CSA 620: Project 6

Inferring and Using Phylogenetic Trees

Due: 2:30 PM, Friday Nov. 15

Setup: For this project you will need:

- The *pickle* library (automatically included).
- The *BioPython* library (can be installed for free from BioPython.org, SourceForge, or several other sources).
- Your code from projects 2, 4, and 5. (You may substitute my solution code if your code is not working.)

Introduction: Our goal is to use our computational tools with the goal of inferring knowledge about the evolutionary history of the SARS virus.

Severe Acute Respiratory Syndrome, or SARS, was first identified in late 2002 and quickly brought international notice due to the apparent seriousness of the pathogen. By April of 2003 the first strain had been sequenced (an isolate collected in Toronto), and computational biologists started looking for the source of the virus. An initial worry was that this was a mutated form of avian flu, but that hypothesis was dismissed when it was established that the SARS genome had essentially the same set of genes, in the same order, as members of the **coronavirus** virus family (a homologous set of virues). However, the sequenced version wasn't particularly similar to other known human coronaviruses – this was new member. This left open the questions: *did SARS represent the result a significant set of mutations to another human coronavirus, or did it originate from a coronavirus that typically infects some other organism but had made a cross-species jump?* If so, what species did it jump from, and when and where did this happen?

In this lab you will identify the source species, and in the next lab you will identify the geographic location and time of the jump. You are going to identify the source species of SARS as follows.

All animal-infesting coronaviruses carry a variant of a gene for the *spike* protein – a protein clearly important enough that it has been largely conserved over time on all lineages. The file **CoV.id** contains the GenBank ids of the associated protein from members of the family infecting several different species. Using the Neighbor-Joining algorithm, generate an unrooted phylogenetic displaying the phylogony of these viruses based on the one gene, and from there determine the source of the human SARs virus.

1. Use BioPython to download the associated protein sequences. (See the appendix for instructions.)
2. Compute a distance matrix as follows. For each pair as follows:
   - Align the sequence using a Smith-Waterman algorithm with affine scoring, using the Blosum62 scoring matrix (from project 5) and gap scores of $o = 7$ and $c = 1$.
   - Calculate the p distance using this alignment, *ignoring* the columns of the alignment that have a gap symbol in one sequence. (Example: an alignment with

90 matches, 10 mis-matches, and 20 gaps would have a p distance of 10/(90+10) = 0.9.)

- Use the Jukes-Cantor correction $d = -0.75 \ln(1 - 4p/3)$ to calculate distance d from the p distance p.

3. Use the distance matrix and the Neighbor-Joining algorithm to find the tree connecting the viruses.
4. Determine the closest neighbor(s) to the human SARS virus (presumably by eye – this doesn't need code), and conjecture from that (to the extent possible) the source species
5. *Note: you are dealing with larger sequences than we have been; don't be surprised if some of the calculations take a while to complete.*

Submission: While I will be looking over your code, there will be no auto-grading this time. And there will be no fixed program format. I want to see your results, and how you got them. You should submit:

- A file **CoV.fa** containing the downloaded fasta sequences.
- A file **CoV.dist** containing a *pickled* version of the distance matrix created for use by the Neighbor-Joining algorithm (see the pickle library) that I can unpickle.
- A file **CoV.tree** containing the Newick representation of your tree.
- A file **CoV.svg** containing a visualization of the tree generated here: http://www.trex.uqam.ca/index.php?action=newick
- All code written for the project. (This may be in a single file or split up between files.)
- A file README.txt containing:
  - Your conclusion as to source species.
  - A brief justification of your conclusion.
  - An explanation of the code. This does not need to be comprehensive documentation, but you do need to give me enough that I can follow what you did.

Do not use 3rd party bioinformatics tools (e.g. alignment or phloygenetic tree software from other sources. All computational problems in this project should be addressed by tools written as part of this course.

Appendix: GenBank / Entrez / BioPython

The NIH GenBank is a *massive* public repository of biological data, including just about all sequence data that has been the subject of any publication in the literature. (Most journals require that any sequences that are the subject of a publication be deposited into the database). Creating and maintaining GenBank is, in itself, a massive database and user interface problem and the product of a huge amount of research in those areas. Many biologists access it through the website (http://www.ncbi.nlm.nih.gov/ Genbank/), using the NIH tools to search for download information. As an example, try going to the site, selecting "Protein" in the search bar and searching for "H1N1". The 77,312 results (as of Sept. 27, 2011) all correspond to protein information having to do with the H1N1 virus.

To provide for automated access, the NIH supports **Entrez**, a data retrieval system that provides access to a number of databases and that can be accessed through BioPython. Using it we can write programs to search, download and parse data that might be needed from a database. The necessary libraries are explained in the BioPython tutorial:

http://BioPython.org/DIST/docs/tutorial/Tutorial.html

Chapter 3 of the tutorial explains the Sequence class, Chapter 4 explains the Sequence Record class, and Chapter 5 explains how to manage Sequence IO.[1] Section 5.2 gives an example of how to retrieve records from GenBank, and Chapter 8 provides a more comprehensive discussion.

WARNING!!! In order to balance workload, Entrez requires that the user not download more than three sequences per second. By default, the BioPython methods regulate the speed for you. **DO NOT OVERIDE THIS.** Failure to follow the Entrez protocols can result in the NIH shutting down Entrez access *to the entire University.*

[1] You are not likely to need to read these chapters in their entirety – there is a lot of extra detail not relevant to this project. Much of the time I can get by on just the code examples.