Phylogenetics: Finding genetic connections & relationships between species. One by comparing specific features — similar species should be genetically close. Phylogeny refers to these relationships in the form of a phylogenetic tree.

Classic phylogenetics dealt with physical/morphological features: size, color etc. We now have DNA and protein sequences: One can align sequences and use "conserved blocks.

One interesting research used phylogenetics to trace the origins of humans via mitochondrial DNA — which is copied from mother to child without recombining with father’s DNA. It was claimed out of phylogenetic analysis that all humans have a common female ancestor from Africa— but it is also claimed that Out-of-Asia/Europe/Africa hypotheses are as plausible.

One problem with sequence based phylogenetics is evolution results in duplications. Copies of genes evolve separately resulting in two or more similar instances of the same gene. When matching genes we must make distinction between 1) homologous matches — both genes are the same in the strong sense and are connected directly, 2) paralogous matches — result of a copy event, and 3) xenologous matches — result of other transfers such as via viruses.

We will first assume there are no paralogues in organisms and later discuss implications of duplicate genes.
Definitions:
Species refer to objects such as genes of different species or paralogous genes on the same genome.
Sites refer to characters which are assigned a position in an alignment.
Phylogenetic trees refer to trees in which each leaf is labeled with the name of one input species and edges represent genetic connections. Internal nodes represent hypothetical species—evolutionary ancestors of species in consideration. Best way to visualize phylogenetic relationships.

Phylogenetic Tree Construction:
Types: 
- Rooted - unrooted
- Binary - general
- Edge lengths - no edge lengths

Data used:
1) Distance matrix (alignment score: pairwise or multiple)
2) Independent characters (each site examined separately)

Character Based Methods:
Input: A set of species and m characters for each species
Output: Fully labeled phylogenetic tree that best explains data via maximizing a target function.
Let $M_{ij}$ represent the $j^{th}$ character of $i^{th}$ species. Assume characters are mutually independent — change in one character has no effect on others. Also, assume once two species diverge in evolution on one character, those characters evolve independently. These assumptions are not necessarily correct but they make life easier.

**Trivial solution**: Enumerate all possible trees to calculate the target function for each one.

**Problem**: Number of non-isomorphic, binary, rooted trees with $n$ leaves are $\frac{1}{2}T(2^{2n-3})$.

For $n=20$, number of $\binom{n}{2}$ such trees are $10^{21}$.

**Parsimony**: An intuitive score for phylogenetic trees is number of changes along edges. The approach of minimizing this score is called parsimony. We're looking for the simplest — minimum number of mutations — explanation for evolution.

Let the vertices of a tree be $V(T)$ and edges be $E(T)$. Let the $j^{th}$ character of vertex $v$ be $v_j$. Then the parsimony score of tree $T$ becomes:

$$S(T) = \sum_{v \in V(T)} \sum_{j=1}^{3} |v_j \neq v_j^{\prime}|$$
Example: Consider the following one-character sequences: C, C, C, T, T. The most parsimonious binary tree for these species are:

- **Total score: 1**

Another tree is less parsimonious:

- **Total score: 2**

Small Parsimony: Given the topology of a rooted phylogenetic tree with labeled leaves:

- What is 1) minimum number of changes on characters?
- 2) optimal internal labeling?

Fitch's method: Under mutual in dependence assumption solve the problem separately for each site.

1) Assign each node $v$ a set $S_v \subseteq \{A, C, T\}$: character of $v$

For each leaf $v$: $S_v = \emptyset$; 3) character of $v$

For each internal node $v$: $S_v = \bigcup S_w$ (if non-empty $S_w \cup S_w$)
2) Given the sets $S_v$, traverse $T$ in preorder fashion; from the root up.
For each internal node $v$, if its parent $u$ satisfies $v \in S_u$, set $c_v \leftarrow c_u$.
Otherwise arbitrarily assign any $t \in S_v$ to $c_v$.

**Example:**

```
          A
         / \  \
        C   G
       / \  /  \
      C   C   C
     / \ / \  / \  /  \
    E   E   E   E   E
```

**Step 1:**

- $c_v = \{A, C\}$
- $c_u = \{C, E\}$
- $t = E$

**Step 2:**

- $c_v = \{C\}$
- $c_u = \{C\}$

**Complexity:**

- $O(k)$ time to compute $S_v$, $O(k)$
- time to compute $c_v$ ($k$ = size of alphabet)
- Do it for each character for each internal node: $O(n \cdot m \cdot k)$
Distance Based Methods:

Without a tree and ancestral sequences, it is not possible to find out the specific edit operations/mutations that diverged two evolutionarily related sequences. However we can assume that some minimum weighted sum of edit penalties well approximate the actual "distance" between two sequences and such distance can be used to build an evolutionary tree.

Given a distance \( d \) on each pair of sequences \( x, y \), the goal of a distance-based phylogenetic tree construction method is to compute the tree with minimum total distance between all pairs of sequences. Here the distance between \( x, y \) on the tree is the sum of the "weights" of all edges on the path between \( x, y \). The distance \( d \) could be the "% divergence" between the sequences under their optimal alignment etc.

**UPGMA Heuristic:** Unweighted Pair Group Method with Arithmetic mean.

1. Initially each sequence \( s_i \) forms a set \( \text{Ci} = \{s_i\} \).
2. Given two sets of sequences \( \text{Ci}, \text{Cj} \) the distance between them \( \text{Di,j} \) can be defined as:

\[
\text{Di,j} = \frac{1}{|\text{Ci}| + |\text{Cj}|} \sum_{p \in \text{Ci}} \sum_{q \in \text{Cj}} d(p,q)
\]
1. Assign a leaf to each singleton set $C_i = \{s_i\}$.
2. Find $i, j$ that has minimum $D_{ij}$.
   Create a new set $C_e = C_i \cup C_j$.
   Create a new node for $C_e$ which becomes the parent of $C_i, C_j$. The weight of the edges between $C_e$ and $C_i, C_j$ becomes $D_{ij}/2$.
3. Compute the distance $D_{ek}$ between $C_e$ and every other set $C_k$:
   $$D_{ek} = \frac{|C_i|}{|C_i| + |C_j|} D_{ik} + \frac{|C_j|}{|C_i| + |C_j|} D_{jk}$$
4. Remove $C_i, C_j$ from the list of sets.
   If the number of sets $\geq 2$ GOTO step 2.

**Running Time:** $O(n^2)$

**Neighbor-Joining - NJ:**

1. For each $s_i$ compute $u_i = \sum_{k \neq i} \frac{D_{ik}}{n-2}$
2. Choose $i, j$ pair for which $D_{ij} - u_i - u_j$ is smallest.
3. Join $C_i, C_j$ to form $C_e$. Compute $d_{ie}, d_{ej}$
   $$D_{ek} = \frac{D_{ij}}{2} + \frac{(u_i - u_j)}{2}$$
4. Delete $C_i, C_j$ from the set list.
   Create a new parent node for $C_i, C_j$.
5. If the remaining sets $\geq 2$ GOTO 2nd step.

**Alternative to NJ:**

1. For each $s_i$ compute $u_i = \sum_{k \neq i} \frac{D_{ik}}{n-2}$
2. Choose $i, j$ pair for which $D_{ij} - u_i - u_j$ is smallest.
3. Join $C_i, C_j$ to form $C_e$. Compute $d_{ie}, d_{ej}$
   $$D_{ek} = \frac{D_{ij}}{2} + \frac{(u_i - u_j)}{2}$$
4. Delete $C_i, C_j$ from the set list.
   Create a new parent node for $C_i, C_j$.
5. If the remaining sets $\geq 2$ GOTO 2nd step.