Practice PRO2

Spring 2020

1. Introduction

In evolutionary biology, anthropology, linguistics and many other scientific disciplines, one of the main problems addressed is the construction of *phylogenetic trees*, diagrams that schematically represent the evolutionary relationships between a set of N entities (in evolutionary biology we usually speak of *species*, although often not species but families or orders), as in the examples in figures 1 and 2. The phylogenetic tree is constructed on the basis of the similarities and differences in the physical or genetic characteristics of the N entities.

In a *rooted* phylogenetic tree each node with descendants represents the most recent common ancestor of its descendants, and it is usual that the length of the edges/branches of the tree is proportional to the time elapsed between the entities represented. The internal nodes are hypothetical entities, as they cannot be directly observed, only the N entities of the leaves of the tree are known to us.

The construction of the phylogenetic tree that best explains the observed data, optimising a certain criterion, is a computationally difficult problem, in the sense that the cost of the necessary calculations grows exponentially with N . In addition, many other problems complicate the obtaining of phylogenetic trees that accurately reflect evolutionary history: inaccurate entity data, hybridisations, convergent evolution, . . .

For this reason, numerous approximate methods have been developed that generate phylogenetic trees efficiently and that approximate the optimal tree very well in most cases, but not always. In this practice our aim will be to build a program and the necessary modules to construct the phylogenetic tree for a set of N species using one of these approximate methods: the method known as WPGMA (*weighted pair group with arithmetic mean*).

2. The WPGMA method

In this subsection we describe how the construction of a phylogenetic tree works using the WPGMA method. We assume that for any two species (or entities) i and j, we have their *distance* $\delta(i, j)$.

In section 4 we give details of how we will calculate the distances between species in our practice. But for the moment let us assume that we have this information: given a set of N species for which we want to construct their phylogenetic tree, we will assume that we have already calculated the distance between any pair of them.

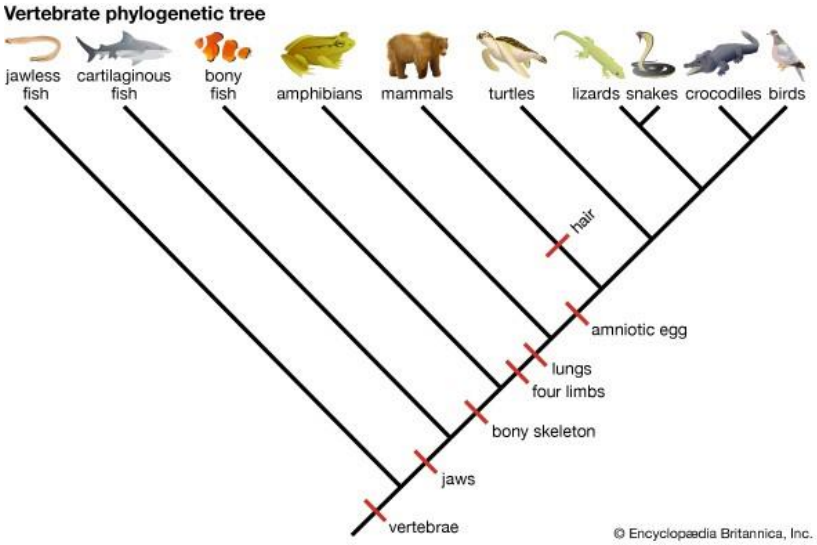

Figure 1: An example of a phylogenetic tree for vertebrates.

Figure 2: An example of a phylogenetic tree for various Indo-European languages.

The WPGMA algorithm builds a binary tree that reflects the hypothetical evolution of the N species in an iterative process in which *clusters* or groups of species are created: each subtree represents a cluster, given two clusters we can join them to form a larger one that brings together the species of both. The root node of each sub-tree represents the most recent common ancestor of all the species in the cluster. The construction of the tree is based on the principle that all distances between the root of a subtree and its leaves are identical, as the length/distance of each branch is proportional to the elapsed time and we will assume that all N species for which we want to construct the tree exist today. For example, in figure 1, the root node of the subtree containing lizards, snakes, crocodiles and birds would represent the common ancestor of all of them (the diapsids). The ancestor of this is the root node of the cluster that also contains turtles (anapsids), so it represents the common ancestor of all sauropsids (basically all reptiles and birds, extinct or not). In the other figure, the phylogenetic tree groups all Latin-derived languages into a cluster, from which a group leading to languages spoken in Sardinia (Nuorese, Cagliari), dialects of Sardinian, is first "split off". From the rest of the cluster, Romanian is later separated, and then two clusters, one with languages spoken in the Iberian Peninsula, and the rest in Italy (Italian, Friulian, Ladin), France (French), Switzerland (Italian, Friulian, Ladin), and the other in Italy (Italian, Friulian, Ladin).

(Romansh), . . .

The WPGMA algorithm starts with a set S of N clusters: each cluster contains a single species and corresponds to a tree consisting of a leaf (a root representing the species) and a tree consisting of a leaf (a root representing the species).

cluster with no descendants). The distance between two clusters $A = \{a\}$ and $B = \{b\}$, each of which

which contains only one species, is given by the distance between the corresponding species:

$$\Delta(A, B) = \delta(a, b).$$

Then, in each iteration of the WPGMA there are three main steps:

1. Identify the two clusters A and B at the shortest distance in S and form a new cluster. $C = A \cup B$.

   The tree associated with cluster C consists of a root that represents cluster C as a whole and left and right subtrees that correspond to clusters A and B. Later we will see how to assign identifiers to the clusters, so that if the identifier of A is smaller than the identifier of B then the left subtree will be A and the right subtree will be B.

   We will also use the order between identifiers to resolve ties between distances. If the situation arises, we will choose the pair of clusters A and B that gives rise to a new cluster C with smaller identifier.

2. Calculate the distance between the new cluster $C = A \cup B$ and the remaining clusters in S. For any cluster D in S the distance between C and D is:

$$\Delta(C, D) = \frac{\Delta(A, D) + \Delta(B, D)}{}$$

3. Add cluster C to S and remove A and B from S.

The algorithm ends when the set of clusters has been reduced to a single element. It will then

have generated the phylogenetic tree of the initially given set of species.

Let us look at an example. Suppose there are $N = 5$ entities or species, a, b, c, d and e. Initially we will have 5 clusters, each with only one species. In the tables of the example we will use, for convenience, the identifiers (a, b, . . .), for the clusters formed by a single species, instead of writing $\{a\}$, $\{b\}$, . . . which would be more accurate. As we have already said the distance between two clusters A and B, each containing only one species is the distance between the two species.

Suppose we have the following distance matrix:

	a	b	c	d	e
a	0		21	31	23
b		0	30		21
c	21	30	0		
d	31			0	43
e	23	21		43	0

In the first iteration the clusters $\{a\}$ and $\{b\}$ form the pair with minimum distance and are merged into a new cluster $\{a, b\}$; the distance table is updated as follows:

	$\{a, b\}$	c	d	e
$\{a, b\}$	0	25.5	32.5	
c	25.5	0		
d	32.5		0	43
e			43	0

In the second iteration the clusters to be joined are $\{a, b\}$ and $\{e\}$, giving rise to a new cluster.

$\{\{a, b\}, e\}$. The updated distance matrix is

	$\{\{a, b\}, e\}$	c	d
$\{\{a, b\}, e\}$	0	32.25	37.75
c	32.25	0	
d	37.75		0

In the third iteration the clusters $\{c\}$ and $\{d\}$ are grouped into a new cluster $\{c, d\}$; after updating the resulting distances are:

	$\{\{a, b\}, e\}$	$\{c, d\}$
$\{\{a, b\}, e\}$	0	35
$\{c, d\}$	35	0

In the final step the last two remaining clusters $\{\{a, b\}, e\}$ and $\{c, d\}$. are merged into a final cluster $\{\{\{a, b\}, e\}, \{c, d\}\}$.

Figure 3 shows the phylogenetic tree constructed by the WPGMA algorithm. The figure takes into account the cluster identifier conventions described in the next section.

3. Clusters

It follows from the above description that a cluster either contains a single element or *species*, or is made up of a pair of clusters: in short, a cluster is assimilated to a

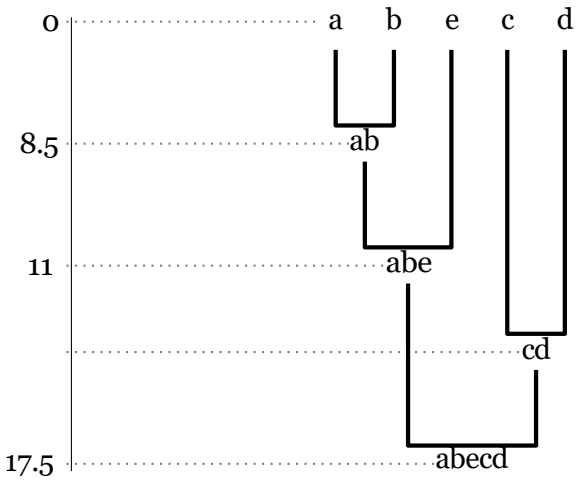

Figure 3: Phylogenetic tree constructed by the WPGMA algorithm.

a binary tree, where we have species at the leaves. Our system must be able to store and dynamically update a "distance table" between clusters. The "rows" and "columns" of such a table are indexed by clusters, and it must be possible to efficiently add and remove distance information between a given pair of clusters.

On the other hand, it must be possible to provide the initial content of such a "distance table" from a set of N species, each of the N species being considered as consisting of the corresponding N clusters, each with a single element.

Although the WPGMA phylogenetic tree construction algorithm does not prescribe who is the "left child" and who is the "right child" (it is irrelevant) in a cluster, in this practice we will adopt the following conventions:

1. A cluster with a single species has the same identifier as the species identifier.
2. Given two clusters with identifiers α and β such that $\alpha < \beta$ in lexicographic order, their combination is a cluster that has as its left child the cluster identified by α, as a right child of the cluster identified by β and its identifier is $\gamma = \alpha$ - β, the concatenation of the identifiers of its children.

In the examples in the previous section we have not used the identifiers of the clusters for ease of explanation1, but let's see now, with the example we have developed, how to use the identifiers of the clus- ters.
previously, how identifiers are assigned to the clusters as the WPGMA algorithm proceeds following the conventions just described. Initially we have the clusters $\{a\}, \ldots,$ $\{e\}$ with identifiers "a", "b", . . . , "e". Then the clusters with identifiers "a" and "b" are combined to form the cluster $\{a, b\}$ whose identifier is "ab". In the second iteration we combine the clusters $\{a, b\}$ and $\{e\}$, and the resulting cluster $\{\{a, b\}, e\}$ is assigned the identifier
"abe".

Note that at the end of the process only one cluster containing all species will remain and its identifier will be a concatenation of the identifiers of all species, irrespective of which is the starting set of species, but the order will be naturally

[1]Figure 3 does reflect the cluster identifier conventions described earlier in this section.

depending on what are the distances between them and what is the resulting tree structure. In our example the identifier of the final cluster (= phylogenetic tree) is "abecd". Note on the other hand that the identifier does not allow us to uniquely reconstruct the tree structure.

4. Species and genes

In the preceding sections we have not gone into detail on what exactly an entity or species is and have assumed that, given two species i and j, we can calculate the distance between them $\delta(i, j)$.

Both the definition of what an entity is and the calculation of distances will vary substantially depending on the application under consideration - whereas the WPGMA construction process is based on the abstraction of species and distance, and is completely independent of what species actually are and how distances are calculated. Your practice, and in particular your modular design, should reflect this.

In our practice we will greatly simplify what we usually find in applications with regard to the information available for each species and the calculation of the distance between two species. Usually for each species we would have a large set of traits or sequenced genes, but we will consider that each species consists of an identifier (a *string*) and a *gene* (another string).

Each gene consists of a (long) sequence of symbols, taken from the only four possible ones: A, C, T or G[2]. So for example a gene could be AACTTGCGAGCTACAACCTGGGATTA.

All that remains is to determine how to compute the distance between two species. There are many different notions of distance between two symbol sequences (the way we represent genes in this practice), but many of them require very expensive computation because the sequences involved can be very long (e.g., the DMD gene contains 2.4 million nucleotides - base pairs in the DNA double helix - so the sequence is 2.4 million symbols in length!)

In this practice we will use a simpler procedure to estimate the distance between two sequences g_1 and g_2 . Given any sequence g and a value $k \geq 1$ we will say that kmer(g, k) is the set of contiguous subsequences of g of length k (the k-numbers of g),
i.e. if n is the length of g then

$$\text{kmer}(g, k) = \{g[i..i + k - 1] \mid 0 \leq i < n + 1 - k\},$$

where $g[i..j]$ denotes the subsequence of g between the *i-th* and *j-th* symbols of g, which we assume to be indexed from 0 to $n - 1$. For example, if g = ACATTATCATGC and $k = 3$ then

$$\text{kmer}(g, 3) = \{\text{ACA}, \text{ATC}, \text{ATG}, \text{ATT}, \text{CAT}^2, \text{TAT}, \text{TCA}, \text{TGC}, \text{TTA}\},$$

where we have written the 3-mer of g in lexicographically increasing order, and the superscripts indicate multiplicity - for example, the 3-mer CAT appears twice in g. The distance between g_1 and g_2 is then defined as

$$\delta_k(g_1, g_2) = \left(1 - \frac{\#(\text{kmer}(g_1, k) \cap \text{kmer}(g_2, k))}{\#(\text{kmer}(g_1, k) \cup \text{kmer}(g_2, k))}\right) \times 100,$$

[2]A gene is a segment in a sequence of nucleotides on a DNA strand. There are only four types of nucleotides in a DNA strand: adenine (A), cytosine (C), guanine (G) and thiamine (T).

where #A denotes the cardinality of the multiset A. Recall that if x appears i times in A ($x^i \in A$) and j times in B ($x^j \in B$) then $A \cap B$ contains $\min(i, j)$ occurrences of x, and that $A \cup B$ contains $\max(i, j)$ occurrences of x.

The value of k is a parameter that will be set globally at the beginning of the execution of the programme and will be used to calculate all the distances between pairs of species. It is appropriate to think of it as a value associated with the table of distances of a set of species and consequently it is a known value even before setting the set or sets of species from which we intend to build their phylogenetic tree.

Although your practice should work well with any value of $k > 0$, the best results in real situations are obtained with values of k between 5 and 15 (we
we are going to use "short" genes in the test sets, so a value of k in the low range will be more appropriate). The set kmer(g, k) has at most $|g| - k + 1$ distinct elements and therefore a very high value of k would be of little use as there would be too few elements.
and all of them would have very low multiplicity, as a rule. At the other extreme, a very low value of k would not be useful either, because in kmer(g, k) we would end up with all possible *k-numbers* (there are 4 possible^k) and the estimation of the distance between two species would then be very crude.

5. Functionalities

5.1. Decisions on data

1. All identifiers are strings consisting exclusively of lowercase and uppercase letters (A to Z, no special characters, no accented vowels, . . .), digits and the underscore character '_'.

2. No species identifier is a prefix of any other species identifier. This ensures that cluster identifiers that are created by concatenating identifiers reflect which species are part of the cluster (not its tree structure). It also ensures the validity of the inter-cluster tie-breaker we have defined.

3. There will be no species sharing the same gene or genes of length $<k$. There may be species at distance 0 (i.e. with the same set of *k-numbers*), but for the WPGMA algorithm this is not a special case.

4. All distances, between species or between clusters, are real numbers (double) that are printed as they are in the cout.

5. The entry is syntactically correct; i.e., if an operation says it has two arguments that are identifiers, the entry will contain exactly that, two valid strings as identifiers.

5.2. Main programme commands

Your main program will start by reading the *k-value* that will be used globally for the calculation of all inter-species distances. You start with an empty set of species and an empty set of clusters. In the remainder we will refer to them as **the** set of species and **the** set of clusters.

Once the sets of species and clusters have been created, a loop is entered that processes one option in each iteration. The options are as follows:

1. create_species: Creates a species with the given identifier and gene (two strings). Writes an error message if a species with the same identifier already exists. The created species, if there is no error, is added to the species set.

2. get_gen: Given a species identifier, prints the gene associated with the species. Writes an error message if no species exists with the given identifier.

3. distance: Given two species identifiers, prints the distance between the two species. An error message is written if either of the two species whose identifiers are given do not exist.

4. removes_species: Given the identifier of a species *and* removes it from the species set. Writes an error message if the species with the given identifier does not exist.

5. exists_species: Given the identifier of a species *and* prints an indication of whether that species exists (i.e. is part of the species set).

6. reads_cjt_species: Reads from the standard input channel an integer $n \geq 0$ and then a sequence of n species (identifier-gene pairs). The given n species have identifiers that are distinct from each other. The previous contents of the species set are discarded. -species cease to exist - and the n species read are added to the set of species.

7. print_cjt_species: Prints in the standard output channel the set of species. If the set is empty, no information is printed.

8. distance_table: Prints the table of distances between each pair of species in the species set. If the set is empty, no information is printed.

9. initialise_clusters: Initialises the set of clusters with the set of species in the current state, and prints the table of distances between clusters. If the set is empty, no information is printed.

10. execute_wpgma_step: executes a step of the WPGMA algorithm (merges the two clusters at smaller distance into a new one) and prints the resulting inter-cluster distance table. In case the number of clusters in the set is less than or equal to one, only an error message shall be printed.

11. print_cluster: given an identifier α, prints the cluster (its "tree structure") with the given identifier, or an error if there is no cluster with that identifier in the set of clusters.

12. print_phylogenetic_tree: prints the phylogenetic tree for the current set of species; this tree is the cluster grouping all species, resulting from applying the WPGMA algorithm. The content of the previous cluster set is discarded and reinitialised with the species set in its current state, and then the algorithm is applied. The final set of clusters is the set of clusters remaining after applying the algorithm.

    If the new set of clusters is empty, only an error message has to be written.

13. end: the execution of the programme ends.

In the two operations that require printing trees, the cluster identifiers (roots of the subtrees) and the distance between each cluster and its descendant leaves (see Figure 3; these distances are the numbers on the left, and can be easily calculated from the distance between the clusters whose combination gives rise to each cluster) will be written. The precise format in which the tree is to be printed will be shown in the public test suite.
Important: the cluster set only records additions and deletions to the species set when it is reinitialised using `initialise_clusters` or `print_phylogenetic_tree`. If, after an application of either of these two operations, the species set undergoes additions or deletions, these do not affect the cluster set until the next reinitialisation.