Week 2: Searching for trees, ancestral states

Genome 570

January, 2010
Greedy search for a maximum

If start here
Greedy search for a maximum
Greedy search for a maximum
Greedy search for a maximum

If start here
Greedy search for a maximum

If start here
Greedy search for a maximum

If start here
Greedy search for a maximum
Greedy search for a maximum

If start here
Greedy search for a maximum

If start here
Greedy search for a maximum

If start here
Greedy search for a maximum
Greedy search for a maximum

If start here
Greedy search for a maximum

If start here
Greedy search for a maximum

If start here
Greedy search for a maximum

If start here
Greedy search for a maximum

If start here

end up here
Greedy search for a maximum

end up here

but global maximum is here

If start here
Nearest-neighbor rearrangements

A subtree

is rearranged by dissolving the connections to an interior branch

and reforming them in one of the two possible alternative ways:
The Schoenberg graph
With numbers of steps of trees
Subtree pruning and regrafting
Subtree pruning and regrafting

Week 2: Searching for trees, ancestral states – p.23/52
Subtree pruning and regrafting

Break a branch, remove a subtree
Subtree pruning and regrafting

1. Break a branch, remove a subtree
2. Add it in, attaching it to one (*) of the other branches

Week 2: Searching for trees, ancestral states – p.25/52
Subtree pruning and regrafting

Break a branch, remove a subtree

Add it in, attaching it to one (*) of the other branches
Subtree pruning and regrafting

Break a branch, remove a subtree

Add it in, attaching it to one (*) of the other branches

Here is the result:
Tree bisection and reconnection
Tree bisection and reconnection
Tree bisection and reconnection

Break a branch, separate the subtrees
Tree bisection and reconnection

Break a branch, separate the subtrees

Connect a branch of one to a branch of the other

Week 2: Searching for trees, ancestral states – p.31/52
Tree bisection and reconnection

Break a branch, separate the subtrees

Connect a branch of one to a branch of the other
Tree bisection and reconnection

Break a branch, separate the subtrees

Connect a branch of one to a branch of the other

Here is the result:
Goloboff's economy in rearranging
Sequential addition
Star decomposition
An example of looking for the Shortest Hamiltonian path

(a) (b) (c) (d)
A search tree of trees

(1,2,3,4,5,6,7,8,10,9)  (1,2,3,4,5,6,7,9,10,8)  (1,2,3,4,5,6,7,10,9,8)
(1,2,3,4,5,6,7,8,9,10)  (1,2,3,4,5,6,7,9,8,10)  (1,2,3,4,5,6,7,10,8,9)
add 10  add 9  add 10  add 8  add 9  add 8
add 9  add 8  add 10  add 8  add 9
add 8  add 9  add 10
add 3  add 4  add 5  etc.
add 2  add 3  add 4  add 5  etc.
add 1  add 2  add 3  etc.

etc.

etc.

etc.

etc.

etc.

etc.
**Time-savings of branch and bound**

Results for this case:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>length</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy search from all points</td>
<td>2.802660</td>
<td>(too fast to measure)</td>
</tr>
<tr>
<td>Exhaustive enumeration</td>
<td>2.781230</td>
<td>10.85 sec</td>
</tr>
<tr>
<td>Branch and bound</td>
<td>2.781230</td>
<td>0.46 sec</td>
</tr>
</tbody>
</table>
Branch and bound on trees

Week 2: Searching for trees, ancestral states – p.40/52
Branch and bound using numbers of steps
Calculating a lower bound on tree score

- The score of the partial tree is a lower bound (since adding more species cannot decrease the number of steps)

- Also can add the number of characters that do not show variation on the species added so far, but will once added (actually, the number of new states that will appear once all species are added – if A and G are there already, will C also appear?)

- Can also take all disjoint pairs of characters that will become incompatible once added, but aren’t incompatible now (this is due to Dave Swofford. Each brings in one more step.)
How does the time taken by an algorithm depend on the size of the problem? If it is a polynomial (even one with big coefficients), with a big enough case it is faster than one that depends on the size exponentially.
NP completeness and NP hardness

P = problems that can be solved by a polynomial time algorithm

NP complete = problems for which a proposed solution can be checked in polynomial time but for which it can be proven that if one of them is in P, all are.

NP hard = problems for which a solution can be checked in polynomial time, but might be not solvable in polynomial time.
Parsimony reconstruction of ancestral states

\[
\begin{align*}
S(1) & \quad S(2) & \quad S(3) & \quad S(4) \\
\end{align*}
\]

\[
\begin{align*}
c_{21} & \quad 0 & \quad c_{23} & \quad c_{24} \\
\end{align*}
\]
Ancestral states in the example

Week 2: Searching for trees, ancestral states – p.46/52
One reconstruction

{C}  {A}  {G}

0  0  0

3.5 3.5 1 5 1 5

3.5 3.5 3.5 4.5

6  6  7  8

2.5

2.5 2.5 3.5 3.5

2.5

2.5

1

0

0

0

1 5 1 5

3.5 3.5 3.5 4.5

0

Week 2: Searching for trees, ancestral states – p.47/52
For the other choice, two possibilities

\begin{align*}
\{C\} & \quad 0 && 0 \\
\{A\} & \quad 0 && 0 \\
\{C\} & \quad 0 && 0 \\
\{A\} & \quad 0 && 0 \\
\{G\} & \quad 0 & & 0 \\
\end{align*}
One of these

\[
\begin{align*}
\{C\} & \quad \{A\} \\
0 & \quad 0 \\
& \quad 0 \\
0 & \quad 0 \\
2.5 & \quad 2.5 \\
3.5 & \quad 3.5 \\
3.5 & \quad 4.5 \\
6 & \quad 6 \\
7 & \quad 8 \\
\{G\} & 
\end{align*}
\]
The other one

```
\{C\}  \{A\}  \{C\}  \{A\}  \{G\}
0     0     0     0     0
2.5   2.5   3.5   3.5   0
2.5   2.5   3.5   3.5   0
0     0     0     0     0
```

Week 2: Searching for trees, ancestral states – p.50/52
There can be multiple tied reconstructions
Average branch lengths over all reconstructions