Homework 3: Phylogeny Inference

CSCI 5481, Computational Techniques for Genomics
University of Minnesota
Instructor: Dan Knights

Instructions

- Please turn this assignment in on the course web page.
- There are multiple files to turn in. All text and code should be placed into a single folder with a name like *lastname_exerciseXX*. The folder should then be compressed and submitted as a single archive (.zip or .tgz)
- You must do this work on your own, although you are encouraged to have general discussions with other students. The work you turn in must be your own. Your code will be checked for overlap and for surprising idiosyncrasies in common with other submissions.
- Please write the names of all students with whom you discussed the assignment at the top of your code.
- Please include copious comments in your code. Full credit will only be given for code that is fully commented, meaning that every line that is not completely obvious needs a comment. Partial credit may be given for broken/non-functioning code if the code is well-commented.
- You may use any programming language you wish.

Background

This homework assignment is an implementation of the Nei-Saitou neighbor-joining algorithm for phylogeny construction, with estimation of bootstrap support.

Datasets

Download and extract the data: https://www.dropbox.com/s/81cqr72rpfgwnhy/Homework3.zip?dl=1.

If you will be using the supplied scripts for visualizing your tree (*hw3-plot-newick.r* or *hw3-plot-edges.r*), then you will need to install *R* (Google it), then install the “*ape*” package and the “*RColorBrewer*” package by running R, and then entering this command:

```
install.packages(c('ape','RColorBrewer'))
```

hw3.fna
File containing a multiple alignment of 61 bacterial 16S subunit ribosomal RNA sequences.

hw-tip-labels.txt
File containing tab-delimited rows of this format:

```
seqID    Phylum    color
```

There is a subfolder called *solution*. This contains examples of correct output:

solution/hw3-solution-genetic-distances.txt
Genetic distances (% different) between every pair of sequences in hw3.fna.

solution/hw3-nj-solution-edges.txt
Correct edges for neighbor-joining tree using R implementation, in preorder traversal order. Your edge order and internal node labels do not need to be exactly the same.

solution/hw3-nj-solution-bootstrap.txt
Example bootstrap support values for the 59 internal nodes in preorder traversal order.

solution/tree.pdf, *solution/tree-bootstrap.pdf*
Example PDF plot of tree showing colored tips and bootstrap support (bonus) for internal nodes.

solution/hw3-nj-solution-tree.tre
Correct solution output tree in NEWICK format. Your node order does not need to be exactly the same.

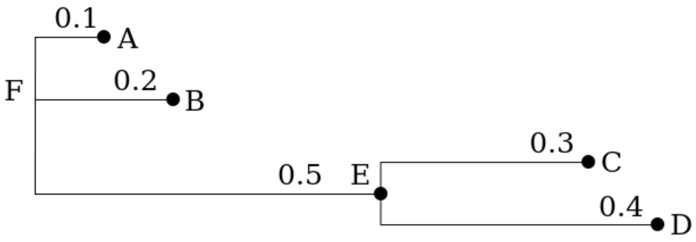
Input and Output Format:

Your program should take one command line argument specifying the name of a sequence file. The command line should be of the form *programName sequence.fasta* or *programName -i sequence.fasta* (it is acceptable to invoke java or another program as part of programName, but only take one argument as input).

Your program should output two files:

edges.txt
This file is tab delimited. Each row describes an edge in the tree. The first column is the ancestor node; the second column is the descendant node; the third column is the edge length. Edges should be in preorder traversal order choosing any internal node as the root. Tips must be indexed starting at 1, with 1 corresponding to the first sequence in the FASTA file, 2 corresponding to the second sequence, and so on. The internal nodes should begin numbering at the root with *ntips + 1* and the numbers should increase according to preorder traversal.

tree.txt
This file is in NEWICK format with all edge distances and with only tips named. For example, The following tree would be encoded (A:0.1,B:0.2,(C:0.3,D:0.4):0.5) :



Problems

1. (25 points): Read in the given FASTA file hw3.fna. Calculate the genetic distance (% dissimilarity) between every pair of sequences, and write this to a tab-delimited file with rows and columns labeled by the sequence identifiers. For pairwise dissimilarity calculations, you may count a gap in both sequences as a similarity.

2. (25 points): Implement Nei-Saitou neighbor joining as described on Wikipedia (https://en.wikipedia.org/wiki/Neighbor_joining) and/or in class notes. Provide extensive comments in your code. I suggest that you store your tree in this data structure, but you can do it however you want:

> `edges,` A 3-column matrix with column 1 representing an ancestor node, column 2 representing the descendant node, and the final column representing the edge length. Tips should be indexed starting at 1. Choose an arbitrary internal node to be the root. The internal nodes should begin numbering with the root as *ntips + 1*. An example is shown in *solution/hw3-nj-solution-edges.txt*.

3. (25 points): Generate the two output files described above for your tree (edges matrix and NEWICK tree file). For the edges matrix you will need to perform a preorder traversal. For the NEWICK file you will need to perform a postorder traversal. These should be generated using recursive functions.

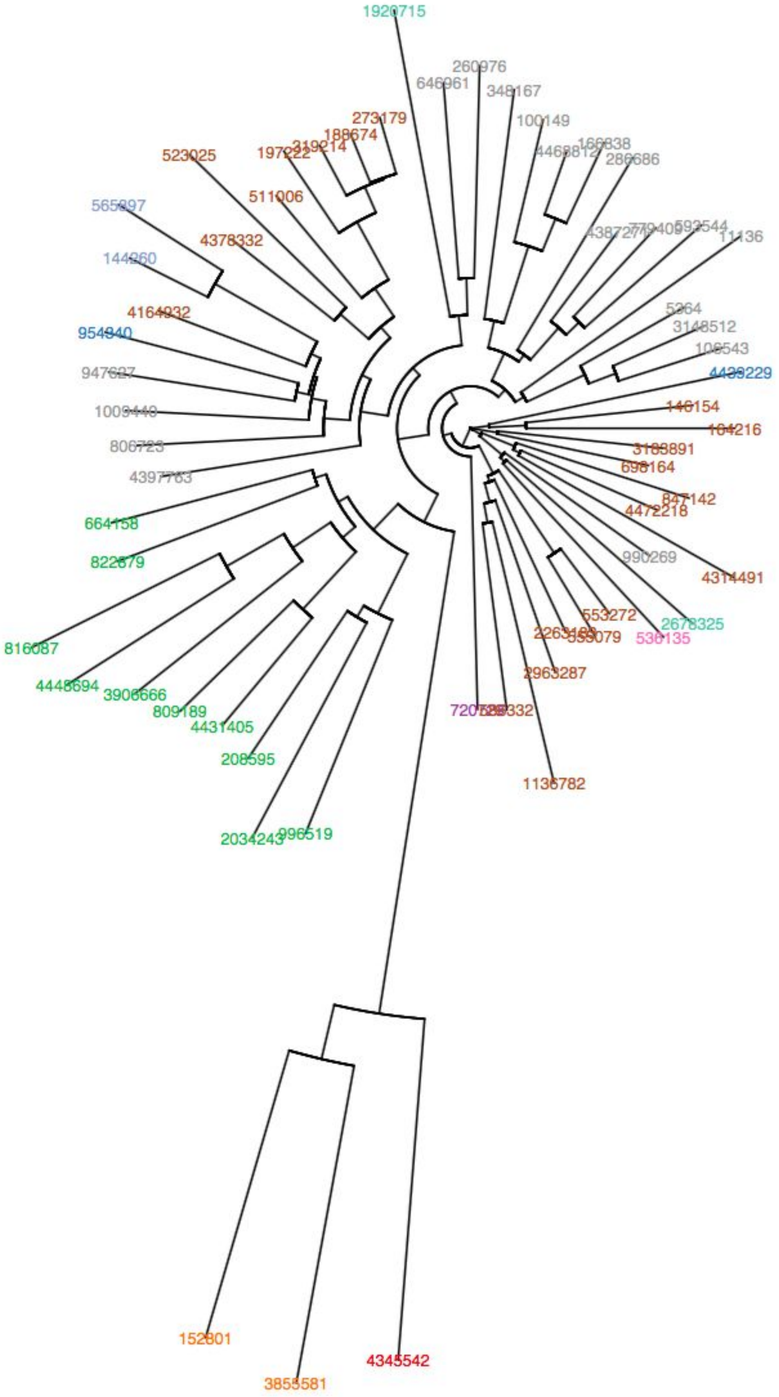
4. (25 points): Find a way to visualize your trees from step (3). There are many NEWICK-based viewers online. Colors for the tips are provided in *hw-tip-labels.txt*. Colors are nice but optional. You must include labels for your tips. ASCII art is acceptable but not preferred. If you made the output correctly for step (3), you can use either of the included R scripts (after installing the “ape” and the “RColorBrewer” package as described above):

```
Rscript hw3-plot-newick.r hw3-nj-solution-tree.txt hw3-tip-labels.txt
```

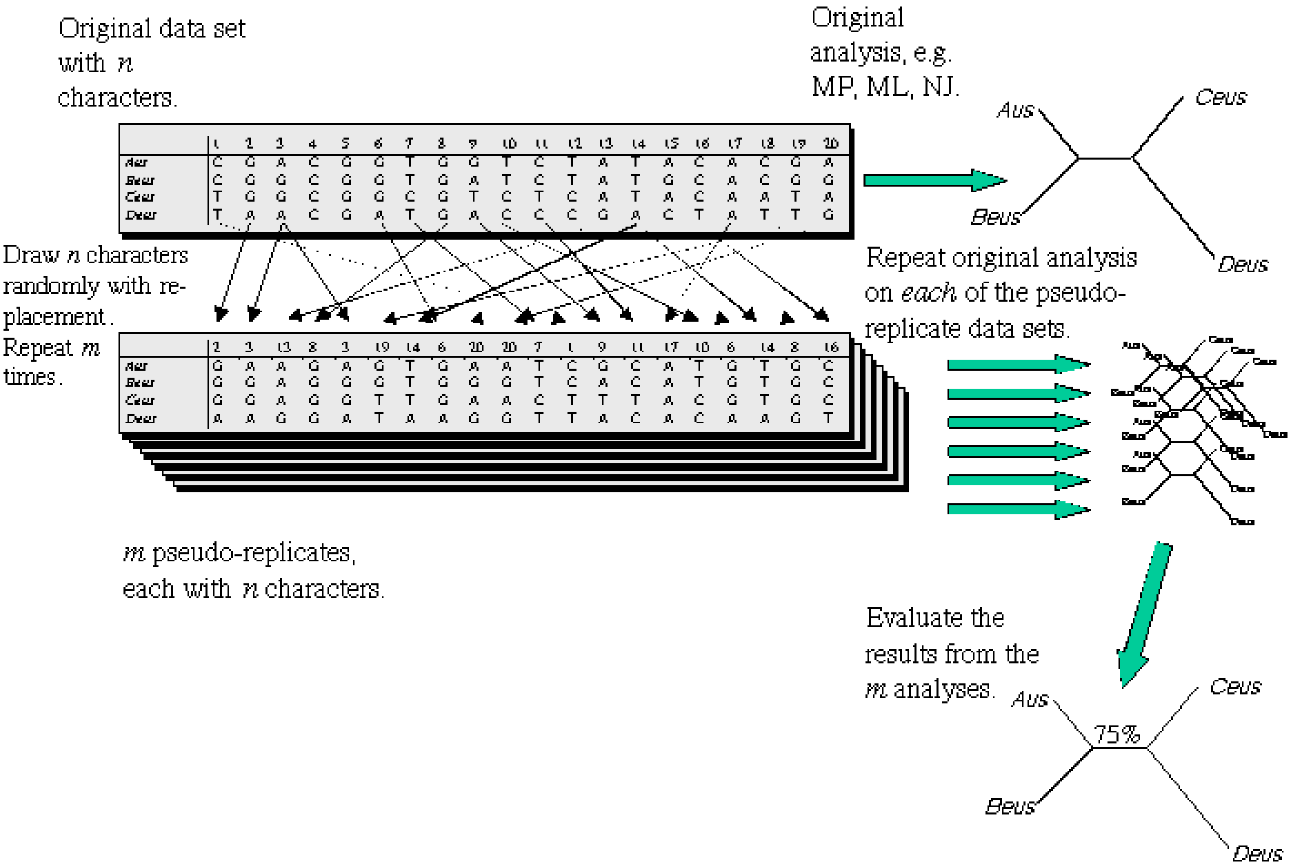
or

```
Rscript hw3-plot-edges.r hw3-nj-solution-edges.txt hw3-tip-labels.txt
```

Here is what my Nei-Saitou tree looks like:


1920715
260976
646961
348167
100149
166838
4468812
286686
273179
188674
319214
197222
523025
511006
565897
4378332
144260
4164932
954940
947627
1009440
806723
4397763
4387271
779409
593544
11136
5264
3148512
106543
4439229
146154
164216
3183891
698164
847142
4472218
990269
4314491
553272
2678325
2263189
555079
536135
2963287
720
585332
1136782
664158
822679
816087
4448694
3906666
809189
4431405
208595
2034243
996519
152801
3855581
4345542

BONUS (10 points): Perform 100 inferences of the tree in step (3) using bootstrap samples. Each bootstrap sample of the DNA sequences should contain a random rearrangement of the original DNA columns selected by repeated sampling with replacement until there are as many columns as in the original input. This means that some columns will appear more than once in a bootstrap sample. Here is a nice depiction of the resampling that Pin-Kuang showed us in class:

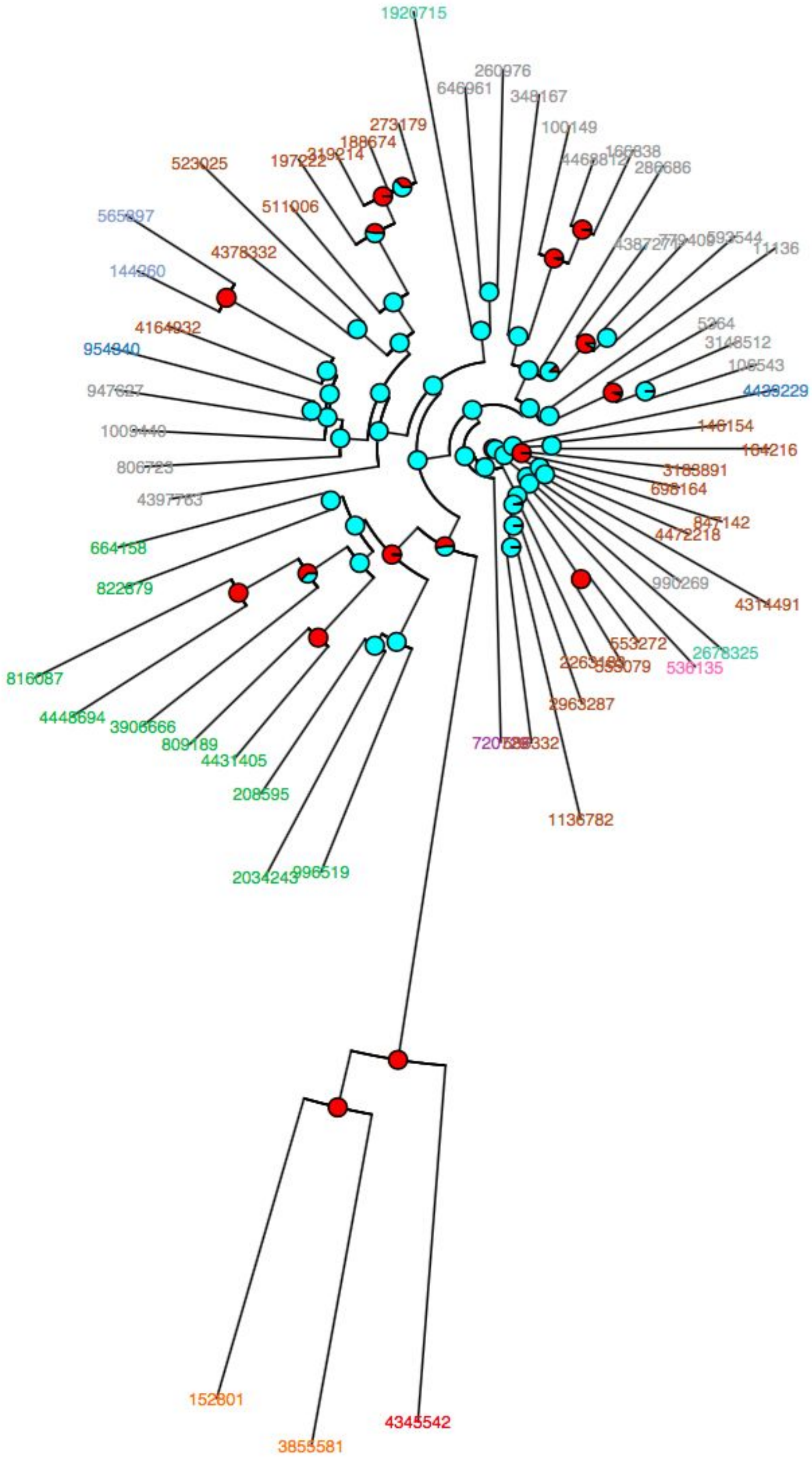

Original data set with *n* characters.
Original analysis, e.g. MP, ML, NJ.
Aus
Ceus
Beus
Deus
Draw *n* characters randomly with replacement. Repeat *m* times.
Repeat original analysis on *each* of the pseudo-replicate data sets.
m pseudo-replicates, each with *n* characters.
Evaluate the results from the *m* analyses.
75%

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Aus	C	G	A	C	G	G	T	G	G	T	C	T	A	T	A	C	A	C	G	A
Beus	C	G	G	C	G	G	T	G	A	T	C	T	A	T	G	C	A	C	G	G
Ceus	T	G	G	C	G	G	C	G	T	C	T	C	A	T	A	C	A	A	T	A
Deus	T	A	A	C	G	A	T	G	A	C	C	C	G	A	C	T	A	T	T	G

	2	3	13	8	3	19	14	6	20	20	7	1	9	11	17	10	6	14	8	16
Aus	G	A	A	G	A	G	T	G	A	A	T	C	G	C	A	T	G	T	G	C
Beus	G	G	A	G	G	G	T	G	G	G	T	C	A	C	A	T	G	T	G	C
Ceus	G	G	A	G	G	T	T	G	A	A	C	T	T	T	A	C	G	T	G	C
Deus	A	A	G	G	A	T	A	A	G	G	T	T	A	C	A	C	A	A	G	T

For each node in the original tree, get a list of the tips that are partitioned below it. Count the fraction of bootstrap trees in which the exact same partition was made. That is the bootstrap confidence for the given internal node. You may simply print the bootstrap values to a text file in the order of the indices of your internal nodes as shown in your edges file from step (3). You can also plot the bootstrap confidence using the supplied R scripts (red is fraction with bootstrap support):

```
Rscript hw3-plot-edges.r hw3-nj-solution-edges.txt hw3-tip-labels.txt
boots.txt
```


1920715
260976
646961
348167
100149
4468812
166838
286686
273179
188674
319214
197222
523025
511006
565897
4378332
144260
4164932
954940
947627
1009440
806723
4397763
4387277
779409
593544
11136
5264
3148512
106543
4439229
146154
164216
3183891
698164
847142
4472218
990269
4314491
553272
2678325
536135
553079
2963287
1136782
664158
822679
816087
4448694
3906666
809189
4431405
208595
2034243
996519
152801
3855581
4345542

Deliverables

Source files (your code for Step 1, 2, 3)
Readme file explaining how to use your code (text)
Step 1 distances file
Step 3 outputs (edges.txt, tree.txt)
Visualization of Step 3 tree
Code for and visualization of Bonus (optional)

All files and source code should be added to a folder with your x500 username as the name of the folder. Then, zip this folder and upload it on Moodle.