differences (genetic distance) that existed between the sequences in the alignment. This method is implemented in R using the `infer.phylogeny` function.

```
coii.tree <- infer.phylogeny(coii.alignment)
Inferred phylogeny for 6 aligned sequences using NJ with model Blosum62
```



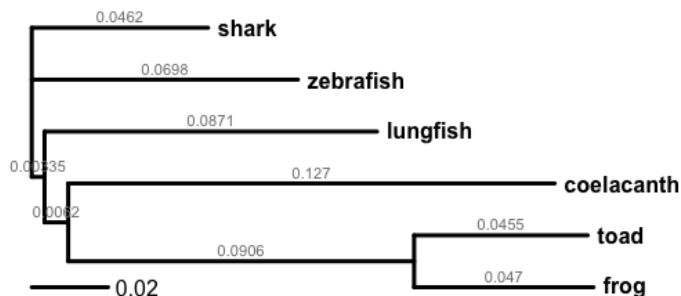
Genetic distance is a measure of percent change in the sequence data from the most recent ancestral node to the taxa at the tips of the tree. Therefore it can range from 0 to 1. Genetic distance is used in the NJ method to calculate the lengths of the tree branches.

10. Let's do some things to clean up this tree and make it easier to interpret. First, let's **make tip labels simpler**. We don't need the NCBI accession numbers anymore, and we know these are all COII sequences. So let's just use the short common names for each taxon. Let's check the current names, in `coii.tree$tip.label`, and then reassign them. (Just be sure to maintain the same order.)

```
coii.tree$tip.label
[1] "shark NP_008526.1 COII [Squalus acanthias]"
[2] "coelacanth NP_008332.1 COII [Latimeria chalumnae]"
[3] "lungfish NP_008267.1 COII [Protopterus dolloi]"
[4] "zebrafish NP_059334.1 COII [Danio rerio]"
[5] "toad YP_001122775.1 COII [Bombina variegata]"
[6] "frog P00407.1 COII [Xenopus laevis]"
coii.tree$tip.label <- c("shark","coelacanth","lungfish","zebrafish","toad","frog")
```

11. Next, notice that the shark, which is ancestral to the bony fish, appears in the middle of the tree. It's our outgroup, so let's **rotate nodes** in the tree to place the outgroup at the top, using `rotateConstr`. (Some people prefer the outgroup to be at the bottom. This is a subjective preference, but it's helpful to put the outgroup to one extreme or the other.) This rotation won't change any relationships shown in the tree, but it should make it easier to interpret relationships.

```
coii.tree <- rotateConstr(
  coii.tree,
  constraint = c("frog","toad","coelacanth","lungfish","zebrafish","shark")
)
draw.tree(coii.tree)
```

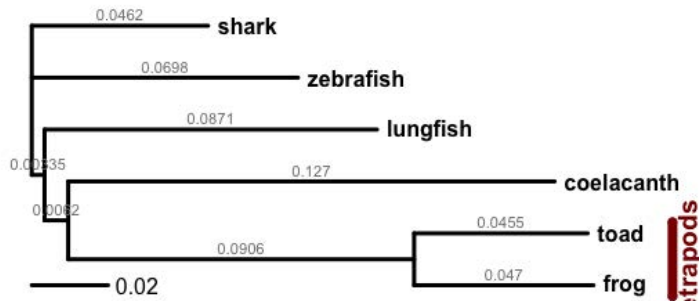


12. Finally, we can label the clade that contains our tetrapod species, the frog and toad, using the `label.clade` function.

```

draw.tree(coii.tree)
label.clade(
  tree = coii.tree,
  clade = c("frog","toad"),
  text = "tetrapods",
  offset = 2
)

```



Questions

Use this tree to **answer the questions below** regarding relationships among fish and tetrapods. Check your conclusions with your instructor.

- According to this tree, which is the group of fish most closely related to the tetrapods?
- Explain what the genetic distance for the Lungfish represents.
- Based on this tree, are the bony fish a monophyletic group? Explain.