Probabilistic methods for phylogenetic tree reconstruction

BMI/CS 576
www.biostat.wisc.edu/bmi576
Colin Dewey
colin.dewey@wisc.edu

Downsides to parsimony methods

- Scoring function parameters (costs for substitutions) are rather arbitrary
  - The most "parsimonious" tree critically depends on these parameters
- Parsimony methods require assignments of character states to the ancestral nodes
  - Only considers score of best assignment, which may not be the true one

Alternative to parsimony: probabilistic-model based tree scoring

- Instead of cost *S(a,b)* of a substitution occurring along a branch, we will use a probability *P(child = a | parent = b)*
- For a given tree, instead of finding a *minimal cost assignment* to the ancestral nodes, we will *sum the probabilities of all possible ancestral states*
- Instead of finding a tree with *minimum cost* will will find a tree the *maximizes likelihood* (probability of the data given the tree)

Probabilistic model setup

- We observe *n* sequences, $x^1,\ldots,x^n$
- We are given a tree *T* and want to model $P(x^1,\ldots,x^n \mid T)$
  - This is the *likelihood* (probability of the observed sequences given the model, the tree)
- For simplicity, we'll just consider the case that our sequences are of length 1 (just one character)
- To generalize to longer sequences, we assume *independence* of each position (each column of an ungapped multiple alignment)
  - Probability of sequences = product of probability of each position/column

Probabilistic model details

- It will be easier to first consider a model in which we represent the states of the internal nodes of the tree with random variables: $X^{n+1},\ldots,X^{2n-1}$ (assuming rooted binary tree)
- Then the probability of any particular configuration of states at all nodes in the tree will be defined as

$$P(x^1,\ldots,x^{2n-1} \mid T) = q_{x^{2n-1}} \prod_{i=1}^{2n-2} P(x^i \mid x^{\alpha(i)})$$

- $q_{x^{2n-1}}$ is the prior probability of the state of the root node
- $\alpha(i)$ is the index of the parent node of node *i*
- Key assumption: state of node *i* is conditionally independent of the states of its ancestors given the state of its parent
- For simplicity, we are ignoring branch lengths for now

The likelihood

- We only care about the probability of the observed (extant) sequences
- Need to marginalize (sum over possible values of ancestral states) to obtain the likelihood

$$P(x^1,\ldots,x^n \mid T) = \sum_{x^{n+1},\ldots,x^{2n-1}} q_{x^{2n-1}} \prod_{i=1}^{2n-2} P(x^i \mid x^{\alpha(i)})$$

- But there is an exponential number of terms in this sum!

Felsenstein's algorithm

- Dynamic programming to the rescue once again!
- Subproblem: $P(L_k|a)$: probability of the leaves below node k, given that the residue at k is a
- Recurrence:

$$P(L_k \mid a) = \sum_{b,c} P(b \mid a) P(L_i \mid b) P(c \mid a) P(L_j \mid c)$$

$$= \sum_{b} P(b \mid a) P(L_i \mid b) \sum_{c} P(c \mid a) P(L_j \mid c)$$

- where i and j are the children nodes of k
- b and c represent the states of node i and node j, respectively

Felsenstein's algorithm

- Initialize: $k=2n-1$
- Recursion:
  - If k is a leaf node,

$$P(L_k|a) = \begin{cases} 1 \text{ if } a = x^k \\ 0 \text{ otherwise} \end{cases}$$

  - Else, compute $P(L_i|a)$ and $P(L_j|a)$ for all a at daughters i and j

$$P(L_k \mid a) = \sum_{b} P(b \mid a) P(L_i \mid b) \sum_{c} P(c \mid a) P(L_j \mid c)$$

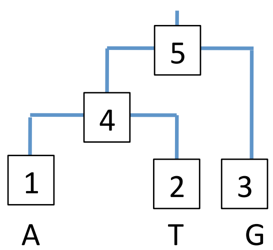
- Termination
  - Likelihood is equal to

$$\sum_{a} P(L^{2n-1}|a) q_a$$

Concluding remarks on probabilistic-model (likelihood) based approach

- Very similar to the weighted parsimony case
  - Main differences are at
    - Leaf nodes
    - Minimization versus summation for internal nodes
- Can it be used to infer ancestral states as well?
  - Instead of summing, we would maximize
  - As in the parsimony case, we would need to keep track of the maximizing assignment
- Substitution probabilities P(a|b) can be derived from principled mathematical models and/or estimated from data

What is probability for the following set of residues



b

a	A	C	G	T
A	0.7	0.1	0.1	0.1
C	0.1	0.7	0.1	0.1
G	0.1	0.1	0.7	0.1
T	0.1	0.1	0.1	0.7

Assume the above conditional probability matrix P(b|a) for all branches

The probabilities computed for each node

	A	C	G	T
$P(L_1\|x)$	1	0	0	0
$P(L_2\|x)$	0	0	0	1
$P(L_3\|x)$	0	0	1	0
$P(L_4\|x)$	0.07	0.01	0.01	0.07
$P(L_5\|x)$	0.0058	0.0022	0.0154	0.0058

Probability of sequence given tree is 0.25(0.0058+0.0022+0.0154 + 0.0058)=0.0073