Comparing and Contrasting Phylogenetic Tree Building Algorithms

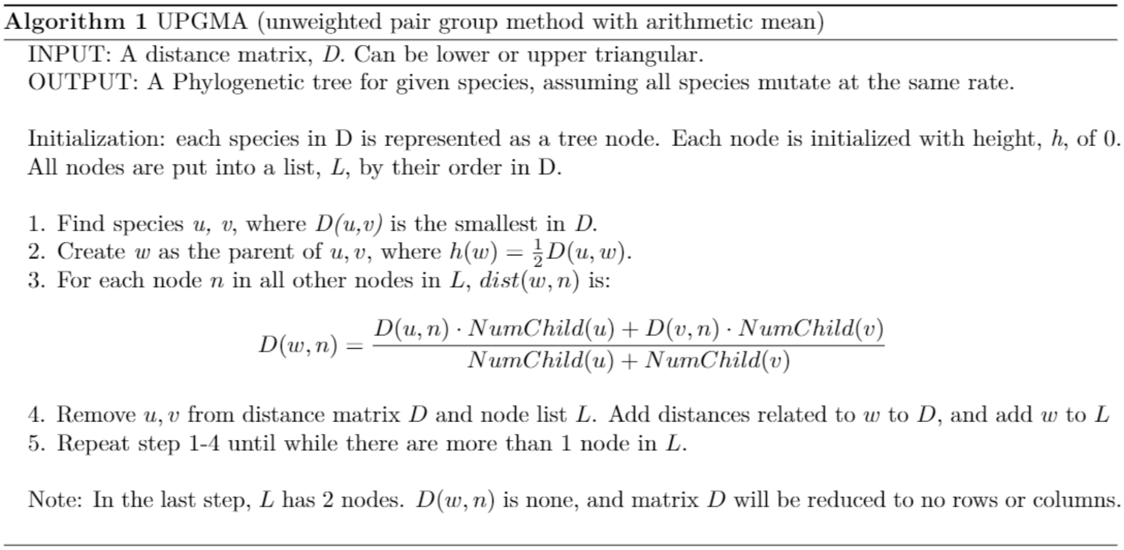
Jimmy Zhong, Zhen Ren's final for Professor Layla Oesper's CS362, Computational Biology

Introduction

Phylogenetic trees, also known as evolutionary trees, are representations of biological relationships between species. They are based on genetic similarities and differences of species. The methods that created phylogenetic trees can be roughly classified as methods that do not consider evolutionary models, such as UPGMA[1] (unweighted pair group method with arithmetic mean) and neighbor-joining [2], or methods that involve evolutionary models, such as maximum parsimony tree and Bayesian methods. In our project, we want to further explore and compare these two kinds of phylogenetic tree reconstruction methods. Considering the project scale and time limitation, we decide to implement UPGMA from the former group and the Sankoff algorithm [3] from the latter group. UPGMA is a hierarchical clustering algorithm that greedily builds a phylogenetic tree from the bottom up. Sankoff algorithm can quickly calculate the parsimony score given a tree shape, so we enumerate all possible tree shapes and find the most parsimonious, or, the maximum parsimony tree. Here, we construct the phylogenetic trees for hemoglobin subunit α-a in cats (*Felis catus*), rhesus macaques (*Macaca mulatta*), crab-eating macaques (*Macaca fascicularis*), chicken (*Gallus gallus*), black and yellow macaws (*Ara ararauna*), and humans (*Homo sapiens*) by both methods and compare the difference between the tree created by UPGMA, Sankoff's algorithm.

UPGMA Algorithm

The input is an *n*n* distance matrix, which can be obtained through pairwise global alignments for each species (for *n* species, *n(n-1)/2* pairwise alignments are needed). The output is an ultrametric tree where leaves are existing species. The idea of the UPGMA algorithm is to recursively collapse the shortest edge on the graph into a union (a greedy algorithm). After collapsing two groups into a union, we define the distance from the new union group to all other nodes by the mean value for all possible pairwise distances between two groups.



Algorithm 1 UPGMA (unweighted pair group method with arithmetic mean)

```
INPUT: A distance matrix, D. Can be lower or upper triangular.
OUTPUT: A Phylogenetic tree for given species, assuming all species mutate at the same rate.

Initialization: each species in D is represented as a tree node. Each node is initialized with height, $h$, of 0.
All nodes are put into a list, $L$, by their order in D.

1. Find species $u$, $v$, where $D(u,v)$ is the smallest in $D$.
2. Create $w$ as the parent of $u, v$, where $h(w) = \frac{1}{2}D(u,w)$.
3. For each node $n$ in all other nodes in $L$, $dist(w,n)$ is:

   $$D(w,n) = \frac{D(u,n) \cdot NumChild(u) + D(v,n) \cdot NumChild(v)}{NumChild(u) + NumChild(v)}$$

4. Remove $u, v$ from distance matrix $D$ and node list $L$. Add distances related to $w$ to $D$, and add $w$ to $L$
5. Repeat step 1-4 until while there are more than 1 node in $L$.

Note: In the last step, $L$ has 2 nodes. $D(w,n)$ is none, and matrix $D$ will be reduced to no rows or columns.
```

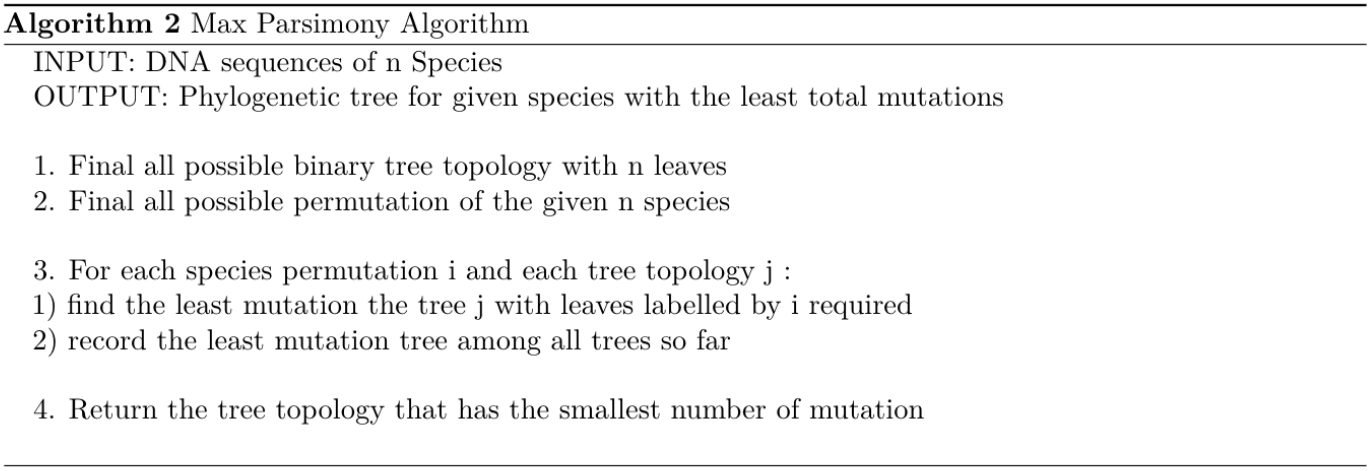
Algorithm 1: The pseudo-code for UPGMA.

The UPGMA algorithm completes in *$O(n^3)$* time, which is fast among tree construction algorithms. However, one of its major pitfalls is that it only generates ultrametric phylogenetic trees, which means each leaf has the same distance to the root node. Biologically, this implies a constant molecular clock, an unrealistic assumption that each species has the same mutation rate. For example, two slow-evolving but taxonomically-distant species (possibly due to late maturation and low error rate in gene duplication) might be clustered together as sister taxa for their proximity to the common ancestors, even though they should be in 2 different branches. Neighbor-Joining (NJ), another greedy (looks for the shortest edge) hierarchical clustering algorithm, improves upon UPGMA by allowing for unequal mutation rates. It seeks to minimize the least-squares error in a distance matrix[2]. However, due to the limitation of time and project scope, we do not further research the NJ algorithm but choose the Max Parsimony as an improved algorithm to study.

Parsimony and Sankoff algorithm

1. Max Parsimony Problem.

"Maximum Parsimony is a character-based approach that infers a phylogenetic tree by minimizing the total number of evolutionary steps required to explain a given set of data assigned on the leaves"[7]. In order to use maximum parsimony to create phylogenetic trees (Algorithm 2), we need to find all possible tree types for a given number of leaves and then for each tree type, find the tree with the least mutation numbers, This is also known as large parsimony problem. Unfortunately, the large parsimony problem is NP-hard. Thus, we solve this problem step by step. For step 1 and step 2 in Max Parsimony (Algorithm 2), since we know that we cannot solve these problems efficiently (NP-hard), we simply find all possible tree topology and all permutations of given *n* species by brutal force. Then, step 3: finding the least mutation tree, also known as the small parsimony problem, can be solved efficiently by dynamic programming. Thus, we mainly focused on how to solve step 3.



```
Algorithm 2 Max Parsimony Algorithm
  INPUT: DNA sequences of n Species
  OUTPUT: Phylogenetic tree for given species with the least total mutations

  1. Final all possible binary tree topology with n leaves
  2. Final all possible permutation of the given n species

  3. For each species permutation i and each tree topology j :
  1) find the least mutation the tree j with leaves labelled by i required
  2) record the least mutation tree among all trees so far

  4. Return the tree topology that has the smallest number of mutation
```

Algorithm 2: The pseudo-code for Max Parsimony.

2. Small Parsimony and Sankoff's Algorithm

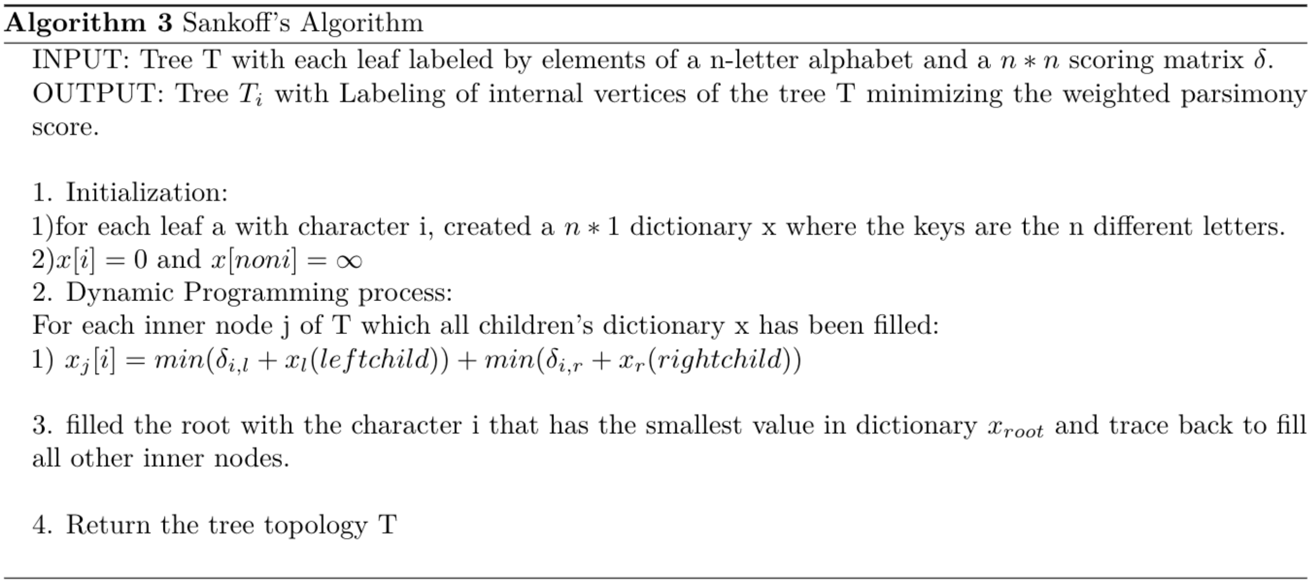
The small parsimony problem is defined as: Given a tree topology and all its leaves, find the minimum number of mutations the tree required. The mutation in the tree is defined as parsimony score *S(T)*:

$$S(T) = \sum\sum d_{v,w} \text{ if } v = w \; d_{v,w} = 0, \text{ if not } d_{v,w} = 1/\delta_{v,w}$$

Where *d* is the score/distance of two nodes *v,w* and *d(v,w)* is decided by whether the sequence on *v* and *w* are the same. And if the scoring matrix is given, the *d(v,w)* could change accordingly.

We implement Sankoff's Algorithm to solve the small parsimony problem. Sankoff's algorithm computes each subtree's smallest parsimony score from the bottom up to the root using dynamic programming. In Sankoff, we assign the whole tree's parsimony score to the root and thus, the inner node's parsimony score is the subtree that is rooted at that inner node's parsimony score. The high-level idea of Sankoff is that when finding one inner node's smallest parsimony score, the algorithm only refers to the children of this node and the parsimony score matrix stored in those children.

According to the definition of parsimony score *S(T)*, we can find that each node *k*'s parsimony score is only related to the node children's parsimony score. More specifically, the *S(k)* of any node *k*, is the sum of the parsimony score of the left child, the parsimony score of the right child, the *d(k,left)*, and the *d(k,right)*. Note: *d* is the distance between the root k and its child. Since the left child's parsimony score and the right child's parsimony score are independent, we can, when minimizing *S(k),* separately minimize the left child's score and right child's score. Thus, if we are given a scoring matrix delta that represents the weight of different characters to characters mutations, we turn min *S(k)* to:



Algorithm 3 Sankoff's Algorithm

```
INPUT: Tree T with each leaf labeled by elements of a n-letter alphabet and a $n * n$ scoring matrix $\delta$.
OUTPUT: Tree $T_i$ with Labeling of internal vertices of the tree T minimizing the weighted parsimony score.

1. Initialization:
1)for each leaf a with character i, created a $n * 1$ dictionary x where the keys are the n different letters.
2)$x[i] = 0$ and $x[noni] = \infty$
2. Dynamic Programming process:
For each inner node j of T which all children's dictionary x has been filled:
1) $x_j[i] = min(\delta_{i,l} + x_l(leftchild)) + min(\delta_{i,r} + x_r(rightchild))$

3. filled the root with the character i that has the smallest value in dictionary $x_{root}$ and trace back to fill all other inner nodes.

4. Return the tree topology T
```

Algorithm 3: The pseudo-code for Sankoff's Algorithm.

$$minS(k) = min(l_l(leftchild) + \delta_{l,k}) + min(l_r(rightchild) + \delta_{r,k})$$

According to Algorithm 3, Sankoff's algorithm takes in a scoring matrix for transforming characters to characters and a given tree topology with each leaf labeled by exactly one species sequence. Then for each node in the tree, a list *L* stores the minimum parsimony score of the subtree rooted at this node after filling the given characters separately. For each leaf, *L(leaf 's character) = 0* and *L(non-leaf's characters) = infinity*.

Finally, from bottom to up, each inner nodes' *L(x) = minS(k)*, where *x* is the character key of the list *L,* by referring to its childrens' *L* using the minimization equation mentioned above.

Snakoff's algorithm takes $O(smn^2)$ time to run once, where m is the sequence length, n is the number of different characters among all sequences, and *s* is the number of species. However, since the whole parsimony algorithm runs Sankoff's on all possible tree topologies and all possible permutations of the species, it runs Sankoff's algorithm for *s!*s!* times. Thus, if the number of species is large, the time complexity of parsimony is factorial time which as mentioned above is NP-hard. Thus, if the number of species is large, the runtime is unfeasible for modern computers.

Experiments and Results

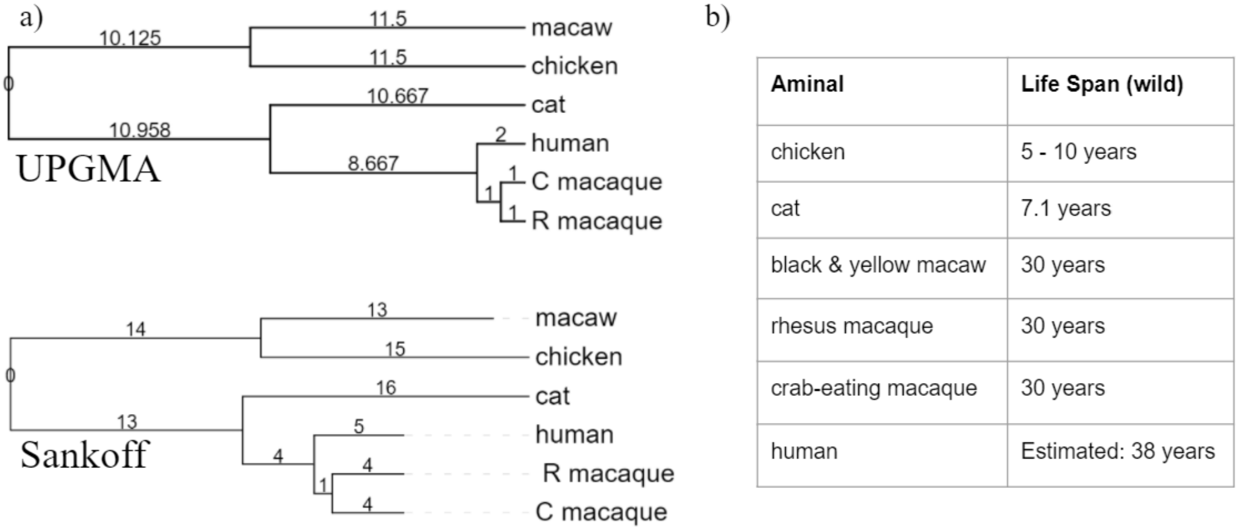
1. Dataset

For phylogenetic tree construction, we use the hemoglobin subunit α-a sequences in cats, humans, rhesus macaques, crab-eating macaques, macaws, and chicken from the UniProt database[8] with accession numbers 07405, 69905, 63108, 21767, 01196, 011994. The hemoglobin subunit α-a helps oxygen transportation from the lung to the body. We choose it because it is highly conserved and has exactly 142 characters for all species we choose. For the scoring matrix, we use the blossom62 matrix to score the similarity and differences between amino acids. We defined a gap score of -5, but our alignments of hemoglobin α-a sequences do not involve gaps.

2. Results and Comparisons

UPGMA and the parsimony approach report similar phylogenetic trees: humans and two types of macaques are close; cats are grouped with other mammals; birds (macaw and chicken) are in another clade (Figure 1a). However, two algorithms show differences in branch length: compared to UPGMA, Sankoff builds a tree where the primates have shorter distances to root.

We conjecture the difference in branch length between two trees is due to whether they represent the evolutionary model. As mentioned in the introduction, UPGMA does not take any evolution model into consideration. When clustering species, it assumes that all species have exactly the same evolution speed and creates an ultrametric tree. However, the parsimony algorithm utilizes the difference in sequence characters to build the parsimony score. Thus, it represents the evolution speed by assuming that the number of mutations has a positive relationship to evolutionary speed. Further, we validate this assumption by differences in generational time. Although the mutation rate for warmblood species is similar, the generational time influences the evolutionary speed and the appearance of new species significantly. Every new birth introduces a large number of germline mutations in a species. That is, species that have longer generational time tend to have slower evolution time and thus have a short distance to root. In Sankoff's phylogenetic tree, Chicken and cat have the longest distance from the root which means they should have short generation times, which are below 10 years (Figure 1b). Primates have shorter distances from the root because they have longer generational times, which are around 30 years (Figure 1b). Macaw is an outlier in the negative association between branch length and generation time, which has a generation time of 30 years but with branch length closer to chicken than to primates (Figure 1a). We conjecture that the outlier in this association is due to species-specific factors and the fact that we are only analyzing small sequences of 142 amino acids. Overall, we believe that generational time difference can explain the distribution of branch length in Sankoff's phylogenetic tree, though more future studies on more species and longer sequences are needed to verify this theory.



Aminal	Life Span (wild)
chicken	5 - 10 years
cat	7.1 years
black & yellow macaw	30 years
rhesus macaque	30 years
crab-eating macaque	30 years
human	Estimated: 38 years

Figure 1. (a) Phylogenetic trees created by UPGMA and Sankoff using hemoglobin subunit α-a. (b) The life spans of chicken, cat, black and yellow macaw, rhesus macaque, crab-eating macaque, and human.

Furthermore, although not shown in graphs, the running time of both algorithms have significant differences. Given a distance matrix, UPGMA takes $O(n^3)$ time to construct a phylogenetic tree, where *n* is the number of species. In our algorithm, generating a distances matrix using pairwise Needleman–Wunsch (global) alignment takes $O(n^2 *m^2)$ time, where *m* is the length of the sequences, and *n* is the number of species. Thus the total runtime of UPGMA in this paper is $O(n^3) + O(n^2 * m^2)$. Furthermore, if other algorithms are used to generate the pairwise distances, the generalized run time of UPGMA is:

$$O(n^3) + O(n^2 * [\textit{runtime of 1 pairwise alignment}])$$

In contrast to the polynomial runtime of UPGMA, getting the optimal tree through the Sankoff method, using a "generate tree -> calculate parsimony score with Sankoff -> find the best score" workflow, takes $O(n! * n! * m)$ time to explore all possible trees, where *n, m* denote the number of species and sequence length, respectively. Thus, the Sankoff parsimony approach would take an unacceptable large amount of time for a phylogenetic tree of many 8 species. In fact, only 8 species will have 660,032 types of rooted trees and 40,320 permutations of species. (26,612,490,240 runs of Sankoff's algorithm required!)[9].

Conclusion and Discussion

The UPGMA and Parsimony approach both provide a reasonable phylogenetic tree for the sequence of hemoglobin subunit α-a in species we examine. Phylogenetic trees built through both methods are meaningful in a biological context: primates are in a cluster; the cluster of mammals contains primates; macaws and chicken are in a non-mammalian cluster.

The ultrametric phylogenetic tree produced by UPGMA does not allow differential evolution speed among species, which can lead to possible inaccuracy of evolutionary relations. Sankoff addresses this problem by considering and minimizing the number of mutations, which implicitly allows for differential evolution speed. However, since large parsimony is an NP-hard problem, the running time is much longer than the polynomial UPGMA. Overall, UPGMA allows for a quick approximation of clustering of species, but the quantitative interpretation of the branch length might be unreliable. In contrast, the parsimony approach is slower especially when there are many species, but the branch lengths of its phylogenetic trees account for differential evolution rate and thus are more biologically meaningful. For instance, UPGMA can answer questions like “are red pandas closer to bears or raccoons”, but the parsimony approach can take a step further and estimate that “the most recent ancestor between red pandas and bears diverged around 30 million years ago”. Thus, when picking an algorithm, researchers should thoroughly consider the advantages and limitations of both algorithms, and choose one that best matches their research goal.

For further research, we want to first explore improvements for both algorithms: the parsimony approach can use the branch-and-bound method to find the best tree when there are less than 20 species, and heuristic methods can find a relatively optimal parsimony tree in a short time. The neighbor-joining algorithm is an improvement over UPGMA because it allows for different evolution rates but preserves the polynomial runtime. Besides 2 approaches explored in this paper, Maximum Likelihood and Bayesian methods are both good candidates for tree-building algorithms. Second, we will include more species, longer sequences, and more algorithms to explore the negative association between evolution rate and generation time.

Reference

1. Sokal, Reuven, R., Michener, D., C. & University of Kansas. *A statistical method for evaluating systematic relationships*. (University of Kansas, 1958).
2. Saitou, N. & Nei, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.* **4**, 406–425 (1987).
3. Sankoff, D. Minimal Mutation Trees of Sequences. *SIAM J. Appl. Math.* **28**, 35–42 (1975).
4. Lee, T.-H., Guo, H., Wang, X., Kim, C. & Paterson, A. H. SNPhylo: a pipeline to construct a phylogenetic tree from huge SNP data. *BMC Genomics* **15**, 162 (2014).
5. Cuadros, A. M., Paulovich, F. V., Minghim, R. & Telles, G. P. Point Placement by Phylogenetic Trees and its Application to Visual Analysis of Document Collections. in *2007 IEEE Symposium on Visual Analytics Science and Technology* 99–106 (2007). doi:10.1109/VAST.2007.4389002.
6. Côté, J. *et al.* Predictors and Evolution of Antiretroviral Therapy Adherence Among Perinatally HIV-Infected Adolescents in Brazil. *J. Adolesc. Health* **59**, 305–310 (2016).
7. Kannan, L. & Wheeler, W. C. Maximum Parsimony on Phylogenetic networks. *Algorithms Mol. Biol.* **7**, 9 (2012).
8. The UniProt Consortium *et al.* UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Res.* **49**, D480–D489 (2020).
9. Wooding, S. Inferring Phylogenies. *Am. J. Hum. Genet.* **74**, 1074 (2004).