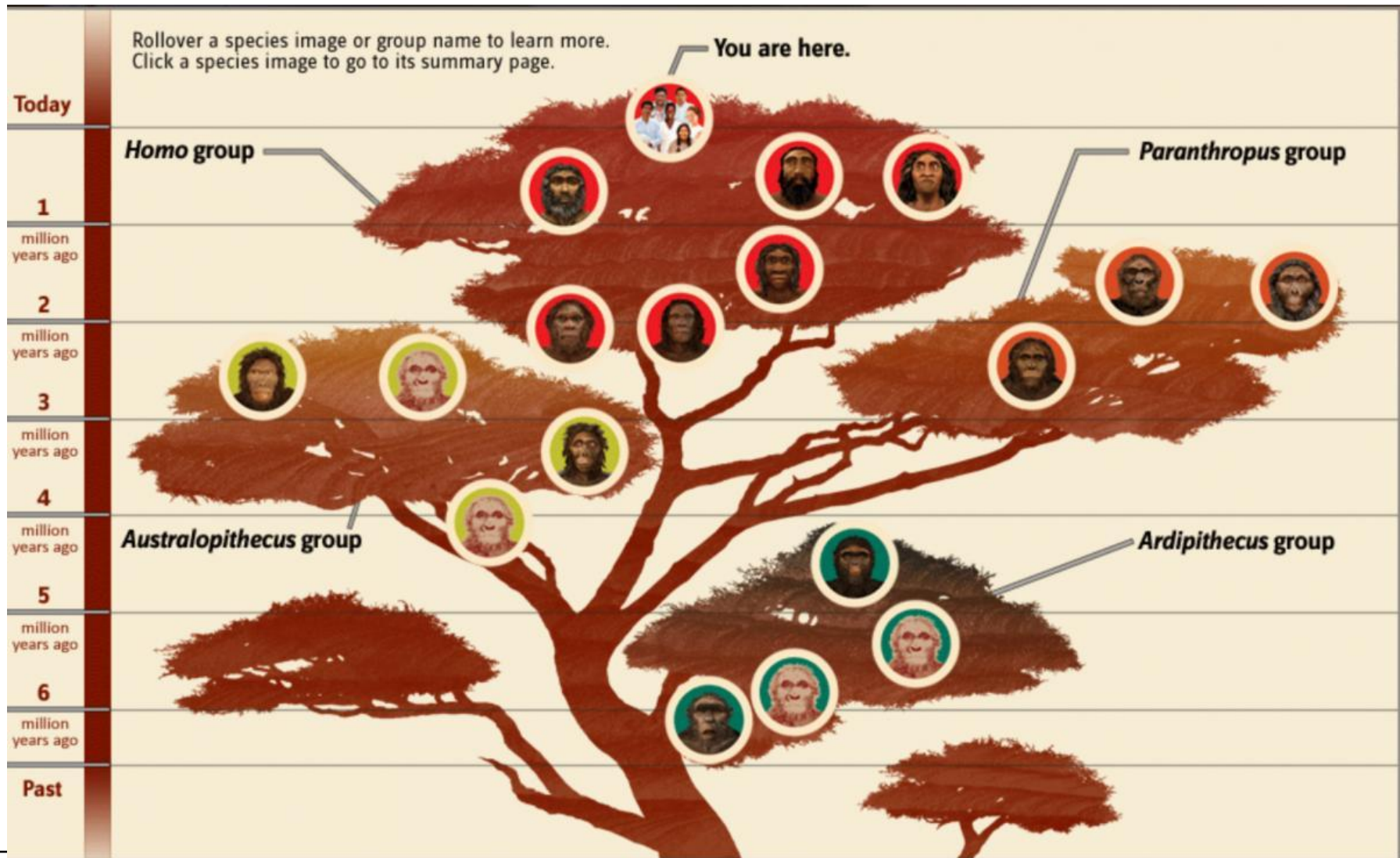


# **Phylogenetic Trees**

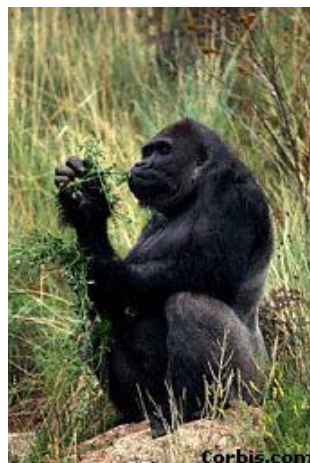
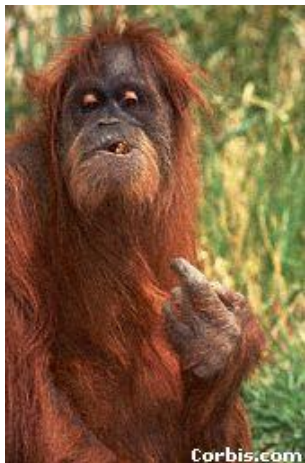
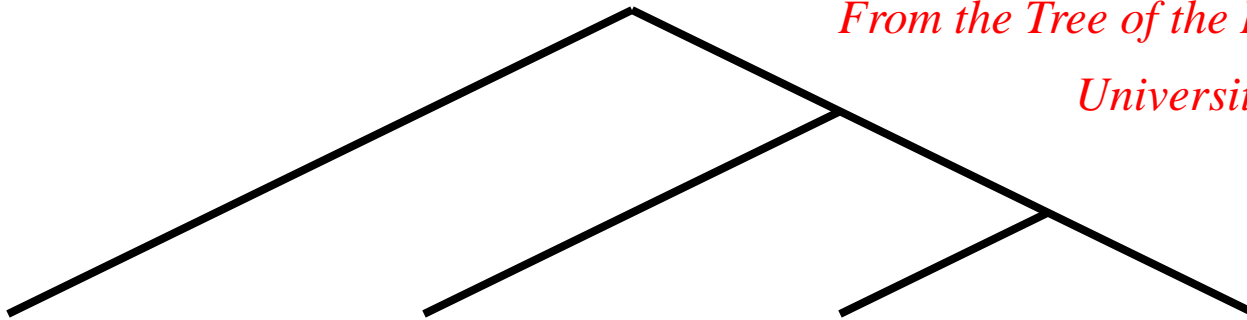
# Phylogenetic Trees

<http://humanorigins.si.edu/evidence/human-family-tree>

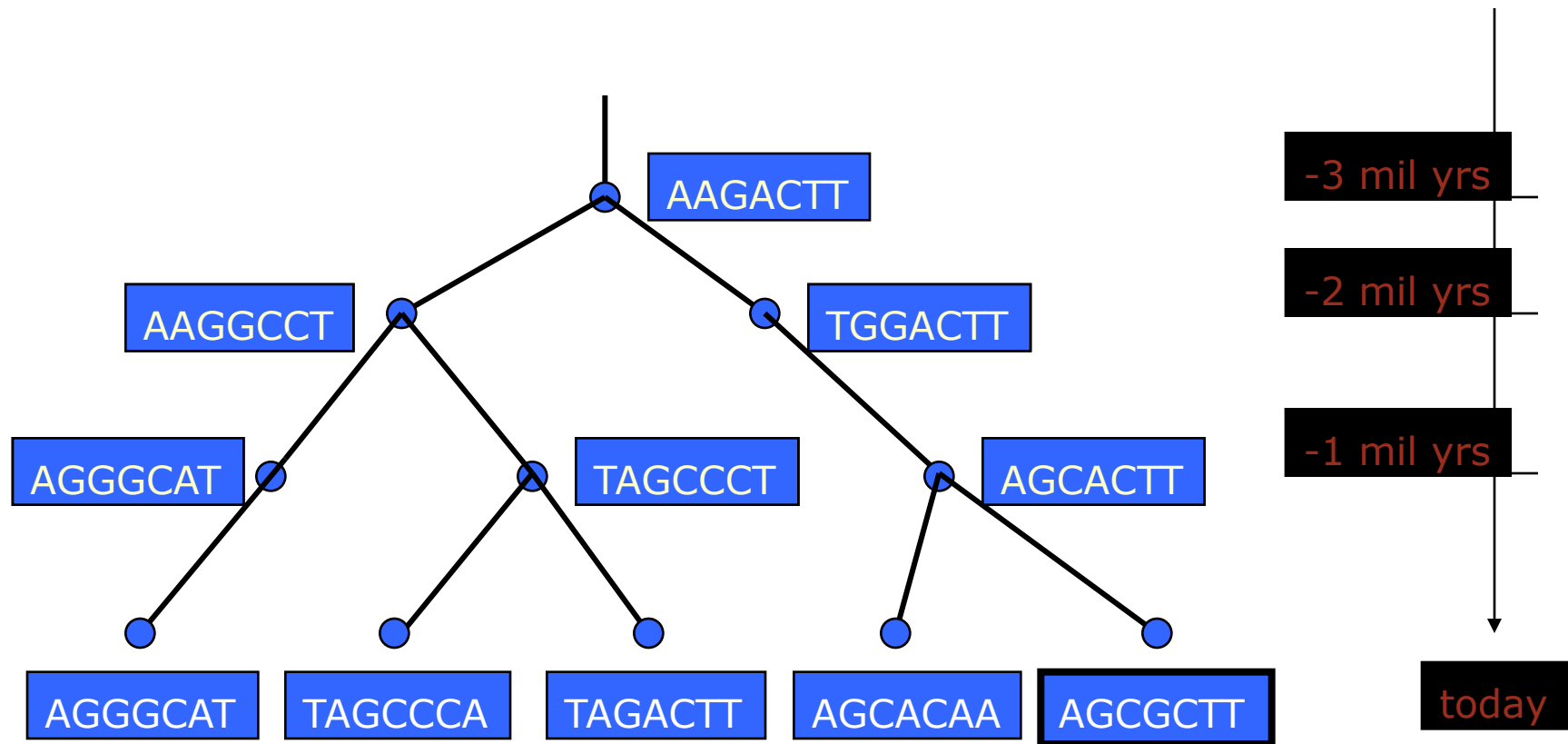


# Phylogeny

*From the Tree of the Life Website,  
University of Arizona*



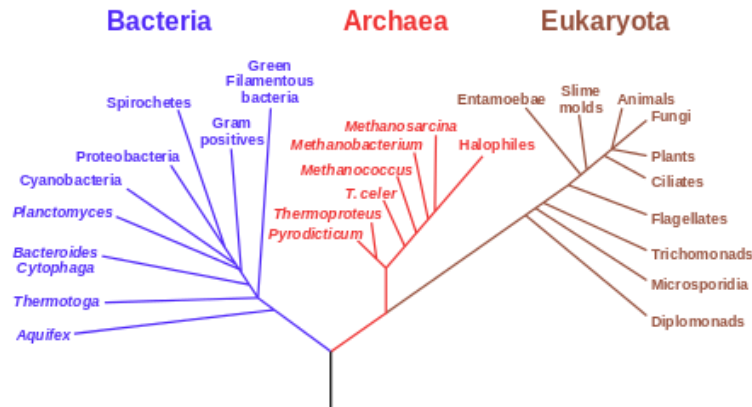
# DNA Sequence Evolution



# Phylogenetic trees

A **phylogenetic tree** or **evolutionary tree** is a branching diagram or "tree" showing the inferred evolutionary relationships among various biological species or other entities—their **phylogeny**—based upon similarities and differences in their physical or genetic characteristics.

Phylogenetic Tree of Life



# Perfect Phylogeny Tree

Defn:

- **Perfect phylogeny** is a term used in computational phylogenetics to denote a phylogenetic tree in which all internal nodes may be labeled such that all characters evolve down the tree **without homoplasy**.
- Characteristics do not hold to **evolutionary convergence**, and do not have **analogous** structures

# Perfect Phylogeny Tree

- **evolutionary convergence**

- **Convergent evolution** is the independent evolution of similar features in species of different lineages.
- Convergent evolution creates **analogous structures** that have similar form or function, but that were not present in the last common ancestor of those groups

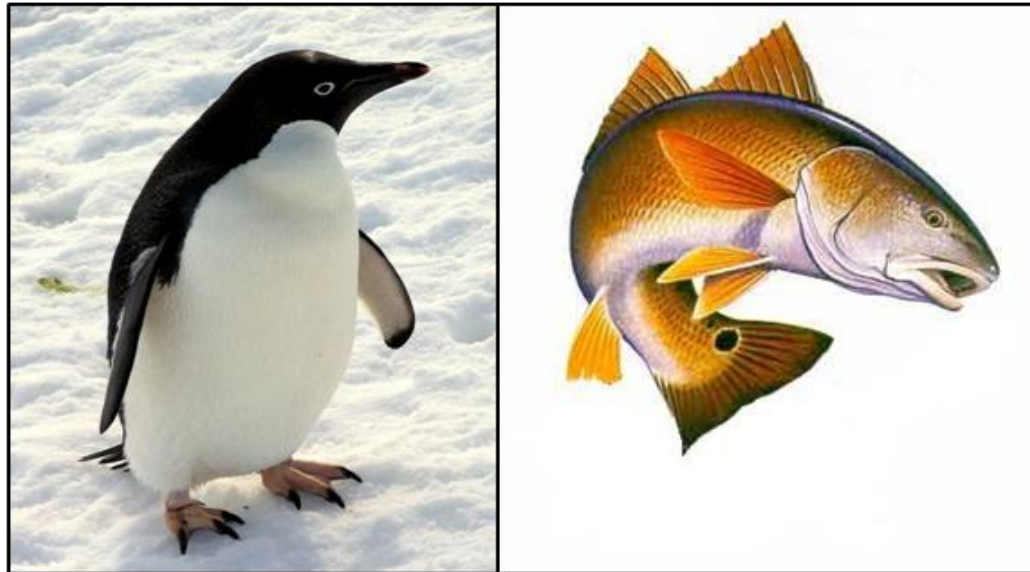


succulent plant genera,  
*Euphorbia* and  
*Astrophytum*, are only  
distantly related, but these  
species within each have  
independently converged  
on a similar body form.



# evolutionary convergence

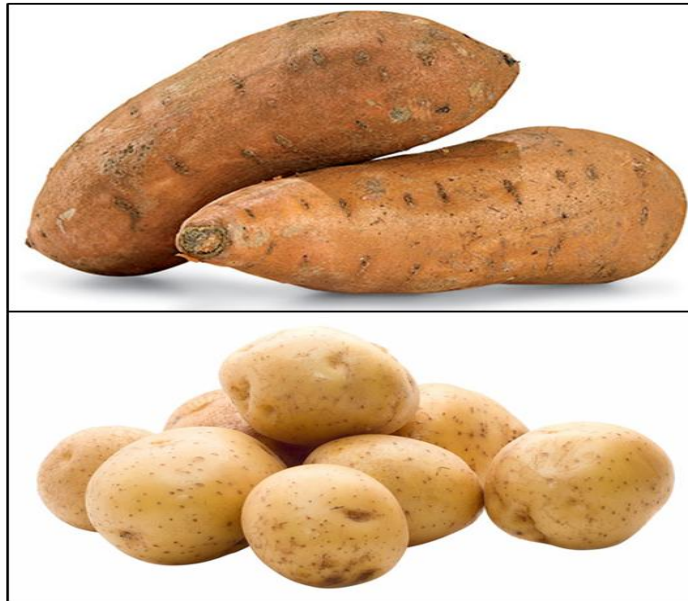
- Animals such as **penguins and fish** both have **fin-like** structures to help them navigate through their aquatic environments. However, because **one is a bird and one is a fish**, it is clear that the fin evolved in these very different species because it was the best **functional feature for the environment** they inhabit instead of from a common ancestor.





# evolutionary convergence

- Sweet potatoes and potatoes have the same function of food storage. The difference is that sweet potatoes are an underground root and potatoes are an underground stem.

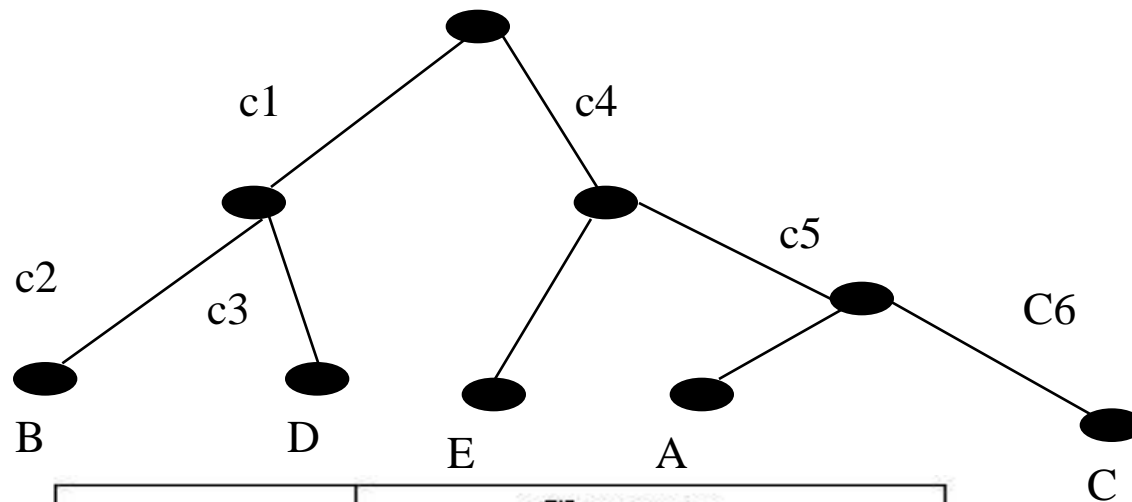


# Perfect Phylogeny Tree

Defn: A tree has perfect phylogeny if

- For each state  $s$  of each character  $c$ , the set of all nodes  $u$  for which the state is  $s$  with respect to  $c$  must form a sub tree of  $T$ . In Particular, the edge  $e$  leading to this sub tree is uniquely associated with a transition from some state  $w$  to state  $s$
- OBEY OUR ASSUMPTIONS

# Ex: Perfect Phylogeny tree



	<i>Character</i>					
<i>Object</i>	<i>c1</i>	<i>c2</i>	<i>c3</i>	<i>c4</i>	<i>c5</i>	<i>c6</i>
A	0	0	0	1	1	0
B	1	1	0	0	0	0
C	0	0	0	1	1	1
D	1	0	1	0	0	0
E	0	0	0	1	0	0

# Perfect Phylogeny Problem

- Instance: A set  $O$  with  $n$  objects, a set  $C$  of  $m$  characters, each character having at most  $r$  states ( $n, m, r$  positive integers)
- Question: Is there a perfect phylogeny for  $O$ ?
- If the character state matrix admits a perfect phylogeny we say that the defining characters are **compatible**

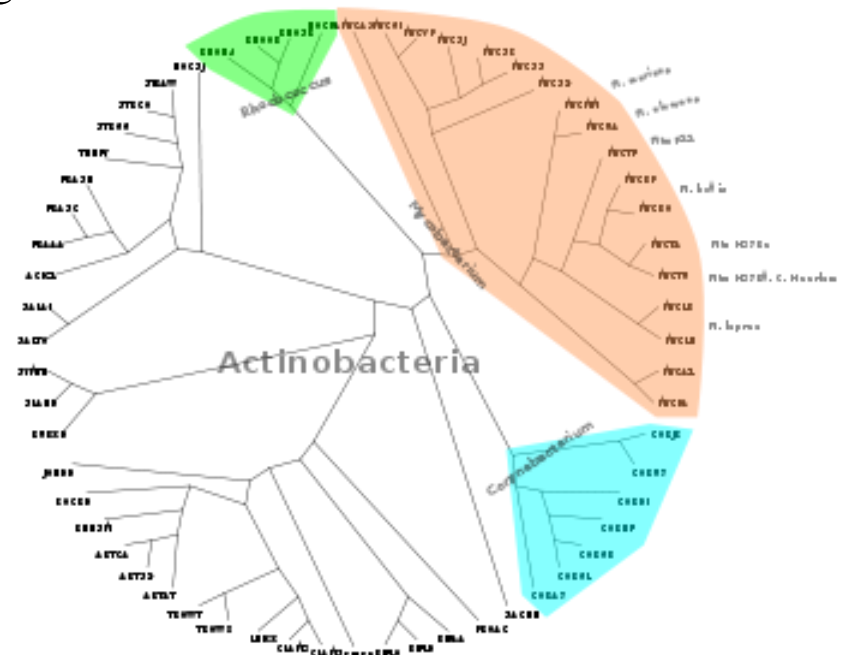
# Perfect Phylogeny Problem

- Instance: A set  $O$  with  $n$  objects, a set  $C$  of  $m$  characters, each character having at most  $r$  states ( $n, m, r$  positive integers)
- Question: Is there a perfect phylogeny for  $O$ ?
- If the character state matrix admits a perfect phylogeny we say that the defining characters are **compatible**

# Perfect Phylogeny Problem

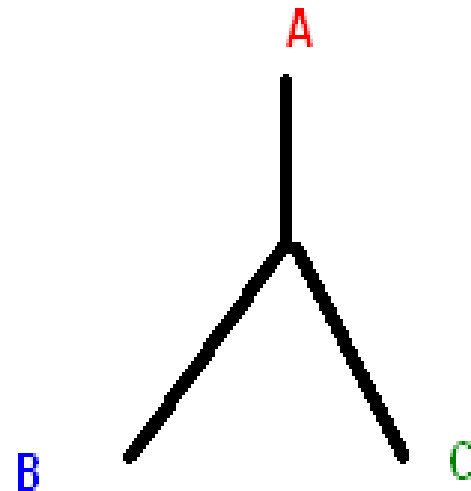
- Can we determine for every problem (input) the root?
- No, we may not have enough information

**Tree will be unrooted !**



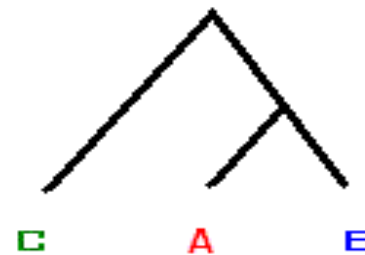
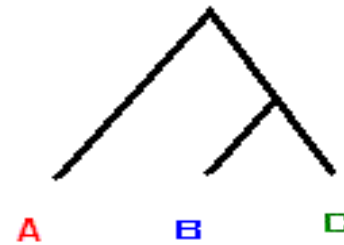
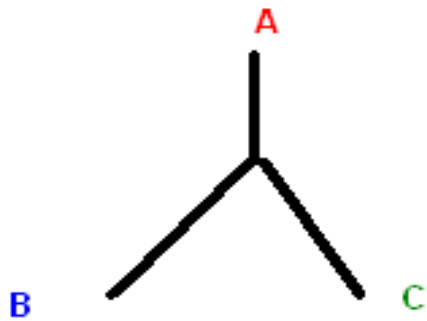
# Ex: Unrooted Binary Tree

- Unrooted Binary tree do not imply a known ancestral root.
- This Tree has 3 possible rooted binary Trees with one common ancestor





# Ex: Unrooted Binary Tree



# Binary Character States

- Defn: For each Column  $j$  of  $M$ , let  $O_j$  be the set of objects whose state is 1 for  $j$ . Let  $\overline{O_j}$  be the set of objects whose state is 0 for  $j$ .

$O_{c1} = ?$

$\overline{O_{c1}} = ?$

	<i>Character</i>					
<i>Object</i>	<i>c1</i>	<i>c2</i>	<i>c3</i>	<i>c4</i>	<i>c5</i>	<i>c6</i>
A	0	0	0	1	1	0
B	1	1	0	0	0	0
C	0	0	0	1	1	1
D	1	0	1	0	0	0
E	0	0	0	1	0	0

# Binary Character States

- Defn: For each Column  $j$  of  $M$ , let  $O_j$  be the set of objects whose state is 1 for  $j$ . Let  $\overline{O}_j$  be the set of objects whose state is 0 for  $j$ .

$$O_{c1} = \{B, D\}$$

$$\overline{O}_{c1} = ?$$

	<i>Character</i>					
<i>Object</i>	<i>c1</i>	<i>c2</i>	<i>c3</i>	<i>c4</i>	<i>c5</i>	<i>c6</i>
A	0	0	0	1	1	0
B	1	1	0	0	0	0
C	0	0	0	1	1	1
D	1	0	1	0	0	0
E	0	0	0	1	0	0

# Binary Character States

- Defn: For each Column  $j$  of  $M$ , let  $O_j$  be the set of objects whose state is 1 for  $j$ . Let  $\overline{O}_j$  be the set of objects whose state is 0 for  $j$ .

$$O_{c1} = \{B, D\}$$

$$\overline{O}_{c1} = \{A, C, E\}$$

	<i>Character</i>					
<i>Object</i>	<i>c1</i>	<i>c2</i>	<i>c3</i>	<i>c4</i>	<i>c5</i>	<i>c6</i>
A	0	0	0	1	1	0
B	1	1	0	0	0	0
C	0	0	0	1	1	1
D	1	0	1	0	0	0
E	0	0	0	1	0	0

# Lemma

- A binary Matrix  $M$  admits a perfect phylogeny if and only if for each pair of characters  $i$  and  $j$  the sets  $O_i$  and  $O_j$  are disjoint or one of them contains each other

# Algorithms

- For Simplicity we assume that the Phylogenetic tree construction works in 2 phases
  - Decision
  - Construction

# Algorithms for Decisions

- The very basic Algorithm:
  - Check if the input Matrix obeys Lemma
  - How would you do that?

	<i>Character</i>					
<i>Object</i>	<i>c1</i>	<i>c2</i>	<i>c3</i>	<i>c4</i>	<i>c5</i>	<i>c6</i>
A	0	0	0	1	1	0
B	1	1	0	0	0	0
C	0	0	0	1	1	1
D	1	0	1	0	0	0
E	0	0	0	1	0	0



# Basic Decision Algorithm

- Check every column pair of being disjoint or if one is the subset of the other
- One of these checks costs us  $O(n)$  we have  $m^2$  column pairs  $O(nm^2)$



	<i>Character</i>					
<i>Object</i>	<i>c1</i>	<i>c2</i>	<i>c3</i>	<i>c4</i>	<i>c5</i>	<i>c6</i>
A	0	0	0	1	1	0
B	1	1	0	0	0	0
C	0	0	0	1	1	1
D	1	0	1	0	0	0
E	0	0	0	1	0	0

# Decision Algorithms

- Improvement
  - Visit every column only once to have Complexity  $O(nm)$
- Process first characters for which the maximum number of objects has state 1
  - All other characters are either subsets of it or are disjoint from it.

# Algorithms Perfect Phylogeny Decision

- Input: Binary Matrix M
- Output: True if M admits perfect phylogeny false otherwise

// Sort column based on #1's

// Initialize auxiliary matrix L

**for** each  $L_{ij}$  **do**

**$L_{ij} \leftarrow 0$**

M	<i>Character</i>					
<i>Object</i>	<i>c1</i>	<i>c2</i>	<i>c3</i>	<i>c4</i>	<i>c5</i>	<i>c6</i>
A	1	0	1	0	0	0
B	0	1	0	1	0	0
C	1	0	1	0	0	1
D	0	1	0	0	1	0
E	1	0	0	0	0	0

L	<i>Character</i>					
<i>Object</i>	<i>c1</i>	<i>c2</i>	<i>c3</i>	<i>c4</i>	<i>c5</i>	<i>c6</i>
A	0	0	0	0	0	0
B	0	0	0	0	0	0
C	0	0	0	0	0	0
D	0	0	0	0	0	0
E	0	0	0	0	0	0

# Algorithms Perfect Phylogeny Decision

- for  $i \leftarrow 1$  to  $n$  do  
 $k \leftarrow -1$   
 for  $j \leftarrow 1$  to  $m$  do  
   if  $M_{ij} = 1$  then  
      $L_{ij} \leftarrow k$   
      $k \leftarrow j$

M	Character					
Object	c1	c2	c3	c4	c5	c6
A	1	0	1	0	0	0
B	0	1	0	1	0	0
C	1	0	1	0	0	1
D	0	1	0	0	1	0
E	1	0	0	0	0	0

L	Character					
Object	c1	c2	c3	c4	c5	c6
A	-1	0	1	0	0	0
B	0	-1	0	2	0	0
C	-1	0	1	0	0	3
D	0	-1	0	0	2	0
E	-1	0	0	0	0	0

# Algorithms Perfect Phylogeny Decision

- **for** each column  $j$  of  $L$  **do**
- **If**  $L_{ij} \neq L_{mj}$  for some  $i$ ,  $m$  and both  $L_{ij}$  and  $L_{mj}$  are both non zero **then return false**
- **return true**

L	<i>Character</i>					
<i>Object</i>	<i>c1</i>	<i>c2</i>	<i>c3</i>	<i>c4</i>	<i>c5</i>	<i>c6</i>
A	-1	0	1	0	0	0
B	0	-1	0	2	0	0
C	-1	0	1	0	0	3
D	0	-1	0	0	2	0
E	-1	0	0	0	0	0
	ok	ok	ok	ok	ok	ok

# Algorithms Perfect Phylogeny Construction

- Input: binary matrix  $M$  with Columns sorted in decreasing order
- Output: perfect phylogeny for  $M$

# Algorithms Perfect Phylogeny Construction

- Create root
  - **for** each object  $i$  **do**
    - $\text{curNode} \leftarrow \text{root}$
    - **For**  $j = 1$  to  $m$  **do**
      - If  $M_{ij} = 1$  then
      - If there already exists edge  $(\text{curNode}, u)$  labeled  $j$  then  
 $\text{curNode} \leftarrow u$
      - **else** Create node  $u$ , Create edge  $(\text{curNode}, u)$  labeled  $j$ ,  
 $\text{curNode} \leftarrow u$
    - Place  $i$  in  $\text{curNode}$
  - **for** each node  $u$  except root **do**
    - Create as many leaves linked to  $u$  as there are objects in  $u$



# Find the phylogenetic tree

M	Character		
Object	x1	x2	x3
A	1	0	0
B	1	1	0
C	0	0	1

Step 1 Decision

Step 2 Construction

# Find the phylogenetic tree

M	Character		
Object	x1	x2	x3
A	1	0	0
B	1	1	0
C	0	0	1

## Step 1 Decision

L	Character		
Object	x1	x2	x3
A	-1	0	0
B	-1	1	0
C	0	0	-1
	Ok	Ok	Ok

# Find the phylogenetic tree

M	Character		
Object	x1	x2	x3
A	1	0	0
B	1	1	0
C	0	0	1

## Step 2 Construction

