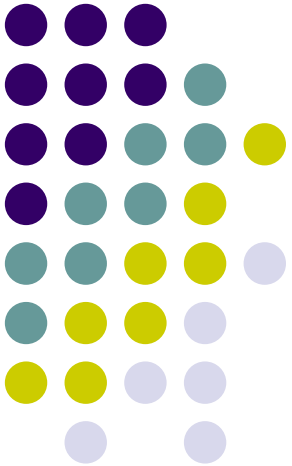
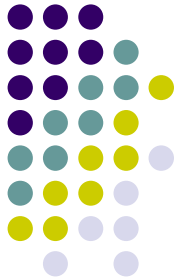
Parsimony and perfect phylogeny

Lecture 14



Phylogeny and evolution



- A phylogeny is a tree representation of the evolutionary history of a set of species, biological sequences, populations or languages
- Phylogeny construction is among the basic computational problems in biology and linguistics

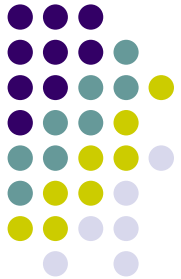
Binary attributes

	move (active)	using sun energy	seeds	eggs	milk	swim (active)	fly (active)
Elephant	1	0	0	0	1	0	0
Snake	1	0	0	1	0	0	0
Whale	1	0	0	0	1	1	0
Fern	0	1	0	0	0	0	0
Eagle	1	0	0	1	0	0	1
Sunflower	0	1	1	0	0	0	0

Perfect phylogeny – the character evolved only once

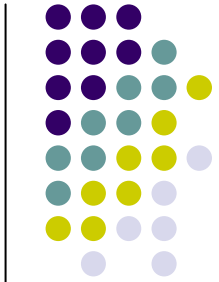
- A very restricted version of phylogeny is called a *perfect phylogeny*
- Many morphological traits evolved independently from different ancestors under the same environmental pressures (wings, fins)
- This is called *homoplasy* and is generally inescapable in real data
- Homoplasy is a poor indicator of evolutionary relationships because similarity does not reflect shared ancestry
- Sets of characters that admit phylogenies without homoplasy are said to be *compatible*
- Phylogenies that avoid homoplasy are called *perfect* and the character compatibility problem is called *the perfect phylogeny problem*

A perfect phylogenetic tree for binary characters



- Let *M* be a binary matrix representing K objects in terms of C characters or traits, which describe the objects. Each character takes one of 2 possible values: 0 or 1, which is recorded in the corresponding cell of M
- Given M for K objects and C characters, *a perfect phylogenetic tree* for M is a rooted directed tree T:
  - with exactly K leaves – 1 leaf per object
  - each character labels exactly 1 edge
  - for any object the characters that label the edges along the path from the root to the parent of a corresponding leaf specify all the characters of this object whose value is 1

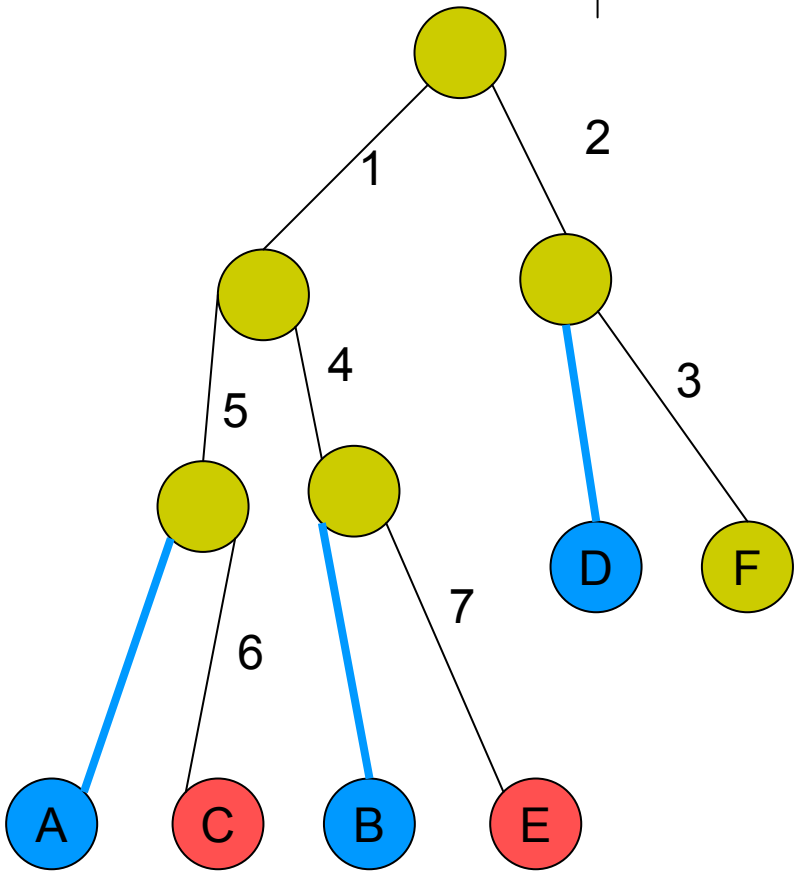
Parsimony



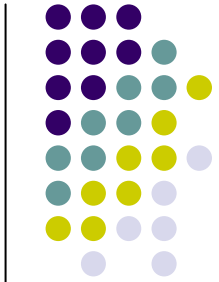
- In science, *parsimony* is preference for the least complex explanation. This is regarded as good when judging hypotheses.
- Occam's razor also states the "principle of parsimony“: *entia non sunt multiplicanda praeter necessitatem*, is the principle that "entities must not be multiplied beyond necessity": the simplest explanation or strategy tends to be the best one
- Under maximum parsimony, the preferred phylogenetic tree is the tree that requires the smallest number of evolutionary changes.

Connection between phylogenetic tree and the parsimony

- The root of the tree represents an ancestral object that has none of the present characters
- Each character changes from the zero state to one state exactly once and never changes back from the 1 state to the zero state: in the tree, any leaf below the node with incoming edge labeled by some character definitely has this character; once acquired, it can not be lost
- If each edge is labeled by each evolutionary event only **once**, the tree has the fewest state changes among all rooted trees for the given set of objects and characters, and thus represents **the most parsimonious tree**



The perfect phylogeny problem:



- Given matrix M, determine whether there is a phylogenetic tree for M and if yes, build it

Pre-processing: reordering the columns

	1	2	3	4	5	6	7
A.	1	0	0	0	1	0	0
B.	1	0	0	1	0	0	0
C.	1	0	0	0	1	1	0
D.	0	1	0	0	0	0	0
E.	1	0	0	1	0	0	1
F.	0	1	1	0	0	0	0

Radix sort (decreasing) of columns as binary numbers

	1	5	4	6	2	7	3
A.	1	1	0	0	0	0	0
B.	1	0	1	0	0	0	0
C.	1	1	0	1	0	0	0
D.	0	0	0	0	1	0	0
E.	1	0	1	0	0	1	0
F.	0	0	0	0	1	0	1

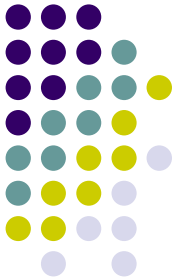
The resulting sets of objects possessing characters:

A={1,5} B={1,4} C={1,5,6} D={2} E={1,4,7} F={2,3}

The resulting sets of characters appear in objects:

1={A,B,C,E} 5={A,C} 4={B,E} 6={C} 2={D,F} 7={E} 3={F}

Testing for perfect phylogeny



The resulting sets of characters appear in objects:

1={A,B,C,E} 5={A,C} 4={B,E} 6={C} 2={D,F} 7={E} 3={F}

- **Theorem: matrix M has a perfect phylogenetic tree if and only if any pair of the character sets is either disjoint, or one is a subset of another.**

Such sets are called *compatible:* Set 1 ∩ Set 2 € {Ø, Set 1, Set 2}

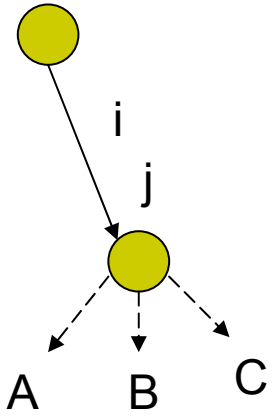
Theorem: matrix M has a perfect phylogenetic tree if and only if any pair of the character sets is compatible

Proof I: **If there is a phylogenetic tree T, then any two character sets are compatible**

- Let ei be the edge of T where character i changes from 0 to 1, and let ej be the similar edge for character j. All the objects that possess character i (or j) are found below these edges. Since in the phylogenetic tree each character labels only 1 edge, there are only

- 4 cases of possible relative topology of ei and ej:
  - (1) ei=ej – the same edge for both characters
  - (2,3) ei is on the path from the root to ej (or vice versa)
  - (4) ei and ej are in separate subtrees (disjoint sets) ■

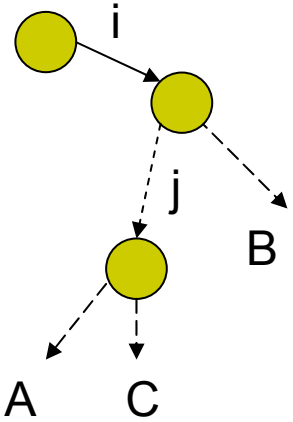
	i	j
A	1	1
B	1	1
C	1	1



Set i € Set j

(1)

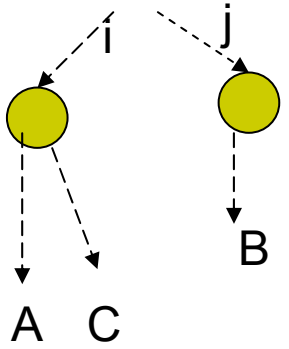
	i	j
A	1	1
B	1	
C	1	1



Set j € Set i

(2)

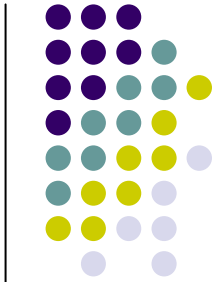
	i	j
A	1	
B		1
C	1	



Set j ∩ Set i=Ø

(3)

Theorem: matrix M has a perfect phylogenetic tree if and only if any pair of the character sets is compatible

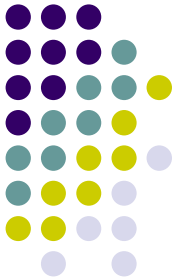


Proof II: **If any two character sets are compatible, then there is a phylogenetic tree T**

- Consider objects B and E, and let k be the largest character (the rightmost in M) that they both possess
- We need to proof that if B possesses character i<k, then E also possesses this character, in order to have a perfect phylogenetic tree
- Since Set i ∩ Set j already has common character k (through object B), then Set i ∩ Set j ≠Ø, and hence Set i is contained in Set j (or vice versa). Therefore, the character i of object E must also be in state 1, and the perfect phylogenetic tree can be constructed ■

	1	2	3	4
A.	1	1	0	0
B.	1	0	1	0
C.	1	1	0	1
D.	0	0	0	0
E.	1	0	1	0

Construction in time O(NM)



The resulting sets of objects possessing characters in the sorted matrix m:

A={1,5} B={1,4} C={1,5,6} D={2} E={1,4,7} F={2,3}

1. Consider each column of M as a binary number. Using radix sort, sort these numbers in non-increasing order
2. Represent each object in a sorted matrix M as a sequence of characters which have state 1
3. Consider each object as a string consisting from this sequence plus sentinel ($)
4. Build *the keyword tree* for all obtained strings. Remove sentinel – obtain a perfect phylogenetic tree

Perfect phylogeny – demo 1

	1	2	3	4	5	6	7
A.	1	0	0	0	1	0	0
B.	1	0	0	1	0	0	0
C.	1	0	0	0	1	1	0
D.	0	1	0	0	0	0	0
E.	1	0	0	1	0	0	1
F.	0	1	1	0	0	0	0

Radix sort (decreasing) of columns as binary numbers

	1	5	4	6	2	7	3
A.	1	1	0	0	0	0	0
B.	1	0	1	0	0	0	0
C.	1	1	0	1	0	0	0
D.	0	0	0	0	1	0	0
E.	1	0	1	0	0	1	0
F.	0	0	0	0	1	0	1

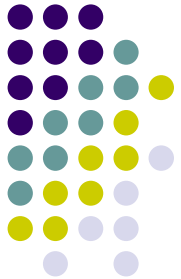
The resulting sets of objects possessing characters:

A={1,5} B={1,4} C={1,5,6} D={2} E={1,4,7} F={2,3}

The resulting sets of characters appearing in objects:

1={A,B,C,E} 5={A,C} 4={B,E} 6={C} 2={D,F} 7={E} 3={F}

Perfect phylogeny – demo 2

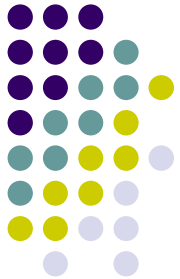


	1	5	4	6	2	7	3
A.	1	1	0	0	0	0	0
B.	1	0	1	0	0	0	0
C.	1	1	0	1	0	0	0
D.	0	0	0	0	1	0	0
E.	1	0	1	0	0	1	0
F.	0	0	0	0	1	0	1

The resulting sets of objects possessing characters:

A={1,5} B={1,4} C={1,5,6} D={2} E={1,4,7} F={2,3}

Perfect phylogeny – demo 3



	1	5	4	6	2	7	3
A.	1	1	0	0	0	0	0
B.	1	0	1	0	0	0	0
C.	1	1	0	1	0	0	0
D.	0	0	0	0	1	0	0
E.	1	0	1	0	0	1	0
F.	0	0	0	0	1	0	1

The resulting strings with sentinel:

A=1 5 $

B=1 4 $

C=1 5 6 $

D=2 $

E=1 4 7 $

F=2 3 $

Perfect phylogeny – demo 4

The resulting strings with sentinel:

A=1 5 $

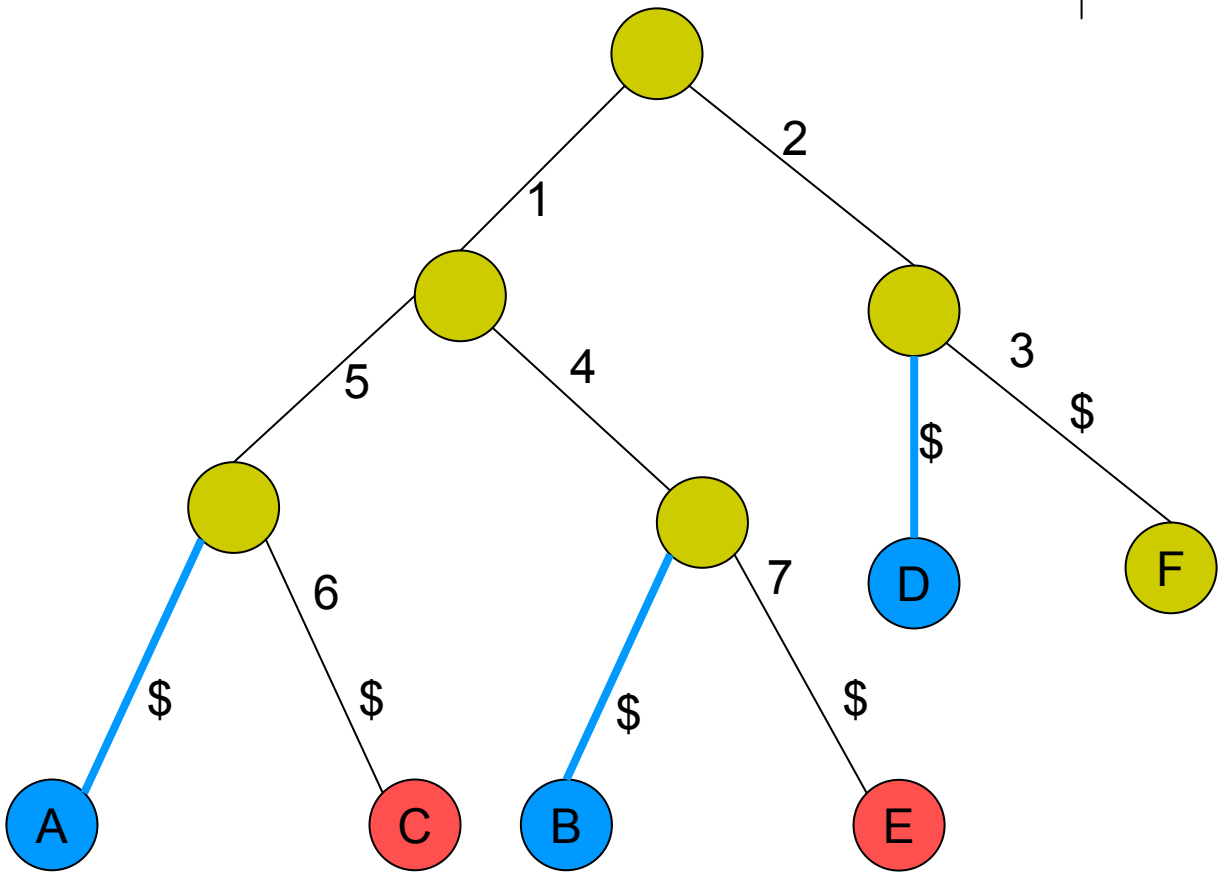
B=1 4 $

C=1 5 6 $

D=2 $

E=1 4 7 $

F=2 3 $



The keyword tree for each string

Perfect phylogeny – tree

The resulting strings with sentinel:

A=1 5 $

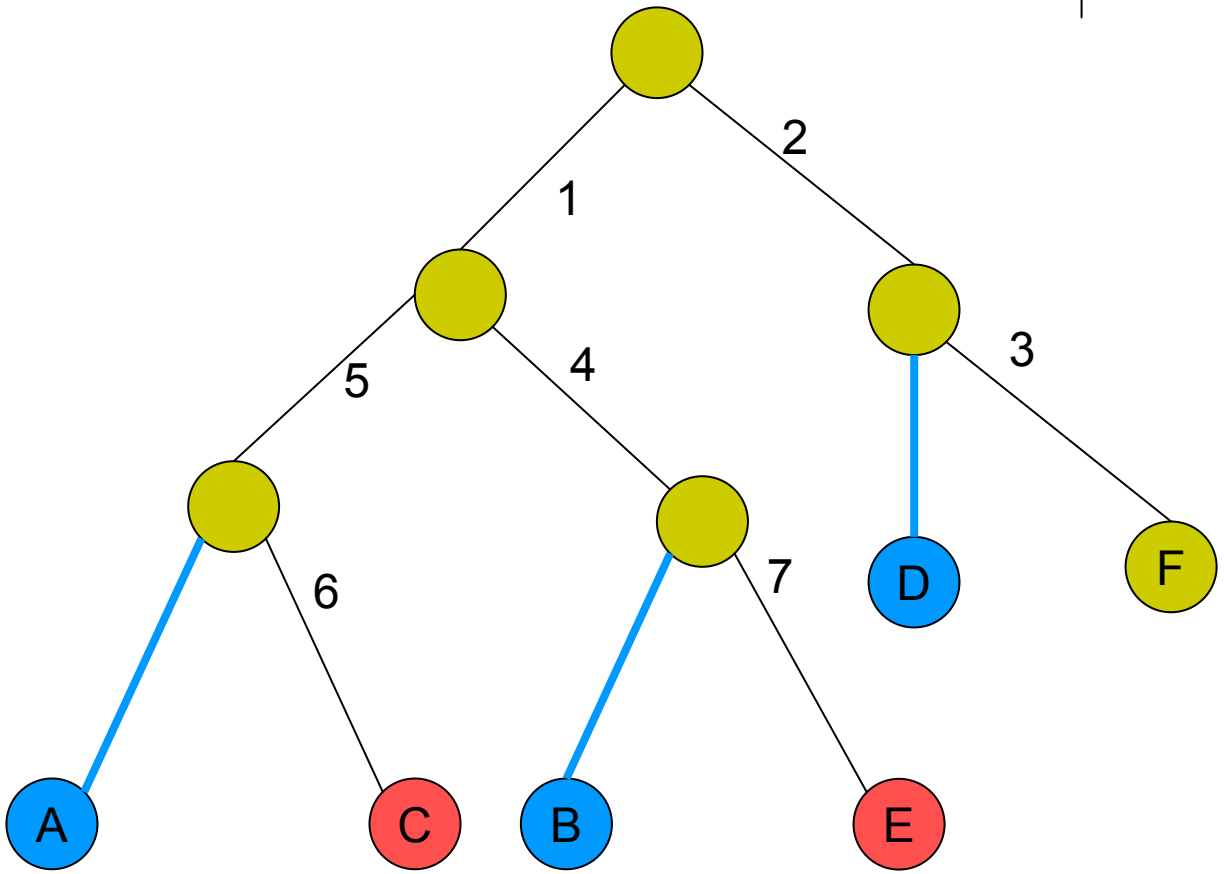
B=1 4 $

C=1 5 6 $

D=2 $

E=1 4 7 $

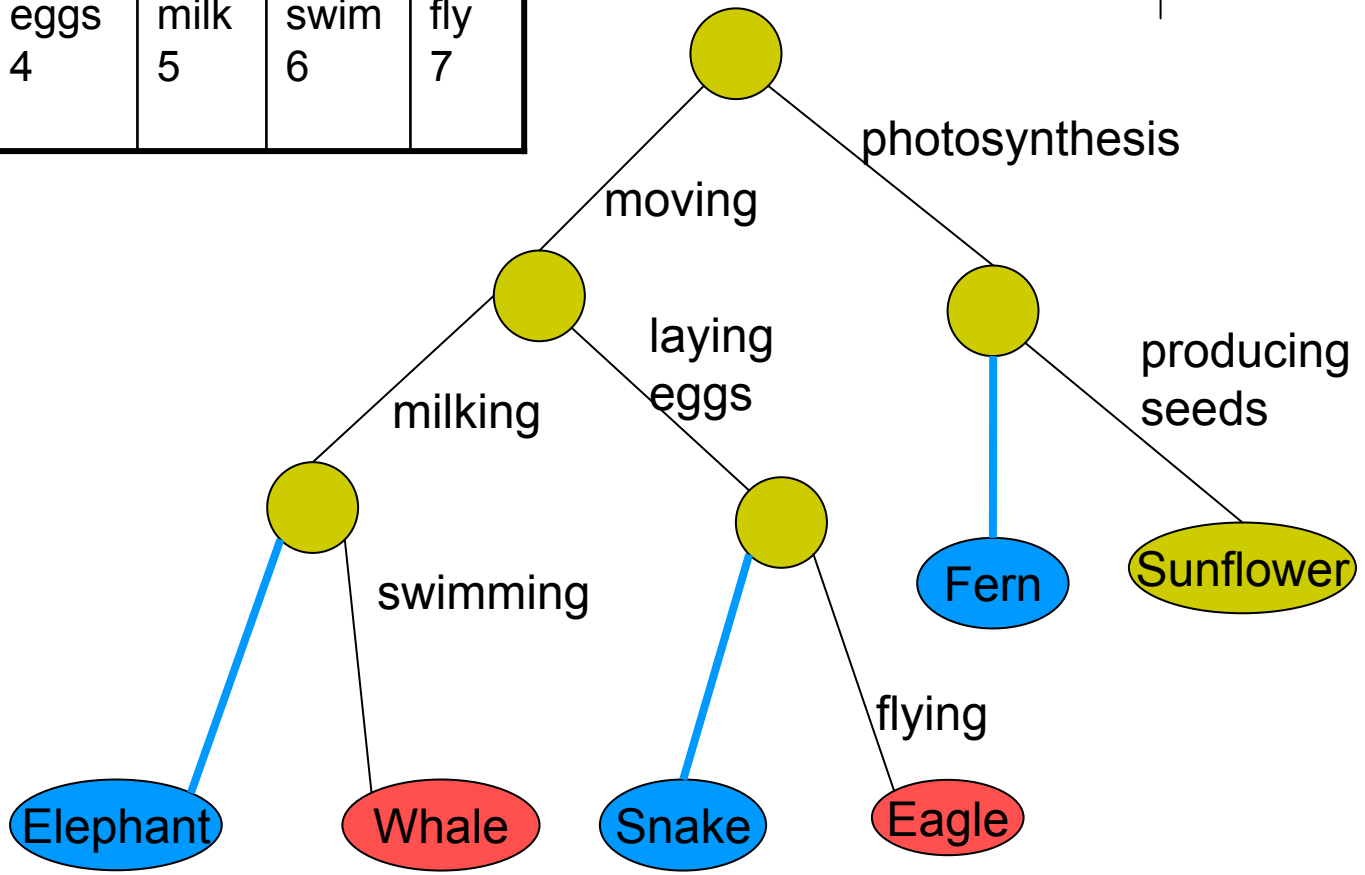
F=2 3 $



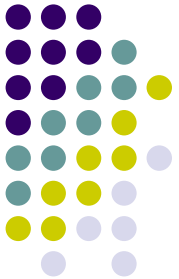
Our first phylogenetic tree

move 1	photo synthe sis 2	seeds 3	eggs 4	milk 5	swim 6	fly 7

A. Elephant
B. Snake
C. Whale
D. Fern
E. Eagle
F. Sunflower



The source of a binary data



- Morphological traits – not a good choice, since a lot of homoplasy – convergent evolution – the same morphological character is acquired more than once and not from the common ancestor
- Biosequences – substrings, special patterns, gaps – better in non-coding regions, since the coding regions do not evolve much with time

Perfect phylogeny for insertions

```
          Ins1          Ins2          Ins3          Ins4
A: RPCVCPKQAVLRQAAQLAQVLQRQI____QQLRRL__AA
B: RPCACP___VLRQVVQ__QALQRQIIQGPQQLRRL__AA
C: KPCLCPKQAAVKQAAHLVQQLYQGQ____KQVRRA__LL
D: KPCVCP___VLRQAAH__QQLYQGQIQGPRQVRRAFRVA
E: KPCVCP___VLRQAAHLVQQLYQGQ____RQVRRLF_AA
```

	Ins 1	Ins 2	Ins 3	Ins 4
A	1	1	0	0
B	0	0	1	0
C	1	1	0	0
D	0	0	1	1
E	0	1	0	0

Exercise:

- Is there a tree?
- If yes, build one