Lecture 15
Molecular Evolution and Phylogenetics

Somewhere, something went wrong…

Module V: Evolution/phylogeny/populations

- Phylogenetics / Phylogenomics
  - Phylogenetics: Evolutionary models, Tree building, Phylo inference
  - Phylogenomics: gene/species trees, reconciliation, coalescent, pops

- Population genomics:
  - Learning population history from genetic data (David Reich)
  - Statistical genetics: disease mapping in populations (Mark Daly)
  - Measuring natural selection in human populations (Pardis Sabeti)
  - The missing heritability in genome-wide associations (Yaniv Erlich)

- And we’re done! Last pset Nov 21st, In-class quiz on Nov 22nd
  - No lab 4! Then entire focus shifts to projects, Thanksgiving, Frontiers
Extinctions part of life

General Problem:
Infer complete ancestry of a set of ‘objects’ based on knowledge of their ‘traits’

‘Objects’ can be: Species, Genes, Cell types, Diseases, Cancers, Languages, Faiths, Cars, Architectural Styles

‘Traits’ can be: Morphological, molecular, gene expression, TF binding, motifs, words…

Historical record varies: Fossils, imprints, timing of geological events, ‘living fossils’, sequencing of extinct species, paintings, stories.

Today: Phylogenies using only extant species data

Gene trees (paralog / ortholog / homolog trees)

Goals for today

1. Basics of phylogeny
   - Characters, traits, nodes, branches, lineages, topology, lengths
   - Gene trees, species trees, cladograms, chronograms, phylograms

2. From alignments to distances: Modeling sequence evolution
   - Turning pairwise sequence alignment data into pairwise distances
   - Probabilistic models of divergence: Jukes Cantor/Kimura/hierarchy

3. From distances to trees: Tree-building algorithms
   - Ultrametric, Additive, General Distances
   - UPGMA, Neighbor Joining, guarantees and limitations
   - Least-squared error, minimum evolution

4. From alignments to trees: Tree proposal and scoring
   - Parsimony methods: set-based vs. dynamic programming
   - Maximum likelihood methods and Maxim-a-Posteriori methods

5. Tree search algorithms
   - Proposal heuristics, tree rearrangements
   - Markov Chain Monte Carlo and sampling from posterior distribution

Introduction: Basics and Definitions

Characters, traits, gene/species trees

Inferring Phylogenies: Traits and Characters

Trees can be inferred by several criteria:
- Traditional traits: Morphology data
  - Kangaroo: ACGTATACCCAAACGT
  - Elephant: ACGTATACCTCAAAACGT
  - Dog: CCTAGTACCTAACAACGA
  - Mouse: CCTAGTACCTAACAACGA
  - Human: CCTAGTACCTAACAACGA

- Modern traits: Molecular data
  - ACGTATACCCAAACGT
  - CCTAGTACCTAACAACGA

From physiological traits to DNA characters

- Traditional phylogenetics
  - Building species trees
  - Small number of traits
    - Hoofs, nails, teeth, horns
  - Well-behaved traits, each arose once
    - Parsimony principle, Occam’s razor

- Modern phylogenetics
  - Building gene trees and species trees
  - Very large number of traits
    - Every DNA base and every protein residue
  - Frequently ill-behaved traits
    - Back-mutations are frequent (convergent evolution)
    - Small number of letters, arise many times independently
Common Phylogenetic Tree Terminology

- **Branches or Lineages**
- **Terminal Nodes**
- **Ancestral Node or ROOT of the Tree**
- **Internal Nodes or Divergence Points** (represent hypothetical ancestors of the taxa)

Represent the **TAXA** (genes, populations, species, etc.) used to infer the phylogeny.

---

Three types of trees

- **Cladogram**
- **Ultrametric tree**
- **Phylogram**

- **Topology only**
- **Topology + Divergence times**
- **Topology + Divergence times + Divergence rates**

---

Inferring a tree from nucleotides/peptides

- **Molecular phylogenetic methods**
- **Sequence data**:
  - Nucleotide alignments
  - Peptide alignments
- **Evolutionary history** represented as a binary tree

---

Two basic approaches for phylogenetic inference

**Distance based**

1. Sequence alignment
2. Pair-wise distance matrix
3. Tree building algorithms
4. Output tree

**Character based**

- **Sequence alignment**
- **Tree proposal and scoring**
- **Output tree**

---

Goals for today

- **Basics of phylogeny**
  - Characters, traits, nodes, branches, lineages, topology, lengths
  - Gene trees, species trees, cladograms, chronograms, phylograms
- **From alignments to distances**:
  - Modeling sequence evolution
  - Turning pairwise sequence alignment data into pairwise distances
  - Probabilistic models of divergence: Jukes Cantor/Kimura/hierarchy
- **From distances to trees**:
  - Tree-building algorithms
  - Ultrametric, Additive, General Distances
  - UPGMA, Neighbor Joining, guarantees and limitations
  - Least-squared error, minimum evolution
- **From alignments to trees**:
  - Tree proposal and scoring
  - Parsimony methods: set-based vs. dynamic programming
  - Maximum likelihood methods and Maxim-a-Posteriori methods
- **Tree search algorithms**
  - Proposal heuristics, tree rearrangements
  - Markov Chain Monte Carlo and sampling from posterior distribution

1. From alignments to distances
   - Modeling evolutionary rates
Measuring evolutionary rates

- Nucleotide divergence
  - Uniform rate. Overall percent identity.
- Transitions and transversions
- Synonymous and non-synonymous substitutions
  - Ka/Ks rates. Amino-acid changing substitutions

\[ N_{\text{substitutions}} > N_{\text{mutations}} \]
  - Some fraction of "conserved" positions mutated twice

'Evolving' a nucleotide under random model

- At time step 0, start with letter A
- At time step 1:
  - Remain A with probability 0.7
  - Change to C,G,T with prob. 0.1 each
- At time step 2:
  - In state A with probability 0.52
  - Remain A with probability 0.7, 0.7 each
  - Go back to A from C,G,T with 0.1 each
  - In states C,G,T with prob. 0.16 each

<table>
<thead>
<tr>
<th>t</th>
<th>A</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>t=1</td>
<td>0.7</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>t=2</td>
<td>0.52</td>
<td>0.16</td>
<td>0.16</td>
</tr>
<tr>
<td>t=3</td>
<td>0.412</td>
<td>0.196</td>
<td>0.2176</td>
</tr>
<tr>
<td>t=4</td>
<td>0.3472</td>
<td>0.196</td>
<td>0.2176</td>
</tr>
<tr>
<td>t=5</td>
<td>0.196</td>
<td>0.2176</td>
<td>0.2176</td>
</tr>
</tbody>
</table>

Modeling Nucleotide Evolution

During infinitesimal time \( \Delta t \), there is not enough time for two substitutions to happen on the same nucleotide. So we can estimate \( P(x \mid y, \Delta t) \), for \( x, y \in \{A, C, G, T\} \)

Then let

\[ S(\Delta t) = \begin{bmatrix} P(A|A, \Delta t) & \ldots & P(A|T, \Delta t) \\ P(T|A, \Delta t) & \ldots & P(T|T, \Delta t) \end{bmatrix} \]

Jukes-Cantor: constant rate of evolution

For short time \( \epsilon \), \( S(\epsilon) = \begin{bmatrix} 1 - 3\epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 1 - 3\epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & 1 - 3\epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & 1 - 3\epsilon \end{bmatrix} \)

Kimura:

Transitions: A/G, C/T
Transversions: A/T, A/C, G/T, C/G

Transitions (rate \( \alpha \)) are much more likely than transversions (rate \( \beta \))

\[ S(t) = \begin{bmatrix} \alpha & \beta & \beta & \beta \\ \beta & \alpha & \beta & \beta \\ \beta & \beta & \alpha & \beta \\ \beta & \beta & \beta & \alpha \end{bmatrix} \]

Where \( s(t) = \frac{1}{4} (1 - e^{-\alpha t}) \)
\( u(t) = \frac{1}{4} (1 - e^{-\beta t}) \)
\( r(t) = 1 - 2s(t) - u(t) \)

\[ \begin{align*}
  r(t) &= \frac{1}{4} (1 + 3 e^{-\alpha t}) \\
  s(t) &= \frac{1}{4} (1 - e^{-\alpha t}) \\
  u(t) &= \frac{1}{4} (1 + e^{-\beta t}) \\
  \alpha &= 1 - 3\epsilon \\
  \beta &= 1 - 3\epsilon \\
  \gamma &= 1 - 3\epsilon \\
  \delta &= 1 - 3\epsilon \\
  \epsilon &= 1 - 3\epsilon \\
  \alpha' &= 1 - 3\epsilon \\
  \beta' &= 1 - 3\epsilon \\
  \gamma' &= 1 - 3\epsilon \\
  \delta' &= 1 - 3\epsilon \\
\end{align*} \]
Distance between two sequences

Given (well-aligned portion of) sequences $x^i$, $x^j$,

Define

$$d_{ij} = \text{distance between the two sequences}$$

One possible definition:

$$d_{ij} = \text{fraction } f \text{ of sites } u \text{ where } x^i[u] \neq x^j[u]$$

Better model (Jukes-Cantor):

$$d_{ij} = -\frac{3}{4} \log(1 - \frac{4f}{3})$$

- $r(t) = \frac{1}{4} \left(1 + 3 e^{-4\alpha t}\right)$
- $s(t) = \frac{1}{4} \left(1 - e^{-4\alpha t}\right)$

Observed $F = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7])$

Actual $D = [0.11, 0.23, 0.38, 0.57, 0.82, 1.21, 2.03]$

Many nucleotide models have been developed

Varying levels of complexity (parameters)

Models also exist for peptides and codons

Goals for today

- Basics of phylogeny
  - Characters, traits, nodes, branches, lineages, topology, lengths
  - Gene trees, species trees, cladograms, chronograms, phylograms
- From alignments to distances: Modeling sequence evolution
  - Turning pairwise sequence alignment data into pairwise distances
  - Probabilistic models of divergence: Jukes Cantor/Kimura/hierarchy
- From distances to trees: Tree-building algorithms
  - Ultrametric, Additive, General Distances
  - UPGMA, Neighbor Joining, guarantees and limitations
  - Least-squared error, minimum evolution
- From alignments to trees: Tree proposal and scoring
  - Parsimony methods: set-based vs. dynamic programming
  - Maximum likelihood methods and Maxim-a-Posteriori methods
- Tree search algorithms
  - Proposal heuristics, tree rearrangements
  - Markov Chain Monte Carlo and sampling from posterior distribution

Distance matrix ↔ Phylogenetic tree

<table>
<thead>
<tr>
<th></th>
<th>Hum</th>
<th>Mou</th>
<th>Rat</th>
<th>Dog</th>
<th>Cat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Mouse</td>
<td>h.y.m</td>
<td>0</td>
<td>3</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Rat</td>
<td>h.y.r</td>
<td>0</td>
<td>9</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Dog</td>
<td>h.z.x.d</td>
<td>m.y.z.x.d</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Cat</td>
<td>h.z.c</td>
<td>m.y.z.c</td>
<td>r.y.z.c</td>
<td>d.c</td>
<td>0</td>
</tr>
</tbody>
</table>

Distance: (a) Ultrametric distances

- For all points $i, j, k$
  - two distances are equal and third is smaller
  - $d(i,j) \leq d(i,k) = d(j,k)$
  - $a + a \leq a + b = a + b$

Goal:

Minimize discrepancy between observed distances and tree-based distances

Result:

- All paths from leaves are equidistant to the root
- Rooted tree with uniform rates of evolution
Distances: (b) Additive distances

- All distances satisfy the four-point condition
  - Any quartet can be labeled i,j,k,l such that:
    - $d(i,j) + d(k,l) \leq d(i,k) + d(j,l) = d(i,l) + d(j,k)$
    - $(a+b)+(c+d) \leq (a+e+c)+(b+e+d) = (a+e+d)+(b+e+c)$

- Result:
  - All pairwise distances obtained by traversing a tree

Distances: (c) General distances

- In practice, a distance matrix is neither ultrametric nor additive
  - Noise
    - Measured distances are not exact
    - Evolutionary model is not exact
  - Fluctuations
    - Regions used to measure distances not representative of the species tree
    - Gene replacement (gene conversion), lateral transfer
    - Varying rates of mutation can lead to discrepancies

- In the general case, tree-building algorithms must handle noisy distance matrices
  - Such a tree can be obtained by
    - Enumeration and scoring of all trees (too expensive)
    - Neighbor-Joining (typically gives a good tree)
    - UPGMA (typically gives a poor tree)

Algorithms: (a) UPGMA (aka Hierarchical Clustering)
(Unweighted Pair Group Method with Arithmetic mean)

Initialization:
Assign each $x_i$ into its own cluster $C_i$
Define one leaf per sequence, height 0

Iteration:
Find two clusters $C_i, C_j$ s.t. $d_{ij}$ is min
Let $C_k = C_i \cup C_j$
Define node connecting $C_i, C_j$, & place it at height $d_{ij}/2$
Delete $C_i, C_j$

Termination:
When two clusters $i, j$ remain, place root at height $d_{ij}/2$

Ultrametric Distances & UPGMA

UPGMA is guaranteed to build the correct tree if distance is ultrametric

Proof:
1. The tree topology is unique, given that the tree is binary
2. UPGMA constructs a tree obeying the pairwise distances

Weakness of UPGMA

Molecular clock assumption:
implies time is constant for all species

However, certain species (e.g., mouse, rat) evolve much faster

Example where UPGMA messes up:

Correct tree

UPGMA

Algorithms: (b) Neighbor-Joining

- Guaranteed to produce the correct tree if distance is additive
- May produce a good tree even when distance is not additive

Step 1: Finding neighboring leaves

Define

$$D_k = d_k - (\epsilon_k + \epsilon_i)$$

Where

$$\epsilon_i = \frac{1}{k + \sum d_k}$$

Claim: The above "magic trick" ensures that $D_k$ is minimal iff $i, j$ are neighbors

Proof: Beyond the scope of this lecture (Durbin book, p. 189)
Algorithm: Neighbor-joining

**Initialization:**
Define T to be the set of leaf nodes, one per sequence
Let L = T

**Iteration:**
Pick i, j s.t. Dij is minimal
Define a new node k, and set dkm = ½ (dim + djm − d ij) for all m ∈ L
Add k to T with edges of lengths d ik = ½ (dij + r i − r j)
Remove i, j from L
Add k to L

**Termination:**
When L consists of two nodes, i, j, and the edge between them of length d ij

Algorithm: (c) Distance-fitting algorithms

- With distance-based algorithms, we can also aim to directly minimize discrepancy between original distance matrix and tree-based distance matrix

**Distance matrix ⇔ Phylogenetic tree**

Goals for today

- Basics of phylogeny
  - Characters, traits, nodes, branches, lineages, topology, lengths
  - Gene trees, species trees, cladograms, chronograms, phylogenograms
- From alignments to distances: Modeling sequence evolution
  1. Turning pairwise sequence alignment data into pairwise distances
  2. Probabilistic models of divergence: Jukes Cantor/Kimura/hierarchy
- From distances to trees: Tree-building algorithms
  1. Ultrametric, Additive, General Distances
  2. UPGMA, Neighbor Joining, guarantees and limitations
- Least-squared error, minimum evolution
- From alignments to trees: Tree proposal and scoring
  3. Parsimony methods: set-based vs. dynamic programming
  4. Maximum likelihood methods and Maxim-a-Posteriori methods
- Tree search algorithms
  5. Proposal heuristics, tree rearrangements
  6. Markov Chain Monte Carlo and sampling from posterior distribution

3. Character-based tree-scoring algorithms

3a: Parsimony (set-based)
3b: Parsimony (Dyn. Prog.)
3c: Maximum Likelihood
Character-based phylogenetic inference

- Really about tree **scoring** techniques, not tree finding techniques
  - Couple them with tree proposal and update and you have an algorithm (part 4 of the lecture)
- Two approaches exist, all use same architecture:
  - Minimize events: Parsimony (union/intersection)
  - Probabilistic: Max Likelihood / MAP

### Parsimony scoring (a): Union and intersection

Given a tree, and an alignment column

- Label internal nodes to minimize the number of required substitutions
- **Initialization:**
  - Set cost \( C = 0 \); \( k = 2N - 1 \)
- **Iteration:**
  - If \( k \) is a leaf, set \( R_k = \{ x_k[u] \} \)
  - If \( k \) is not a leaf, let \( i, j \) be the daughter nodes; set \( R_k = R_i \cap R_j \) if intersection is nonempty
  - Set \( R_k = R_i \cup R_j \) and \( C += 1 \) if intersection is empty
- **Termination:**
  - Minimal cost of tree for column \( u \), = \( C \)

### Parsimony traceback to find ancestral nucleotides

1. Choose an arbitrary nucleotide from \( R_{2N-1} \) for the root
2. Having chosen nucleotide \( r \) for parent \( k \),
   - If \( r \in R_i \) choose \( r \) for daughter \( i \)
   - Else, choose arbitrary nucleotide from \( R_i \)

### Parsimony Scoring (b): Dynamic programming

- Each cell \((N,C)\) represents the min cost of the subtree rooted at \( N \), if the label at \( N \) is \( C \)
- Update table by walking up the tree from the leaves to the root, remembering max choices
- Traceback from root to leaves to construct a min cost assignment

### Goals for today

- Basics of phylogeny
  - Characters, traits, nodes, branches, lineages, topology, lengths
  - Gene trees, species trees, cladograms, chronograms, phylogenomics
- From alignments to distances: Modeling sequence evolution
  - Turning pairwise sequence alignment data into pairwise distances
  - Probabilistic models of divergence: Jukes-Cantor/Kimura/hierarchy
- From distances to trees: Tree-building algorithms
  - Ultrametric, Additive, General Distances
  - UPGMA, Neighbor Joining, guarantees and limitations
  - Least-squared error, minimum evolution
- From alignments to trees: Tree proposal and scoring
  - Parsimony methods: set-based vs. dynamic programming
  - Maximum likelihood methods and Maximum-a-Posteriori methods
- Tree search algorithms
  - Proposal heuristics, tree rearrangements
  - Markov Chain Monte Carlo and sampling from posterior distribution

### Scoring (c) Maximum Likelihood & Max-a-Posteriori

**Input:** Sequence alignment

**Output:** tree with maximum likelihood / max a posteriori prob.

**Search:** Heuristic search for max likelihood tree.

**Maximum Likelihood (ML)**

\[
B^*,T^* = \arg\max_{B,T} P(D|B,T)
\]

**Maximum a Posteriori (MAP)**

\[
B^*,T^* = \arg\max_{B,T} P(B,T|D)
\]

- **Compute recursively using DP**

\[P(D|B,T)\] is the likelihood of data given model

- Use seq evolution model: JC,K2P,HKY.

\[P(B,T)\] is a prior on trees/branch lengths

- Use Yule process, Birth-Death process to model
1. Site evolution over single branch

**Remember: Jukes-Cantor (JC)**

**Discrete MC version**
- Given time $t$, we define a discrete MC with transition matrix $S(t)$, also called a substitution probability matrix.
- Gives the probability of seeing base $a$ given initial base $b$ after duration time $t$.

Use JC to define single site evolution:
$$P(a = “C”|b = “A”, t) = S(t)_{bc}$$

2. Sequence evolution over single branch

- **Assume site independence**
  $$P(x_1, x_2, x_3, x_4|T, t) = \prod_{i=1}^4 P(x_i|T, t)$$

Use product to define sequence evolution:
$$P(x_{1\ldots n}\ T\ t)$$

- **Examples:**
  - $x_1 = “AACTG”$
  - $x_2 = “CAAGTC”$

3. Sequence evolution over entire tree

- **Assume branch independence**
  $$P(x_{1\ldots n}\ T\ t) = \prod_{i=1}^n P(x_i|T, t)$$

4. Integrate (marginalize) over hidden ancestral seqs!

- Notice, all sequences are needed, both internal nodes and leaves:
  $$P(x_{1\ldots n}\ T\ t) = \sum_{x_{1\ldots n}} P(x_{1\ldots n}) P(x_{1\ldots n}|T, t)$$

- But, only leaves are given: $x_1, \ldots, x_n$

- Therefore, need to marginalize (sum) over unknowns: $x_{n+1}, \ldots, x_{2n-1}$

- This looks expensive:
  $$P(x_{1\ldots n}\ T\ t) = \sum_{x_{1\ldots n}} \sum_{x_{n+1}\ldots x_{2n-1}} P(x_{1\ldots n}) P(x_{n+1}\ldots x_{2n-1}|T, t)$$

- Don’t worry, dynamic programming can do it efficiently.

**Basic trick to efficient marginalization**

Apply factorization trick to every internal node in the tree.

| $P(x_{i\ldots j}|x_i, x_j, T, t)$ | $P(x_i|x_j, T, t)$ | $P(x_j|T, t)$ |
|---------------------------------|--------------------|---------------|
| $P(x_1, x_2, x_3, x_4|T, t)$    | $P(x_1|x_2, x_3, x_4|T, t)$ | $P(x_2, x_3, x_4|T, t)$ |
| $P(x_5|x_6, x_7, x_8|T, t)$    | $P(x_5|x_6, x_7|T, t)$ | $P(x_6, x_7|T, t)$ |
| $P(x_9|x_{10}, x_{11}, \ldots, x_{20}|T, t)$ | $P(x_9|x_{10}, x_{11}|T, t)$ | $P(x_{10}, x_{11}|T, t)$ |

**Peeling algorithm**

- Let $P(x_{i\ldots j}|i, j, T, t)$ be the DP table.
- Each entry contains the probability of seeing the leaf data below node $i$ given that node $i$ has base at site $j$.
- The leaves of the table are initialized based on the observed sequence.
- Entries populated in post-order traversal.
- Runtime: $O(n^2 m^2 k^2)$
Use DP to compute $\arg\max P(D|B,T)$ efficiently

- If we know the branch lengths $t_{left}$ & $t_{right}$.
- And we already have the likelihood tables $L_j$ & $L_k$ of left and right subtrees (for each possible ending character at $b, c$).

$\Rightarrow$ Fill in likelihood table $L_i$ for each char $a$ at $i$.

$\text{L}_i[a] = \sum_{b \in \{\text{ACGT}\}} \sum_{c \in \{\text{ACGT}\}} (P(b|a,t_{left}) \cdot L_{left}[b] \cdot P(c|a,t_{right}) \cdot L_{right}[c])$

Char $a$ at node $i$

$t_{left}$

$t_{right}$

$\text{b at } j$

$\text{c at } j$

$\text{L}_j$[b]

$\text{L}_k$[c]

$\text{L}_i[a]$

Advantages/disadvantages of ML/MAP methods

**Advantages:**
- Inherently statistical and evolutionary model-based.
- Usually the most ‘consistent’ of the methods available.
- Used for both character and rate analyses.
- Can be used to infer the sequences of the extinct ancestors.
- Account for branch-length effects in unbalanced trees.
- Nucleotide or amino acid sequences, other types of data.

**Disadvantages:**
- Not as simple and intuitive as many other methods.
- Computationally intense (limits num taxa, sequence length).
- Like parsimony, can be fooled by high levels of homoplasy.
- Violations of model assumptions can lead to incorrect trees.

Initialization and Termination

<table>
<thead>
<tr>
<th>Root</th>
<th>Internal Nodes</th>
<th>Leaves</th>
<th>$z_{n-1}$</th>
<th>i</th>
<th>j</th>
<th>k</th>
<th>...</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Characters at the leaves are already known
  - Their likelihood is 1 or 0, indicating the known char.
- Fill in internal node likelihood vectors iteratively
- Once we reach the root, multiply by the base freqs.
- Maximization over Topologies and Lengths
  $\Rightarrow$ Numerical: gradient descent, Newton’s method.

Algorithms for reconstructing trees

- **UPGMA** (Sokal 1958) [distance]
- Least Square Error (Cavalli-Sforza & Edwards 1963) [dist]
- Maximum Parsimony (Fitch 1977) [character]
  $\Rightarrow$ PHYLIP
- Maximum Likelihood (Felsenstein 1981) [character]
  $\Rightarrow$ RAxML, PHYML, PHYLIP, PAUP
- Neighbor Joining (Saitou 1987) [distance]
  $\Rightarrow$ BIONJ, QuickTree, RapidTree, FJ
- Maximum a posteriori (Rannala & Yang 1996) [character]
  $\Rightarrow$ MrBayes, BEAST
  $\Rightarrow$ BEST, CoalMCMC, CoalHMM, PhyloNet, SRunning, SYNERGY, STIME-GER

Basic algorithms of phylogenetic methods

**Distance based**

1. Sequence alignment
2. Pair-wise distance matrix
3. Tree building algorithms

**Character based**

1. Output tree
2. Tree Proposal Methods
3. Markov-Chain Monte Carlo (MCMC)

Tree space exploration

**Tree Proposal Methods**

Markov-Chain Monte Carlo (MCMC)
Propose (NNI, SPR) Heuristic tree search in character-based reconstruction

- Propose many trees using rearrangements; "score" each one.
- Score can be: parsimony cost, likelihood, posterior probability.
- Algorithms differ on how "score" is used to guide the tree search.

Computing Probability/likelihood $P(\text{Score, likelihood, posterior prob., etc})$

Initial Tree (NJ)

- Propose many trees using rearrangements; "score" each one.
- Score can be: parsimony cost, likelihood, posterior probability.
- Algorithms differ on how "score" is used to guide the tree search.

- Nearest neighbor interchange
- Subtree pruning and regrafting

Other methods include: tree bisection and join (TBJ).

Tree space is big:
- Number of unrooted topologies: $N_u = 3 \times 5 \times \ldots \times (2n - 5) = (2n - 5)!$
- Number of rooted topologies: $N_r = (2n - 3) \times N_u = (2n - 5)!$

Nu(3)=1, Nu(4)=3, Nu(5)=15,
Nu(10)=2e6, Nu(20)=2e20, Nu(64) > googol

Markov Chain Monte Carlo (MCMC)

Exploring your state space in a way that makes it a sample from the posterior (well, after skipping burn-in while likelihood is climbing and only taking 1 in 100)

Metropolis-Hastings details

- The ratio you compute in the Metropolis-Hastings rule is actually more complicated than what previous slide describes. Say current state is $(B_1, T_1)$ and proposed next state is $(B_2, T_1)$. Then the ratio you look at is

$$\alpha = \frac{P(B_1, T_1 | D) P(B_2, T_2 | B_2, T_2)}{P(B_2, T_2 | D) P(B_1, T_1 | B_1, T_1)}$$

- If $\alpha > 1$ then accept $(B_2, T_2)$ no matter what, and if it's less than 1, accept with probability $\alpha$. The term $P(B_1, T_1 | B_2, T_2)$ describes the probability of proposing state 1 given state 2. The Metropolis-Hastings algorithm is a generalization of an early algorithm called the Metropolis algorithm that required your proposal rule to be symmetric. Namely,$P(B_1, T_1 | B_2, T_2)$ is the probability of proposing state 2 given state 1.

- Now notice this is just ratio of prior probabilities. If the prior is flat $(P(B_1, T_1) = P(B_2, T_2))$, then it simplifies even further to a ratio of likelihoods.

$$\alpha = \frac{P(D | B_2, T_2) P(B_2, T_2)}{P(D | B_1, T_1) P(B_1, T_1)}$$

- with $P(D | B_2, T_2)$ which is the ratio that was described in the previous slide.

- Assumptions needed: symmetric proposal rule and flat prior. Whether or not these assumptions are added, the data probability $P(D)$ (often called the "evidence") is still canceled in the ratio.

- Markov Chain Monte Carlo (MCMC) is only used when $P(D)$ can't be computed efficiently. If $P(D)$ can be computed efficiently you would probably just do $P(D | B, T) P(B, T) = P(D) P(B, T)$ and compute posterior probabilities directly.

Tree reliability: Bootstrapping

Main outline of algorithm
1. Make re-sampled alignment by randomly sampling with replacement columns from the real data to create an alignment of equal size.
2. Build a phylogenetic tree based on the random sample from (1)
3. Repeat (1), (2) many (say, 1000) times
4. Output the tree that is constructed most frequently (or a consensus tree)
Basic algorithms of phylogenetic methods

Distance based
- From sequences to distances
- Tree building algorithms

Character based
- From alignments to phylogenies
- Couple to tree proposal and scoring

Summary
- Basics of phylogeny
  - Characters, traits, nodes, branches, lineages, topology, lengths
  - Gene trees, species trees, cladograms, chronograms, phylograms
- From alignments to distances: Modeling sequence evolution
  - Turning pairwise sequence alignment data into pairwise distances
  - Probabilistic models of divergence: Jukes Cantor/Kimura/hierarchy
- From distances to trees: Tree-building algorithms
  - Ultrametric, Additive, General Distances
  - UPGMA, Neighbor Joining, guarantees and limitations
  - Least-squared error, minimum evolution
- From alignments to trees: Tree proposal and scoring
  - Parsimony methods: set-based vs. dynamic programming
    - Maximum likelihood methods and Maxim-a-Posteriori methods
- Tree search algorithms
  - Proposal heuristics, tree rearrangements
  - Markov Chain Monte Carlo and sampling from posterior distribution

Module V: Evolution/phylogeny/populations

- Phylogenetics / Phylogenomics
  - Phylogenetics: Evolutionary models, Tree building, Phylo inference
  - Phylogenomics: gene/species trees, reconciliation, coalescent, pops
- Population genomics:
  - Learning population history from genetic data (David Reich)
  - Statistical genetics: disease mapping in populations (Mark Daly)
  - Measuring natural selection in human populations (Pardis Sabeti)
  - The missing heritability in genome-wide associations (Yaniv Erlich)
- And we’re done! Last pset Nov 21st, In-class quiz on Nov 22nd
  - No lab 4! Then entire focus shifts to projects, Thanksgiving, Frontiers