Who am I?

- Faculty at Genome Sciences
- Computational systems biologist
- Training: CS, high-tech, biology
- Research interests: Complex biological networks | Evolutionary dynamics | Microbial communities and metagenomics

What will change?

- Not much!
- **Informatics**: From sequence to genes and to systems
- **Programming**:
  - More emphasis on design and coding practices
  - Tip of the day
  - Coding style
A quick review

Trees:
- Represent sequence relationships
- A sequence tree has a topology and branch lengths (distances)
- The number of tree topologies grows very fast!

Distance trees
- Compute pairwise corrected distances
- Build tree by sequential clustering algorithm (UPGMA or Neighbor-Joining).
- These algorithms don't consider all tree topologies, so they are very fast, even for large trees.
“Maximum Parsimony Algorithm”

A fundamentally different method:

Instead of **reconstructing** a tree, we will **search** for the best tree.
“Pluralitas non est ponenda sine necessitate”
(Maximum) Parsimony Principle

- "Pluralitas non est ponenda sine necessitate" (plurality should not be posited without necessity)
  
- Occam’s Razor: Of two equivalent theories or explanations, all other things being equal, the simpler one is to be preferred.

- "when you hear hoof beats, think horses, not zebras"
  
- The KISS principle: "Keep It Simple, Stupid!"

- “Make everything as simple as possible, but not simpler”
Parsimony principle for Phylogenetic Trees

Find the tree that requires the fewest evolutionary changes!
Consider 4 Species

human
chimp
gorilla
orangutan
Consider 4 Species

Sequence data:

<table>
<thead>
<tr>
<th></th>
<th>human</th>
<th>chimp</th>
<th>gorilla</th>
<th>orangutan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>agtctc</td>
<td>agagtc</td>
<td>cggcag</td>
<td>cgggac</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The same approach would work for any discrete property that can be associated with the various species:

- Gene content (presence/absence of each gene)
- Morphological features (e.g., “has wings”, purple or white flowers)
- Numerical features (e.g., number of bristles)
Consider 4 Species

Sequence data:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>human</td>
<td>agtctc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>chimp</td>
<td>agagtc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gorilla</td>
<td>cggcag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>orangutan</td>
<td>cgggac</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

positions in alignment (usually called "sites")

Parsimony Algorithm

1) Construct all possible trees
2) For each site in the alignment and for each tree count the minimal number of changes required
3) Add all sites up to obtain the total number of changes for each tree
4) Pick the tree with the lowest score
Consider 4 Species

Sequence data:

<table>
<thead>
<tr>
<th></th>
<th>human</th>
<th>123456</th>
</tr>
</thead>
<tbody>
<tr>
<td>chimp</td>
<td>agtctc</td>
<td></td>
</tr>
<tr>
<td>gorilla</td>
<td>cgccag</td>
<td></td>
</tr>
<tr>
<td>orangutan</td>
<td>cgggac</td>
<td></td>
</tr>
</tbody>
</table>

All possible unrooted trees:

- H closest to C
- H closest to G
- H closest to O
Consider 4 Species

Sequence data:

All possible unrooted trees:

For each site and for each tree count the minimal number of changes required:
Consider site 1

What is the minimal number of evolutionary changes that can account for the observed pattern?

(Note: This is the “small parsimony” problem)
Consider site 1

What is the minimal number of evolutionary changes that can account for the observed pattern?

(Note: This is the “small parsimony” problem)
Consider site 1
Consider site 1
Consider site 2

Uninformative (no changes)
Consider site 3

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>human</td>
<td>a</td>
<td>g</td>
<td>t</td>
<td>c</td>
<td>t</td>
<td>c</td>
</tr>
<tr>
<td>chimp</td>
<td>a</td>
<td>g</td>
<td>a</td>
<td>g</td>
<td>t</td>
<td>c</td>
</tr>
<tr>
<td>gorilla</td>
<td>c</td>
<td>g</td>
<td>g</td>
<td>c</td>
<td>a</td>
<td>g</td>
</tr>
<tr>
<td>orangutan</td>
<td>c</td>
<td>g</td>
<td>g</td>
<td>a</td>
<td>c</td>
<td></td>
</tr>
</tbody>
</table>
Put sites 1 and 3 together.
Now put all of them together

Which tree is the most parsimonious?

parsimony score
The parsimony algorithm

1) *Construct all possible trees*

2) *For each site in the alignment and for each tree count the minimal number of changes required*

3) *Add all sites up to obtain the total number of changes for each tree*

4) *Pick the tree with the lowest score*
The parsimony algorithm

1) Construct all possible trees

2) For each site in the alignment and for each tree count the minimal number of changes required

3) Add all sites up to obtain the total number of changes for each tree

4) Pick the tree with the lowest score
The parsimony algorithm

1) Construct all possible trees

2) For each site in the alignment and for each tree count the minimal number of changes required

3) Add all sites up to obtain the total number of changes for each tree

4) Pick the tree with the lowest score

Too many!

How?
The parsimony algorithm

1) Construct all possible trees

2) For each site in the alignment and for each tree count the minimal number of changes required

3) Add all sites up to obtain the total number of changes for each tree

4) Pick the tree with the lowest score

Too many!

Search algorithm

How?

Fitch’s algorithm