

MS2a, Exercises Week 5, Model Solution

Rune Lyngsø

November 12, 2009

A Score Based Alignment

Define a similarity score w on the four nucleotides such that

$$w(X, Y) = \begin{cases} 10 & \text{if } X = Y \\ 2 & \text{if } X \neq Y \text{ but } X \text{ can be changed to } Y \text{ by a transition} \\ 0 & \text{otherwise} \end{cases}$$

Furthermore, let an indel have a *dissimilarity* of $g = 10$.

To find the maximum 'similarity' between two sequences, $s_1 = \text{CTAGGA}$ and $s_2 = \text{TTGTG}$, (taken over all possible alignments) you should use the recursion

$$S_{i,j} = \max \{S_{i-1,j-1} + w(s_1[i], s_2[j]), S_{i,j-1} - g, S_{i-1,j} - g\}$$

With initial conditions

$$S_{i,j} = \begin{cases} 0 & \text{if } i = j = 0 \\ -\infty & \text{if } i < 0 \text{ or } j < 0 \end{cases}$$

a. Fill out the following table according to the recursion

	-50	-38	-18	-6	14	24	14
G					↘	↘	←
	-40	-28	-8	4	14	12	2
T				↘	↘	↘	
	-30	-18	2	14	12	2	-8
G			↘	↘	↘		
	-20	-8	12	2	-8	-18	-28
T		↘	←				
	-10	2	0	-10	-20	-30	-40
T		↘					
	0	-10	-20	-30	-40	-50	-60
	C	T	A	G	G	A	

- b. What is the maximum similarity score between the two sequences s_1 and s_2 ?

The maximum similarity score is the score in the upper, right hand entry of the table, *i.e.* the maximum similarity score is 14.

- c. Find an alignment with this similarity score.

The tracebacks of the maximum similarity score is indicated by arrows in the alignment table. The alignments corresponding to the two possible tracebacks are

C T A G G A	C T A G G A
T T – G T G	T T G T G –

- d. Is the alignment you found unique, or are there more than one alignment achieving the maximum similarity score?

There are two distinct alignments maximising the similarity score, each having two identities, two transitions, and one indel.

B Recombination

- a. Can we find a tree for the data set

Pan	TTATCC
Gorilla	TTGTTC
Pongo	CCACCC
Hylobates	CCGTTC

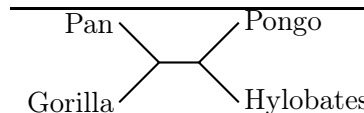
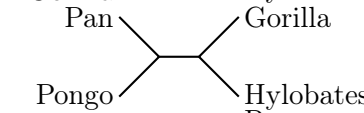
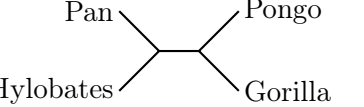
such that only one substitution is required in each position? If yes, provide such a tree. If no, why not?

We cannot. The first two sites require the tree to group Pan and Gorilla together, against Pongo and Hylobates, to be able to explain them with just one substitution. However, positions 3 and 5 require that we group Pan and Pongo together against Gorilla and Hylobates to explain them with just one substitution.

- b. Compute the minimum number of substitutions required for the above data set for each of the three possible unrooted tree topologies, e.g. by using Fitch's algorithm (or just eye-balling it if you feel confident about doing this).

Independent of topology, site 4 will always require 1 substitution and site 6 will not require any substitutions. For the remaining four sites,

the cost depends on whether the topology has the right grouping into pairs. Hence, we get the following minimum number of substitutions:

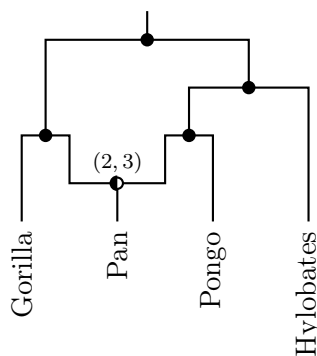
	Site	1	2	3	4	5	6	Total
		1	1	2	1	2	0	7
		2	2	1	1	1	0	7
		2	2	2	1	2	0	9

- c. Assume that apart from substitutions, you are also allowed events arbitrarily changing the tree topology between consecutive sites (this is a simplification of recombination events – recombination events only allow certain changes to tree topology). What is the minimum number of events you need to explain the above data set.

If we start with the topology grouping Pan with Gorilla, but between sites 2 and 3 swap to the topology grouping Pan with Pongo, we get a total of 5 substitutions and one change of topology for a total of 6 events.

Give an ancestral recombination graph explaining the data set with this number of events.

There are many possible ancestral recombination graphs possible, but if we assume that Pan is the recombinant species with closest relative Gorilla for the first two positions and closest relative Pongo for the remaining positions, we get the following



where the partly filled node represents the recombination. At a recombination node we choose the branch in the direction of the filled half of the node for positions less than or equal to the first position in the pair indicating the recombination point, otherwise we choose the branch in the direction of the hollow half of the node.

- d. How many recombination nodes are there in the ancestral recombination graph (ARG) you constructed in c?

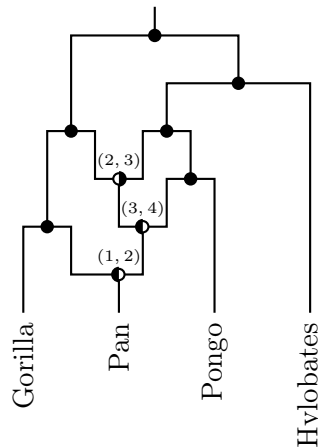
There is one recombination node in the ARG above.

Can you construct a data set by permuting the columns in the above data set that requires more recombination nodes for any ARG explaining it? If yes, give an example.

One example would be the data set obtained by swapping columns 2 and 3:

Pan	TATTCC
Gorilla	TGTTTC
Pongo	CACCCC
Hylobates	CGCTTC

Now the first site requires that Pan and Gorilla are grouped together, the second site that Pan and Pongo are grouped together, the third site that Pan and Gorilla are grouped together, and the fifth site that Pan and Pongo are grouped together. Hence, we will need recombinations between sites 1 and 2, between sites 2 and 3, and between sites 3 and 5. An ARG yielding this relationship would be



How many different marginal trees does the ARG you constructed in *c* have (a marginal tree is the tree relating the species at a particular position)?

For positions 1 and 2 we get the tree grouping Pan and Gorilla, while for positions 3 to 6 we get the tree grouping Pan and Pongo. This gives two marginal trees.

Can you construct a data set by permuting the columns in the above data set that has more different marginal trees in any ARG explaining it? If yes, give an example.

All informative sites in the data set groups Pan with either Gorilla or Pongo. Hence we do not need other marginal trees to explain the data. We can always construct an ARG with any given sequence of marginal trees: first add recombinations nodes above each species to split its sequence into single positions, then connect the lineages corresponding to the same position according to the marginal tree desired for that position, and finally connect the lineages for each position in arbitrary order. It follows that any permutation of the columns will allow an ARG with just the two marginal trees of the ARG in *c*.