Statistical Alignment with *k*-Restricted Steiner Trees

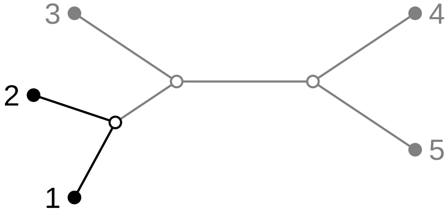
R.B. Lyngsø[1], J. Nielsen[2], C.N.S. Pedersen[2] & J. Hein[1]

[1]University of Oxford [2]Aarhus University

27th of June, 2008

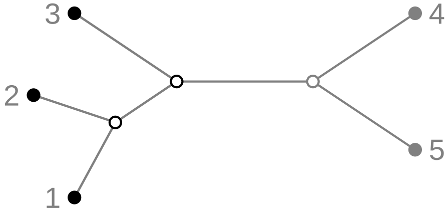
Bayesian Phylogenetics

Statistical Alignment. . .



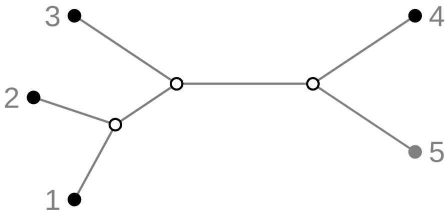
$$P(D) = \sum_{x} P(x \rightarrow 1) P(x \rightarrow 2)$$

Statistical Alignment. . .



$$P(D) = \sum_{x} P(x \rightarrow 1)P(x \rightarrow 2) \sum_{y} P(y \rightarrow x)P(y \rightarrow 3)$$

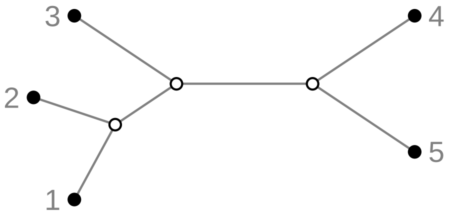
Statistical Alignment. . .



$$P(D) = \sum_x P(x \to 1)P(x \to 2) \sum_y P(y \to x)P(y \to 3)$$

$$\sum_z (z \to y)P(z \to 4)$$

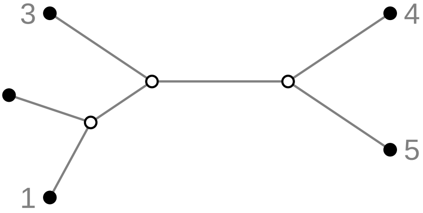
Statistical Alignment. . .



$$P(D) = \sum_x P(x \to 1) P(x \to 2) \sum_y P(y \to x) P(y \to 3)$$

$$\sum_z (z \to y) P(z \to 4) P(5 \to z) P(5)$$

Statistical Alignment. . .

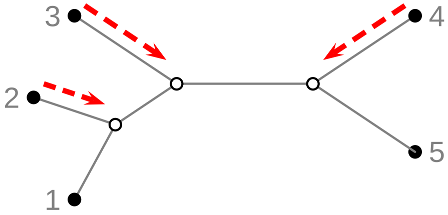


The infinite summations can be handled relatively efficiently, but still cause run-time to be exponential in number of sequences

$$P(D) = \sum_x P(x \to 1)P(x \to 2) \sum_y P(y \to x)P(y \to 3)$$

$$\sum_z (z \to y)P(z \to 4)P(5 \to z)P(5)$$

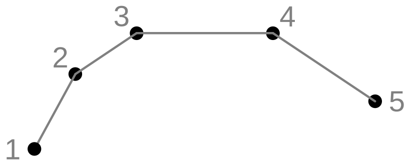
. . . with 2-Restricted Steiner Trees



$$P(D) = \sum_x P(x \to 1)P(x \to 2) \sum_y P(y \to x)P(y \to 3)$$

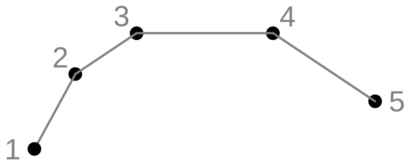
$$\sum_z (z \to y)P(z \to 4)P(5 \to z)P(5)$$

. . . with 2-Restricted Steiner Trees



$$P(D) = P(2 \to 1)P(3 \to 2)P(4 \to 3)P(5 \to 4)P(5)$$

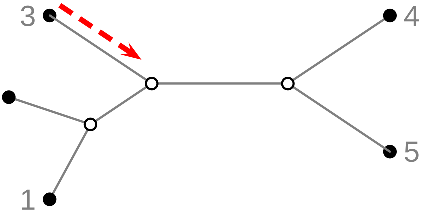
. . . with 2-Restricted Steiner Trees



By ‘moving’ observed sequences to internal nodes the problem simplifies to a series of pairwise alignments

$$P(D) = P(2 \to 1)P(3 \to 2)P(4 \to 3)P(5 \to 4)P(5)$$

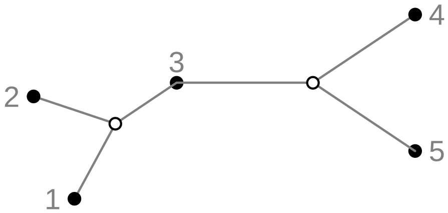
. . . with *k*-Restricted Steiner Trees



$$P(D) = \sum_x P(x \to 1)P(x \to 2) \sum_y P(y \to x)P(y \to 3)$$

$$\sum_z (z \to y)P(z \to 4)P(5 \to z)P(5)$$

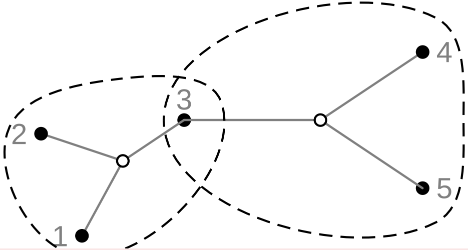
. . . with k-Restricted Steiner Trees



$$P(D) = \sum_x P(x \to 1)P(x \to 2)P(3 \to x)$$

$$\sum_z P(z \to 3)P(z \to 4)P(5 \to z)P(5)$$

. . . with k-Restricted Steiner Trees



If we can align k sequences we only need to 'move' enough sequences to separate the full phylogeny into components with at most k observed sequences

$$P(D) = \sum_x P(x \rightarrow 1)P(x \rightarrow 2)P(3 \rightarrow x)$$

$$\sum_z P(z \rightarrow 3)P(z \rightarrow 4)P(5 \rightarrow z)P(5)$$

Steiner Tree Appr. (Robins & Zelikovsky, 2000)

```
Input: A set of terminal nodes $S$ and optimal Steiner trees on all subsets of $S$ of size at most $k$
$T$ = minimum spanning tree on $S$
$H$ = complete graph on $S$
while there is a $k$-restricted full component $K$ with $\text{gain}_T(K) > 0$
    Find $k$-restricted full component $K$ with maximal $\text{gain}_K(T) / \text{loss}(K)$
    $H = H \cup K$
    $T$ = minimum spanning tree on $T \cup K_{\text{loss-contracted}}$
Output the minimum spanning tree on $H$
```

Steiner Tree Appr. (Robins & Zelikovsky, 2000)

Input: A set of terminal nodes S and optimal Steiner trees on all subsets of S of size at most k

```
$T$ = minimum spanning tree on $S$
$H$ = complete graph on $S$
while there is a $k$-restricted full component $K$ with $\mathrm{gain}_T(K) > 0$
    Find $k$-restricted full component $K$ with maximal $\mathrm{gain}_K(T)/\mathrm{loss}(K)$
    $H = H \cup K$
    $T$ = minimum spanning tree on $T \cup K_{\text{loss-contracted}}$
Output the minimum spanning tree on $H$
```

$\mathrm{gain}_K(T)$ and $\mathrm{loss}(K)$ measures improvement and worst case eventual loss of adding component K to current solution T. They can be computed from the log probabilities of evolution on edges.

Steiner Tree Appr. (Robins & Zelikovsky, 2000)

Input: A set of terminal nodes S and optimal Steiner trees on all subsets of S of size at most k

```
$T$ = minimum spanning tree on $S$
$H$ = complete graph on $S$
while there is a $k$-restricted full component $K$ with $\mathrm{gain}_T(K) > 0$
    Find $k$-restricted full component $K$ with maximal $\mathrm{gain}_K(T)/\mathrm{loss}(K)$
    $H = H \cup K$
    $T$ = minimum spanning tree on $T \cup K_{\text{loss-contracted}}$
Output the minimum spanning tree on $H$
```

Guaranteed approximation ratio of 1.55 – currently being implemented for statistical alignment

Drawbacks of Algorithm

- Need alignments of all subsets of up to k observed sequences, so for m sequences of length n running time is $\sum_{i=2}^{k} \binom{m}{i} n^i$
- Approximation ratio is on log scale, so we are only guaranteed to find a tree with probability at least $P^{1/1.55}$ where P is the data probability computed on the optimal tree
- Approximation ratio of 1.55 is obtained for $k \to \infty$; for e.g. $k = 8$ the approximation ratio is 1.74

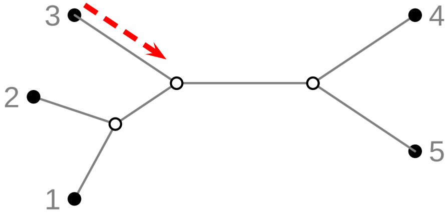
Drawbacks of Algorithm

- Need alignments of all subsets of up to k observed sequences, so for m sequences of length n running time is $\sum_{i=2}^{k} \binom{m}{i} n^i$
- Approximation ratio is on log scale, so we are only guaranteed to find a tree with probability at least $P^{1/1.55}$ where P is the data probability computed on the optimal tree
- Approximation ratio of 1.55 is obtained for $k \to \infty$; for e.g. $k = 8$ the approximation ratio is 1.74

Drawbacks of Algorithm

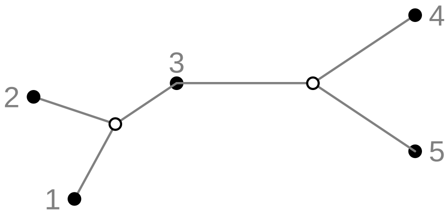
- Need alignments of all subsets of up to k observed sequences, so for m sequences of length n running time is $\sum_{i=2}^{k} \binom{m}{i} n^i$
- Approximation ratio is on log scale, so we are only guaranteed to find a tree with probability at least $P^{1/1.55}$ where P is the data probability computed on the optimal tree
- Approximation ratio of 1.55 is obtained for $k \rightarrow \infty$; for e.g. $k = 8$ the approximation ratio is 1.74

Alternative Approach



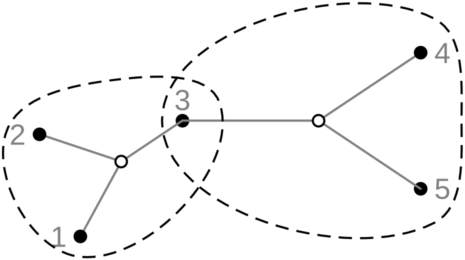
Start from externally predicted phylogeny, e.g. obtained using neighbour-joining on pairwise distances

Alternative Approach



Find a series of edge contractions such that the final tree is a *k*-restricted Steiner tree, i.e. the largest set of sequences that needs to be aligned is *k*

Alternative Approach



Align each subset of sequences thus identified and combine alignments to a global alignment

Choosing Edge Contractions

Observation

Contracting an edge not incident to an observed sequence will not alter the subsets of sequences that have to be aligned (and we should never contract an edge incident to two observed sequences)

General Procedure

As we may postpone contracting an edge until it is incident to an observed sequence, for each observed sequence we can identify the edges it has allowed contraction of. These will constitute a subtree rooted at the observed sequence, and iff the set of edge contractions is optimal this forest will be a minimal *almost* spanning forest

Choosing Edge Contractions

Observation

Contracting an edge not incident to an observed sequence will not alter the subsets of sequences that have to be aligned (and we should never contract an edge incident to two observed sequences)

General Procedure

As we may postpone contracting an edge until it is incident to an observed sequence, for each observed sequence we can identify the edges it has allowed contraction of. These will constitute a subtree rooted at the observed sequence, and iff the set of edge contractions is optimal this forest will be a minimal *almost* spanning forest

Integer Linear Programming Formulation

For edge e and observed sequence s let x_e indicate whether e is contracted and $y_{e,s}$ whether s allowed the contraction of e. Let $\mathcal{T}$ denote the set of minimal subtrees in the original phylogeny that need to contain at least one contracted edge.

Then we need to maximise

$$\sum_e \text{weight}_e x_e$$

subject to constraints

$\forall e : \sum_s y_{e,s} = x_e$
$\forall e, s : y_{e,s} \leq y_{e',s}$ where e' is the next edge from e towards s
$\forall T \in \mathcal{T} : \sum_{e \in T} x_e \geq 1$

Conclusions

- General method is standard technique in Steiner tree approximation
- Starting from an external phylogeny, in particular, will make it feasible to align tens or hundreds of sequences (depending on subset size) within a rigorous framework
- Quality of inferences are likely to be highest for parameters, lowest for trees, with alignments somewhere in between
- Software still in development