Inferring Phylogenetic Trees

(Slides courtesy of Dr. Mark Craven)
Phylogenetic Inference: Task Definition

• Given
  – data characterizing a set of species/genes

• Do
  – infer a *phylogenetic tree* that accurately characterizes the evolutionary lineages among the species/genes
Motivation

• why construct trees?
  – to understand lineage of various species
  – to understand how various functions evolved
  – to inform multiple alignments
  – to identify what is most conserved/important in some class of sequences
Example Species Tree: Ichthyosaurs

Image from the Ichthyosaur Page (www.ucmp.berkeley.edu/people/motani/ichthyo/)
Phylogenetic Tree Basics

- leaves represent things (genes, individuals/strains, species) being compared
  - the term *taxon* (*taxa* plural) is used to refer to these when they represent species and broader classifications of organisms
- internal nodes are hypothetical ancestral units
- in a *rooted* tree, path from root to a node represents an evolutionary path
  - the root represents the common ancestor
- an *unrooted* tree specifies relationships among things, but not evolutionary paths
Data for Building Trees

• trees can be constructed from various types of data
  – *distance-based*: measures of distance between species/genes
  – *character-based*: morphological features (e.g. # legs), DNA/protein sequences
  – *gene-order*: linear order of orthologous genes in given genomes
Rooted vs. Unrooted Trees

![Diagram of rooted and unrooted trees](image-url)
Phylogenetic Tree Approaches

- three general types of methods
  - *distance*: find tree that accounts for estimated evolutionary distances
  - *parsimony*: find the tree that requires minimum number of changes to explain the data
  - *maximum likelihood*: find the tree that maximizes the likelihood of the data
**Distance Based Approaches**

- **given**: an \( n \times n \) matrix \( M \) where \( M_{ij} \) is the distance between objects \( i \) and \( j \)
- **do**: build an edge-weighted tree such that the distances between leaves \( i \) and \( j \) correspond to \( M_{ij} \)

\[
\begin{array}{cccccc}
& A & B & C & D & E \\
A & 0 & 8 & 8 & 5 & 3 \\
B & & 0 & 3 & 8 & 8 \\
C & & & 0 & 8 & 8 \\
D & & & & 0 & 5 \\
E & & & & & 0 \\
\end{array}
\]
Parsimony Based Approaches

given: character-based data

do: find tree that explains the data with a minimal number of changes

• focus is on finding the right tree topology, not on estimating branch lengths
Parsimony Example

- there are various trees that could explain the phylogeny of the sequences AAG, AAA, GGA, AGA including these two:

- parsimony prefers the first tree because it requires fewer substitution events
Parsimony Based Approaches

• usually these approaches involve two separate components
  – a search through the space of trees
  – a procedure to find the minimum number of changes needed to explain the data (for a given tree topology)

• first, we’ll talk about the latter aspect, and then we’ll talk about the search process
Finding Minimum Number of Changes for a Given Tree

• Fitch’s algorithm [1971]
  – assumes any state (e.g. nucleotide, amino acid) can convert to any other state
  – assumes positions are independent
Fitch’s Algorithm

1. traverse tree from leaves to root determining set of possible states (e.g. nucleotides) for each internal node

2. traverse tree from root to leaves picking ancestral states for internal nodes
Fitch’s Algorithm: Step 1
Possible States for Internal Nodes

• do a post-order (from leaves to root) traversal of tree
• determine possible states $R_i$ of internal node $i$ with children $j$ and $k$

$$R_i = \begin{cases} R_j \cup R_k, & \text{if } R_j \cap R_k = \emptyset \\ R_j \cap R_k, & \text{otherwise} \end{cases}$$

• this step calculates the number of changes required
  $\# \text{ of changes} = \# \text{ union operations}$
Fitch’s Algorithm: Step 1 Example

\{C \ T\} \cap \{A \ G \ T\} = \{T\}

\{C\} \cup \{T\} = \{CT\}

{CT} → T

{A \ G \ T} = \{T\}

{C} \cup \{T\} = \{CT\}

CT → T

GT → A

A GT → T
Fitch’s Algorithm: Step 2
Select States for Internal Nodes

- do a pre-order (from root to leaves) traversal of tree
- select state \( r_j \) of internal node \( j \) with parent \( i \)

\[
r_j = \begin{cases} r_i, & \text{if } r_i \in R_j \\ \text{arbitrary state} \in R_j, & \text{otherwise} \end{cases}
\]
Fitch’s Algorithm: Step 2