CS 466
Introduction to Bioinformatics
Lecture 16

Mohammed El-Kebir
October 31, 2018
Course Announcements

Discuss HW3 Grading: Thursday, Nov 1, 11-12
(whiteboard on 3rd floor by elevator)
Outline

• Two-State Perfect Phylogeny
• Multi-State Perfect Phylogeny
• Large Maximum Parsimony Phylogeny Problem
• Summary

Reading:
• Lecture notes
**Binary Characters**

<table>
<thead>
<tr>
<th>Species</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0 1 1 0 0</td>
</tr>
<tr>
<td>B</td>
<td>0 0 1 1 0</td>
</tr>
<tr>
<td>C</td>
<td>1 1 1 1 0</td>
</tr>
<tr>
<td>D</td>
<td>1 1 0 1 1</td>
</tr>
</tbody>
</table>

**Question**: Given $n$ binary characters, what is the smallest parsimony score?

Characters only have two possible states

- Possible Encoding: 
  0 : not-mutated
  1 : mutated

- Possible Encoding: 
  0 : no wings
  1 : wings
Infinite Sites Model = Two-state Perfect Phylogeny

The genome is large

Mutations are rare

**Infinite sites model:** multiple mutations never occur at the same position

[Kimura, 1969]

<table>
<thead>
<tr>
<th>Species (cancer cells)</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutated Loci</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0: not</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1: mutated</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

All sites are bi-allelic: mutated or not.
Two-state Perfect Phylogeny

Matrix $M \in \{0, 1\}^{n \times m}$ has $n$ taxa and $m$ characters

- Taxon $f$ has state 1 for character $c$ $\iff f$ possesses character $c$

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
<th>$c_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$r_4$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$r_5$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Definition

A perfect phylogeny for $M$ is a rooted tree $T$ with $n$ leaves such that:

1. Each taxon labels only one leaf
2. Each character labels only one edge
3. Character possessed by a taxon are on unique path to root

Root node is all zero ancestor
Two-state Perfect Phylogeny Problem

**Input:**

Matrix $M \in \{0, 1\}^{n \times m}$ has $n$ taxa and $m$ characters

- Taxon $f$ has state 1 for character $c$ if and only if $f$ possesses character $c$

\[
\begin{array}{|c|ccccc|}
\hline
& c_1 & c_2 & c_3 & c_4 & c_5 \\
\hline
r_1 & 1 & 1 & 0 & 0 & 0 \\
r_2 & 0 & 0 & 1 & 0 & 0 \\
r_3 & 1 & 1 & 0 & 0 & 1 \\
r_4 & 0 & 0 & 1 & 1 & 0 \\
r_5 & 0 & 1 & 0 & 0 & 0 \\
\hline
\end{array}
\]

**Problem**

*Given* $M \in \{0, 1\}^{n \times m}$ *does* $M$ *have a perfect phylogeny?*
Try it yourself!

Only one of these matrices can be used to build a perfect phylogeny.

(1) As a group, **decide on an approach** to try to determine which one is which.

(2) Try out your approach to see if you can construct the tree.

(3) What did you learn from your attempt?

\[
\begin{array}{c|ccccc}
\text{Species} & A & B & C & D & E \\
\hline
\text{Characters} & & & & & \\
C_1 & 0 & 1 & 1 & 0 & 1 \\
C_2 & 0 & 0 & 1 & 1 & 0 \\
C_3 & 1 & 1 & 0 & 0 & 1 \\
C_4 & 0 & 0 & 1 & 1 & 0 \\
C_5 & 1 & 1 & 0 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{c|ccccc}
\text{Species} & A & B & C & D & E \\
\hline
\text{Characters} & & & & & \\
C_1 & 0 & 0 & 1 & 1 & 0 \\
C_2 & 0 & 0 & 1 & 0 & 1 \\
C_3 & 1 & 1 & 0 & 0 & 1 \\
C_4 & 1 & 1 & 0 & 0 & 0 \\
C_5 & 0 & 1 & 0 & 0 & 1 \\
\end{array}
\]
The Perfect Phylogeny Problem – Preliminaries

Problem

Given $M \in \{0, 1\}^{n \times m}$ does $M$ have a perfect phylogeny?

Definition

$I(c)$ is the set of taxa that possess character $c$; and $\sigma(f)$ is the set of characters possessed by taxon $f$.

\[
\begin{array}{c|ccccc}
 r_1 & c_1 & c_2 & c_3 & c_4 & c_5 \\
 r_2 & 0 & 0 & 1 & 0 & 0 \\
r_3 & 1 & 1 & 0 & 0 & 1 \\
r_4 & 0 & 0 & 1 & 1 & 0 \\
r_5 & 0 & 1 & 0 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{c|ccccc}
 r_1 & c_1 & c_2 & c_3 & c_4 & c_5 \\
 r_2 & 1 & 1 & 0 & 0 & 0 \\
r_3 & 0 & 0 & 1 & 0 & 0 \\
r_4 & 1 & 1 & 0 & 1 & 0 \\
r_5 & 0 & 0 & 1 & 0 & 1 \\
\end{array}
\]

$I(c_1) = \{r_1, r_3\}$

$\sigma(r_1) = \{c_1, c_2\}$

Sort columns of $M$ s.t. $c < d$ iff $|I(c)| \geq |I(d)|$. Break ties arbitrarily.
Consider rows of $M$ iteratively
- $T_i$ is tree of first $i$ rows of $M$

$T_1$ is a path graph
- Terminal nodes $r$ and 1
- $|\sigma(1)| + 1$ edges labeled by $\sigma(1)$

\[
\begin{array}{ccccc}
\text{r}_1 & c_1 & c_2 & c_3 & c_4 & c_5 \\
1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

$c < d$ iff $|l(c)| \geq |l(d)|$
Consider rows of $M$ iteratively
- $T_i$ is tree of first $i$ rows of $M$

$T_1$ is a path graph
- Terminal nodes $r$ and 1
- $|\sigma(1)| + 1$ edges labeled by $\sigma(1)$

$T_{i+1}$ is a supertree of $T_i$
- Let $v$ be last node on walk from $r$ matching characters $\sigma(i + 1)$
  - Character $d$ is the last match
  - Unmatched characters $\tau(i + 1)$

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
<th>$c_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$r_4$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$r_5$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

$c < d$ iff $|l(c)| \geq |l(d)|$
Consider rows of $M$ iteratively
  - $T_i$ is tree of first $i$ rows of $M$

$T_1$ is a path graph
  - Terminal nodes $r$ and $1$
  - $|\sigma(1)| + 1$ edges labeled by $\sigma(1)$

$T_{i+1}$ is a supertree of $T_i$
  - Let $v$ be last node on walk from $r$
    matching characters $\sigma(i + 1)$
      - Character $d$ is the last match
      - Unmatched characters $\tau(i + 1)$
  - Extend $T_i$ with path $\Pi$
    - $\Pi$ has terminals $v$ and $i + 1$
    - $\Pi$ has $|\tau(i + 1)| + 1$ edges labeled by $\tau(i + 1)$

$c < d$ iff $|I(c)| \geq |I(d)|$
Consider rows of $M$ iteratively

- $T_i$ is tree of first $i$ rows of $M$

$T_1$ is a path graph

- Terminal nodes $r$ and 1
- $|\sigma(1)| + 1$ edges labeled by $\sigma(1)$

$T_{i+1}$ is a supertree of $T_i$

- Let $v$ be last node on walk from $r$ matching characters $\sigma(i + 1)$
  - Character $d$ is the last match
  - Unmatched characters $\tau(i + 1)$
- Extend $T_i$ with path $\Pi$
  - $\Pi$ has terminals $v$ and $i + 1$
  - $\Pi$ has $|\tau(i + 1)| + 1$ edges labeled by $\tau(i + 1)$

$c < d$ iff $|l(c)| \geq |l(d)|$
Consider rows of $M$ iteratively
  ▶ $T_i$ is tree of first $i$ rows of $M$

$T_1$ is a path graph
  ▶ Terminal nodes $r$ and 1
  ▶ $|\sigma(1)| + 1$ edges labeled by $\sigma(1)$

$T_{i+1}$ is a supertree of $T_i$
  ▶ Let $v$ be last node on walk from $r$ matching characters $\sigma(i + 1)$
    ★ Character $d$ is the last match
    ★ Unmatched characters $\tau(i + 1)$
  ▶ Extend $T_i$ with path $\Pi$
    ★ $\Pi$ has terminals $v$ and $i + 1$
    ★ $\Pi$ has $|\tau(i + 1)| + 1$ edges labeled by $\tau(i + 1)$

$\ c < d \text{ iff } |l(c)| \geq |l(d)|$

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
<th>$c_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_3$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$r_4$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$r_5$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Lemma**

Let $M_i \in 0, 1^{i \times m}$ be a submatrix of $M$. If $M$ is conflict-free then $T_i$ is a perfect phylogeny for $M_i$. 
Outline

• Two-State Perfect Phylogeny
• Multi-State Perfect Phylogeny
• Large Maximum Parsimony Phylogeny Problem
• Summary

Reading:
• Lecture notes
### Integer Characters

#### Question: Given $n$ integer characters with $k$ states, what is the smallest parsimony score?

<table>
<thead>
<tr>
<th>Species</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Characters have $k$ possible states.
Infinite Alleles Model = Multi-state Perfect Phylogeny

**Infinite alleles model:**
- For any mutation, there are an infinite number of possibilities of what mutation looks like (states).
- So, the same position can be mutated multiple times, but it never mutates to the same “allele” or state.

**Site History:**
- Characters have integer states
Infinite Alleles Model = Multi-state Perfect Phylogeny

Infinite alleles model:
• For any mutation, there are an infinite number of possibilities of what mutation looks like (states).
• So, the same position can be mutated multiple times, but it never mutates to the same “allele” or state.

Site History:
Characters have integer states
Infinite Alleles Model = Multi-state Perfect Phylogeny

**Infinite alleles model:**
- For any mutation, there are an infinite number of possibilities of what mutation looks like (states).
- So, the same position can be mutated multiple times, but it never mutates to the same “allele” or state.

**Site History:**
- Characters have integer states
Multi-state Perfect Phylogeny

Matrix $M \in \{0, \ldots, k - 1\}^{n \times m}$ has $n$ taxa and $m$ characters

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1$</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$r_2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_3$</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>$r_4$</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$r_5$</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Definition

A multi-state perfect phylogeny for $M$ is a tree $T$ with $n$ leaves such that:

1. Each taxon labels exactly one leaf
2. Each node is labeled by $\{0, \ldots, k - 1\}^m$
3. Nodes labeled with state $i$ for character $c$ form a connected subtree $T_c(i)$

Theorem (Bodlaender et al., 1992) [Bodlaender, Fellows and Warnow]

For general $k$, the multi-state perfect phylogeny problem is NP-complete
Cladistic vs. Qualitative Characters

Definition

A multi-state perfect phylogeny for $M$ is a tree $T$ with $n$ leaves such that:

1. Each taxon labels exactly one leaf
2. Each node is labeled by $\{0, \ldots, k - 1\}^m$
3. Nodes with state $i$ for character $c$ form a connected subtree $T_c(i)$

A cladistic character $c$ has a state tree $t_c$ on its states

A phylogeny $T$ is consistent if the reduced tree $\sigma(T, c)$ is identical with $t_c$ for all $c$
Cladistic vs. Qualitative Characters

Definition

A multi-state perfect phylogeny for $M$ is a tree $T$ with $n$ leaves such that:

1. Each taxon labels exactly one leaf
2. Each node is labeled by $\{0, \ldots, k-1\}^m$
3. Nodes with state $i$ for character $c$ form a connected subtree $T_c(i)$

A cladistic character $c$ has a state tree $t_c$ on its states

A phylogeny $T$ is consistent if the reduced tree $\sigma(T, c)$ is identical with $t_c$ for all $c$
Cladistic vs. Qualitative Characters

**Definition**

A multi-state perfect phylogeny for \( M \) is a tree \( T \) with \( n \) leaves such that:

1. Each taxon labels exactly one leaf
2. Each node is labeled by \( \{0, \ldots, k - 1\}^m \)
3. Nodes with state \( i \) for character \( c \) form a connected subtree \( T_c(i) \)

A cladistic character \( c \) has a state tree \( t_c \) on its states

A phylogeny \( T \) is consistent if the reduced tree \( \sigma(T, c) \) is identical with \( t_c \) for all \( c \)
Multi-state Cladistic Perfect Phylogeny
Outline

• Two-State Perfect Phylogeny
• Multi-State Perfect Phylogeny
• Large Maximum Parsimony Phylogeny Problem
• Summary

Reading:
• Lecture notes
Small and a Large Problem

Small Maximum Parsimony Phylogeny Problem:
Given \( m \times n \) matrix \( A = [a_{i,j}] \) and tree \( T \) with \( m \) leaves, find assignment of character states to each internal vertex of \( T \) with minimum parsimony score.

Large Maximum Parsimony Phylogeny Problem:
Given \( m \times n \) matrix \( A = [a_{i,j}] \), find a tree \( T \) with \( m \) leaves labeled according to \( A \) and an assignment of character states to each internal vertex of \( T \) with minimum parsimony score.
General Large Maximum Parsimony Phylogeny

- This problem is NP-hard
- Heuristics using local search (tree moves)

1. Start with an arbitrary tree $T$.
2. Check “neighbors” of $T$.
3. Move to a neighbor if it provides the best improvement in parsimony/likelihood score.

Caveats:
Could be stuck in **local** optimum, and not achieve global optimum
Example: Nearest-Neighbor Interchange (NNI)

Rearrange four subtrees defined by one internal edge

Figure: Jones and Pevzner
Outline

• Two-State Perfect Phylogeny
• Multi-State Perfect Phylogeny
• Large Maximum Parsimony Phylogeny Problem
• Summary

Reading:
• Lecture notes
Distance-based Phylogeny

• Small additive distance phylogeny problem
  • In P
  • Recursive algorithm using neighboring leaves

• Large additive distance phylogeny problem
  • In P -- two algorithms:
    1. Find degenerate triples and resolve these
    2. Neighbor joining: identifies neighboring leaves even when tree is not given
  • Complete characterization of additive matrices using the four-point condition

Character-based Phylogeny

• Small maximum parsimony problem
  • Sankoff algorithm: dynamic programming

• Two-state perfect phylogeny problem
  • In P: O(mn) time
  • Complete characterization as conflict free binary matrices

• Multi-state perfect phylogeny problem
  • NP-hard in general
  • In P given state trees

• Large maximum parsimony problem
  • NP-hard
  • Heuristic using local search
Discuss HW3 Grading: Thursday, Nov 1, 11-12
(whiteboard on 3rd floor by elevator)