CS 173, Lecture B
August 25, 2015

Professor Tandy Warnow

Websites

- http://tandy.cs.illinois.edu/cs173-warnow.html - this is the **Course Webpage**, for nearly everything
- Piazza – really just for you
- Moodle – for homeworks

Please look at the course webpage the day before class for the PDF of the upcoming lecture, announcements, reading and homework assignments, etc.

Grading

- Lab notebook (for discussion section): 5 pts
- Homework: 9 pts (due Mondays at 10PM on moodle, bottom homework dropped)
- Reading quizzes: 5 pts (due Wednesdays at 10PM on moodle, bottom quiz dropped)
- Examlets: 21 pts (8 exams in class, 3 pts each, worst examlet dropped)
- Midterm (October 6, in class): 20 pts
- Final exam (December 11, 8-11 AM): 40 pts

Syllabus

- Logic (2 lectures)
- Sets (2 lectures)
- Functions (1 lecture)
- Relations (1 lecture)
- Proof techniques (4 lectures)
- Combinatorial counting (1 lecture)
- Problems and algorithms (1 lecture)
- Big-O and running time analysis (1 lecture)
- Graphs and trees (6 lectures)
- NP, P, and NP-hard (1 lecture)
- Dealing with NP-hard problems (3 lectures)
- Countability and uncountability (1 lecture)

Differences between Lectures A and B

- Similarities:
  - We will both use Margaret Fleck's Building Blocks
  - We will both have homework submitted through Moodle
  - The discussion sections in both courses will be very similar
- Differences:
  - I will not cover number theory or state diagrams (but Fleck will)
  - I will cover "trees" differently (as handled by Rosen)
  - I will give examples from computational biology to illustrate techniques and concepts
  - I will have a midterm, but Fleck will not
  - Fleck has more examlets

Two-person games

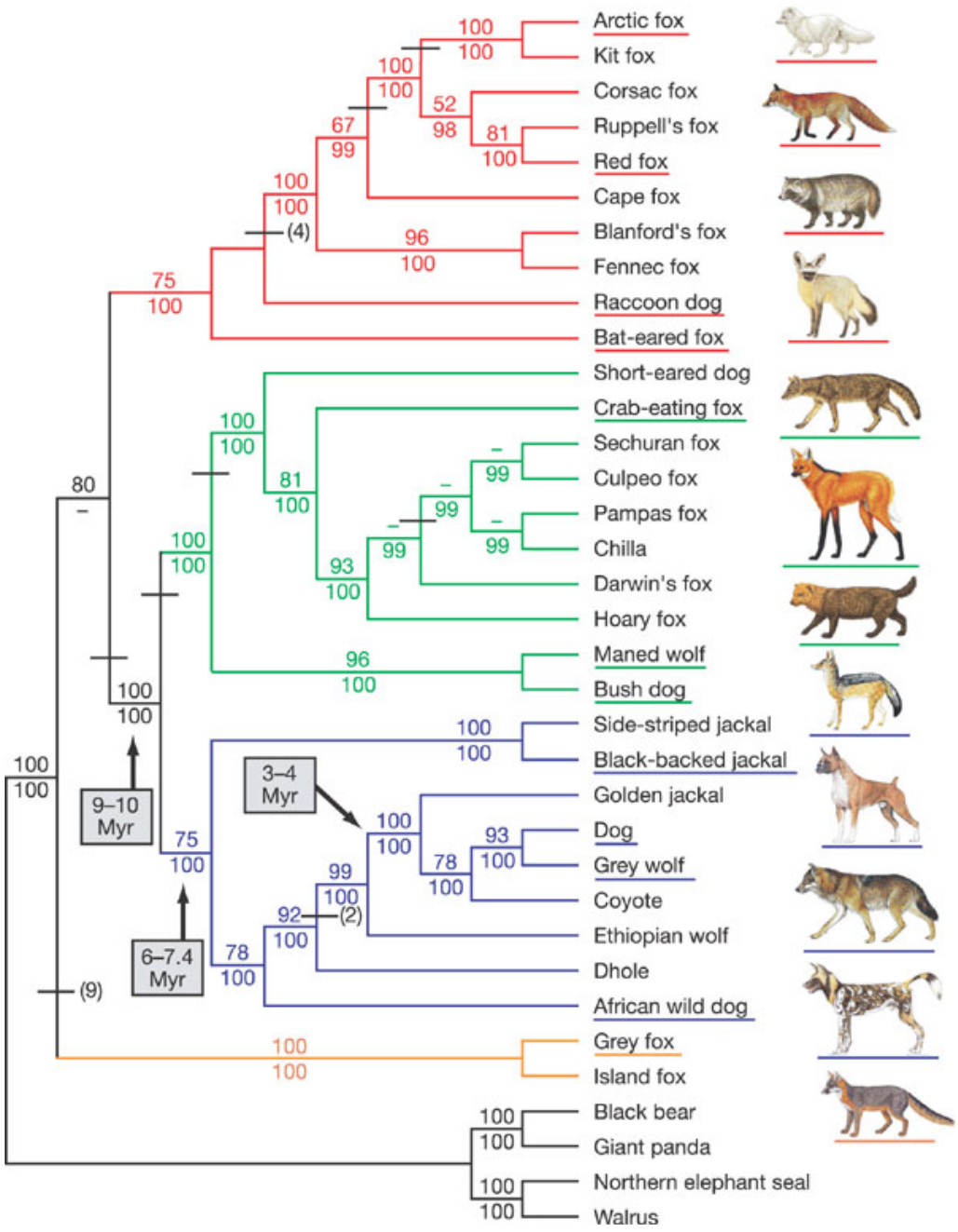
- Two players, A and B. A starts.
- In the beginning there are two piles of stones, with K and L stones respectively.
- During a turn, a player must take at least one stone – the choice is between one stone off of both piles, or one stone off of one of the two piles. The person who takes the last stone wins.
- Who wins when
  - K = 1 and L = 1?
  - K = 2 and L = 1?
  - K = 3 and L = 3?
  - K = 4 and L = 16?
- You can probably figure out a pattern here… but see if you can try to *prove* that you are right. (This is something you'll learn how to do in this class.)
- Spoiler: this can be solved using dynamic programming, and the proof of correctness uses induction

Another two-person game

- Again two players, A and B. A begins. The starting position has two piles of stones, with K and L stones.
- During a turn, the player can take 1 or 2 stones off in total, and these can be from the same pile, or from different piles.
- Who wins
  - K=2 and L=1?
  - K=2 and L=2?
  - K=101 and L =47?
- Figuring out who has a winning strategy is harder here, but still feasible. You'll learn how to do this, and prove you are correct, in this class.
- Spoiler: this can be solved using dynamic programming and the proof of correctness uses induction.

Something perhaps more realistic

- Biologists often try to infer how evolution occurred.



Lindblad-Toh et al., Nature 2005

Part of the data matrix of aligned nucleotide sequences for the malaria parasite Plasmodium.

	Site:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	Species	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	Pre (Chimp)	C	T	T	G	A	G	A	A	A	A	T	T	C	T	T	A	G	A	T	A
2	Pme (Lizard)	T	C	T	A	A	A	A	G	A	T	T	A	T	A	T	A	G	A	T	A
3	Pma (Human)	T	T	T	A	A	G	G	A	A	A	T	T	C	T	T	A	A	A	T	T
4	Pfa (Human)	T	T	T	G	A	G	A	A	A	A	T	T	C	T	T	A	G	A	T	A
5	Pbe (Rodent)	T	T	T	A	A	G	A	A	A	A	T	T	T	A	T	A	A	A	T	A
6	Plo (Bird)	T	T	T	A	A	G	A	A	A	A	C	T	C	A	C	A	A	A	T	C
7	Pfr (Monkey)	C	T	T	A	A	G	A	A	G	A	T	T	C	T	T	A	G	G	A	A
8	Pkn (Monkey)	C	T	T	A	A	G	A	A	A	G	T	T	C	T	T	A	G	A	T	A
9	Pcy (Monkey)	C	T	C	A	T	G	A	A	A	A	T	T	C	T	T	A	G	A	T	A
10	Pv (Human)	C	T	T	A	T	G	A	A	A	A	T	T	C	T	C	G	G	A	T	A
11	Pga (Bird)	T	T	T	A	A	G	A	A	A	A	T	T	T	T	C	A	A	A	T	C

Bradley Efron et al. PNAS 1996;93:13429

How do biologists compute evolutionary trees?

Input: unaligned sequences

```
S1 = AGGCTATCACCTGACCTCCA
S2 = TAGCTATCACGACCGC
S3 = TAGCTGACCGC
S4 = TCACGACCGACA
```

Phase 1: Alignment

```
S1 = AGGCTATCACCTGACCTCCA
S2 = TAGCTATCACGACCGC
S3 = TAGCTGACCGC
S4 = TCACGACCGACA
```



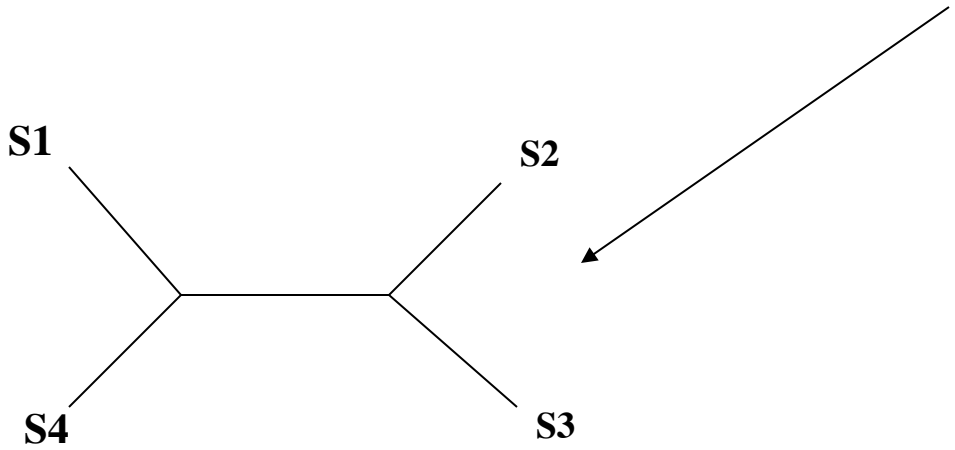
```
S1 = -AGGCTATCACCTGACCTCCA
S2 = TAG-CTATCAC--GACCGC--
S3 = TAG-CT-------GACCGC--
S4 = -------TCAC--GACCGACA
```

Phase 2: Construct tree

```
S1 = AGGCTATCACCTGACCTCCA
S2 = TAGCTATCACGACCGC
S3 = TAGCTGACCGC
S4 = TCACGACCGACA
```

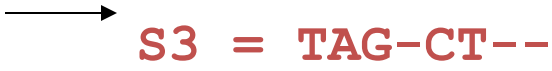
→

```
S1 = -AGGCTATCACCTGACCTCCA
S2 = TAG-CTATCAC--GACCGC--
S3 = TAG-CT-------GACCGC--
S4 = -------TCAC--GACCGACA
```

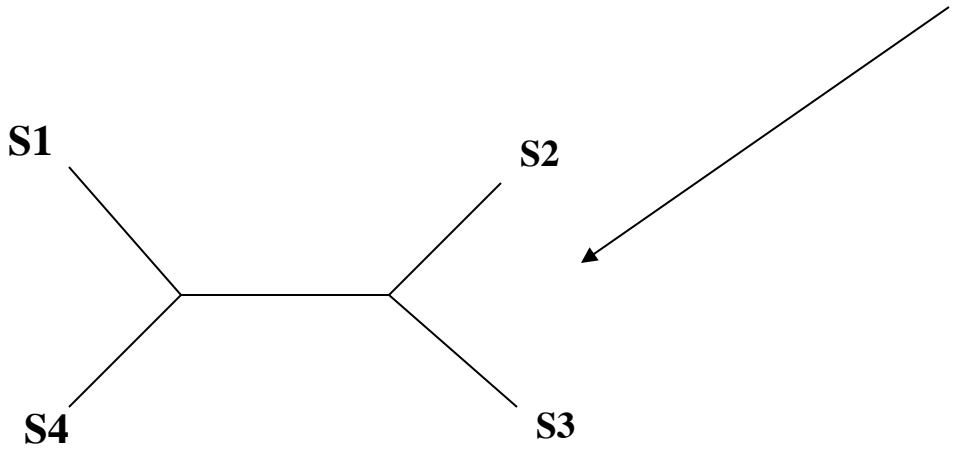


First Align, then Compute the Tree

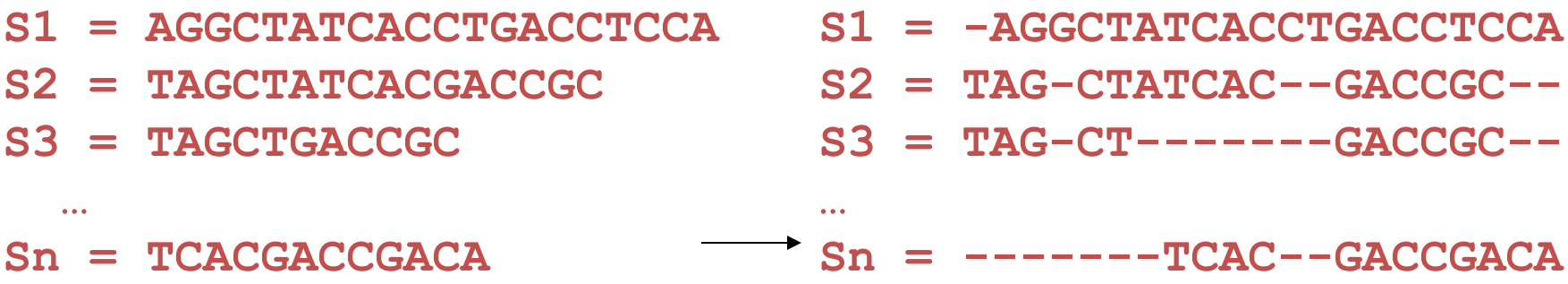
```
S1 = AGGCTATCACCTGACCTCCA
S2 = TAGCTATCACGACCGC
S3 = TAGCTGACCGC
S4 = TCACGACCGACA
```



```
S1 = -AGGCTATCACCTGACCTCCA
S2 = TAG-CTATCAC--GACCGC--
S3 = TAG-CT-------GACCGC--
S4 = -------TCAC--GACCGACA
```



Multiple Sequence Alignment (MSA): *another grand challenge*[1]



Novel techniques needed for scalability and accuracy

NP-hard problems and large datasets

Current methods do not provide good accuracy

Few methods can analyze even moderately large datasets

Many important applications besides phylogenetic estimation

[1] Frontiers in Massive Data Analysis, National Academies Press, 2013

Maximum Parsimony

- **Input**: Set *S* of *n* aligned sequences of length k
- **Output**:
  - A phylogenetic tree T leaf-labeled by sequences in *S*
  - additional sequences of length *k* labeling the internal nodes of *T*

such that $\sum_{(i,j)\in E(T)} H(i,j)$

is minimized, where H(i,j) denotes the Hamming distance between sequences at nodes i and j

Maximum Parsimony

- **Input**: Set *S* of *n* aligned sequences of length k
- **Output**:
  - A phylogenetic tree T leaf-labeled by sequences in *S*
  - additional sequences of length *k* labeling the internal nodes of *T*

such that $\sum_{(i,j)\in E(T)} H(i,j)$

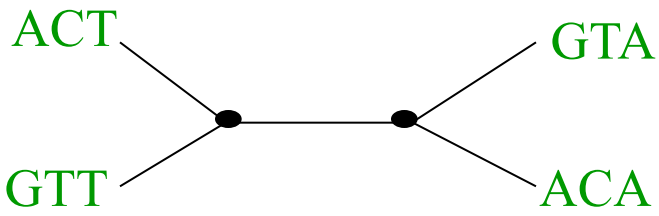
Note: E(T) is a set, denoting the edges of a tree.

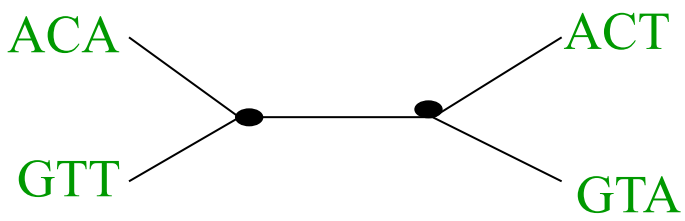
is minimized, where H(i,j) denotes the Hamming distance between sequences at nodes i and j

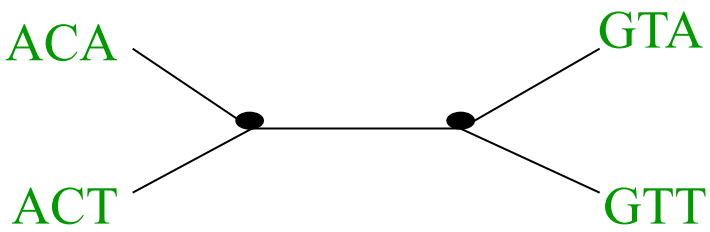
Maximum parsimony (example)

- **Input**: Four sequences
  - ACT
  - ACA
  - GTT
  - GTA
- **Question**: which of the three trees has the best MP scores?

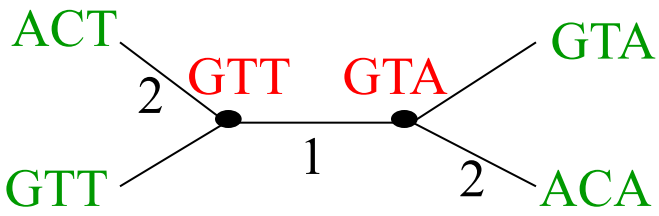
Maximum Parsimony



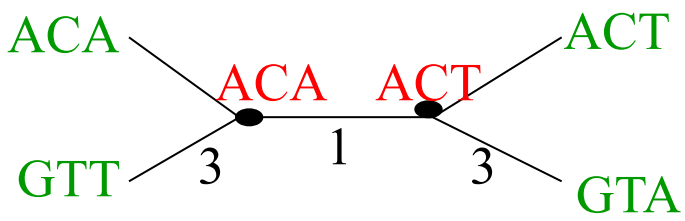




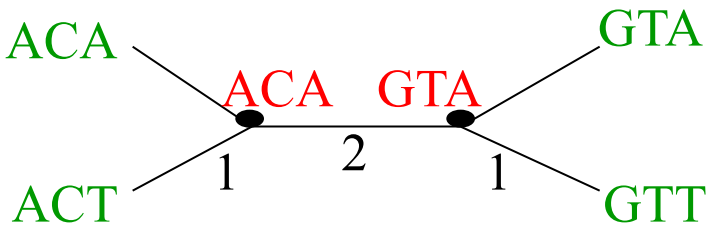
Maximum Parsimony



MP score = 5



MP score = 7

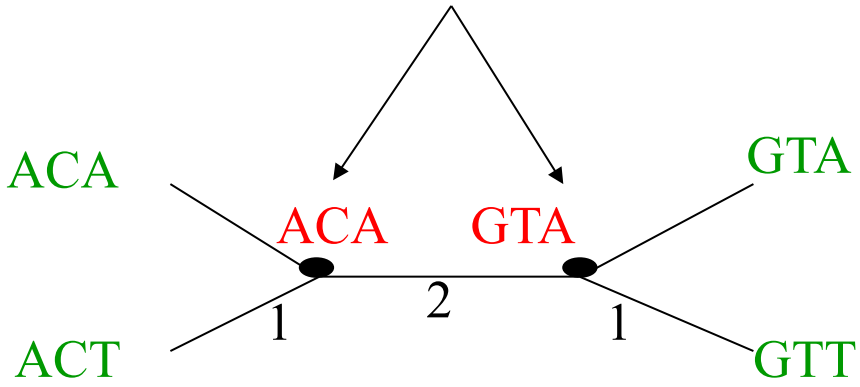


MP score = 4

Optimal MP tree

Maximum Parsimony: computational complexity

Optimal labeling can be computed in linear time O(nk)



MP score = 4

Finding the optimal MP tree is **NP-hard**

NP-hardness and Algorithms

- Finding the best possible parsimony score of a given tree T (with leaves labelled by DNA sequences) can be computed in polynomial time using an algorithmic technique called “Dynamic Programming”. You will learn how to design dynamic programming algorithms in this course.
- But finding the best possible tree for the sequences is NP-hard. You will learn what that means, and how to design methods that address NP-hard problems.

Concepts (so far)

- Sets
- Trees (a special kind of graph)
- Running time – and "Big-O" notation
- NP-hardness
- Dynamic programming
- Proofs by induction
- Two-person games
- Evolutionary trees and multiple sequence alignments

Upcoming Assignments

- Wednesday 10 PM (tomorrow!) – reading quiz due (see Moodle).
- Monday 10 PM (next week) – homework assignment due (see Moodle) on proofs by contradiction.
- You are expected to read Chapter 17 in advance of Thursday's class!