Lecture 16

• Evolutionary trees

• Tree-based probabilities for aligned sequences
Evolutionary trees

• Binary tree with
  – $n_{\text{leaf}}$ leaf nodes (observed individuals)
  – $n_{\text{anc}}$ ancestral nodes (unobserved)
• Each ancestral node has two descendants (‘left’ and ‘right’); leaves have none
• # edges:
  • # edge starts = 2 $n_{\text{anc}}$
  • # edge ends = $n_{\text{leaf}} + n_{\text{anc}} - 1$ (every node except root)
  • $2 n_{\text{anc}} = n_{\text{leaf}} + n_{\text{anc}} - 1$
  • $n_{\text{anc}} = n_{\text{leaf}} - 1$,  # edges = 2 $n_{\text{leaf}} - 2$
ancestral nodes

root node

leaf nodes
• Want to compute *probabilities* of observed leaf sequences, given tree
  – Allows discriminating between possible trees

• Requires
  – considering possible sequences at ancestral nodes
    • # grows exponentially in both $n_{anc}$ and sequence length !!
  – a probability model for change along edges
Mutational model for tree

• Will assume independent evolution at each sequence position
  – Doesn’t allow for context effects (e.g. CpG hotspots!)
• Mutations along an edge $e$:
  \[ P_e(s \mid r) = \text{prob a residue } r \text{ at beginning of } e \text{ is } s \text{ at end} \]
• ‘Background’ residue freqs at the root:
  \[ P_{\text{root}}(r) \]
• **Simplifying assumptions:**
  
  – (for DNA) \( P_e(s^\wedge \mid r^\wedge) = P_e(s \mid r) \)
    
    • \(^\wedge = \) complementary nuc
    
    • so each \( P_e \) has 6 independent params
  
  – A *single, reversible, infinitesimal* (~per small time unit) mutation model \( P_{inf} \) applies across entire tree
    
    • \( P_e = (P_{inf})^t \) where \( t = \) time along \( e \)
    
    • Reversibility implies root can’t be uniquely placed
    
    • This is model assumed by Siepel *et al.*
Probability calculations on tree

• Given:
  1. a set of observed residues at the leaves (a gap-free alignment column of the sequences)
  2. \( \{ P_e(s \mid r) \} \) and \( \{ P_{\text{root}}(s) \} \)

compute prob of observed residues

• Still exponentially many (in \( n_{\text{anc}} \)) possibilities for ancestral residues!

• But can use dynamic programming on a WDAG...
Evolut tree → WDAG

- Each *ancestral node* in tree becomes 4 nodes in WDAG
  - labelled with the 4 nucs
- *leaf nodes* remain unchanged
  - labelled with observed nuc
- Two nodes in WDAG are connected by an *edge*
  - if corresponding tree nodes are (but reverse direction)
    - weight = $P_e(s \mid r)$ where $e = \text{tree edge}$, $r, s = \text{node labels}$
- ‘urnode’
  - unlabelled
  - 4 edges coming from the 4 root nodes
  - weights = $P_{\text{root}}(s)$
urnode

$$P_{\text{root}}(s)$$

$$P_g(s \mid r)$$

$$P_h(G \mid r)$$

$$P_e(A \mid r)$$

$$P_f(G \mid r)$$
• Size of WDAG is linear in $n_{\text{leaf}}$
  – # nodes: $n_{\text{leaf}} + 4 \ n_{\text{anc}} + 1$
  – # edges: $4 \ n_{\text{leaf}} + 16 \ (n_{\text{anc}} - 1) + 4$

• Edges in tree point down; in WDAG, up
  – so WDAG ‘parents’ are below
cf. WDAG for 3-state HMM length n sequence (lecture 14)

weights are emission probabilities $e_k(b_i)$ for $i^{th}$ residue $b_i$

weights are transition probabilities $a_{kl}$

$b_{i-1}$ position $i-1$

$b_i$ position $i$

$b_{i+1}$ position $i+1$
For each vertex $v$, let $f(v) = \sum_{\text{paths } p \text{ ending at } v} \text{weight}(p)$, where \text{weight}(p) = \text{product} of edge weights in $p$. Only consider paths starting at ‘begin’ node.

Compute $f(v)$ by dynam. prog: $f(v) = \sum_i w_i f(v_i)$, where $v_i$ ranges over the parents of $v$, and $w_i = \text{weight of the edge from } v_i \text{ to } v$.

Similarly for $b(v) = \sum_{p \text{ beginning at } v} \text{weight}(p)$

The paths $\text{beginning}$ at $v$ are the ones $\text{ending}$ at $v$ in the reverse (or inverted) graph.
\[ f(v)b(v) = \text{sum of the path weights of all paths through } v. \]

\[ f(v')wb(v) = \text{sum of the path weights of all paths through the edge } (v',v) \]
• Compute overall *probability* of leaf residues (nucleotides) by *dynamic programming* on WDAG:

• Let, for each node $v$, $f(v) =$ prob of leaf nuc *below* $v$ (i.e. tree-descendants, or WDAG-ancestors, of $v$), given $v$’s nuc

  $f_{\text{left}}(v) =$ prob of leaf nuc *below* and to *left*

  $f_{\text{right}}(v) =$ prob of leaf nuc *below* and to *right*

  then $f(v) = f_{\text{left}}(v) f_{\text{right}}(v)$
• Compute these values node-by-node, visiting (WDAG-)parents before children:
  – starting at leaf nodes (setting \( f(v) = 1 \)), ending at urnode

\[
f_{left}(v) = \sum_{left - u} w(u, v) f(u)
\]

where

– \( u \) ranges over parent nodes to the left
– \( w(u, v) \) = weight on edge from \( u \) to \( v \)

(= mutation prob from \( v \) to \( u \))

Similarly for \( f_{right}(v) \)

\[
f(v) = f_{left}(v) f_{right}(v)
\]

– For \( v = \) urnode, view all parents as being to ‘left’ and \( f(v) = f_{left}(v) \)

• \( f(urnode) = \) probability of the observed leaf nucs
• a ‘forward-backward’ calc gives posterior prob of having
  – a particular nuc at an ancestral node, or
  – a particular mutational change along an edge
• can use these as *fractional counts* to estimate $P$’s (EM algorithm)
Probability models & alignments

• Getting the probability model $P_e$ requires a multiple alignment

• But optimal (LLR) scoring for alignment uses $P_e$:
  \[
  \log\left(\frac{\text{prob of col} \mid P_e \text{ model}}{\text{prob of col} \mid \text{background}}\right)
  \]

• Find $P_e$, alignment jointly & iteratively (Sankoff):
  – crude alignment $\rightarrow P_e \rightarrow$ scores $\rightarrow$ better alignment etc