Distance-based methods for phylogenetic tree reconstruction

Colin Dewey
BMI/CS 576
www.biostat.wisc.edu/bmi576/
colin.dewey@wisc.edu
Fall 2015
Basic idea of distance-based methods

• Suppose we can compute a “distance”, $d_{ij}$, between each pair of taxa based on some data (e.g., sequences)
• Can we come up with a tree structure (with lengths assigned to branches) that accurately reflect the pairwise distances?

$$d_{15} = L_{16} + L_{68} + L_{58}$$
Distance-based methods for phylogenetic tree reconstruction

- Given $n \times n$ distance matrix for $n$ units, construct the tree for these $n$ units
- Algorithms
  - UPGMA
  - Neighbor joining
- Assume additivity and sometimes a “molecular clock”
- Additivity means we can add up the branch lengths of the tree connecting two nodes and get their distances
  - In other words, “additivity” of the distances means that there exists some tree that perfectly explains these distances
Defining distance between sequences

• Fractional alignment mismatch for two sequences $i$ and $j$
  
  $- p_{ij} = m_{ij}/L_{ij}$
  
  • Gives an estimate of changes per site
  
  $- m_{ij}$: Number of mismatches between sequences $i$ and $j$
  
  $- L_{ij}$: Number of aligned positions between sequences $i$ and $j$
  
  • Assumes that changes have happened only once
    
    • Underestimates the distance between sequences

• Jukes Cantor distance

  $-$ Removes assumption above

  $- The simplest evolutionary distance $d_{ij}$ between sequences $i$ and $j$, where $p_{ij}$ is the fractional mismatch defined above

  $$d_{ij} = - \frac{3}{4} \ln(1 - \frac{4}{3} p_{ij})$$
UPGMA algorithm for phylogenetic tree reconstruction

- UPGMA: Unweighted pair group method using arithmetic averages
- Represent all sequences as the leaf nodes of a tree
- Start with just the leaf nodes
- At each step, merge two closest nodes to create a new node in the tree
  - Set new node at height determined by nodes being merged
  - Recompute distance between new node and all other nodes
- Intermediate nodes will correspond to a set of sequences
- We will call sequences associated with an intermediate node $i$ cluster $C_i$
- Need to compute
  - Distance between two clusters of sequences
  - Height
Computing distance between clusters

- Let $i$ and $j$ be two nodes
- Let $C_i$ be the cluster of sequences for node $i$
- Let $C_j$ be the cluster of sequences for node $j$
- $|C_j|$: Number of sequences in $C_j$
- Distance between nodes $i$ and $j$

$$d_{ij} = \frac{1}{|C_i||C_j|} \sum_{p \in C_i, q \in C_j} d_{pq}$$
Computing distance from a new node

- Let $k$ be a new node to be created from merging $i$ and $j$
- Let $C_i$ be the cluster of sequences for node $i$
- Let $C_j$ be the cluster of sequences for node $j$
- Distance $d_{kl}$ between nodes $k$ and $l$, $l \neq i$ and $l \neq j$

\[
d_{kl} = \frac{1}{|C_k||C_l|} \sum_{p \in C_k, q \in C_l} d_{pq}
\]

- This is equal to

\[
d_{kl} = \frac{d_{il}|C_i| + d_{jl}|C_j|}{|C_i| + |C_j|}
\]
UPGMA algorithm

• Input
  – $n$ sequences
  – Distance matrix for all pairs of $n$ sequences, $d_{ij}$

• Output
  – Tree $T$

• Initialization
  – Assign each sequence $i$ to its own cluster $C_i$
  – Define one leaf of $T$ for each sequence

• Iterate until only two clusters remain
  – Find two nodes $C_i$ and $C_j$ that have the smallest $d_{ij}$
  – Define new cluster $C_k = C_i \cup C_j$
  – Define daughters of $k$ as $i$ and $j$, place at height $d_{ij}/2$
  – Add $k$ to cluster set. Remove $i$ and $j$ from the set of clusters

• Terminate
  – When only two clusters $C_i$ and $C_j$ remain, place root at $d_{ij}/2$
UPGMA example

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>3</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Initial state

<table>
<thead>
<tr>
<th></th>
<th>AE</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>3</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

After one merge

Example calculation

\[
d_{(A,E)B} = \frac{d_{AB} + d_{EB}}{1 + 1} = \frac{16}{2}
\]
UPGMA example (cont.)

after two merges

<table>
<thead>
<tr>
<th></th>
<th>AE</th>
<th>BC</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>0</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>BC</td>
<td>0</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

after three merges

<table>
<thead>
<tr>
<th></th>
<th>AED</th>
<th>BC</th>
</tr>
</thead>
<tbody>
<tr>
<td>AED</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>BC</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

final state
UPGMA relies on the molecular clock assumption

- Sequences diverge at the same rate at all points in the phylogeny
- Distance from any leaf to root is the same.
- If this is true the distances are said to have an “ultrametric” property
- This assumption is rarely true in practice
The molecular clock assumption & ultrametric data

- Ultrametric data: for any triplet of sequences, $i, j, k$, the distances are either all equal, or two are equal and the remaining one is smaller.
Problem with UPGMA when the molecular clock assumption does not hold

Actual tree

![Actual tree](image1)

Constructed by UPGMA

![Constructed by UPGMA](image2)
Neighbor joining

- The assumption about the ultra-metric property is too strong
  - Most sequences diverge at different rates
- A more relaxed requirement is that of additivity
  - Distance between a pair of species/nodes is equal to the sum of the branch lengths
- Neighbor Joining (NJ) Uses a similar idea to construct trees as UPGMA
  - That is, consider pairs of nodes and joins them
- Produces unrooted trees
How to select nodes for joining?

- Given all pairwise distances for $n$ sequences
- $d_{ij}$ denote the distance between node $i$ and $j$
- Should we select node pairs with the smallest $d_{ij}$?

This will give us an incorrect tree
Selecting nodes to join

- Neighbor joining requires us to correct the distance to account for distances from all other nodes.
- The corrected distance is denoted as $D_{ij}$

$$D_{ij} = d_{ij} - (r_i + r_j)$$

$$r_i = \frac{1}{L-2} \sum_{1 \leq k \leq L} d_{ik}$$

$L$: number of leaves

$r_i$: “Average” distance from all other leaves
Defining the distance to a new node

Given \(d_{ij}, d_{im}, d_{jm}\), how to calculate distance of existing node \(m\) to new node \(k\)?

\[
d_{km} = \frac{d_{im} + d_{jm} - d_{ij}}{2}
\]
Updating Distances in Neighbor Joining

- can calculate the distance from a leaf to its parent node in the same way

\[
d_{ik} = \frac{1}{2} (d_{ij} + d_{im} - d_{jm})
\]

\[
d_{jk} = d_{ij} - d_{ik}
\]
Updating Distances in Neighbor Joining

- we can generalize this so that we take into account the distance to all other leaves

\[ d_{ik} = \frac{1}{2} (d_{ij} + r_i - r_j) \]

where

\[ r_i = \frac{1}{|L| - 2} \sum_{m \in L} d_{im} \]

and \( L \) is the set of leaves

- this is more robust if data aren’t strictly additive
Algorithm for NJ

• Initialization
  – $T$ be set the of leaf nodes
  – $L = T$
  – Compute $r_i$ for all $i$ in $L$
  – Compute $D_{ij}$

• Iteration
  – Pick a pair $i, j$ from $L$ such that $D_{ij}$ is smallest
  – Define new node $k$
  – Compute $d_{ik}, d_{jk}$, add edge between $k$ and $i$, and between $k$ to $j$
  – Add $k$ to $L$, remove $i$ and $j$ from $L$
  – Compute $D_{mn}$ for all nodes $m, n$ in $L$

• Terminate
  – If $L$ has two nodes, add the edge between these two.
Can we check for additivity?

Check for additivity: For four leaves, \(i, j, k, l\) and the distances \(d_{ij}, d_{ik}, d_{il}, d_{jk}, d_{jl}, d_{kl}\)

The three sums of two distances

Should be such that two of these are equal, and larger than the third.
Comparing NJ and UPGMA

- **UPGMA**
  - Rooted tree
  - Assumptions: Molecular clock assumption/ultrametric distance and additivity

- **NJ**
  - Unrooted tree
  - Assumption: Additivity
Rooting a tree

- An unrooted tree can be converted to a rooted tree using an outgroup species.
- Outgroup: a species known to be more distantly related to all the species than each of the species themselves.
- Find the branch where the outgroup is selected to be added.
- That gives the root.

![Diagram of a rooted tree with an outgroup.]