Today’s material (from Chapters 1-3)

- Newick strings
- Representing rooted trees using clades and rooted triplet trees
- Constructing a rooted tree from its set of clades using Hasse Diagrams
- Constructing a rooted tree from rooted triplet trees using Aho, Sagiv, Szymanski, and Ullman
- Constructing a rooted tree from rooted subtrees of any size
Today’s material (from Chapters 1-3)

- Newick strings
- Representing rooted trees using clades and rooted triplet trees
- Constructing a rooted tree from its set of clades using Hasse Diagrams
- Constructing a rooted tree from rooted triplet trees using Aho, Sagiv, Szymanski, and Ullman
- Constructing a rooted tree from rooted subtrees of any size
Newick representations

• For a rooted tree, we represent a graph with a string with the taxa, commas, and nested parentheses.

• For example, what tree is represented by $(a,(b,(c,((d,e),(f,g))))))$?

• How do we represent an unrooted tree? (Easy - root it somewhere, and write down the Newick representation of the rooted version.)
(U,((V,W),(X,Y)))

or

((X,Y),(U,(V,W)))

or ...

U

V  W

X  Y
Rooted vs. unrooted

• Task: be able to move between rooted and unrooted representations of trees
• Task: be able to compare two trees and see if they are different or the same
Today’s material (from Chapters 1-3)

• Newick strings
• Representing rooted trees using clades and rooted triplet trees
• Constructing a rooted tree from its set of clades using Hasse Diagrams
• Constructing a rooted tree from rooted triplet trees using Aho, Sagiv, Szymanski, and Ullman
• Constructing a rooted tree from rooted subtrees of any size
Triplet Trees

Definition:
Let T be a rooted tree leaf-labelled by S.
A triplet tree is a rooted 3-leaf subtree of T, such as \(((a,b),c)\) (also written as \(ab|c\)).

The set of all triplet trees of T is denoted \(\text{Triplets}(T)\).
Triplet trees of a rooted tree

6 \binom{3}{3} = 20 \text{ triplet trees}

ab|c, ab|d, ab|e, ab|f
ef|a, ef|b, ef|c, ef|d
de|a, de|b, de|c
df|a, df|b, df|c
cd|a, cd|b
ce|a, ce|b
cf|a, cf|b
Clades

Definition:
Let $T$ be a rooted tree leaf-labelled by $S$, let $v$ an internal node in $T$, and let $X_v$ be the set of leaves in $T$ below $v$.

Let $\text{Clades}(T) = \{X_v: v \text{ in } V(T)\}$.

Note: $X_v$ is also called the “cluster” at node $v$, so this is sometimes called Clusters($T$).
Clades of a rooted tree

Singletons \{a\}, \{b\}, ..., \{f\}
Full set \{a,b,c,d,e,f\}

Non-trivial clades:
\{e,f\}, \{a,b\}, \{d,e,f\}, \{c,d,e,f\}
Constructing trees from clades or triplet trees

• Problem 1: Given a set of subsets. Is there a tree that has all these subsets as clades?
• Problem 2: Given a set of triplet trees. Is there a tree that has all the triplet trees?

Note (important): We would like a polynomial time algorithm that does not require that we have ALL the clades or ALL the triplet trees.
Computing rooted trees from clades

• (If necessary): add in the full set S and singletons
• Partially order the set of clades by containment,
• and compute the Hasse Diagram of the resultant poset (partially ordered subset)

Note: Hasse Diagrams and Partially Ordered Sets are explained in Appendix B in the textbook.
Example: Constructing from clades

Algorithm: Construct “Hasse Diagram”

1. Draw directed graph (nodes are clades, directed edge reflects subset relationship)
   • Given clades $X$, $Y$, put an edge $X \to Y$ if $X$ is a proper subset of $Y$.

2. Remove the “unnecessary edges” (implied by transitivity, e.g., delete $\{e,f\} \to \{c,d,e,f\}$)

What does this graph look like?
Di-Graph for Partially Ordered Set

Singletons \{a\}, \{b\}, ..., \{f\}

Full set \{a, b, c, d, e, f\}

Non-trivial clades:
\{e, f\}, \{a, b\}, \{d, e, f\}, \{c, d, e, f\}

Red edges can be eraased
Hasse diagram!

Singletons \{a\}, \{b\}, ..., \{f\}

Full set \{a,b,c,d,e,f\}

Non-trivial clades:
\{e,f\}, \{a,b\}, \{d,e,f\}, \{c,d,e,f\}
Clades of T -> Hasse Diagram -> T

Singletons \{a\}, \{b\}, ..., \{f\}
Full set \{a,b,c,d,e,f\}

Non-trivial clades:
\{e,f\}, \{a,b\}, \{d,e,f\}, \{c,d,e,f\}
Clade compatibility

• Theorem: Let $X$ be a set of subsets of $S$ (containing all singletons and the full set). Then there exists a tree $T$ such that $X = \text{Clades}(T)$ if and only if for all $A, B$ in $X$, either $A$ and $B$ are disjoint, or one contains the other.

Proof: One direction is easy. The other direction is slightly harder.

**Corollary 1:** Given $X = \text{Clades}(T)$, the Hasse Diagram is $T$

**Corollary 2:** The Hasse Diagram algorithm answers correctly whether $X$ is a compatible set of clades
Question

• Suppose I give you an arbitrary set of subsets, and ask you: Is there a rooted tree that has these subsets as clades?
  – Example: \{a,b\}, \{b,c\}, \{a,b,c\}, \{a\}, \{b\}, \{c\}

• Can you answer this problem correctly?

• What is the running time?
Small changes

• Suppose you only have a subset of the non-trivial clades. For example: \{a,b\}, \{e,f\}, \{c,d,e,f\}
  – What would happen? What does the HASSE diagram look like?

• Suppose you have this input: Non-trivial clades: \{a,b\}, \{e,f\}, \{d,e,f\}, \{c,d,e,f\}, \{a,d,e\}
  – What would happen? What does the HASSE diagram look like?
Tree construction from clades

We presented an algorithm (Hasse Diagram)

Questions:

• Accuracy if given all the clades?
• Does it always produce binary trees?
• What if you are missing some clades? (for example, singletons?)
• What is the running time?
Today’s material (from Chapters 1-3)

• Newick strings
• Representing rooted trees using clades and rooted triplet trees
• **Constructing a rooted tree from its set of clades using Hasse Diagrams**
• Constructing a rooted tree from rooted triplet trees using Aho, Sagiv, Szymanski, and Ullman
• Constructing a rooted tree from rooted subtrees of any size
Today’s material (from Chapters 1-3)

• Newick strings
• Representing rooted trees using clades and rooted triplet trees
• Constructing a rooted tree from its set of clades using Hasse Diagrams
• Constructing a rooted tree from rooted triplet trees using Aho, Sagiv, Szymanski, and Ullman
• Constructing a rooted tree from rooted subtrees of any size
Triplet trees of a rooted tree

6 \binom{3}{3} = 20 triplet trees

ab|c, ab|d, ab|e, ab|f
ef|a, ef|b, ef|c, ef|d
de|a, de|b, de|c
df|a, df|b, df|c
cd|a, cd|b
ce|a, ce|b
cf|a, cf|b
Suppose you see only the rooted triplet trees (and maybe not all the rooted trees).

I ask: Is there a tree that has all these rooted triplet trees?

Can you answer this question correctly, and in polynomial time?

6 \binom{3}{3} = 20 \text{ triplet trees}

ab|c, ab|d, ab|e, ab|f
ef|a, ef|b, ef|c, ef|d
de|a, de|b, de|c
df|a, df|b, df|c
cd|a, cd|b
ce|a, ce|b
cf|a, cf|b
Rooted Tree Compatibility

• Input: Set $X$ of rooted trees, not all on the same set of leaves.
• Output: Tree $T$ (if it exists) that agrees with all the trees in $X$, and otherwise “Fail”

This problem is solvable in polynomial time. Proof: the Aho, Sagiv, Szymanski, and Ullman (ASSU) algorithm!
ASSU

• Given set of rooted triplet trees, we want to know if there is a rooted tree that agrees with all the triplet trees.
  – Example: ab | c, bc | d, cd | e: YES
  – Example: ab | c, bc | d, ad | c: NO
ASSU

• Given set of rooted triplet trees, we want to know if there is a rooted tree that agrees with all the triplet trees.

• Approach: Assume there is a binary tree that agrees with the triplet trees, and try to construct it. Then check.
• Given set of rooted triplet trees, we want to know if there is a rooted tree that agrees with all the triplet trees.

• Key insight: If $xy|z$ is a triplet tree, then $x$ and $y$ must be on the same side of the root of the binary tree that agrees with $xy|z$.  
  – Why?
ASSU

• Given set of rooted triplet trees, we want to know if there is a rooted tree that agrees with all the triplet trees.

• Key insight: Make a graph $G=(V,E)$ with every "species" a vertex in $V$ and include edge $(x,y)$ if some triplet tree $xy|z$ is in the input. This graph must have at least two components. – Why?
ASSU

• Algorithm:
  – If number of leaves is 2, return sibling pair
  – Else: construct graph (previous slide)
  – If there is only one component, reject and exit (no tree exists)
  – Else:
    • make two groups A and B of taxa (each component in one group)
    • Recurse on each group (only including triplets that are contained within a single group), producing trees $T_A$ and $T_B$
    • Return tree obtained by making $T_A$ and $T_B$ both children of the root
ASSU

• Try the algorithm on this input:
  – AB|C, BC|D, AD|H, AH|E, EF|G, FG|H
• Your first graph should have components
  – {A,B,C,D,H} and {E,F,G}
• When you recurse on {E,F,G} there is only one triplet that matters: EF|G
• When you recurse on {A,B,C,D,H} the triplets that matter are AB|C, BC|D, and AD|H
ASSU algorithm (alternative version, produces non-binary trees)

Given set X of k triplet trees on n species:

• If n>1, then construct graph with each species one of the vertices, and edges (a,b) for triplets ab|c.

• If the graph has a single component, reject (the set is not compatible); else recurse on each component, and return tree formed by making the rooted trees on the components each a subtree off the root of the returned tree.
Today’s material (from Chapters 1-3)

• Newick strings
• Representing rooted trees using clades and rooted triplet trees
• Constructing a rooted tree from its set of clades using Hasse Diagrams
• Constructing a rooted tree from rooted triplet trees using Aho, Sagiv, Szymanski, and Ullman
• Constructing a rooted tree from rooted subtrees of any size
Today’s material (from Chapters 1-3)

• Newick strings
• Representing rooted trees using clades and rooted triplet trees
• Constructing a rooted tree from its set of clades using Hasse Diagrams
• Constructing a rooted tree from rooted triplet trees using Aho, Sagiv, Szymanski, and Ullman
• Constructing a rooted tree from rooted subtrees of any size
Compatibility of rooted trees

• Suppose the input is a set $X$ of rooted trees (not necessarily triplet trees).
Compatibility of rooted trees

• Suppose the input is a set $X$ of rooted trees (not necessarily triplet trees).
• Can we use ASSU to determine if $X$ is compatible, and to compute a compatibility supertree for $X$?
Compatibility of rooted trees

• Suppose the input is a set $X$ of rooted trees (not necessarily triplet trees).

• Can we use ASSU to determine if $X$ is compatible, and to compute a compatibility supertree for $X$?

• Solution: YES, just encode each rooted tree in $X$ by its set of rooted triplet trees (or some subset of these that suffices to define each tree in $X$), and then run ASSU.
Summary of today’s lecture

• We have seen how to construct a rooted tree from its set of clades or triplet trees.
• We have seen how to test compatibility of a set of clades or rooted trees.
Parting thoughts: Rooted != Unrooted

ALSO: Please think about corresponding approaches for unrooted trees.

• Remember the All Quartets algorithm.
• What if we only had a subset of the quartet trees?
• Would it still work?