Phylogeny
Nov 29, 2016

- Slides:
  - Adi Akavia
  - Nir Friedman’s slides at HUJI (based on ALGMB 98)
  - Anders Gorm Pedersen, Technical University of Denmark
- Sources: Joe Felsenstein “Inferring Phylogenies” (2004)
Phylogeny

- **Phylogeny**: the ancestral relationship of a set of species.
- Represented by a **phylogenetic tree**

- **Leaves** - contemporary
- **Internal nodes** - ancestral
- **Branch length** - distance between sequences
Classical vs. Modern Phylogeny schools

- Classical – morphological characters
- Modern – molecular sequences.
Page from Darwin's notebooks around July 1837 showing the first-known sketch by Charles Darwin of an evolutionary tree describing the relationships among groups of organisms.
Trees and Models

- rooted / unrooted
- topology / distance
- binary / general
To root or not to root?

- **Unrooted** tree: phylogeny without direction.
Rooting an Unrooted Tree

- We can estimate the position of the root by introducing an **outgroup**: a species that is definitely most distant from all the species of interest.

Proposed root

- Aardvark
- Bison
- Chimp
- Dog
- Elephant
- **Falcon**
HOW DO WE FIGURE OUT THESE TREES? TIMES?
Dangers of Paralogs

- Right species topology: (1,(2,3))

Sequence Homology Caused By:
- Orthologs - speciation,
- Paralogs - duplication
- Xenologs - horizontal (e.g., by virus)
Dangers of Paralogs

- Right species distance: (1,(2,3))
- If we only consider 1A, 2B, and 3A: ((1,3),2)
Type of Data

• **Distance-based**
  - Input: matrix of distances between species
  - Distance can be
    • fraction of residue they disagree on,
    • alignment score between them,
    • ...

• **Character-based**
  - Examine each character (e.g., residue) separately
Distance Based Methods
Tree based distances

- \( d(i,j) = \text{sum of arc lengths on the path } i \xleftrightarrow{} j \)

- Given \( d \), can we find
  - an exactly matching tree?
  - An approximately matching tree?
The Problem

The least squares criteria

Input: matrix \( d \) of distances between species

Goal: Find a tree with leaves=chars and edge distances, matching \( d \) best.

Quality measure: sum of squares:

\[
SSQ(T) = \sum_{i} \sum_{j \neq i} w_{ij} (d_{ij} - t_{ij})^2
\]

\( t_{ij} \): distance in the tree

\( w_{ij} \): pair weighting. Options: (1) \( \equiv 1 \) (2) \( 1/d_{ij} \) (3) \( 1/d_{ij}^2 \)

NP-hard (Day '86). We'll describe common heuristics
UPGMA Clustering (Sokal & Michener 1958)
(Unweighted pair-group method with arithmetic mean)

• Approach: Form a tree; closer species according to input distances should be closer in the tree
• Build the tree bottom up, each time merging two smaller trees
• All leaves are at same distance from the root
UPGMA Algorithm
UPGMA Algorithm
UPGMA Algorithm
UPGMA Algorithm
UPGMA Algorithm
Efficiency lemma

• Approach: gradually form clusters: sets of species
• Repeatedly identify two clusters and merge them.
• For clusters $C_i$, $C_j$, define the distance between them to be the average dist betw their members:

$$d(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{p \in C_i} \sum_{q \in C_j} d(p, q)$$

• Lemma: If $C_k$ is formed by merging $C_i$ and $C_j$ then for every other cluster $C_l$

$$d(C_k, C_l) = (|C_i| \cdot d(C_i, C_l) + |C_j| \cdot d(C_j, C_l)) / (|C_i| + |C_j|) \quad \text{(ex.)}$$

→ Can update distances between clusters in time prop. to the number of clusters.
UPGMA algorithm

**Initialize:** each node is a cluster $C_i=\{i\}$. $d(C_i,C_j)=d(i,j)$ set $\text{height}(i)=0 \ \forall i$

**Iterate:**
- Find $C_i,C_j$ with smallest $d(C_i,C_j)$
- Introduce a new cluster node $C_k$ that replaces $C_i$ and $C_j$
  // $C_k$ represents all the leaves in clusters $C_i$ and $C_j$
- Introduce a new tree node $A_{ij}$ with $\text{height}(A_{ij}) = d(C_i,C_j)/2$
  // $d(C_i,C_j)$ is the average dist among leaves of $C_i$ and $C_j$
- Connect the corresponding tree nodes $C_i,C_j$ to $A_{ij}$ with
  $\text{length}(C_i,A_{ij}) = \text{height}(A_{ij}) - \text{height}(C_i)$
  $\text{length}(C_j,A_{ij}) = \text{height}(A_{ij}) - \text{height}(C_j)$
- For all other $C_l$:
  $d(C_k,C_l) = (|C_i|*d(C_k,C_l) + |C_j|*d(C_j,C_l)) / (|C_i| + |C_j|)$
  // dist to any old cluster is the ave dist between its leaves and leaves in $C_i, C_j$

**Time:** Naïve: $O(n^3)$; Can show $O(n^2 \log n)$ (ex.) and $O(n^2)$ (harder ex.)
UPGMA alg (2)

- Orange nodes represent the groups of nodes that they replaced, and maintain the average dist of the set from other leaf nodes/clusters.
Molecular Clock

• UPGMA assumes the tree has equal leaf-root non-negative distances \(\Rightarrow\) common uniform clock. Such a tree is called ultrametric*

• Thm: If the distance matrix \(D\) corresponds to an ultrametric, then applying UPGMA to \(D\) provides a solution that is an ultrametric tree, and the tree is unique up to isomorphism

• \(\Rightarrow\) UPGMA gives the correct solution on such perfect data.

*or a particular type of ultrametric, according to others.
..and in real life?

UPGMA works reasonably well for nearby species or near-clock-like matrices, but can fail in other situations.

A more flexible model is needed in such cases.
Additivity

• An additivity assumption: distances between species are the sum of distances between intermediate nodes, i.e. the sum of edge length in the path connecting them.

• Holds for ultrametric trees - and more generally

\[ d(i, j) = a + b \]
\[ d(i, k) = a + c \]
\[ d(j, k) = b + c \]

If the distance matrix is an exact reflection of a true tree, then additivity holds.
Consequences of Additivity

• Suppose input distances are additive
• For any three leaves

\[ d(i, j) = a + b \]
\[ d(i, m) = a + c \]
\[ d(j, m) = b + c \]

• Thus

\[ d(i, k) = \frac{1}{2} (d(i, m) + d(i, j) - d(m, j)) \]
\[ d(m, k) = \frac{1}{2} (d(i, m) + d(j, m) - d(i, j)) \]
Consequences of Additivity II

- If we can identify neighbor leaves, then we can use pairwise distances to reconstruct the tree:
  - Remove neighbors i, j from the leaf set
  - Add k
  - Set $d_{km} = (d_{im} + d_{jm} - d_{ij})/2$
    
    $d_{ik} = d_{im} - d_{km} = (d_{im} - d_{jm} + d_{ij})/2$

How do we find neighbor leaves?
Problem: The closest pair of nodes may not be neighbors!

- Closest pair: k and j
Neighbor Joining (Saitou-Nei ’87)

• Let

\[ D(i, j) = d(i, j) - (r_i + r_j) \]

where

\[ r_i = \frac{1}{|L| - 2} \sum_k d(i, k) \]  

“Corrected” average distance of i from all other nodes

**Theorem:** if \( D(i, j) \) is minimum among all pairs of leaves, then i and j are neighbors in the tree.
Neighbor Joining algorithm

- Set $L$ to contain all leaves

Iteration:
- Choose $i,j$ such that $D(i,j)$ is minimum
- Create new node $k$, and set
  \[ d(i,k) = \left( d(i,j) + r_i - r_j \right) / 2 \]
  \[ d(j,k) = \left( d(i,j) + r_j - r_i \right) / 2 \]
  \[ d(k,m) = \left( d(i,m) + d(j,m) - d(i,j) \right) / 2 \]
- remove $i,j$ from $L$, and add $k$
- Update $r$, $D$
- Termination: when $|L| = 2$ connect the two nodes

Thm: Opt tree guaranteed if distances match a tree

Time: $O(n^3)$

Ex.
An example

True tree

Distance matrix

UPGMA tree
Neighbor Joining Algorithm

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>17</td>
<td>21</td>
<td>27</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>12</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>-</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
## Neighbor Joining Algorithm

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>17</td>
<td>21</td>
<td>27</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>12</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[i = \frac{(A + B + C) / 2}{32.5} = 23.5\]

\[i = \frac{(17 + 12 + 18) / 2}{23.5} = 23.5\]

\[i = \frac{(21 + 12 + 14) / 2}{23.5} = 23.5\]

\[i = \frac{(27 + 18 + 14) / 2}{29.5} = 29.5\]
# Neighbor Joining Algorithm

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>17</td>
<td>21</td>
<td>27</td>
</tr>
<tr>
<td>B</td>
<td>12</td>
<td></td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
d_{ij} = r_i - r_j
\]

<table>
<thead>
<tr>
<th>i</th>
<th>( r_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>( \frac{(17+21+27)}{2} = 32.5 )</td>
</tr>
<tr>
<td>B</td>
<td>( \frac{(17+12+18)}{2} = 23.5 )</td>
</tr>
<tr>
<td>C</td>
<td>( \frac{(21+12+14)}{2} = 23.5 )</td>
</tr>
<tr>
<td>D</td>
<td>( \frac{(27+18+14)}{2} = 29.5 )</td>
</tr>
</tbody>
</table>
Neighbor Joining Algorithm

\[
\begin{array}{c|cccc}
  & A & B & C & D \\
\hline
  A & - & 17 & 21 & 27 \\
  B & - & 12 & 18 & \\
  C & - & 14 & & \\
  D & - & & & \\
\end{array}
\]

\[
\begin{array}{c|c}
  i & r_i \\
\hline
  A & \frac{(17+21+27)}{2}=32.5 \\
  B & \frac{(17+12+18)}{2}=23.5 \\
  C & \frac{(21+12+14)}{2}=23.5 \\
  D & \frac{(27+18+14)}{2}=29.5 \\
\end{array}
\]

\[
\begin{array}{c|cccc}
  & A & B & C & D \\
\hline
  B & - & -35 & -35 & \\
  C & - & -35 & -35 & \\
  D & - & -39 & & \\
\end{array}
\]

\[d_{ij} = r_i - r_j\]
**Neighbor Joining Algorithm**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>17</td>
<td>21</td>
<td>27</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>12</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ d_{ij} = r_i - r_j \]

<table>
<thead>
<tr>
<th>i</th>
<th>( r_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>( \frac{(17+21+27)}{2} = 32.5 )</td>
</tr>
<tr>
<td>B</td>
<td>( \frac{(17+12+18)}{2} = 23.5 )</td>
</tr>
<tr>
<td>C</td>
<td>( \frac{(21+12+14)}{2} = 23.5 )</td>
</tr>
<tr>
<td>D</td>
<td>( \frac{(27+18+14)}{2} = 29.5 )</td>
</tr>
</tbody>
</table>
Neighbor Joining Algorithm

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>17</td>
<td>21</td>
<td>27</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>12</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ i \quad r_i \]

\[
\begin{align*}
A & \quad \frac{(17+21+27)}{2} = 32.5 \\
B & \quad \frac{(17+12+18)}{2} = 23.5 \\
C & \quad \frac{(21+12+14)}{2} = 23.5 \\
D & \quad \frac{(27+18+14)}{2} = 29.5
\end{align*}
\]

\[ d_{ij} - r_i - r_j \]

\[
\begin{align*}
d_{CD} & = 0.5 \times 14 + 0.5 \times (23.5 - 29.5) = 4 \\
v_C & = 0.5 \times 14 + 0.5 \times (23.5 - 29.5) = 10
\end{align*}
\]
### Neighbor Joining Algorithm

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>17</td>
<td>21</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>12</td>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td></td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Neighbor Joining Tree Diagram](image)
## Neighbor Joining Algorithm

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>17</td>
<td>21</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>12</td>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td></td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
D_{XA} = \frac{(D_{CA} + D_{DA} - D_{CD})}{2} = \frac{(21 + 27 - 14)}{2} = 17
\]

\[
D_{XB} = \frac{(D_{CB} + D_{DB} - D_{CD})}{2} = \frac{(12 + 18 - 14)}{2} = 8
\]
Neighbor Joining Algorithm

\[
d_{\text{XA}} = \frac{(d_{\text{CA}} + d_{\text{DA}} - d_{\text{CD}})}{2}
\]
\[
= \frac{(21 + 27 - 14)}{2}
\]
\[
= 17
\]

\[
d_{\text{XB}} = \frac{(d_{\text{CB}} + d_{\text{DB}} - d_{\text{CD}})}{2}
\]
\[
= \frac{(12 + 18 - 14)}{2}
\]
\[
= 8
\]
Neighbor Joining Algorithm

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$D_{XA} = \frac{(d_{CA} + d_{DA} - d_{CD})}{2}$

$= \frac{(21 + 27 - 14)}{2}$

$= 17$

$d_{XB} = \frac{(d_{CB} + d_{DB} - d_{CD})}{2}$

$= \frac{(12 + 18 - 14)}{2}$

$= 8$
Neighbor Joining Algorithm

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>i</th>
<th>r_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>(17+17)/1 = 34</td>
</tr>
<tr>
<td>B</td>
<td>(17+8)/1 = 25</td>
</tr>
<tr>
<td>X</td>
<td>(17+8)/1 = 25</td>
</tr>
</tbody>
</table>
Neighbor Joining Algorithm

\[
\begin{array}{c|ccc}
  & A & B & X \\
\hline
A & - & 17 & 17 \\
B & - & 8 & - \\
X & - & - & - \\
\end{array}
\]

\[
\begin{array}{c|c}
  i & u_i \\
\hline
A & \frac{(17+17)}{1} = 34 \\
B & \frac{(17+8)}{1} = 25 \\
X & \frac{(17+8)}{1} = 25 \\
\end{array}
\]

\[d_{ij} - r_i - r_j\]
Neighbor Joining Algorithm

\[ d_{ij} - r_i - r_j \]

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

\[ r_i = \frac{(17 + 17)}{1} = 34 \]
\[ r_i = \frac{(17 + 8)}{1} = 25 \]

<table>
<thead>
<tr>
<th>i</th>
<th>[ r_i ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>\frac{(17+17)}{1} = 34</td>
</tr>
<tr>
<td>B</td>
<td>\frac{(17+8)}{1} = 25</td>
</tr>
<tr>
<td>X</td>
<td>\frac{(17+8)}{1} = 25</td>
</tr>
</tbody>
</table>
Neighbor Joining Algorithm

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-17</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>-8</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>

\[ d_{ij} = r_i - r_j \]

\[
\begin{align*}
    i & \quad r_i \\
    A & \quad \frac{(17+17)}{1} = 34 \\
    B & \quad \frac{(17+8)}{1} = 25 \\
    X & \quad \frac{(17+8)}{1} = 25 \\
\end{align*}
\]

\[
\begin{align*}
    v_A & = 0.5 \times 17 + 0.5 \times (34-25) = 13 \\
    v_D & = 0.5 \times 17 + 0.5 \times (25-34) = 4 \\
\end{align*}
\]
# Neighbor Joining Algorithm

A B X Y
---
A | - | 17 | 17
B | - | 8 |
X | - |   |
Y |   |   |

\[ \begin{array}{cccc}
\text{C} & \text{D} \\
4 & 10
\end{array} \quad \begin{array}{cccc}
\text{A} & \text{B} \\
13 & 4
\end{array} \]
Neighbor Joining Algorithm

\[
d_{YX} = \frac{(d_{AX} + d_{BX} - d_{AB})}{2}
\]

\[
= \frac{(17 + 8 - 17)}{2}
\]

\[
= 4
\]
Neighbor Joining Algorithm

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>-4</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>

\[
d_{YX} = \frac{(d_{AX} + d_{BX} - d_{AB})}{2}
\]
\[
= \frac{(17 + 8 - 17)}{2}
\]
\[
= 4
\]
Neighbor Joining Algorithm

\[
d_{YX} = \frac{d_{AX} + d_{BX} - d_{AB}}{2}
\]
\[
= \frac{17 + 8 - 17}{2}
\]
\[
= 4
\]
Neighbor Joining Algorithm

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>17</td>
<td>21</td>
<td>27</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>12</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>-</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
Naruya Saitou

- Division of Population Genetics, National Institute of Genetics
  & Department of Genetics, Graduate University for Advanced Studies (Sokendai)
  Mishima, 411-8540, Japan
The fuller history of NJ

• Saitou and Ney (MBE 87): Introduced the neighbor-joining alg, $O(n^5)$ time, incorrect proof.
• Studier and Keppler (MBE 88): Reduced run time to $O(n^3)$, gave a correct proof

• Alg is highly popular – effective in many situations where the distance matrix is not tree-like
Character Based Methods
Inferring a Phylogenetic Tree

**Generic problem: Optimal Phylogenetic Tree:**

- **Input:**
  - \( n \) species,
  - set of characters,
  - for each species, the state of each of the characters.
  - (parameters)
- **Goal:** find a fully-labeled phylogenetic tree that best explains the data. (maximizes a target function).

**Assumptions:**
- characters are mutually independent
- species evolve independently

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>CAGGTA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>CAGACA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>CGGGTA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>TGCACCT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>TGCGTA</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A Simple Example

• Five species, three have 'C' and two 'T' at a specified position.
• A minimal tree has one evolutionary change:

```
C
C
C
C
C
↔
C
T
T
T
```

Inferring a Phylogenetic Tree

Naive Solution - Enumeration:

- No. of non-isomorphic, labeled, binary, rooted trees, containing $n$ leaves: $(2n - 3)!! = \prod_{i=3}^{n} (2i-3)$
- Unrooted: $(2n-5)!!$

Each new species adds 2 new edges

Adding 3rd species

for $n=20$ this is $10^{21}$!
Parsimony

- **Goal**: explain data with min. no. of evolutionary changes ("mutations", or mismatches)

- **Parsimony**: $S(T) = \sum_j \sum_{\{v, u\} \in E(T)} |\{j: v_j \neq u_j\}|$

- "Small parsimony problem":
  - Input: leaf sequences + a leaf-labeled tree $T$
  - Goal: Find ancestral sequences implying minimum no. of changes (most parsimonious)

- "Large parsimony problem":
  - Input: leaf sequences
  - Goal: Find a most parsimonious tree (topology, leaf labeling and internal seqs.)

- Note: Leaf seqs are assumed to be already aligned
Algorithm for the Small Parsimony Problem (Fitch `71)

- Consider each site in a sequence separately
- Initialization: scan T in post-order, assign:
  - leaf vertex \( m \): \( S_m = \{ \text{state at node } m \} \)
  - internal node \( m \) with children \( l, r \):
    \[
    S_m = \begin{cases} 
    S_l \cup S_r & \text{if } S_l \cap S_r = \emptyset \quad (i) \\
    S_l \cap S_r & \text{o/w} \quad (ii)
    \end{cases}
    \]
- Solution Construction: scan tree in preorder, choose:
  - for the root choose \( x \in S_{root} \)
  - at node \( m \) with parent \( k \) (already constructed)
    pick same state as \( k \) if possible; o/w - pick arbitrarily

Time: \( O(m \cdot n \cdot k) \),
\( k = \# \text{states } s; \ n = \# \text{nodes}; \ m = \# \text{sites} \)
Fitch’s Alg for small parsimony
Walter Fitch
May 21, 1929 - March 10, 2011

• One of the most influential evolutionary biologists in the world, who established a new scientific field: molecular phylogenetics. He was a member in the National Academy of Sciences, the American Academy of Arts and Sciences and the American Philosophical Society. He co-founded and was the first president of the Society for Molecular Biology, which established the annually awarded Fitch Prize. Additionally, he was a founding editor of Molecular Biology and Evolution.

• Fitch was at the University of California, Irvine, until his death, preceded by three years at the University of Southern California and 24 years University of Wisconsin-Madison.
Weighted Small Parsimony

$k$ states $S_1, ..., S_k$

$C_{ij} =$ cost of changing from state $i$ to $j$

Algorithm (Sankoff-Cedergren '88):

• Need: $S_k(x)$ - best cost for the subtree rooted at $x$ if state at $x$ is $k$

• For leaf $x$,
  
  $$S_k(x) = \begin{cases} 
  0 & \text{if state of } x \text{ is } k \\
  \infty & \text{o/w}
  \end{cases}$$

• Scan $T$ in postorder. At node $a$ with children $l$, $r$

  $$S_k(a) = \min_m (S_m(l) + C_{mk}) + \min_m (S_m(r) + C_{mk})$$

• $\text{Opt} = \min_m (S_m(\text{root}))$

  time: $O(n \cdot k^2)$

In Ex.
Over the past 30 years, Sankoff formulated and contributed to many of the fundamental problems in computational biology. In **sequence comparison**, he introduced the quadratic version of the Needleman-Wunsch algorithm, developed the first statistical test for alignments, initiated the study of the limit behavior of random sequences with Vaclav Chvatal and described the **multiple alignment problem**, based on minimum evolution over a phylogenetic tree. In the study of **RNA secondary structure**, he developed algorithms based on general energy functions for **multiple loops** and for **simultaneous folding and alignment**, and performed the earliest studies of parametric folding and **automated phylogenetic filtering**.

Sankoff and Robert Cedergren collaborated on the first studies of the **evolution of the genetic code** based on tRNA sequences. His contributions to **phylogenetics** include early models for **horizontal transfer**, a general approach for optimizing the nodes of a given tree, a method for rapid bootstrap calculations, a generalization of the **nearest neighbor interchange** heuristic, various constraint, consensus and supertree problems, the computational complexity of several phylogeny problems with William Day, and a general technique for **phylogenetic invariants** with Vincent Ferretti. Over the last fifteen years he has focused on the evolution of genomes as the result of **chromosomal rearrangement** processes. Here he introduced the computational analysis of **genomic edit distances**, including parametric versions, the distribution of gene numbers in conserved segments in a random model with Joseph Nadeau, phylogeny based on **gene order** with Mathieu Blanchette and David Bryant, generalizations to include multi-gene families, including algorithms for analyzing **genome duplication** and hybridization with Nadia El-Mabrouk, and the statistical analysis of gene clusters with Dannie Durand. Sankoff is also well known in **linguistics** for his methods of studying grammatical variation and change in speech communities, the quantification of discourse analysis and production models of bilingual speech.
Large Parsimony Problem

**Input**: n x m matrix $M$:
- $M_{ij} =$ state of $j^{th}$ character of species $i$.
- $M_{i.} =$ label of $i$ (all labels are distinct)

**Goal**:
Construct a phylogenetic tree $T$ with $n$ leaves and a label for each node, s.t.
- 1-1 correspondence of leaves and labels
- cost of tree is minimum.
- NP-hard
Branch & Bound (Hendy-Penny ’89)

• enumerate all unrooted trees with increasing no. of leaves

\[ \text{Note: cost of tree with all leaves} \geq \text{cost of subtree with some leaves pruned (and same labeling)} \]
\[ \Rightarrow \text{If cost of subtree} \geq \text{best cost for full tree so far, then: can prune (ignore) all refinements of the subtree.} \]

enumeration & pruning can be done in \( O(1) \) time per visited subtree.
Branch swapping

Each internal edge defines 4 sub-trees:

Can swap two such non-adjacent sub-trees
Nearest Neighbor Interchanges

- handles n-labeled trees
- $T$ and $T'$ are **neighbors** if one can get $T'$ by following operation on $T$:

Use the neighborhood structure on the set of solutions (all trees) via *hill climbing, annealing, other heuristics*...
Probabilistic approaches
Likelihood of a Tree

• Given:
  - $n$ aligned sequences $M = X^1, \ldots, X^n$
  - A tree $T$, leaves labeled with $X^1, \ldots, X^n$

• reconstruction $t$:
  - labeling of internal nodes
  - branch lengths

• Goal: Find optimal reconstruction $t^*$: One maximizing the likelihood $P(M/T, t^*)$
Likelihood (2)

• We need a model for computing $P(M/T, t^*)$

• Assumptions:
  - Each character is independent
  - The branching is a Markov process:
    • The probability that a node $x$ has a specific char. is only a function of the char at the parent node $y$ and the branch length $t$ between them.
  - The probabilities $P(x/y, t)$ are known
Modeling phylogeny as a Bayesian network

- BN with variables $x_1$-$x_5$ and local distributions

\[
P(x_i \mid Pa_i, t_i) = P_{Pa_i \rightarrow x_i}(t_i)
\]

\[
P(x^1, \ldots, x^5 \mid T, t^*) =
\]

\[
P(x^1 \mid x^4, t_1)P(x^2 \mid x^4, t_2)P(x^3 \mid x^5, t_3)P(x^4 \mid x^5, t_4)P(x^5) =
\]

\[
P(root) \prod_{\text{edge } u \rightarrow v} p_{u \rightarrow v}(t_{uv})
\]
Calculating the Likelihood - Example

This is a special case of BN – in general there could be multiple parents for each node (more later in the course).

\[ L = P(M|T) = \sum_r \sum_v P(r) \cdot P_{r\rightarrow s}(t_{rs}) \cdot P_{r\rightarrow v}(t_{rv}) \cdot P_{v\rightarrow u}(t_{vu}) \cdot P_{v\rightarrow w}(t_{vw}) \]
Calculating the Likelihood – General equation

Assume that the branch lengths $t_{uv}$ are known.

Let $\overline{t}$ be the branch lengths and $R$ the rest of the reconstruction = the internal node labels.

\[
P(M|T, \overline{t}) = \prod_{\text{character } j} \left\{ \sum_{\text{reconstruction } R} P(M_{.,j}, R | T, \overline{t}) \right\}
\]

\[
= \prod_{\text{character } j} \left\{ \sum_{\text{reconstruction } R} \left( P(\text{root}) \prod_{\text{edge } u \rightarrow v} p_{u \rightarrow v}(t_{uv}) \right) \right\}
\]

Independence of sites

Markov property independence of each branch
Additional Assumed Properties

- **Additivity:**
  \[ P_{x\rightarrow z}(s + t) = \sum_{b} P_{x\rightarrow b}(s)P_{b\rightarrow z}(t) \]

- **Reversibility (symmetry):**
  \[ P(x)P_{x\rightarrow y}(t) = P(y)P_{y\rightarrow x}(t) \]

- Provable under broad and reasonable assumptions
- Allows one to freely move the root
Efficient Likelihood Calculation  
(Felsenstein ’73)

Use dynamic programming

Define $S_j(a,v) = \Pr(\text{subtree rooted in } v \mid v_j = a)$

Initialization:

∀ leaf $v$ set $S_j(a,v) = 1$ if $v$ is labeled by $a$, else $S_j(a,v) = 0$

Recursion:

Traverse the tree in postorder: for each node $v$ with children $u$ and $w$, for each state $x$

$$S_j(x,v) = \left( \sum_y S_j(y,u)p_{x\rightarrow y}(t_{vu}) \right) \left( \sum_y S_j(y,w)p_{x\rightarrow y}(t_{vw}) \right)$$

Final Soln:  
$$L = \prod_j \left( \sum_x S_j(x,\text{root})P(x) \right)$$

Complexity:  
$O(nmk^2)$

$n$ species, $m$ chars, $k$ states
Finding Optimal Branch lengths

\[ L = \sum_{x, y} P(B \mid v = y)P(v = y \mid z = x, t)P(z = x)P(A \mid z = x) \]

\[ S^z(y, v) \quad p(x) \quad S^v(x, z) \]

\[ p_{x \rightarrow y}(t) \]
Finding Optimal Branch lengths

Optimizing the length of a single branch \( z-v \) can be done using standard optimization techniques

\[
\log L = \sum_{j=1,...,m} \log \sum_{x,y} S_j^z(y,v) P_{x\rightarrow y}(t) P(x) S_j^v(x,z)
\]

- Under the symmetry assumption, each node can be made (temporarily) the root
- To heuristically optimize all the branch lengths: repeatedly optimize one branch at a time
  - No guaranteed convergence, but often works
HOW DO WE FIGURE OUT THE TIMES?

Calculating $P_{u \rightarrow v}(t_{uv})$
Jukes-Cantor Model (J-K ’69)

• Assumptions:
  - Each base in sequence has equal chance of changing
  - Each base changes to one of the other 3 bases with equal probability

• Characteristics
  - Each base appears with equal frequency in DNA
  - The quantity $a$ is the rate of change
  - During each infinitesimal time $\Delta t$ a substitution occurs with probability $3a\Delta t$
Jukes-Cantor Model (J-K '69)
Jukes-Cantor Model

- $P_{A(t)}$: prob of A in the character at time $t$
- Discrete case: Prob. change $A \rightarrow B$ in one time unit is $r$
  - $P_{A(t+1)} = (1-3r) P_{A(t)} + r (1- P_{A(t)})$
  - $P_{A(t+1)} - P_{A(t)} = -3r \ P_{A(t)} + r (1- P_{A(t)})$
- Continuous case:
  - $\Delta P_{A(t)} = -3r \ P_{A(t)} + r (1- P_{A(t)}) = -4r \ P_{A(t)} + r$
  - $\frac{dP_{A(t)}}{dt} = -4rP_{A(t)} + r$
- Solution to the differential equation:
  - $P_{A(t)} = \frac{1}{4} (1+3e^{-4rt})$
Jukes-Cantor Model (J-K '69)

• prob. that the nucleotide remains unchanged over \( t \) time units:
  \[
  P_{\text{same}} = \frac{1}{4} + \frac{3}{4} e^{-4at}
  \]

• Probability of specific change:
  \[
  P_{A \rightarrow B} = \frac{1}{4} - \frac{1}{4} e^{-4at}
  \]

• Probability of change:
  \[
  P_{\text{change}} = \frac{3}{4} - \frac{3}{4} e^{-4at}
  \]

• Note: For \( t \rightarrow \infty \)
  \[
  P_{\text{change}} \rightarrow \frac{3}{4}
  \]
Prob of each of the four nucleotides at equilibrium (\( t = \infty \)) is 1/4.

\[
P_A = P_G = P_C = P_T = \frac{1}{4}
\]

\[
P_{A(0)} = 1 : P_{A(t)} = \frac{1}{4} + \frac{3}{4} e^{-4 \alpha t}
\]

\[
P_{A(0)} = 0 : P_{A(t)} = \frac{1}{4} - \frac{1}{4} e^{-4 \alpha t}
\]

http://nsmn1.uh.edu/dgraur/molevol/fall2010/
Charles Cantor

Boston University
Professor Emeritus, Biomedical Engineering
Professor of Pharmacology, School of Medicine
Ph.D., Biophysical Chemistry, University of California, Berkeley

CSO Sequenom, San Diego, California.
Other Models

• Kimura’s 2-parameter model:
  - A,G - purines; C,T - pyrmidines
  - Two different rates
    • purine-purine or pyrmididine-pyrimidine (transitions)
    • purine-pyrmididine or pyrmidine-purine (transversions)

• Felsenstein '84 and Yano, Hasegawa & Kishino '85 extend the Kimura model to asymmetric base frequencies.
FIN