Bioinformatics Supervision

Michaelmas Term 2017

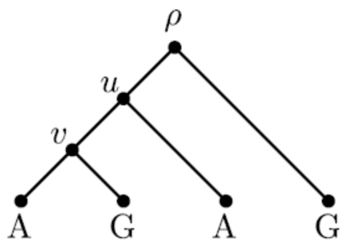
–Problem Sheet 2–

Supervisor: Sebastian Müller (Department of Plant Sciences)

Please hand in your work 24 hours prior to the supervision either to `sm934@cam.ac.uk` or at the Plant Sciences Department reception (make sure my name is on it). Feel free to team up with other group members, the main aim is to understand the material. If you hand in electronically, please name the file `group<x>_<crsid>_problemsheet<x>.<x>`

Phylogenetics

1. Considerable recent Bioinformatics research has focused on phylogenetics. What is the motivation for this work?
2. Describe the differences in complexity and performance between parsimony and two distance phylogenetic methods. Also try to find another tree construction method not mentioned in the lecture and describe it conceptually.
3. Commonly used methods for traversing a binary tree include pre-order, in-order, and post-order. Suppose we need to implement the SmallParsimony algorithm using one of these traversal methods. Which one(s) would be suitable for our implementation? Explain your choice.
4. How is the score matrix used in phylogenetic tree building techniques? Hint: A prominent example is the PAM score matrix described on page 285 of Vol.I (Compeau and Pevzner 2nd Edition).
5. You are given the tree below with single-letter sequences at its leaves. Use the SmallParsimony algorithm to find the minimum parsimony cost for the given tree and identify the optimal state assignments for each node with $c(A;G) = 1$ and $c(A;A) = c(G;G) = 0$.



6. Given the following distance matrix, calculate an evolutionary tree using UPGMA:

	A	B	C	D	E	F
A	0					
B	2	0				
C	4	4	0			
D	6	6	6	0		
E	6	6	6	4	0	
F	8	8	8	8	8	0

7. Given the following distance matrix, calculate an evolutionary tree using Neighbour-Joining algorithm:

	A	B	C	D	E
A	0				
B	5	0			
C	4	7	0		
D	7	10	7	0	
E	6	9	6	5	0

Pattern Matching

8. Implement *TRIE(patterns)* in a language of your choice using the patterns given in the book (Compeau, 2nd Edition, Vol II, page 124)

   patterns = "ananas", "and", "antenna", "banana", "bandana", "nab", "nana", "pan"

9. Implement a function *PrefixTrieMatching(Text, Trie)* which checks whether any string from *Patterns* matches a prefix of the string *Text* = "panamabananas" (see page 125). Also try with *Text* = "namabananas".

10. Discuss the main features of the Burrows-Wheeler transform (BWT) using the following example: T = `GATATCA$`. Also, explain the reversibility of BWT.

11. **OPTIONAL** Implement *TrieMatching(Text,Trie)* to allow pattern matching also within *Text* as opposed to only the prefix.