6 Evolutionary Trees

Adapted from a problem in DPV. Reconstructing evolutionary trees by maximum parsimony

Suppose we manage to sequence a particular gene across a whole bunch of different species. For concreteness, say there are n species, and the seqeuences are strings of length k over alphabet $\Sigma = \{A, C, G, T\}$. How can we use this information to reconstruct the evolutionary history of these species?

Evolutionary history is commonly represented by a tree whose leaves are the different species, whose root is their common ancestor, and whose internal branches represent speciation events (that is, moments when a new species broke off from an existing one). Thus we need to find the following:

- a (binary) evolutionary tree with the given species at the leaves
- For each internal node, a string of length k: the gene sequence for that particular ancestor

We will assume that a speciation event occurs when at least one base pair changes from the parent species. (Note that in practice, this is wildly impractical assumption, as there are many possible alterations to a DNA strand including deletions, additions, swaps, exchanges, and other local changes.) By this assumption, we can assume that all species have exactly k base pairs in their DNA.

For each possible tree T, annotated with sequences $s(u) \in \Sigma^k$ at each of its nodes u, we can assign a score based on the principle of parsimony: fewer mutations are more likely.

$$score(T) = \sum_{\{u,v\} \in E(T)} (\text{number of positions on which } s(u) \text{ and } s(v) \text{ agree})$$

Given a list of DNA sequences and a tree, find a labelling of the inner nodes with maximum parsimony.

The input file, called genetics.txt, will be a text file with first line that describes the tree by indicating the steps of a depth-first search from the root. L stands for left, R stands for right, and U stands for up. The next several lines will be the DNA strings that appear on the leaves of the tree from left to right. (HINT: Consider one string position at a time.)

Your output (tree.txt) will be the description of the tree with instructions for how to place DNA strings at every node. For the formatting, please see the example below.

For example, assume that the file genetics looks like the following.

LLLURUURLURUUURLURLURUUU

GCTC

CCAC

ACGA

ACCG

AACC

AGTG

TGGC

If you follow the instructions for creating the tree in the first line (left, left, left, up, right, up, up, right...), the tree that you create should have seven leaves. The DNA strings given should be placed inside those leaves from left to right. Your job will be to fill in the internal nodes of the tree and send the resulting completed tree.

One possible solution for this tree (tree.txt) is as follows.

:AACC

0:ACCC

00:ACAC

000:GCTC

001:CCAC

01:ACCA

010:ACGA

011:ACCG

1:AACC

10:AACC

11:AGCC

110:AGTG

111:TGGC

To determine where a particular DNA strand should be placed in the tree, follow the instructions given on the line. 0 stands for left, and 1 stands for right. The first line is the root (empty instructions) and indicates that the DNA string AACC should be placed inside the root. The second line indicates that the internal node to the left of the root should contain the DNA ACCC. And so on. For this example, the parsimony score is 13.

Every node must be filled with a DNA string.

Note that it is theoretically possible for one or both children to have exactly the same DNA as the parent.