AAGB - TME 3 - Sankoff Algorithm

December 4, 2020

1 Problem : Pokémon Phylogeny

We would like to study a particular branch of the Pokémon phylogenetic tree. For each species studied, a well-conserved position in a gene of their DNA is recovered. Here are the different species considered, as well as the associated nucleotide:

Probopass : A
Aggron : T
Bastiodon : T
Regirock : G
Registeel : G
Regice : G
Klingklang : G
Metagross : C
Genesect : A
Porygon−Z : C
Magnezone : C
Forretress : T
Eletrode : A
Ferrothorn : G

Two groups of scientists propose different phylogenetic trees to explain the evolution of Pokémon. We recovered the structure of the two trees in the Newick format:

N1 = "((((Electrode , Magnezone) ,Porygon−Z) , ((((Aggron , Bastiodon) , Forretress) , Ferrothorn) , (((((Regirock , Regice) , Registeel) , Metagross) , Klingklang) , Genesect))) , Probopass);"
N2 = "(((((Regirock , Regice) , Registeel) , ((Metagross , Klingklang) , Genesect)) , (((Aggron , Bastiodon) ,(Forretress , Ferrothorn)) , Probopass)) ,(Porygon−Z,(Magnezone , Electrode)));"

The goal is to determine the most probable tree, using Sankoff's algorithm to find which has the lowest parsimony score.

2 Sankoff's Algorithm

An efficient method would be to use a tree structure, available for example with the ete3 package, and to browse the graph from the leaves to calculate the parsimony scores, and from the root for the traceback. We would not like to use any particular package here, so we provide a guide for developing the Sankoff algorithm without resorting to it.

2.1 Tree Visualization

View the two proposed trees, for example using the following site: etetoolkit treeview.

2.2 Computation of the internal nodes' scores

A strategy may be, as a first step, to create a dictionary matching the name of the nodes to their labeling. For example, we can create an initialization function for such a dictionary by all the leaves, which we progressively update with each new internal node. Then code a function allowing, given two nodes and their associated score vector, to return their common ancestor and its parsimony score.

We can then parse the character string corresponding to the Newick format to gradually cluster each pair of leaves / internal nodes.

2.3 Traceback

We can build the traceback in parallel, by storing progressively in a list all the pairs of nodes that we have clustered, as well as the argmin which made it possible to obtain the various parsimony scores. You just have to go through the list in the opposite direction to start from the root and gradually descend into the tree. The idea is to have at the output of the program a character string with the labeling of the internal nodes, as well as the parsimony score of the tree.

2.4 Verdict

Which tree gives the minimum parsimony score? (See the image for the complete Pokémon phylogenetic tree...)