Proposal

The overall goal of this project has two parts. First, I want to create a pipeline that takes in fasta data and automatically runs through a bunch of scripts to have an overall output of an alignment and a tree. The second part of the project will focus on displaying the data in terminal in a simple way. Once the user is able to see the tree, I want them to be able to run a simple script that takes the tip label they want to select, and shows any of the data generated from previous blasts/alignment, as well as shows hits from the tip sequence against the NCBI database. This would help simplify finding outgroups within trees so we don't have to go find sequences and blast them individually ourselves. If I am able to solve this part, I would want to create a simple TUI that displays the tree output which would have clickable tip labels, and would display the data right away instead of having to run a script for each tip you want to look at, going back and forth between command line and tree viewer.

Input:

correctly formatted fasta files in a directory

Output:

formatted directory of outputs of each step of pipeline (blast hits, sequences of blast hits, alignment, tree)
TUI output of tree and any sequence data produced from pipeline

Options:

some will be user options, others such as output names of each steps, will be automatically set by overall program (most ones I will need are listed below in scripts)

Data:

I will be using my own Hagfish assemblies from Dr. Plachetzki's lab

Testing Strategy:

I have already run each step of this pipeline individually. I will compare overall output of the pipeline before moving onto the next part, to make sure each output occurs, and there are no errors between programs due to misplaced files/ other.

I will test the Tip data script for many of the tips, and compare data output to actually going into the files and checking to make sure it is displaying the correct items. If there are any summary stats, I will use smaller test fastas and do the calculations myself for stuff like gc content/ gaps/ stop codon counts/ etc

Strategy:

Pipeline Script

Takes in a correctly formatted directory of pastas, and runs through the script to correctly format output for further use by next programs

mkblastdb

1. directory
2. fasta type (nuc/prot)

blast

1. query
2. fasta database (from mkblastdb)
3. output file names
4. format of output (BioPython may need specific?)
5. target seqs
6. e value
7. threads

selectseqs (take headers from blast out and grab hit seqs)

1. directory of blast hits
2. output fasta with hit seqs

cdhit to remove dupes

1. input fasta (from selectseqs)
2. output fasta with no dupes

alignment (MACSE)

1. alignment program
2. cdhit fasta input
3. output alignment

tree (IQtree)

1. input alignment
2. Tree algorithm

Tip Data Script

Displays data in an easy to read table output in terminal

- gc content
- stop codon in sequence?
- gaps in alignment
- blast scores/ etc

Can be used by user directly to show in terminal output, but I am hoping to make it used in TUI below if I have time

1. tip label
2. type of data you want output (blast summary, blast against NCBI + summary, alignment summary, etc)

Phylogenetic Tree Visualization (If the others get finished)

Take pipeline output and Tip Data Script and makes it a TUI Integrate biopython phylo.draw output with pytermgui to make it window scrollable and have clickable tips to show (and maybe run subscripts to make?) data

PyTermGUI

- format windows:
- tree in 3/4 top
- if tree tip clicked
- bot left -> blast against ncbi + show table
- bot right -> show stats of alignment with original seq

Biopython phylo.draw (ascii tree)

- make sure Macse output is formatted correctly
    - has some weird format for frameshift not usual to tree files
- format names to be Button class in pytermgui
- figure out how to identify button names so that I can run tip data script and push output to bot right/left windows of TUI

Future Directions:

It would be cool to be able to show all the stats for a tip, and if the tip had bad stats (stop codon in alignment, lots of gaps, etc) to be able to remove it directly in TUI, and at the end of analysis remake the tree without removed tips. This is definitely outside the scope of the current project, but would be another cool thing to implement.