Goals:
1. Review dynamic programming algorithm for parsimony scoring
2. Relate inference on phylogenetic trees to HMMs, networks
3. Derive models of nucleotide substitution

Maximum parsimony scoring
- **Problem:** how well does a tree explain the data (multiple alignment)?
- **Idea:** use minimum number of substitutions $\Rightarrow$ best explanation
- **Greedy algorithm** (Fitch's algorithm): build up sets of possibilities at each internal node. Add cost of substitution if this is not possible. Recover min cost assignment starting from arbitrary nucleotide at root.
- This doesn't work for non-uniform substitution costs.

- Dynamic program:

$$C(j, i) = \begin{cases} 
0, & \text{if } j \text{ is a leaf and data has nucleotide } i \\
\infty, & \text{if } i \neq i \\
\min_k \left\{ C(\text{left}(j), k) + S(i, k) \right\} \\
+ \min_k \left\{ C(\text{right}(j), k) + S(i, k) \right\}, & \text{otherwise}
\end{cases}$$

- Implementation detail: easiest to annotate nodes with cost instead of building a matrix. Pre-order traversal ensures subproblems are complete as needed.
Maximum likelihood parameter estimation (Jukes-Cantor's algorithm)

- consider single site

- Phylogenetic tree is a bayesian network (children conditionally independent given parents)

- Conditional distributions come from nucleotide substitution model.

- Inference problem: \( \arg \max \ P(x_1, \ldots, x_n | \theta) \)

\( \theta \) = the branch lengths

- Observations: This is the same problem solved by Baum-Welch, HMMs are trees, so Baum-Welch & Jukes-Cantor are special cases of EM.
- In the **E** step, evaluate $p(x_1, \ldots, x_q | t_1, \ldots, t_8, S)$

- **Problem:** $x_6, \ldots, x_q$ unobserved

- **Solution:** Marginalize (sum over all values)

- **Problem:** This takes exponential time

- **Solution:** Dynamic programming

$$p(x_1, \ldots, x_q | t_1, \ldots, t_8, S) = p(x_q) p(x_1 | x_q, t_1) \ldots$$

$$\sum_{x_6, \ldots, x_q} p(x_1, \ldots, x_q | t_1, \ldots, t_8, S) = \sum_{x_q} p(x_q) \sum_{x_1, x_q} p(x_1 | x_q, t_1) \sum_{x_8} p(x_8 | x_q, t_8) \ldots$$

- **Idea:** each inner sum is a message (message passing, belief propagation)

  1. pass messages from root to leaves
  2. leaves to root
  3. marginalize by multiplying message and normalizing

- **Observation:** For HMMs: 1 = forward-algorithm, 2 = backward algorithm

- **Problem:** tree topology needs to be parameter. Exponentially many trees.

- **Idea:** Heuristic search (local edit operations: neighbor swapping)
- **Fucos–Cantor model**

- continuous time Markov chain:
  - transition rate matrix $Q$
  - probability matrix $P$ such that $\frac{d}{dt} P(t) = P(t) Q$
  - states $\{A, C, G, T\}$. Assume one site w/ A
  - rate matrix
    
    \[
    \begin{bmatrix}
    0 & 0 & \mu / 4 & 0 \\
    0 & 0 & 0 & \mu / 4 \\
    \mu / 4 & 0 & 0 & 0 \\
    0 & \mu / 4 & 0 & 0
    \end{bmatrix}
    \]

- initial state distribution $[\frac{1}{4}, \ldots, \frac{1}{4}]$

- assume $P(t) = P$ (substitution of nucle i $\rightarrow$ j in time $t$) constant

- for small $\varepsilon$, no back mutations $\Rightarrow$

  
  \[
  P(\varepsilon) = \begin{bmatrix}
  1 - 3d\varepsilon & d\varepsilon \\
  1 - 3d\varepsilon & d\varepsilon \\
  d\varepsilon & \ldots \\
  d\varepsilon & \ldots
  \end{bmatrix}
  \]

- by Markov property, $P(t + \varepsilon) = P(t) P(\varepsilon) = 

  \[
  \begin{bmatrix}
  r(t) & s(t) \\
  r(t) & s(t) \\
  s(t) & r(t)
  \end{bmatrix}
  \]

  \[
  r(t + \varepsilon) = r(t) (1 - 3d\varepsilon) + 3(d\varepsilon) s(t)
  
  s(t + \varepsilon) = s(t) (1 - d\varepsilon) + d\varepsilon r(t)
  \]

- from the def'n of derivative,

  \[
  r'(t) = 3d\varepsilon (s(t) - r(t))
  
  s'(t) = d\varepsilon (r(t) - s(t))
  \]
- \( r(0) = 1 \), \( s(0) = 0 \)

\[
\Rightarrow \quad r(t) = \frac{1}{4} \left( 1 + 3 \exp(-4at) \right)
\]

\[
s(t) = \frac{1}{4} \left( 1 - \exp(-4at) \right)
\]

- Recall: evolutionary distance from alignment
  - (no substitution: divergence \( t \) units ago)
  
  \[
  = r(2t) = \frac{1}{4} + \frac{3}{4} \exp(-8at)
  \]
  
  Evolution is time reversible

- Bases are independent, so proportion of substitutions

  \[
  d = s(2t) = \frac{3}{4} \left( 1 - \exp(-8at) \right)
  \]

- The expected number of substitutions

  \[
  k = 3d(2t)
  \]

- From \( d \), we can estimate \( k = 4dt \)

  \[
  \exp(-8at) = 1 - \frac{4}{3} d
  \]

  \[
  8at = -\ln \left( 1 - \frac{4}{3} d \right)
  \]

  \[
  k = -\frac{3}{4} \ln \left( 1 - \frac{4}{3} d \right)
  \]

- Note: PhyloCSF extends this to codon substitution model.