BioE 131: Intro to Computational Biology Fall 2020

Phylogeny

Professor: Ian Holmes

Notes written by Vikram Shivakumar

1 Introduction

Often, multiple alignments are constructed with the ultimate goal of studying the relationships between sequences (and species), which we can visualize using **Phylogenetic trees**. In this note, we will explore methods for building and analyzing phylogenies, as well as the Jukes-Cantor model for estimating evolution distance.

2 fUn WiTh GrApHs

Phylogenetic trees (or dendograms, or cladograms), like all trees, are special types of **graphs**. Graphs are a set of **vertices** (or **nodes**), connected by **edges**. Edges can be **directed** or **undirected**, as well as labeled with weights or distances. The node **degree** is the number of neighbors of the node (for directed graphs, each node has an in-degree and out-degree). Graphs can be **connected**, if there is a *path* between any two vertices. They can also be

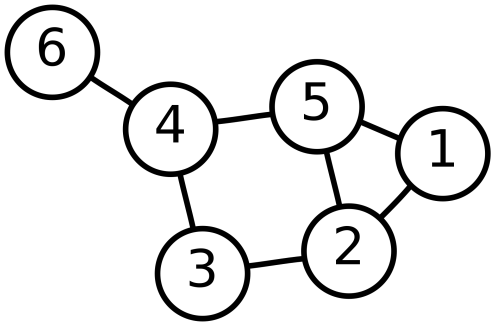

Figure 1: Example of an undirected cyclic graph with 6 vertices

complete, if there is an edge between every pair of vertices. Lastly, graphs can be **acyclic** if there are no cycles (a path from a vertex to itself).

A tree is a special type of graph that is connected and acyclic. Trees are also **minimally**

connected, with exactly $|V| - 1$ edges, where $|V|$ is the number of vertices. A **binary tree** is a tree where all nodes are internal (degree 3), leaves (degree 1), or the root (degree 2). Phylogenetic trees had edge labels (representing evolutionary distances), and node labels (gene or taxons). Lastly, a root node is specified, which implies directionality in the graph (edges are directed *away* from the root).

3 Phylogenetic Trees

Phylogenetic trees are often rooted such that the tree is directed. In phylogenies of taxons, the root represents a common ancestor of the leaf nodes, and in phylogenies of genes, the root node represents the **ancestral sequence**. **Outgroups**, species which are distantly related to the rest of the tree, can be included to root the tree. Often times the root can be ambiguous, e.g in the tree of life, there is no outgroup to determine the root.

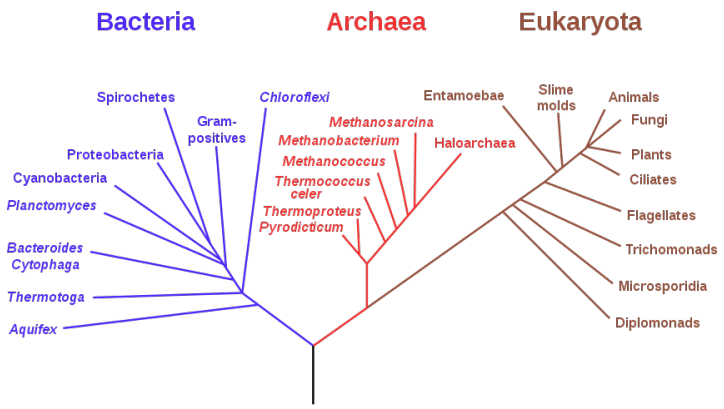

Figure 2: Tree of life with a possible root (between Bacteria and Archeae)

Lastly, trees can be **ultrametric**, where the leaf nodes are the same distance from the root. One example of an ultrametric tree is from the **coalescent process**, which models running time backwards in the Wright Fisher Model (see the note on Probability). Non-ultrametric trees can result from data where the leaf nodes are not contemporaneous (e.g. sequencing ancient DNA). Branch lengths can also vary between contemporaneous taxa, where factors like metabolic rate and mutation rate vary.

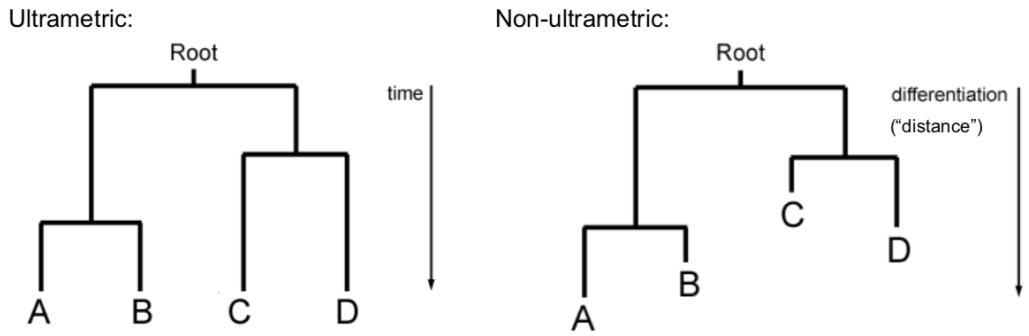

Figure 3: Ultrametric vs Non-ultrametric trees

4 Algorithms for Phylogenetic Reconstruction

Now let's look at a few methods to construct phylogenetic trees from multiple alignments. In general, the substitutions (and in some cases the indels) in the MSA are used to build and evaluate a tree.

4.1 Parsimony

Parsimony trees group taxa such that the number of substitutions is minimized. These trees are the simplest type of phylogenetic trees, and can be solved using various optimization methods (like the **branch-and-bound** method). However, this method ignores different types of substitutions, treating them all the same. It also does not account for **back-substitutions**, where a mutation reverts a previous mutation (and the final nucleotide appears unchanged).

4.2 Distance Matrix

We can also construct a phylogeny (or any distance tree) from a **distance matrix**, which contains the "distance" between each taxon or gene. This works well if distances are **additive** (not the case with back-substitutions!). This method can be a quick approximation for likelihood methods (which we will explore later), but can be prone to certain types of error.

One type of error is **long branch attraction**. When two species are on long branches in a phylogenetic tree, there can be chance similarities due to a long time to accumulate sequence or morphological changes. This can cause distantly related species to appear more related!

4.2.1 UPGMA algorithm

Using a multiple alignment, we can build a matrix of pairwise distances, and construct an ultrametric tree using the **UPGMA** algorithm.

The general idea of the algorithm is:

1. Pick the closest two nodes, and group them under an ancestral node
2. The distance between **ancestral nodes** is the average over the distances between all descendants
3. Repeat until all nodes are included

We can describe the UPGMA algorithm in pseudo-code:

Algorithm 1 UPGMA algorithm

Input: Distance matrix, D_{ij}
Let N be a set of nodes
Let $C(i)$ be the set of descendants of node i
for nodes $\in N$ **do**
  $C(i) \leftarrow \{i\}$
end for
while N contains nodes **do**
  $(i, j) \leftarrow \arg\min_{i,j} D_{i,j}$
  Create node k
  $C(k) \leftarrow C(i) \bigcup C(j)$
  for nodes $\in N$ **do**
    $D_{kn} \leftarrow \langle D_{xy} \rangle_{x \in C(k), y \in C(n)}$
  end for
end while

4.2.2 Runtime Complexity of UPGMA

Looking at the pseudocode, the initialization takes $O(N-1)$ steps. The *while* loop runs in $O(N)$ time, and in each loop, we find the smallest item in the matrix D, which naively takes $O(N^2)$ time (though this can be reduced by storing the smallest value). Thus the while loop takes $O(N^3)$ time, which is the overall runtime of the algorithm. The memory complexity is simply $O(N^2)$, since the algorithm stores the distance matrix in memory.

4.3 Phylogenetic Likelihood

Phylogenetic likelihood methods for tree construction find the tree with the most likely substitutions. These trees are more realistic than those from distance matrix or parsimony approaches, but they can be much slower, as they evaluate the likelihood of many tree topologies.

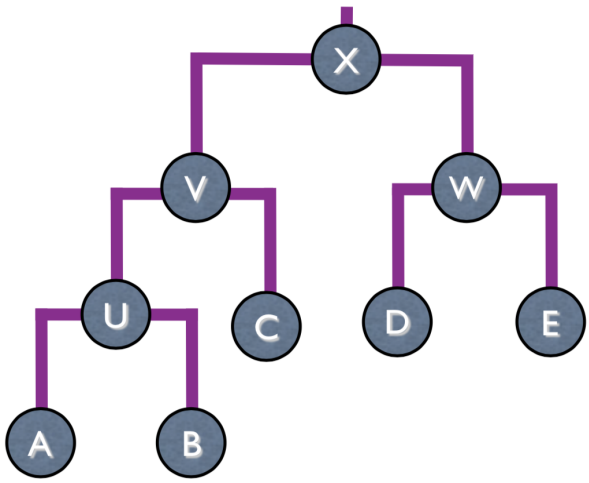

Figure 4: Example of a phylogenetic tree

4.3.1 Likelihood

We can calculate the likelihood of a tree using the probability distribution over the root $P(x)$, and the conditional distribution of a node, $P(w|x)$, the probability of a node given its parent. Thus for the tree in Figure 4, the likelihood would be:

$$L_0 = P(x) \cdot P(v|x) \cdot P(u|v) \cdot P(a|u) \cdot P(b|u) \cdot P(c|v) \cdot P(w|x) \cdot P(d|w) \cdot P(e|w)$$

However, we don't knnow the ancestral states (in this case nodes u, v, w, and x). Thus we can sum over all possible states of these nodes.

$$L = \sum_u \sum_v \sum_w \sum_x L_0$$

Since there are order $O(N)$ internal nodes in a tree, iterating through all possible states for all ancestral nodes would take $O(A^N)$ times (A is the alphabet size, e.g. 4 for nucleotide sequences).

We can use dynamic programming to reduce the runtime to $O(A^2N)$! The idea is to compute

the likehood of each subtree, and recursively calculate the final likelihood. This is equivalent to rearranging the likelihood sum:

$$\sum_x P(x) \left(\sum_v P(v|x) \left(\sum_u P(u|v) \cdot P(a|u) \cdot P(b|u) \right) P(x|v) \right) \left(\sum_w P(w|x) \cdot P(d|w) \cdot P(e|w) \right)$$

Now, each sum represents sum in parentheses represents a partial solution $P(\text{subtree}|\text{root})$.

4.3.2 Confidence estimates

Often phylogenies include the confidence of branches, which can be determined directly from likelihood methods (like MCMC sampling), or by **bootstrapping**. Bootstrapping involves sampling a random set of columns from the multiple alignment (*with* replacement), and building a tree from just that subset of data. We can repeat this many times, and find the percent of trees which include a certain branch in their topology.

4.3.3 Other methods

Other algorithms have been developed to build phylogenies from multiple alignments, which approximate likelihood methods. **Neighbor-joining** is an algorithm that extends the ideas from UPGMA, but allows for siblings to be non-equidistant from the parent. Thus, neighbor-joining methods can also produce non-ultrametric trees. **Weighted neighbor-joining** improves on normal neighbor-joining by correcting for long-branch estimation error. **Quartet-puzzling** algorithms look at sets of 4 nodes in the tree, and finds the best arrangement for each local set of 4, as opposed to comparing pairs of nodes

Lastly, **MCMC sampling** is an important algorithm for approximating the likelihood of a tree. This method stochastically generates trees from the underlying probability distribution of trees, which (after enough trees have been generated) can be used to calculate the probabilities of each tree. This method is slow, but the longer it runs, the more accurate an approximation to the maximum likelihood tree it can provide.

5 Jukes Cantor Model

The Jukes Cantor model was developed at Berkeley in 1969 as a method to estimate evolutionary distances between sequences. We'll derive the equation for the distance estimate in this section.

Let's assume in a sequence, there are randomly timed replacement events, where a nucleotide is replaced by any of the 4 nucleotides with uniform probability (so there is a 1/4 probability

of the nucleotide remaining the same). We can model these replacements with a **Poisson** distribution, with a mean of RT, where R is the rate of replacement events, and T is time. *Note*: the rate of *substitutions* is $\lambda = \frac{3}{4}R$.

Now let $X(t)$ be the state of the process at time t, and $Q(t)$ be the probability that there are *no replacements* from time 0 to t. Since R is the rate of replacement events, we can derive an expression for $Q(t)$:

$$\frac{dQ}{dt} = -RQ \tag{1}$$

Since $Q(0) = 1$, we can solve the differential equation:

$$Q(t) = \exp(-Rt) \tag{2}$$

Now let's find the probability that the state of the process at time t is the same as the initial state, i.e. $P\left(X\left(t\right) = X\left(0\right)\right)$. If the state of the process remains the same, then there were EITHER no replacement events in time t OR the replacements did not change the nucleotide (with probability 1/4):

$$P\left(X\left(t\right) = X\left(0\right)\right) = Q\left(t\right) + \frac{1 - Q\left(t\right)}{4} \tag{3}$$

We can further simplify this equation:

$$P\left(X\left(t\right) = X\left(0\right)\right) = \frac{1}{4}\left(1 + 3\exp\left(-Rt\right)\right) \tag{4}$$

Now let's look at a pairwise alignment of two sequences with length L. Assume the sequences are separated by **evolutionary distance** t. If we assume that each nucleotide is independent, then we can find the **expected number of matches** M (same nucleotide in both sequences at a position):

$$M = L \times \frac{1}{4}\left(1 + 3\exp\left(-Rt\right)\right) \tag{5}$$

But we are trying to find an equation to estimate the evolutionary distance t! Let's rearrange equation (5) to find t:

$$t = -\frac{1}{R}\log\left(\frac{1}{3}\left(4\frac{M}{L} - 1\right)\right) \tag{6}$$

Lastly, we can calibrate time such that the rate of substitutions is 1, i.e. $\lambda = 1$, which implies that the rate of replacement $R = \frac{4}{3}$. We can plug in this rate to derive the full Jukes-Cantor distance estimate:

$$t = -\frac{3}{4}\log\left(\frac{4}{3}\frac{M}{L} - \frac{1}{3}\right) \tag{7}$$

We can also rewrite this equation in terms of the number of *mismatches* instead of matches, $q = 1 - M/L$:

$$t = -\frac{3}{4}\log\left(1 - \frac{4}{3}q\right) \tag{8}$$

6 Virus design

One problem in bioinformatics is designing therapeutic viruses for various purposes like delivering genetic material or lysing specific cells. These designed viruses need to have a few properties, like safety (not involved in disease), stability, and ease of transformation. Another property that is important, related to evolution, is the rate of **nonsynonymous mutations**. These mutations change the amino acid sequence of the expressed protein, unlike **synonymous mutations**, which are silent, and cause no change in the overall protein sequence. One common metric to calculate is the **Ka/Ks ratio**, or the dN/ds ratio. This is the ratio of nonsynonymous to synonymous mutations, and can reveal information about the selective pressures driving evolution of the gene (or in this case, viral genome). There are ranges of values for the ratio:

1. Ka/Ks > 1: **diversifying selection**, the case in pathogens undergoing immune system evasion
2. Ka/Ks ≈ 1: **neutral selection**
3. Ka/Ks < 1: **purifying selection**, such as in the case of housekeeping genes

By measuring the Ka/Ks ratio of a virus, we can study the selective evolutionary pressures that a virus is undergoing, and better understand how viruses evolve.

7 Summary

Phylogenetic trees are an important tool to understand the evolutionary relationships between taxa or sequences. We can used properties of graphs to study phylogenies, and reconstruct trees using various methods. We can also use probabilistic models like the Jukes-Cantor model to estimate the evolutionary distance between sequences. Lastly, evolutionary models can be useful in the study of viruses, which undergo selective pressures that drive genetic change.