Lecture 17
Molecular Evolution and Phylogenetics

Somewhere, something went wrong…
<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Topic</th>
<th>Lection</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Thu Sep 8</td>
<td>Introduction</td>
<td>L1</td>
<td>Intro: Biology, Algorithms, Machine Learning, Course Overview</td>
</tr>
<tr>
<td></td>
<td>Fri Sep 9</td>
<td>Introduction</td>
<td>R1</td>
<td>Recitation 1: Biology and Probability Review</td>
</tr>
<tr>
<td>2</td>
<td>Tue Sep 13</td>
<td>Module I: Aligning and Modeling Genomes</td>
<td>L2</td>
<td>Alignment I: Dynamic Programming, Global and local alignment</td>
</tr>
<tr>
<td></td>
<td>Tue Sep 16</td>
<td>Module I: Aligning and Modeling Genomes</td>
<td>L3</td>
<td>Alignment II: Database search, Rapid string matching, BLAST, BLOSUM</td>
</tr>
<tr>
<td></td>
<td>Tue Sep 20</td>
<td>Module I: Aligning and Modeling Genomes</td>
<td>R2</td>
<td>Recitation 2: Deriving Parameters of Alignment, Multiple Alignment</td>
</tr>
<tr>
<td>3</td>
<td>Thu Sep 22</td>
<td>Module I: Aligning and Modeling Genomes</td>
<td>L4</td>
<td>Hidden Markov Models Part 1: Evaluation/Parsing, Viterbi, Forward algorithms</td>
</tr>
<tr>
<td></td>
<td>Thu Sep 23</td>
<td>Module I: Aligning and Modeling Genomes</td>
<td>L5</td>
<td>Hidden Markov Models Part 2: Posterior Decoding, Learning, Baum-Welch</td>
</tr>
<tr>
<td></td>
<td>Fri Sep 23</td>
<td>No classes - student holiday</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Thu Sep 27</td>
<td>Module II: Gene Expression and Networks</td>
<td>L6</td>
<td>Expression Analysis: Clustering/Classification, K-means, Hierarchical, Bayesian</td>
</tr>
<tr>
<td></td>
<td>Thu Sep 29</td>
<td>Module II: Gene Expression and Networks</td>
<td>L7</td>
<td>Transcript structure: GenScan, RNA-seq, Mapping, De novo Assembly, Diff Expr</td>
</tr>
<tr>
<td></td>
<td>Fri Sep 30</td>
<td>Module II: Gene Expression and Networks</td>
<td>R3</td>
<td>Recitation 3: Affinity Propagation Clustering and Random Forest Classification</td>
</tr>
<tr>
<td>5</td>
<td>Tue Oct 4</td>
<td>Module II: Gene Expression and Networks</td>
<td>L8</td>
<td>Epigenomics: ChIP-Seq, Read mapping, Peak calling, IDR, Chromatin states</td>
</tr>
<tr>
<td></td>
<td>Fri Oct 7</td>
<td>Module II: Gene Expression and Networks</td>
<td>L9</td>
<td>Three-dimensional chromatin interactions: 3C, 5C, HiC, ChIA-Pet</td>
</tr>
<tr>
<td></td>
<td>Fri Oct 7</td>
<td>Module II: Gene Expression and Networks</td>
<td>R4</td>
<td>Recitation 4: ENCODE, Epigenome Roadmap, ChromHMM, ChromImpute</td>
</tr>
<tr>
<td>6</td>
<td>Thu Oct 11</td>
<td>No classes - Columbus Day Holiday</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Fri Oct 21</td>
<td>No classes - Columbus Day Holiday</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Tue Oct 25</td>
<td>Module IV: Population and Disease Genetics</td>
<td>L13</td>
<td>Population genetics: Linkage disequilibrium, pop struct, 1000GENOMES, allele freq</td>
</tr>
<tr>
<td></td>
<td>Fri Oct 28</td>
<td>Module IV: Population and Disease Genetics</td>
<td>L14</td>
<td>Disease Association Mapping, GWAS, organismal phenotypes</td>
</tr>
<tr>
<td></td>
<td>Tue Nov 1</td>
<td>Module IV: Population and Disease Genetics</td>
<td>R7</td>
<td>Recitation 7: Linkage Disequilibrium, Haplotype Phasing, Genotype Imputation</td>
</tr>
<tr>
<td>9</td>
<td>Thu Nov 3</td>
<td>Module IV: Population and Disease Genetics</td>
<td>L15</td>
<td>Quantitative trait mapping, molecular traits, eQTLs, mediation analysis, IMWAS</td>
</tr>
<tr>
<td></td>
<td>Fri Nov 4</td>
<td>Module IV: Population and Disease Genetics</td>
<td>L16</td>
<td>Missing Heritability, Complex Traits, Interpret GWAS, Rank-based enrichment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Module V: Comparative Genomics and Evolution</td>
<td>L17</td>
<td>Comparative genomics and evolutionary signatures</td>
</tr>
<tr>
<td>10</td>
<td>Tue Nov 8</td>
<td>Module V: Comparative Genomics and Evolution</td>
<td>L18</td>
<td>Genome Scale Evolution, Genome Duplication</td>
</tr>
<tr>
<td></td>
<td>Thu Nov 10</td>
<td>Module V: Comparative Genomics and Evolution</td>
<td>L19</td>
<td>Phylogenetics: Molecular evolution, Tree building, Phylogenetic inference</td>
</tr>
<tr>
<td></td>
<td>Thu Nov 15</td>
<td>Module V: Comparative Genomics and Evolution</td>
<td>L20</td>
<td>Phylogenomics: Gene/species trees, reconciliation, coalescent, ARGs</td>
</tr>
<tr>
<td></td>
<td>Thu Nov 17</td>
<td>Module V: Comparative Genomics and Evolution</td>
<td>R8</td>
<td>Recitation 8: Phylogenetic-distance metrics, Coalescent Process</td>
</tr>
<tr>
<td>11</td>
<td>Thu Nov 18</td>
<td>Module VI: Current Research Directions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fri Nov 11</td>
<td>Module VI: Current Research Directions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Thu Nov 22</td>
<td>No more sets! (work on your final project)</td>
<td>L21</td>
<td>Single-cell genomics: technology, breakthroughs, microfluidics, applications, insights</td>
</tr>
<tr>
<td></td>
<td>Thu Nov 24</td>
<td>No more sets! (work on your final project)</td>
<td>L22</td>
<td>Mining human phenotypes, PhEWS, UK Biobank, meta-phenotypes-imputation</td>
</tr>
<tr>
<td>13</td>
<td>Thu Dec 1</td>
<td>No more sets! (work on your final project)</td>
<td>R10</td>
<td>Recitation 10: Project Feedback, results, interpretations, directions</td>
</tr>
<tr>
<td>14</td>
<td>Thu Dec 6</td>
<td>Recitation 11: Presentation Tips - Intro, discussion, Slides, Presentation skills</td>
<td>L23</td>
<td>Cancer Genomics, Single-cell Sequencing, Tumor-Immune Interface</td>
</tr>
<tr>
<td></td>
<td>Thu Dec 8</td>
<td>Recitation 11: Presentation Tips - Intro, discussion, Slides, Presentation skills</td>
<td>L24</td>
<td>Genome Engineering with CRISPR/Cas9 and related technologies</td>
</tr>
<tr>
<td>15</td>
<td>Fri Dec 9</td>
<td>Recitation 11: Presentation Tips - Intro, discussion, Slides, Presentation skills</td>
<td>L25</td>
<td>Final Presentations - Part I (11am) 32-8 G reading room</td>
</tr>
<tr>
<td></td>
<td>Fri Dec 13</td>
<td>Recitation 11: Presentation Tips - Intro, discussion, Slides, Presentation skills</td>
<td>L25</td>
<td>Final Presentations - Part I (1pm) 32-141</td>
</tr>
</tbody>
</table>

**Project Profile due Tue 9/27**

- Identify previous project proposals, recent papers, and potential partners that match your areas of interest. List initial project ideas and partners.
- Form teams of two; specify project goals, division of work, milestones, datasets, challenges. Prepare slide presentation for the class and the mentors. **Project proposal due Tue 10/18. Presented to mentor on Fri 10/21.**

**Evaluation/Discuss three peer proposals, NIH review format. Reviews due Mon 10/31. Reviews returned Thu 11/3.**

- Address peer evaluations, revise aims, scope, list of final deliverables/goals. **Response due Thu 11/10.**

**Midcourse report due Wed 11/23**

- Complete making substantial progress on proposed milestones. Write outline of final report. As part of report, comment on your overall project experience. **Written report due Sun 12/11.**

**Conference report format slide presentation. Talks on Tue 12/13.**
Module V: Comparative genomics and evolution

- **L17: Comparative genomics & evolutionary signatures**
  - Genome-wide studies → infer signatures → annotate elements
  - Protein-coding, non-coding RNA, microRNA, regulatory motifs

- **L18: Genomic rearrangements and genome duplication**
  - Synteny-based alignment, Beyond nucleotide mutation
  - Detecting structural changes: genome assembly/alignment

- **L19: Phylogenetics**
  - Evolutionary rates and models of evolution
  - Distance-based trees, DP on a tree, Bayesian models

- **L20: Phylogenomics**
  - Gene trees vs. species trees, reconciliation, coalescence
Extinctions part of life
Phylogenetics

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: Phylogenetics

0. Basics of phylogeny: Introduction and definitions
   - Characters, traits, nodes, branches, lineages, topology, lengths
   - Gene trees, species trees, cladograms, chronograms, phylograms

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

2. From distances to trees: Tree-building algorithms
   - Tree types: Ultrametric, Additive, General Distances
   - Algorithms: UPGMA, Neighbor Joining, guarantees and limitations
   - Optimality: Least-squared error, minimum evolution (require search)

3. From alignments to trees: Alignment scoring given a tree
   - Parsimony: greedy (union/intersection) vs. DP (summing cost)
   - ML/MAP (includes back-mutations, lengths): peeling algorithm (DP)

4. Tree exploration: Markov Chain Monte Carlo tree search
   - Proposal operations: NNI neighbor interchange, SPR prune/regraft
   - MCMC: Metropolis-Hastings, sample posterior \( P(B,T|D) \) without \( P(D) \)
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

– Modern traits: Molecular data

Kangaroo: ACAGTGACGCCCCAAACGT
Elephant: ACAGTGACGCTACAAACGT
Dog: CCTGTGACGTAACAAACGA
Mouse: CCTGTGACGTAGCAAACGA
Human: CCTGTGACGTAGCAAACGA
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

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

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

**Cladogram**
- Taxon A
- Taxon B
- Taxon C
- Taxon D

**Chronogram**
- Taxon B
- Taxon C
- Taxon A
- Taxon D

**Phylogram**
- 6
- Taxon B
- 3
- 1
- 1
- Taxon C
- 5
- 1
- Taxon A
- 3
- Taxon D

**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. From Sequences To Distances
   - Sequence alignment
   - Pair-wise distance matrix
   - Tree building algorithms
   - Output tree

Character based

2. From Sequences To Distances
   - Sequence alignment
   - Pair-wise distance matrix
   - Tree building algorithms
   - Output tree

3. From alignments To phylogenies
   - Sequence alignment
   - Pair-wise distance matrix
   - Tree building algorithms
   - Output tree

4. Couple to tree proposal and scoring
   - Sequence alignment
   - Pair-wise distance matrix
   - Tree building algorithms
   - Output tree
Goals for today: Phylogenetics

- Basics of phylogeny: Introduction and definitions
  - Characters, traits, nodes, branches, lineages, topology, lengths
  - Gene trees, species trees, cladograms, chronograms, phylograms

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

2. From distances to trees: Tree-building algorithms
   - Tree types: Ultrametric, Additive, General Distances
   - Algorithms: UPGMA, Neighbor Joining, guarantees and limitations
   - Optimality: Least-squared error, minimum evolution (require search)

3. From alignments to trees: Alignment scoring given a tree
   - Parsimony: greedy (union/intersection) vs. DP (summing cost)
   - ML/MAP (includes back-mutations, lengths): peeling algorithm (DP)

4. Tree exploration: Markov Chain Monte Carlo tree search
   - Proposal operations: NNI neighbor interchange, SPR prune/regraft
   - MCMC: Metropolis-Hastings, sample posterior $P(B,T|D)$ without $P(D)$
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

- $N_{\text{actual mutations}} > N_{\text{observed substitutions}}$
  - 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
    - Go back to A from C,G,T with 0.1*0.1 each
  - In states C,G,T with prob. 0.16 each

<table>
<thead>
<tr>
<th></th>
<th>t=1</th>
<th>t=2</th>
<th>t=3</th>
<th>t=4</th>
<th>t=5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>0.7</td>
<td>0.52</td>
<td>0.412</td>
<td>0.3472</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0.1</td>
<td>0.16</td>
<td>0.196</td>
<td>0.2176</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0.1</td>
<td>0.16</td>
<td>0.196</td>
<td>0.2176</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>0.1</td>
<td>0.16</td>
<td>0.196</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

$$S(\Delta t) = \begin{bmatrix}
P(A|A, \Delta t) & \ldots & P(A|T, \Delta t) \\
\ldots & \ldots & \ldots \\
P(T|A, \Delta t) & \ldots & P(T|T, \Delta t)
\end{bmatrix}$$
Modeling Nucleotide Evolution

Reasonable assumption: multiplicative
(implying a stationary Markov process)

\[ S(t+t') = S(t)S(t') \]

That is, \[ P(x \mid y, t+t') = \sum_z P(x \mid z, t) \, P(z \mid y, t') \]

Jukes-Cantor: constant rate of evolution

For short time \( \varepsilon \), \( S(\varepsilon) = \begin{pmatrix}
1 - 3\alpha\varepsilon & \alpha\varepsilon & \alpha\varepsilon & \alpha\varepsilon \\
\alpha\varepsilon & 1 - 3\alpha\varepsilon & \alpha\varepsilon & \alpha\varepsilon \\
\alpha\varepsilon & \alpha\varepsilon & 1 - 3\alpha\varepsilon & \alpha\varepsilon \\
\alpha\varepsilon & \alpha\varepsilon & \alpha\varepsilon & 1 - 3\alpha\varepsilon
\end{pmatrix} \]
Modeling Nucleotide Evolution

**Jukes-Cantor:**

For longer times,

\[
S(t) = \begin{pmatrix}
    r(t) & s(t) & s(t) & s(t) \\
    s(t) & r(t) & s(t) & s(t) \\
    s(t) & s(t) & r(t) & s(t) \\
    s(t) & s(t) & s(t) & r(t)
\end{pmatrix}
\]

Where we can derive:

\[
r(t) = \frac{1}{4} \left( 1 + 3 \, e^{-4\alpha t} \right)
\]

\[
s(t) = \frac{1}{4} \left( 1 - e^{-4\alpha t} \right)
\]

Geometric asymptote to 1/4
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 $\beta$)

\[
S(t) = \begin{pmatrix}
A & G & C & T \\
A & r(t) & s(t) & u(t) & u(t) \\
G & s(t) & r(t) & u(t) & u(t) \\
C & u(t) & u(t) & r(t) & s(t) \\
T & u(t) & u(t) & s(t) & r(t)
\end{pmatrix}
\]

Where

\[
\begin{align*}
s(t) &= \frac{1}{4} (1 - e^{-4\beta t}) \\
u(t) &= \frac{1}{4} (1 + e^{-4\beta t} - e^{-2(\alpha+\beta)t}) \\
r(t) &= 1 - 2s(t) - u(t)
\end{align*}
\]
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):

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

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

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

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)

<table>
<thead>
<tr>
<th>Unequal Base Frequency</th>
<th>Equal Base Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 substitution types</td>
<td></td>
</tr>
<tr>
<td>GTR (general time reversible)</td>
<td>SYM (symmetric)</td>
</tr>
<tr>
<td>TrN (Tamura-Nei)</td>
<td>K3ST (Kimura 3 sub. type)</td>
</tr>
<tr>
<td>HKY85 (Hasegawa-Kishino-Yano)</td>
<td>1 transition</td>
</tr>
<tr>
<td>F84 (Felsenstein)</td>
<td>2 transversions</td>
</tr>
<tr>
<td></td>
<td>1 transition</td>
</tr>
<tr>
<td>3 substitution types</td>
<td></td>
</tr>
<tr>
<td>HKY85 (Hasegawa-Kishino-Yano)</td>
<td></td>
</tr>
<tr>
<td>F84 (Felsenstein)</td>
<td></td>
</tr>
<tr>
<td>2 substitution types</td>
<td></td>
</tr>
<tr>
<td>1 transition</td>
<td></td>
</tr>
<tr>
<td>1 transversion</td>
<td></td>
</tr>
<tr>
<td>1 substitution type</td>
<td></td>
</tr>
<tr>
<td>F81 (Felsenstein)</td>
<td></td>
</tr>
<tr>
<td>1 subsitution type</td>
<td></td>
</tr>
<tr>
<td>1 substitution type</td>
<td></td>
</tr>
<tr>
<td>F81 (Felsenstein)</td>
<td></td>
</tr>
<tr>
<td>1 substitution type</td>
<td></td>
</tr>
<tr>
<td>1 substitution type</td>
<td></td>
</tr>
</tbody>
</table>

Models also exist for peptides and codons
Goals for today: Phylogenetics

• Basics of phylogeny: Introduction and definitions
  – Characters, traits, nodes, branches, lineages, topology, lengths
  – Gene trees, species trees, cladograms, chronograms, phylograms

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

2. From distances to trees: Tree-building algorithms
  – Tree types: Ultrametric, Additive, General Distances
  – Algorithms: UPGMA, Neighbor Joining, guarantees and limitations
  – Optimality: Least-squared error, minimum evolution (require search)

3. From alignments to trees: Alignment scoring given a tree
  – Parsimony: greedy (union/intersection) vs. DP (summing cost)
  – ML/MAP (includes back-mutations, lengths): peeling algorithm (DP)

4. Tree exploration: Markov Chain Monte Carlo tree search
  – Proposal operations: NNI neighbor interchange, SPR prune/regraft
  – MCMC: Metropolis-Hastings, sample posterior \( P(B,T|D) \) without \( P(D) \)
2. Distance-based tree-building algorithms

Mapping a distance matrix to a tree

UPGMA, NJ, LSE, ME
### 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><strong>Human</strong></td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td><strong>Mouse</strong></td>
<td>h.y.m</td>
<td>0</td>
<td>3</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td><strong>Rat</strong></td>
<td>h.y.r</td>
<td>m.r</td>
<td>0</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td><strong>Dog</strong></td>
<td>h.z.x.d</td>
<td>m.y.z.x.d</td>
<td>r.y.z.x.d</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td><strong>Cat</strong></td>
<td>h.z.x.c</td>
<td>m.y.z.x.c</td>
<td>r.y.z.x.c</td>
<td>d.c</td>
<td>0</td>
</tr>
</tbody>
</table>

**Goal:**

Minimize discrepancy between observed distances and tree-based distances

**Tree implies a distance matrix**

\[ M_{ij} \]

**Map distances** \( D_{ij} \) **to a tree**

\[ \min \sum_{ij} (D_{ij} - M_{ij})^2 \]
Distances: (a) Ultrametric distances

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

  ![Diagram of ultrametric distances]

  where \( a \leq b \)

• 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:
    - $d(i,j) + d(k,l) \leq d(i,k) + d(j,l) = d(i,l) + d(j,k)$
    - $(a+b)+(c+d) \leq (a+m+c)+(b+m+d) = (a+m+d)+(b+m+c)$

- 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 $d_{ij}/2$
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:
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

\[ D_{ij} = d_{ij} - (r_i + r_j) \]

Where

\[ r_i = \frac{\sum_k d_{ik}}{|L| - 2} \]

**Claim:** The above “magic trick” ensures that \( D_{ij} \) is minimal \textit{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. $D_{ij}$ is minimal
Define a new node $k$, and set $d_{km} = \frac{1}{2} (d_{im} + d_{jm} - d_{ij})$ for all $m \in L$

Add $k$ to $T$, with edges of lengths $d_{ik} = \frac{1}{2} (d_{ij} + r_i - r_j)$
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 $d_{ij}$
Algorithms: (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

<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>m.r</td>
<td>0</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Dog</td>
<td>h.z.x.d</td>
<td>m.y.z.x.d</td>
<td>r.y.z.x.d</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Cat</td>
<td>h.z.x.c</td>
<td>m.y.z.x.c</td>
<td>r.y.z.x.c</td>
<td>d.c</td>
<td>0</td>
</tr>
</tbody>
</table>

Goal:
Minimize discrepancy between observed distances and tree-based distances.

Tree implies a distance matrix $M_{ij}$

Map distances $D_{ij}$ to a tree

$\min \sum_{ij} (D_{ij} - M_{ij})^2$
Goals for today: Phylogenetics

• Basics of phylogeny: Introduction and definitions
  – Characters, traits, nodes, branches, lineages, topology, lengths
  – Gene trees, species trees, cladograms, chronograms, phylograms

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

2. From distances to trees: Tree-building algorithms
  – Tree types: Ultrametric, Additive, General Distances
  – Algorithms: UPGMA, Neighbor Joining, guarantees and limitations
  – Optimality: Least-squared error, minimum evolution (require search)

3. From alignments to trees: Alignment scoring given a tree
  – Parsimony: greedy (union/intersection) vs. DP (summing cost)
  – ML/MAP (includes back-mutations, lengths): peeling algorithm (DP)

4. Tree exploration: Markov Chain Monte Carlo tree search
  – Proposal operations: NNI neighbor interchange, SPR prune/regraft
  – MCMC: Metropolis-Hastings, sample posterior P(B,T|D) without P(D)
3. Character-based tree-scoring algorithms
   3a: Parsimony (set-based)
   3b: Parsimony (Dyn. Prog.)
   3c: Maximum Likelihood
Basic algorithms of phylogenetic methods

Distance based

1. From Sequences To Distances
   - Sequence alignment
   - Pair-wise distance matrix
   - Output tree

Character based

2. Tree building algorithms
   - From alignments To phylogenies
   - Coupled to tree proposal and scoring
   - Output tree
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 $u$, $= C$
**Traceback:**

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 $R_i$

Easy to see that this traceback produces some assignment of cost $C$
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: Phylogenetics

• Basics of phylogeny: Introduction and definitions
  – Characters, traits, nodes, branches, lineages, topology, lengths
  – Gene trees, species trees, cladograms, chronograms, phylograms

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

2. From distances to trees: Tree-building algorithms
  – Tree types: Ultrametric, Additive, General Distances
  – Algorithms: UPGMA, Neighbor Joining, guarantees and limitations
  – Optimality: Least-squared error, minimum evolution (require search)

3. From alignments to trees: Alignment scoring given a tree
  – Parsimony: greedy (union/intersection) vs. DP (summing cost)
  – ML/MAP (includes back-mutations, lengths): peeling algorithm (DP)

4. Tree exploration: Markov Chain Monte Carlo tree search
  – Proposal operations: NNI neighbor interchange, SPR prune/regraft
  – MCMC: Metropolis-Hastings, sample posterior P(B,T|D) without P(D)
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)

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

- **D** = seq. alignment data
- **B** = branch lengths
- **T** = topology

### Maximum a Posteriori (MAP)

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

---

- **\(P(D|B,T)\)** 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

---

Compute recursively using DP
1. **Assume sites** $j$ **evolve independently.**
   - Treat each column of the alignment in isolation

2. **Assume branch independence**, conditioned on parent
   - Expand total joint probability into prod of $P(x_i|x_{\text{parent}(i)}, t_i)$
   - Only $P(x_{2n-1})$ remains, root prior, background nucl. freq.

3. **We know how to compute** $P(x_i|x_{\text{parent}(i)}, t_i)$ **for fixed pair**
   - Defined by our sequence model (JC, K2P, HKY, etc)
   - Easily calculate for any given assignment of internal nodes

4. **As internal node values are not known** → **marginalize**
   - Sum over all possible values of all internal/root nodes
   - Let $x_{n+1}, \ldots, x_{2n-1}$ represent seqs of $n-1$ internal nodes
1. Site evolution over single branch

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

JC is a Continuous-Time Markov Chain (CTMC)
- Defines instantaneous rates of transition between states (bases)

Discrete MC version
- Given time $t$, we define a discrete MC with transition matrix is $S(t)$, 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**:

$$P(a|b,t) = S(t) = \begin{pmatrix} r_t & s_t & s_t & s_t \\ s_t & r_t & s_t & s_t \\ s_t & s_t & r_t & s_t \\ s_t & s_t & s_t & r_t \end{pmatrix}$$

$$r_t = \frac{1}{4} (1 + 3e^{-4\alpha t})$$
$$s_t = \frac{1}{4} (1 - e^{-4\alpha t})$$

![Diagram of base transition](image.png)
2. Sequence evolution over single branch

- Assume site independence
  \[ P(x_i | x_k, t_i) = \prod_j P(b=x_{ij} | a=x_{kj}, t_i) \]

Use product to define **sequence evolution**:

\[ x_k = \text{“AAACTG”} \]
\[ x_i = \text{“CAAGTC”} \]
\[ t_i \]

\[ P(x_i | x_k, t_i) \]
3. Sequence evolution over entire tree

- Assume branch independence
  \[ P(x_1, \ldots, x_n, \ldots, x_{2n-1} | T, t) = P(x_{2n-1}) \prod_i P(x_i | x_{\text{parent}(i)}, t_i) \]

- Assume prior on root sequence, e.g.
  \[ P(x_{2n-1}) = P(x_{2n-1,j}) = (1/4)^m \] for sequence length \( m \)

Use product and prior to define \textit{sequence evolution over tree}:

\[ x_9 = \text{“AAACTG”} \]

\[ P(x_1, \ldots, x_n, \ldots, x_{2n-1} | T, t) \]
4. Integrate (marginalize) over hidden ancestral seqs!

- Notice, all sequences are needed, both internal nodes and leaves
  \[ P(x_1, \ldots, x_n, \ldots, x_{2n-1} | T, t) \]
- 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!
  \[ P(x_1, \ldots, x_n | T, t) = \sum_{x_{n+1}} \ldots \sum_{x_{2n-1}} P(x_1, \ldots, x_n, \ldots, x_{2n-1} | T, t) \]
- Don’t worry, dynamic programming can do it efficiently.
Basic trick to efficient marginalization

Apply factorization trick to every internal node in the tree.

\[
P(x_1, x_2, x_3, x_4 | T, t) = \sum_{x_5} \sum_{x_6} \sum_{x_7} P(x_1, x_2, x_3, x_4, x_5, x_6, x_7 | T, t)
\]
\[
= \sum_{x_5} \sum_{x_6} P(x_1 | x_5, t_1) P(x_2 | x_5, t_1) P(x_3 | x_6, t_3) P(x_4 | x_6, t_4)
\]
\[
= \sum_{x_7} P(x_7) \left[ \sum_{x_5} P(x_5 | x_7, t_5) P(x_1 | x_5, t_1) P(x_2 | x_5, t_1) \right]
\]
\[
[ \sum_{x_5} P(x_5 | x_7, t_5) P(x_1 | x_5, t_1) P(x_2 | x_5, t_1) ]
\]
\[
[ \sum_{x_6} P(x_6 | x_7, t_6) P(x_3 | x_6, t_3) P(x_4 | x_6, t_4) ]
\]

Peeling algorithm

- L(i,j,a) is the DP table.
- Each entry contains the probability of seeing the leaf data below node i, given that node i has base a at site j.
- The leaves of the table are initialized based on the observed sequence. Entries populated in post-order traversal.
- Runtime: O(2^n * k^2)
Use DP to compute argmax P(D|B,T) efficiently

• If we know the branch lengths $t_{\text{left}}$ & $t_{\text{right}}$.
• 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 $i$

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

$\text{Prob}(a \rightarrow b)$ $\text{Prob}(a \rightarrow c)$
### Initialization and Termination

- 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
  - Numerical: gradient descent, Newton’s method

<table>
<thead>
<tr>
<th></th>
<th>2n-1</th>
<th>...</th>
<th>...</th>
<th>i</th>
<th>j</th>
<th>k</th>
<th>...</th>
<th>n+1</th>
<th>n</th>
<th>...</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</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
  - Numerical: gradient descent, Newton’s method
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 intuitive as parsimony (e.g. may choose more events if they’re more likely in our probabilistic model).
- 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.
Goals for today: Phylogenetics

• Basics of phylogeny: Introduction and definitions
  – Characters, traits, nodes, branches, lineages, topology, lengths
  – Gene trees, species trees, cladograms, chronograms, phylograms

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

2. From distances to trees: Tree-building algorithms
  – Tree types: Ultrametric, Additive, General Distances
  – Algorithms: UPGMA, Neighbor Joining, guarantees and limitations
  – Optimality: Least-squared error, minimum evolution (require search)

3. From alignments to trees: Alignment scoring given a tree
  – Parsimony: greedy (union/intersection) vs. DP (summing cost)
  – ML/MAP (includes back-mutations, lengths): peeling algorithm (DP)

4. Tree exploration: Markov Chain Monte Carlo tree search
  – Proposal operations: NNI neighbor interchange, SPR prune/regraft
  – MCMC: Metropolis-Hastings, sample posterior $P(B, T|D)$ without $P(D)$
Basic algorithms of phylogenetic methods

Distance based

1. From Sequences To Distances
   - Sequence alignment
   - Pair-wise distance matrix
   - Tree building algorithms
   - Output tree

Character based

3. From alignments To phylogenies
   - Sequence alignment
   - Couple to tree proposal and scoring
   - Output tree
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

Initial Tree (NJ) → Propose Tree (NNI, SPR) → Compute Probability/likelihood → P (Score, likelihood, posterior prob., etc)
Searching through topologies: operations

Nearest neighbor interchange

Subtree pruning and regrafting

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

Note: NNIs alone can explore entire tree space

Tree space is big:
Number of unrooted topologies: \( N_u = 3 \times 5 \times 7 \times \ldots \times (2n - 5) = (2n - 5)!!. \)
Number of rooted topologies: \( N_r = (2n - 3) \times N_u = (2n - 3) \times (2n - 5)!! \)
\( N_u(3) = 1, \ N_u(4) = 3, \ N_u(5) = 15, \)
\( N_u(10) = 2\times10^6, \ N_u(20) = 2\times10^{20}, \ N_u(64) \ > \ googol \)
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

- Lets you convert any proposal rule that is a connected one (can reach every state) into one that visits the states according to their posterior
- The rule is:
  - If better: always take it
  - If worse:
    - Compute likelihood ratio $p \sim P(\text{proposal}) / P(\text{current})$ (if no priors)
    - With $p$: next becomes proposal, with $(1-p)$: next becomes current
  - Why ratio: cuz we don’t need to compute evidence $P(D)$ (see next slide)
- While any connected proposal rule is okay, in practice proposal strategy is an art.
  - Need good mixing $\rightarrow$ Big steps good
  - BUT: Don’t fall off cliffs $\rightarrow$ Small steps good
    - Ensure low rejection rate, i.e. your step size is not soooo big that you typically fall off the cliffs
  - Good rule of thumb is try it, if low rejection rate $\rightarrow$ good
The ratio you compute in the Metropolis-Hasting rule is actually more complicated than what previous slide describes. Say current state is \((B_1, T_1)\) and proposed next state is \((B_2, T_2)\). Then the ratio you look at is

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

If \(\alpha > 1\) then accept \((B_2, T_2)\) no matter what, and if its 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 be symmetric. Namely,

\[
P(B_1, T_1|B_2, T_2) = P(B_2, T_2|B_1, T_2)
\]

which means the Metropolis algorithm (no Hastings) is a special case of MH where we have

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

Now notice this is just ratio of posterior 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:

\[
\alpha = \frac{P(D, B_2, T_2)/P(D)}{P(D, B_1, T_1)/P(D)}
\]

\[
= \frac{P(D|B_2, T_2)P(B_2, T_2)}{P(D|B_1, T_1)P(B_1, T_1)}
\]

and its this 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; and that's the true advantage of MCMC, because \(P(D)\) is usually tough to calculate. In fact, MCMC is only used when \(P(D)\) can't be computed efficiently. If \(P(D)\) can be computed efficiently you would could just do

\[
P(D|B, T)P(B, T)/P(D) = P(B, T|D)
\]

and compute posterior probabilities directly.
Tree reliability: Bootstrapping

1. **Re-sample alignments:**
   - Randomly sample alignment columns with replacement
   - Create many alignments of equal size.

2. **Build a phylogenetic tree for each sample**

3. **Repeat (1) and (2) many times**
   - 1000s of times

4. **Output summary tree**
   - Tree constructed most frequently
   - Consensus tree (even if not most freq)
   - Other options

5. **Report observation frequency of each branch**
   - Each branch is a binary split
Basic algorithms of phylogenetic methods

Distance based

1. From Sequences To Distances
   - Sequence alignment
   - Pair-wise distance matrix
   - Tree building algorithms
   - Output tree

Character based

2. Tree building algorithms
3. From alignments To phylogenies
   - Sequence alignment
   - Output tree

4. Couple to tree proposal and scoring
Algorithms for reconstructing trees

- UPGMA (Sokal 1958) [distance]
- Least Square Error (Cavalli-Sforza & Edwards 1963) [dist]
- Maximum Parsimony (Fitch 1977) [character]
  - PHYLIP
- Maximum Likelihood (Felsenstein 1981) [character]
  - RAxML, PHYML, PHYLIP, PAUP
- Neighbor Joining (Saitou 1987) [distance]
  - BIONJ, QuickTree, RapidTree, FNJ
- Maximum a posteriori (Rannala & Yang 1996) [character]
  - MrBayes, BEAST
  - BEST, CoalMCMC, CoalHMM, PhyloNet, SPIMAP, SYNERGY, PrIME-GSR
Goals for today: Phylogenetics

- Basics of phylogeny: Introduction and definitions
  - Characters, traits, nodes, branches, lineages, topology, lengths
  - Gene trees, species trees, cladograms, chronograms, phylograms

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

2. From distances to trees: Tree-building algorithms
   - Tree types: Ultrametric, Additive, General Distances
   - Algorithms: UPGMA, Neighbor Joining, guarantees and limitations
   - Optimality: Least-squared error, minimum evolution (require search)

3. From alignments to trees: Alignment scoring given a tree
   - Parsimony: greedy (union/intersection) vs. DP (summing cost)
   - ML/MAP (includes back-mutations, lengths): peeling algorithm (DP)

4. Tree exploration: Markov Chain Monte Carlo tree search
   - Proposal operations: NNI neighbor interchange, SPR prune/regraft
   - MCMC: Metropolis-Hastings, sample posterior P(B,T|D) without P(D)