BMI/CS 776 Spring 2008
Homework #1

Prof. Colin Dewey

Due Tuesday, February 5th, 2008 by 11:59pm

This homework consists of a few problems to get you thinking about homology forests and some programming tasks to get you warmed-up for later assignments.

To turn in your assignment, copy all relevant files to the directory:
`/u/medinfo/handin/bmi776/hw1/USERNAME`
where `USERNAME` is your account name for the BMI network. You must submit a file named `README` to this directory, which gives directions on how to compile (if necessary) and run your programs. For each question below, the `README` file should list the files relevant to that question (e.g., code, other files with written answers). I recommend that you use the programming language(s) that you intend to use for the rest of the semester, as later assignments will build on code from earlier ones.

1. Given a set of labels $X = \{A, B, C, D, E, F, G\}$, and a laminar family, H, on X:

$$H = \{\{A, B, C\}, \{A\}, \{B, C\}, \{C\}, \{D\}, \{E, F, G\}, \{E\}, \{F, G\}\}$$

   (a) Give the rooted X-forest that corresponds to H.

   (b) Are there any edges that can be added to H and still have H be a laminar family? Either give such an edge or prove that none exist.

2. How many laminar families are possible for the set $X = \{A, B, C\}$? (Assume that each vertex must be a member of at least one edge)

3. Write a program, `randseq`, that outputs random DNA sequences. The program should be run as:

   ```
   randseq N L S output.fasta
   ```

   where `N` is the number of sequences to generate, `L` is the length of each sequence, `S` is an integer seed for your random number generator, and `output.fasta` is the name of the file in which to write the sequences. The sequences should be written

to the output file in FASTA format (see Wikipedia for the specification), which is the most common format in which biological sequences are stored. We will be using FASTA files throughout the semester, so you will want to write a FASTA reader and writer that you can reuse (there are software libraries out there that have such facilities, but I'd like for you to write your own). DNA sequences should be generated with equal probability for each base at each position.

4. Write a program, `invertsim`, that takes as input a single DNA sequence and outputs that sequence after it has undergone a fixed number of random inversions due to chromosome breakage. The program will be run as:

   ```
   invertsim N S input.fasta output.fasta
   ```

   where `N` is the number of inversions to simulate, `S` is an integer seed for your random number generator, `input.fasta` is the file containing the input sequence, and `output.fasta` is the file to which the output (mutated) sequence will be written. Model each inversion as two breaks chosen uniformly at random along the sequence followed by an inversion of the sequence in between the breakpoints. You should consider the ends as positions at which a break will occur (in these cases an inversion of a prefix or suffix will occur). Note that an inversion does not simply reverse a DNA sequence.