CS 466
Introduction to Bioinformatics
Lecture 17

Mohammed El-Kebir

October 22, 2021

Outline

- Two-State Perfect Phylogeny
- Multi-State Perfect Phylogeny
- Large Maximum Parsimony Phylogeny Problem
- Summary

Reading:

- Lecture notes

Maximum Parsimony

Small Maximum Parsimony Phylogeny Problem:

Given $m \times n$ matrix $A = [a_{i,j}]$ and tree T with m leaves, find assignment of character states to each internal vertex of T with minimum parsimony score.

Large Maximum Parsimony Phylogeny Problem:

Given $m \times n$ matrix $A = [a_{i,j}]$, find a tree T with m leaves labeled according to A and an assignment of character states to each internal vertex of T with minimum parsimony score.

Binary Characters

Species	1	2	3	4	5
	Characters				
A	0	1	1	0	0
B	0	0	1	1	0
C	1	1	1	1	0
D	1	1	0	1	1

Characters only have two possible states

Possible Encoding:
0 : not-mutated
1 : mutated

Possible Encoding:
0 : no wings
1 : wings

Binary Characters

Species	1	2	3	4	5
	Characters				
A	0	1	1	0	0
B	0	0	1	1	0
C	1	1	1	1	0
D	1	1	0	1	1

Characters only have two possible states

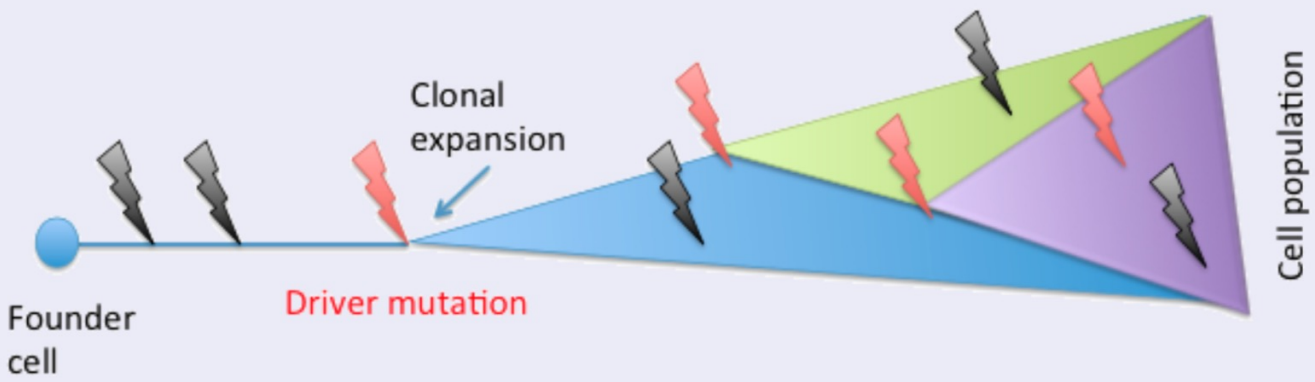
Possible Encoding:
0 : not-mutated
1 : mutated

Possible Encoding:
0 : no wings
1 : wings

Question: Given *n* binary characters, what is the smallest parsimony score?

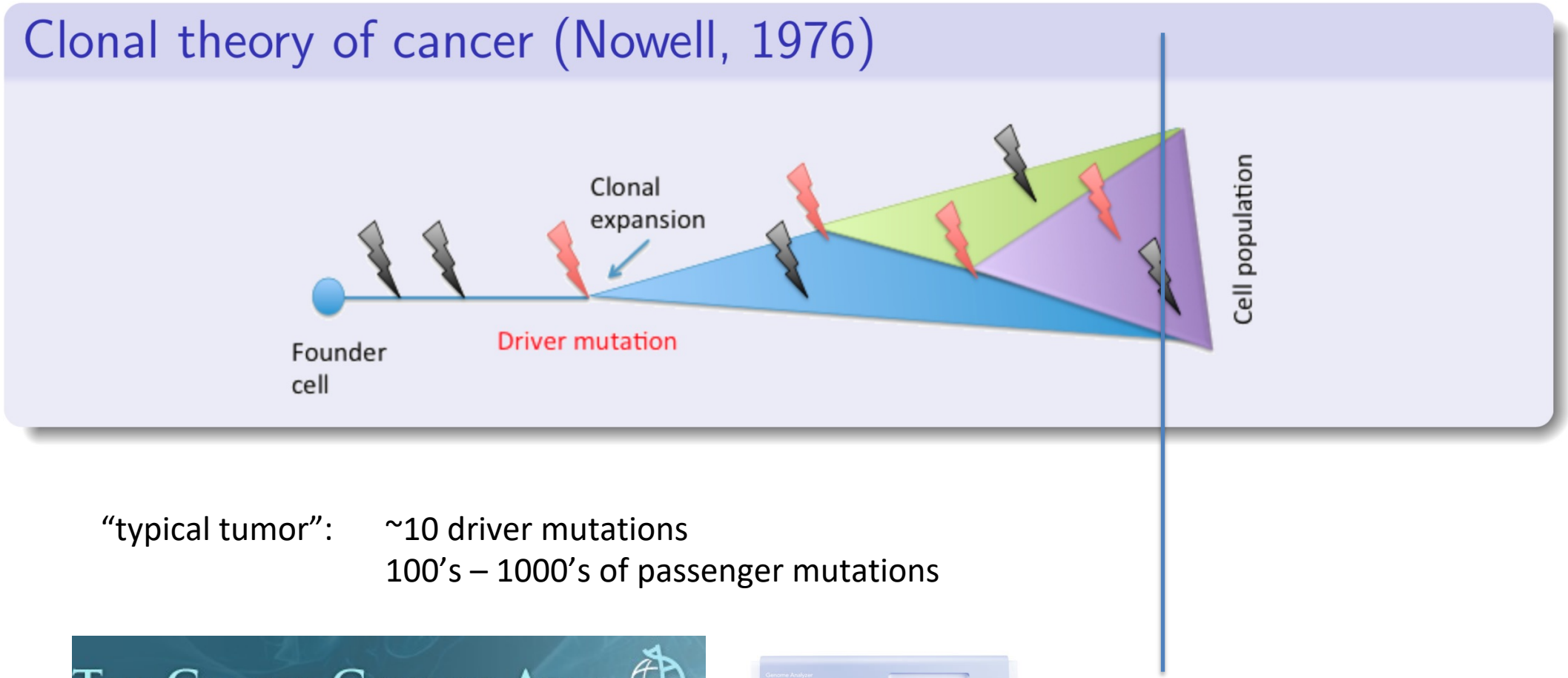
Somatic Mutations and Cancer

Clonal theory of cancer (Nowell, 1976)



“typical tumor”: ~10 driver mutations
100’s – 1000’s of passenger mutations

Somatic Mutations and Cancer



"typical tumor": ~10 driver mutations
100's – 1000's of passenger mutations

THE CANCER GENOME ATLAS

International Cancer Genome Consortium

Progression of Somatic Mutations

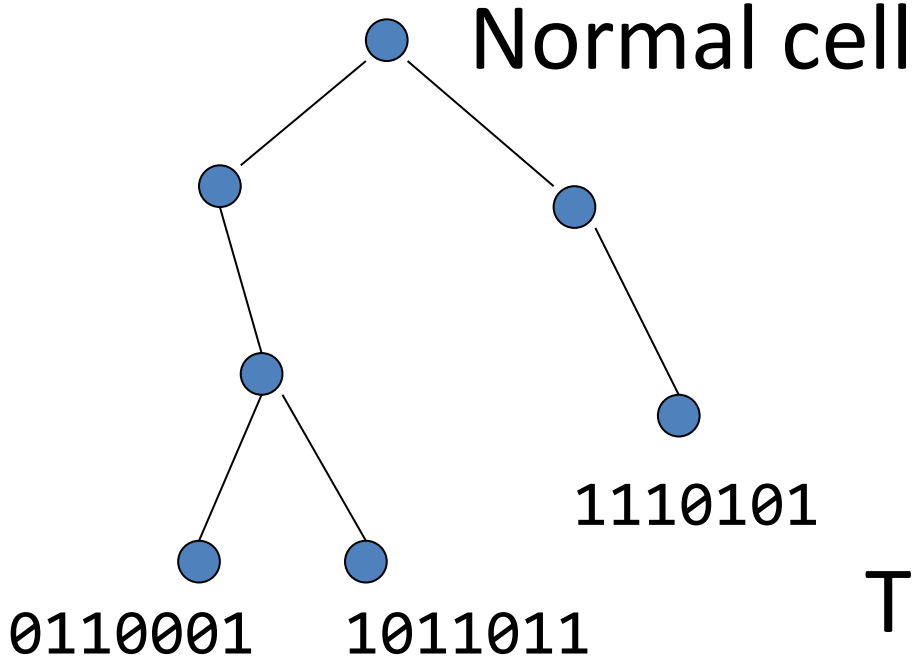
Single nucleotide mutation

… CGTAATTAG …

… CGTCATTAG …

0 = normal

1 = mutated



Root is the normal, founder cell and leaves are cells in tumor.

Progression of Somatic Mutations

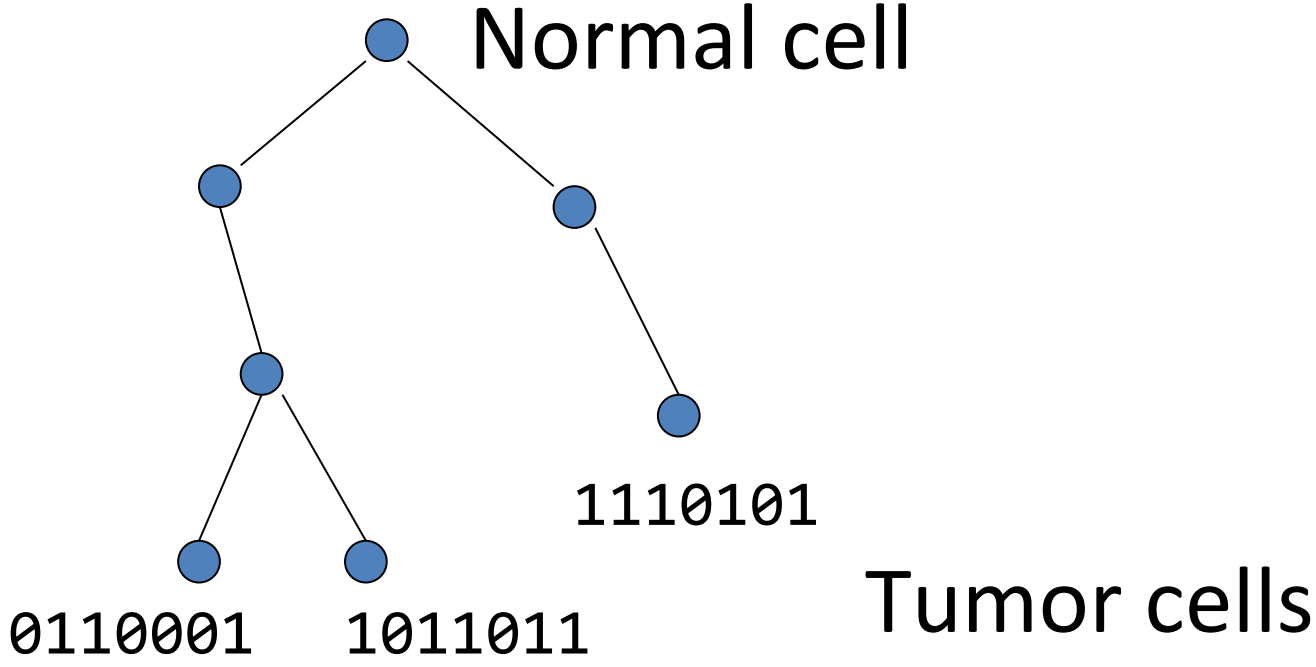
Single nucleotide mutation

… CGTAATTAG …

… CGTCATTAG …

0 = normal

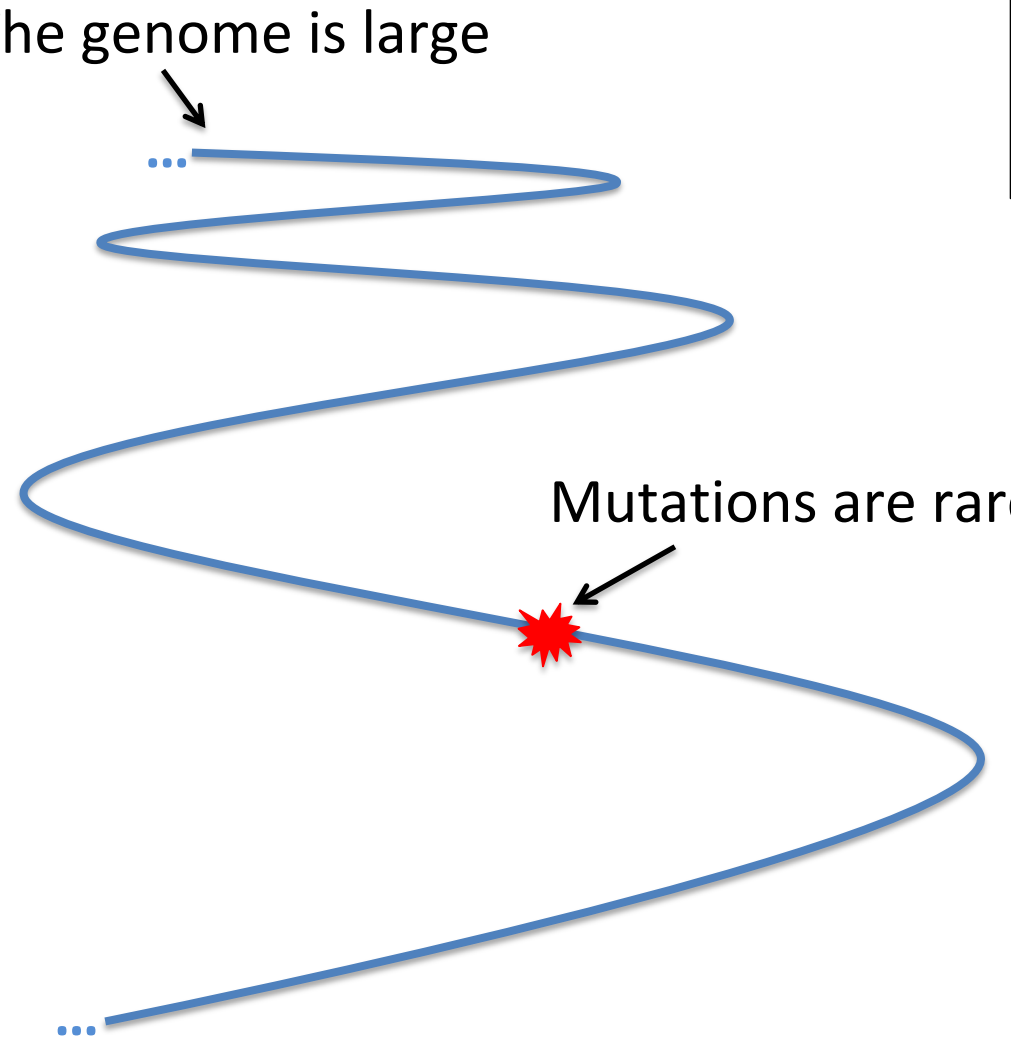
1 = mutated



Root is the normal, founder cell and leaves are cells in tumor.

Infinite sites assumption: each locus mutates only once.

Infinite Sites Model = Two-state Perfect Phylogeny



[Kimura, 1969]

Infinite sites model: multiple mutations never occur at the same position

Mutated Loci

Species (cancer cells)						
A	0	0	0	0	1	1
B	0	0	0	1	1	1
C	0	0	1	0	1	0
D	1	0	0	0	0	0
E	1	1	0	0	0	0

1: mutated
0: not

All sites are bi-allelic: mutated or not.

Two-state Perfect Phylogeny

Matrix $M \in \{0,1\}^{n \times m}$ has n taxa and m characters

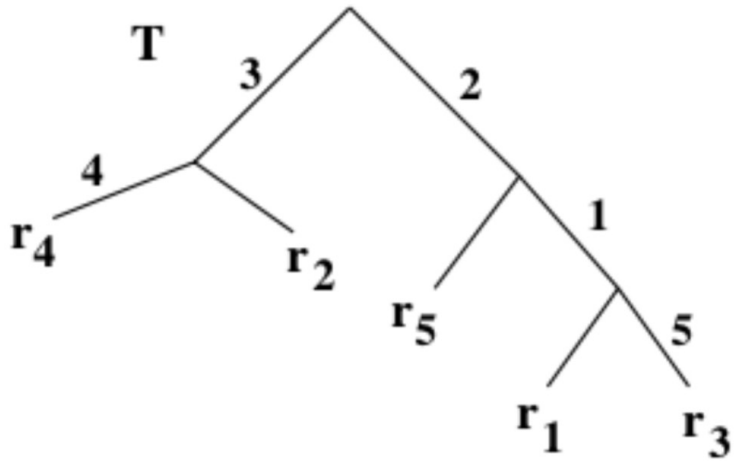
- Taxon f has state 1 for character c $\Leftrightarrow$ f possesses character c

	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	0	1
r_4	0	0	1	1	0
r_5	0	1	0	0	0

Definition

A perfect phylogeny for M is a rooted tree T with n leaves such that:

1. Each taxon labels only one leaf
2. Each character labels only one edge
3. Character possessed by a taxon are on unique path to root



Root node is all zero ancestor

Two-state Perfect Phylogeny Problem

Input:

Matrix $M \in \{0,1\}^{n \times m}$ has n taxa and m characters

- Taxon f has state 1 for character c $\Leftrightarrow$ f possesses character c

	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	0	1
r_4	0	0	1	1	0
r_5	0	1	0	0	0

Problem

Given $M \in \{0,1\}^{n \times m}$ does M have a perfect phylogeny?

Try it yourself!

Only one of these matrices can be used to build a perfect phylogeny.

(1) As a group, **decide on an approach** to try to determine which one is which.

(2) Try out your approach to see if you can construct the tree.

(3) What did you learn from your attempt?

M_1 =

Species	C_1	C_2	C_3	C_4	C_5
A	0	1	0	0	0
B	0	0	1	0	0
C	1	1	0	0	0
D	0	0	1	1	0
E	1	1	0	0	1

Characters

M_2 =

Species	C_1	C_2	C_3	C_4	C_5
A	0	0	1	1	0
B	0	0	1	0	1
C	1	1	0	0	1
D	1	1	0	0	0
E	0	1	0	0	1

Characters

The Perfect Phylogeny Problem – Preliminaries

Problem

Given $M \in \{0,1\}^{n \times m}$ does M have a perfect phylogeny?

Definition

$I(c)$ is the set of taxa that possess character c; and $\sigma(f)$ is the set of characters possessed by taxon f.

	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	0	1
r_4	0	0	1	1	0
r_5	0	1	0	0	0

$\Rightarrow$

	c_1 (2)	c_2 (1)	c_3 (3)	c_4 (5)	c_5 (4)
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	1	0
r_4	0	0	1	0	1
r_5	1	0	0	0	0

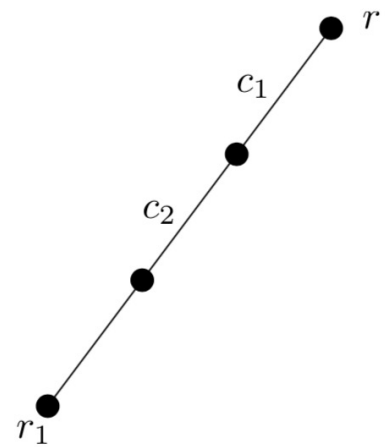
$$I(c_1) = \{r_1, r_3\}$$
$$\sigma(r_1) = \{c_1, c_2\}$$

Sort columns of M s.t. $c < d$ iff $|I(c)| \geq |I(d)|$. Break ties arbitrarily.

- Consider rows of M iteratively
  - T_i is tree of first i rows of M
- T_1 is a path graph
  - Terminal nodes r and 1
  - $|\sigma(1)| + 1$ edges labeled by $\sigma(1)$

$$c < d \text{ iff } |I(c)| \geq |I(d)|$$

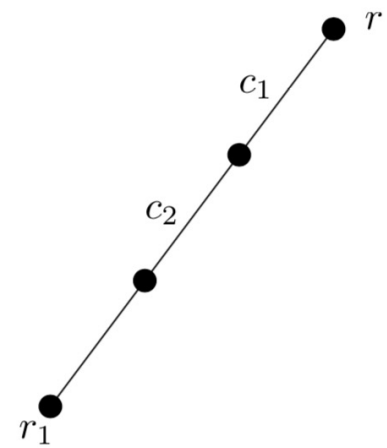
	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	1	0
r_4	0	0	1	0	1
r_5	1	0	0	0	0



- Consider rows of M iteratively
  - T_i is tree of first i rows of M
- T_1 is a path graph
  - Terminal nodes r and 1
  - $|\sigma(1)| + 1$ edges labeled by $\sigma(1)$
- T_{i+1} is a supertree of T_i
  - Let v be last node on walk from r matching characters $\sigma(i+1)$
    - ★ Character d is the last match
    - ★ Unmatched characters $\tau(i+1)$

$$c < d \text{ iff } |I(c)| \geq |I(d)|$$

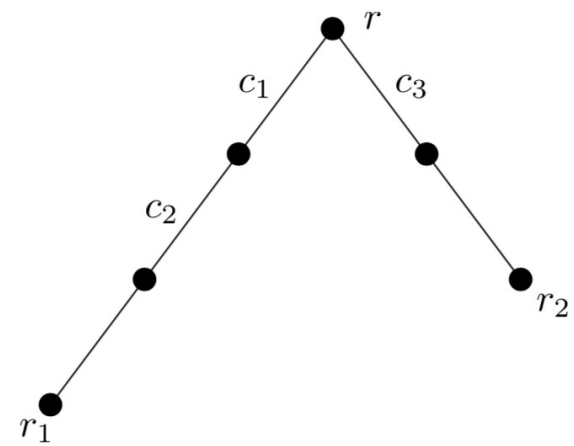
	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	1	0
r_4	0	0	1	0	1
r_5	1	0	0	0	0



- Consider rows of M iteratively
  - T_i is tree of first i rows of M
- T_1 is a path graph
  - Terminal nodes r and 1
  - $|\sigma(1)| + 1$ edges labeled by $\sigma(1)$
- T_{i+1} is a supertree of T_i
  - Let v be last node on walk from r matching characters $\sigma(i+1)$
    - ★ Character d is the last match
    - ★ Unmatched characters $\tau(i+1)$
  - Extend T_i with path Π
    - ★ Π has terminals v and $i+1$
    - ★ Π has $|\tau(i+1)| + 1$ edges labeled by $\tau(i+1)$

$$c < d \text{ iff } |I(c)| \geq |I(d)|$$

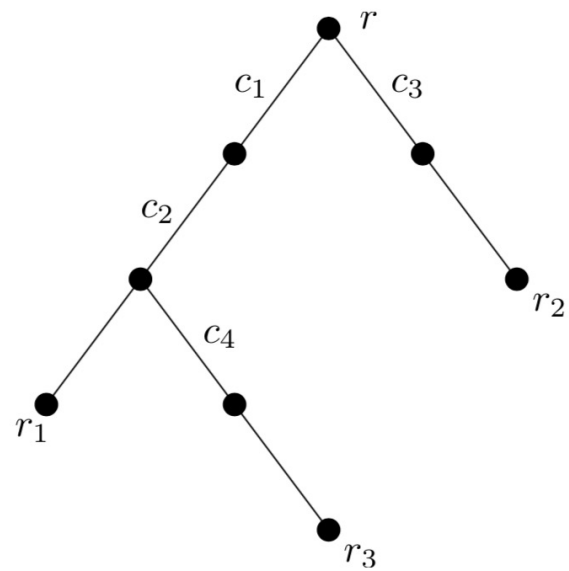
	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	1	0
r_4	0	0	1	0	1
r_5	1	0	0	0	0



- Consider rows of M iteratively
  - T_i is tree of first i rows of M
- T_1 is a path graph
  - Terminal nodes r and 1
  - $|\sigma(1)| + 1$ edges labeled by $\sigma(1)$
- T_{i+1} is a supertree of T_i
  - Let v be last node on walk from r matching characters $\sigma(i+1)$
    - ★ Character d is the last match
    - ★ Unmatched characters $\tau(i+1)$
  - Extend T_i with path Π
    - ★ Π has terminals v and $i+1$
    - ★ Π has $|\tau(i+1)| + 1$ edges labeled by $\tau(i+1)$

$$c < d \text{ iff } |I(c)| \geq |I(d)|$$

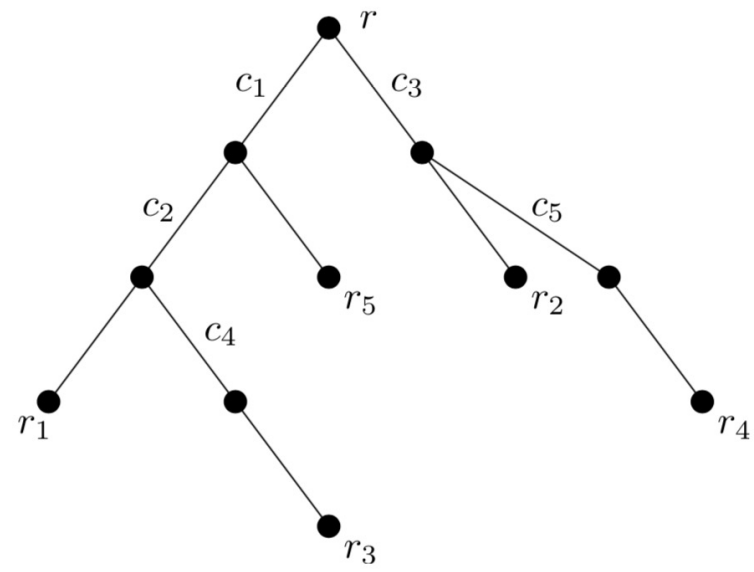
	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	1	0
r_4	0	0	1	0	1
r_5	1	0	0	0	0



- Consider rows of M iteratively
  - T_i is tree of first i rows of M
- T_1 is a path graph
  - Terminal nodes r and 1
  - $|\sigma(1)| + 1$ edges labeled by $\sigma(1)$
- T_{i+1} is a supertree of T_i
  - Let v be last node on walk from r matching characters $\sigma(i+1)$
    - ★ Character d is the last match
    - ★ Unmatched characters $\tau(i+1)$
  - Extend T_i with path Π
    - ★ Π has terminals v and $i+1$
    - ★ Π has $|\tau(i+1)| + 1$ edges labeled by $\tau(i+1)$

$$c < d \text{ iff } |I(c)| \geq |I(d)|$$

	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	1	0
r_4	0	0	1	0	1
r_5	1	0	0	0	0



Lemma

Let $M_i \in 0,1^{i \times m}$ be a submatrix of M. If M is conflict-free then T_i is a perfect phylogeny for M_i.

Outline

- Two-State Perfect Phylogeny
- Multi-State Perfect Phylogeny
- Large Maximum Parsimony Phylogeny Problem
- Summary

Reading:

- Lecture notes

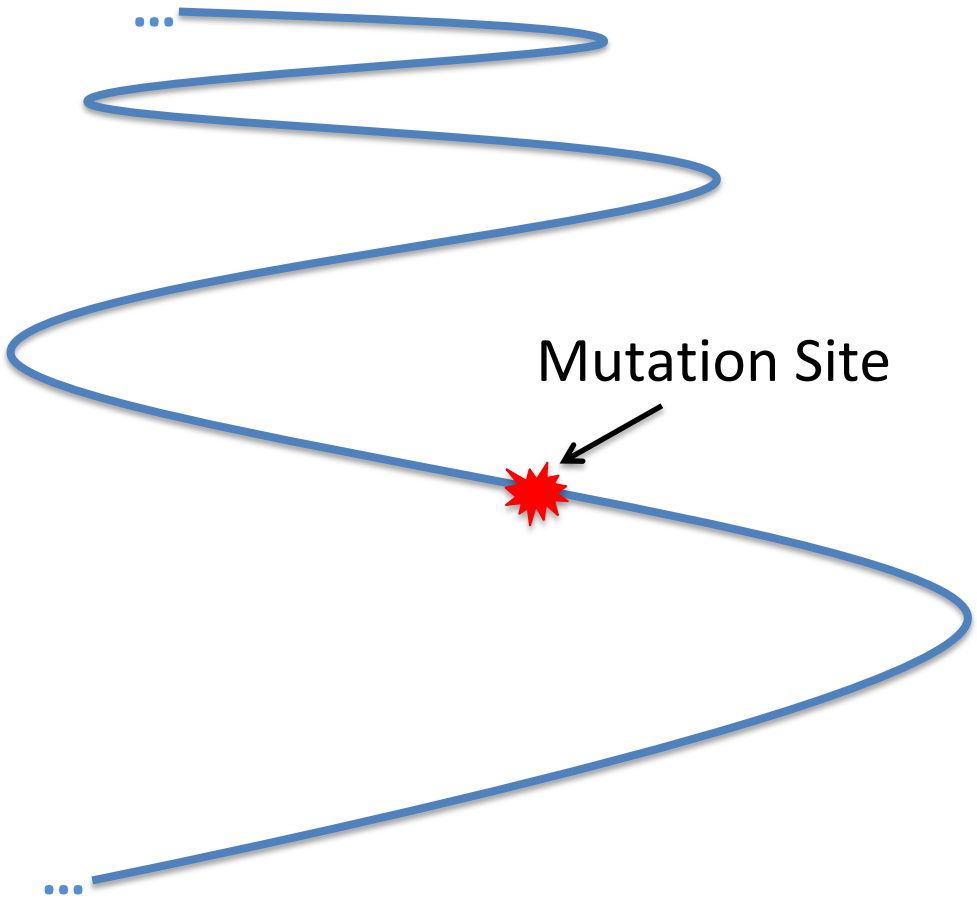
Integer Characters

Species	1	2	3	4	5
	Characters				
A	2	1	1	0	0
B	0	2	1	2	2
C	1	2	1	1	1
D	1	1	0	1	2

Characters have ***k*** possible states

Question: Given *n* integer characters with *k* states, what is the smallest parsimony score?

Infinite Alleles Model = Multi-state Perfect Phylogeny

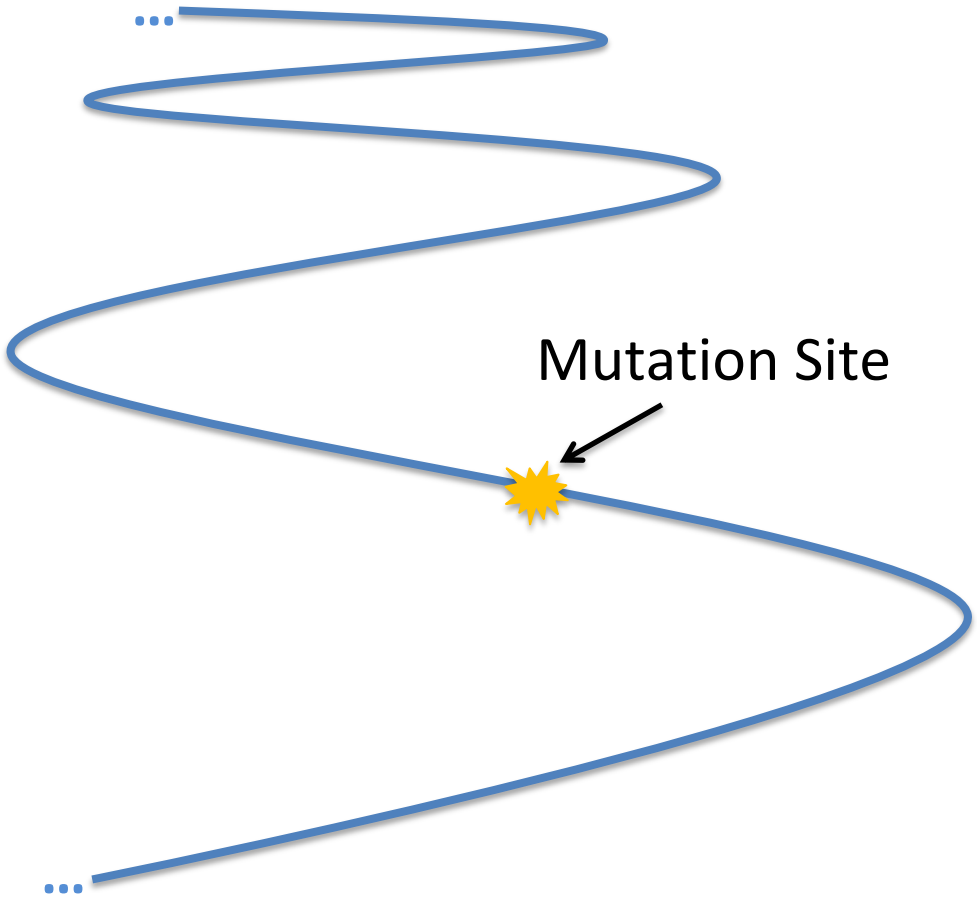


Infinite alleles model:

- For any mutation, there are an infinite number of possibilities of what mutation looks like (states).
- So, the same position can be mutated multiple times, but it never mutates to the same "allele" or state.

Characters have integer states

Infinite Alleles Model = Multi-state Perfect Phylogeny

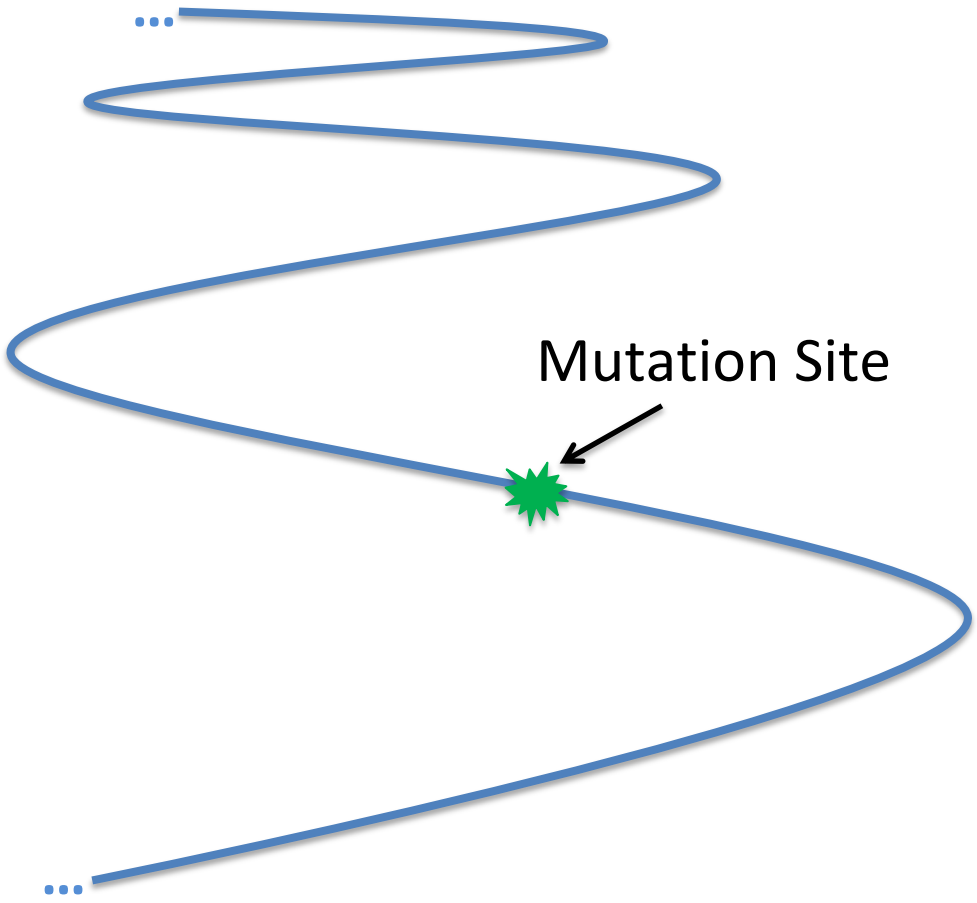


Infinite alleles model:

- For any mutation, there are an infinite number of possibilities of what mutation looks like (states).
- So, the same position can be mutated multiple times, but it never mutates to the same "allele" or state.

Characters have integer states

Infinite Alleles Model = Multi-state Perfect Phylogeny



Infinite alleles model:

- For any mutation, there are an infinite number of possibilities of what mutation looks like (states).
- So, the same position can be mutated multiple times, but it never mutates to the same "allele" or state.

Characters have integer states

Multi-state Perfect Phylogeny

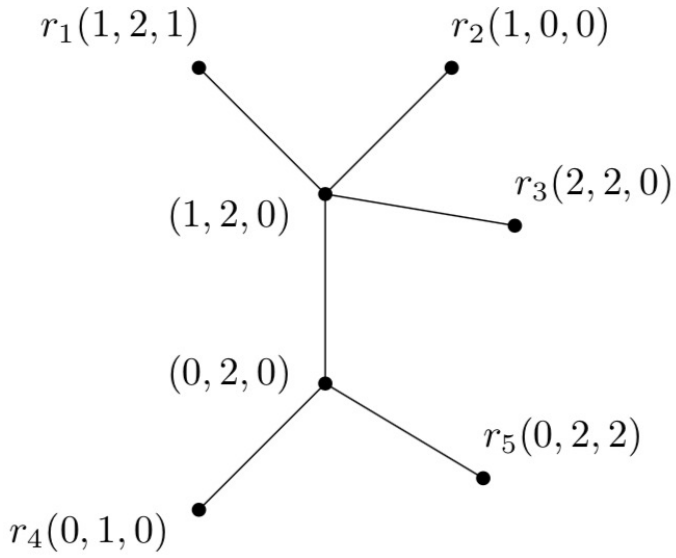
Matrix $M \in \{0, \ldots, k-1\}^{n \times m}$ has
n taxa and m characters

	c_1	c_2	c_3
r_1	1	2	1
r_2	1	0	0
r_3	2	2	0
r_4	0	1	0
r_5	0	2	2

Definition

A multi-state perfect phylogeny for M is a tree T with n leaves such that:

1. Each taxon labels exactly one leaf
2. Each node is labeled by $\{0, \ldots, k-1\}^m$
3. Nodes labeled with state i for character c form a connected subtree $T_c(i)$



Theorem (Bodlaender et al., 1992) [Bodlaender, Fellows and Warnow]

For general k, the multi-state perfect phylogeny problem is NP-complete

Cladistic vs. Qualitative Characters

Definition

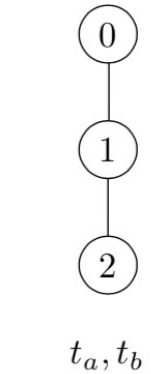
A multi-state perfect phylogeny for M is a tree T with n leaves such that:

1. Each taxon labels exactly one leaf
2. Each node is labeled by $\{0, \ldots, k-1\}^m$
3. Nodes with state i for character c form a connected subtree $T_c(i)$

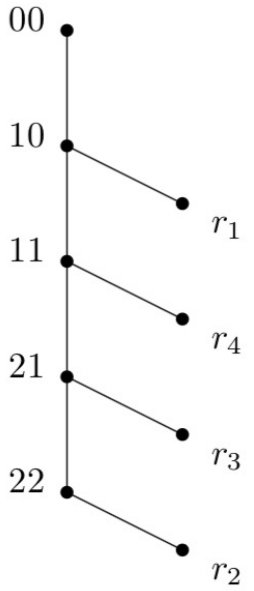
A cladistic character c has a state tree t_c on its states

A phylogeny T is consistent if the reduced tree $\sigma(T, c)$ is identical with t_c for all c

	a	b
r_1	1	0
r_2	2	2
r_3	2	1
r_4	1	1



t_a, t_b



Cladistic vs. Qualitative Characters

Definition

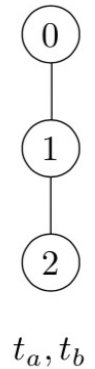
A multi-state perfect phylogeny for M is a tree T with n leaves such that:

1. Each taxon labels exactly one leaf
2. Each node is labeled by $\{0, \ldots, k-1\}^m$
3. Nodes with state i for character c form a connected subtree $T_c(i)$

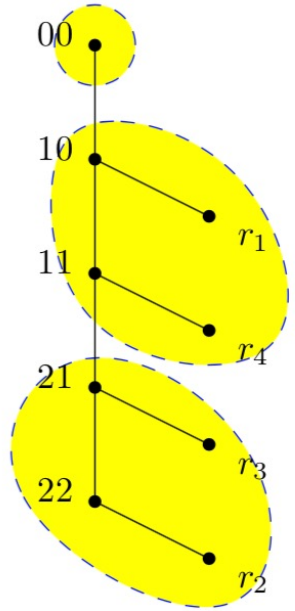
A cladistic character c has a state tree t_c on its states

A phylogeny T is consistent if the reduced tree $\sigma(T, c)$ is identical with t_c for all c

	a	b
r_1	1	0
r_2	2	2
r_3	2	1
r_4	1	1



t_a, t_b



Cladistic vs. Qualitative Characters

Definition

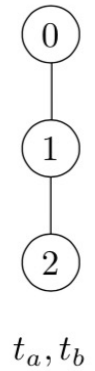
A multi-state perfect phylogeny for M is a tree T with n leaves such that:

1. Each taxon labels exactly one leaf
2. Each node is labeled by $\{0, \ldots, k-1\}^m$
3. Nodes with state i for character c form a connected subtree $T_c(i)$

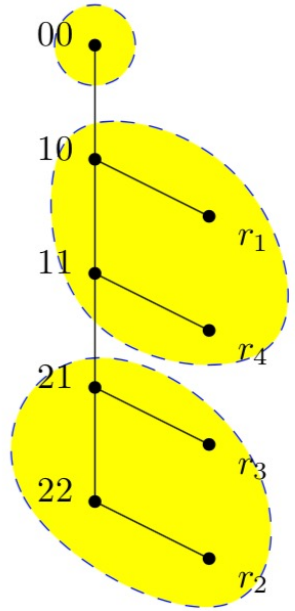
A cladistic character c has a state tree t_c on its states

A phylogeny T is consistent if the reduced tree $\sigma(T, c)$ is identical with t_c for all c

	a	b
r_1	1	0
r_2	2	2
r_3	2	1
r_4	1	1



t_a, t_b



Multi-state Cladistic Perfect Phylogeny

Outline

- Two-State Perfect Phylogeny
- Multi-State Perfect Phylogeny
- Large Maximum Parsimony Phylogeny Problem
- Summary

Reading:

- Lecture notes

Small and a Large Problem

Small Maximum Parsimony Phylogeny Problem:

Given $m \times n$ matrix $A = [a_{i,j}]$ and tree T with m leaves, find assignment of character states to each internal vertex of T with minimum parsimony score.

Large Maximum Parsimony Phylogeny Problem:

Given $m \times n$ matrix $A = [a_{i,j}]$, find a tree T with m leaves labeled according to A and an assignment of character states to each internal vertex of T with minimum parsimony score.

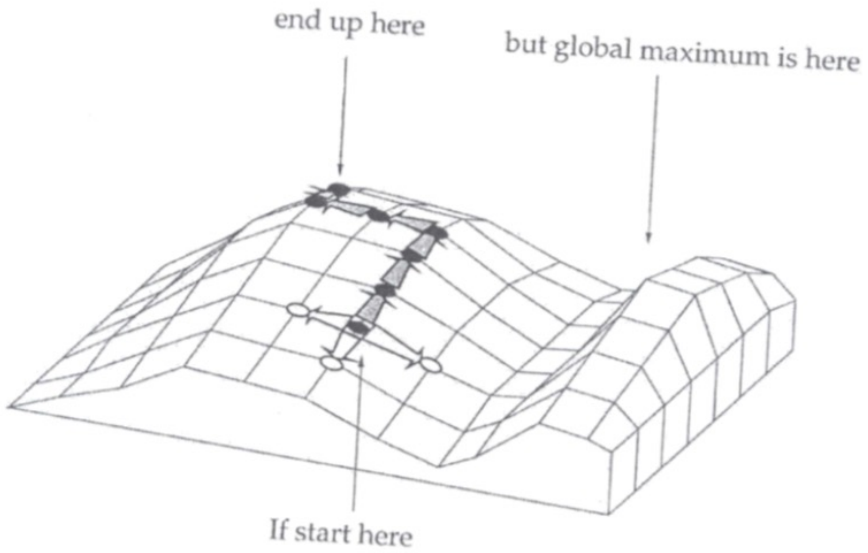
General Large Maximum Parsimony Phylogeny

- This problem is NP-hard
- Heuristics using local search (tree moves)

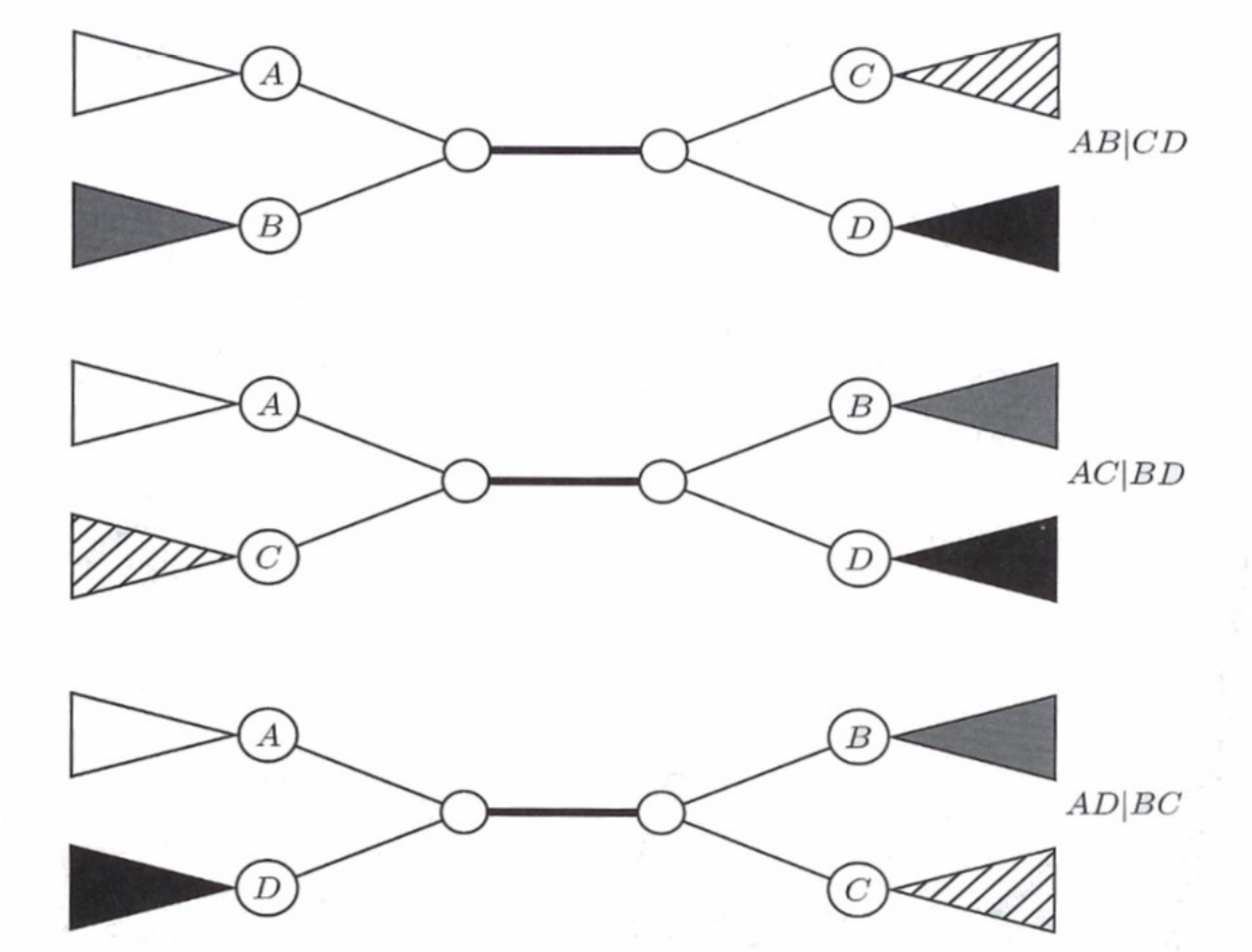
1. Start with an arbitrary tree *T.*
2. Check "neighbors" of *T*.
3. Move to a neighbor if it provides the best improvement in parsimony/likelihood score.

Caveats:

Could be stuck in **local** optimum, and not achieve global optimum



Example: Nearest-Neighbor Interchange (NNI)



Rearrange four subtrees defined by one internal edge

Figure: Jones and Pevzner

Outline

- Two-State Perfect Phylogeny
- Multi-State Perfect Phylogeny
- Large Maximum Parsimony Phylogeny Problem
- Summary

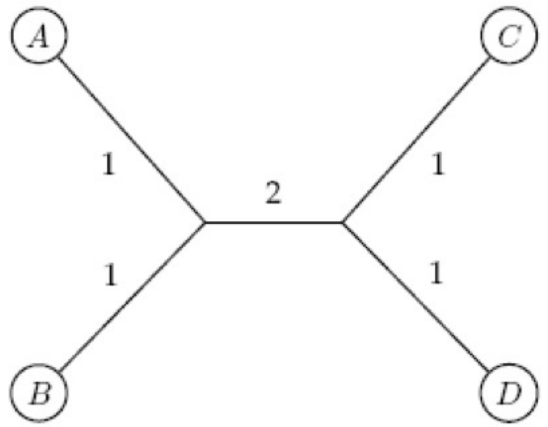
Reading:

- Lecture notes

Distance-based Phylogeny

- Small additive distance phylogeny problem
  - In P
  - Recursive algorithm using neighboring leaves
- Large additive distance phylogeny problem
  - In P -- two algorithms:
    1. Find degenerate triples and resolve these
    2. Neighbor joining: identifies neighboring leaves even when tree is not given
  - Complete characterization of additive matrices using the four-point condition

	A	B	C	D
A	0	2	4	4
B	2	0	4	4
C	4	4	0	2
D	4	4	2	0



Character-based Phylogeny

- Small maximum parsimony problem
  - Sankoff algorithm: dynamic programming
- Two-state perfect phylogeny problem
  - In P: O(mn) time
  - Complete characterization as conflict free binary matrices
- Multi-state perfect phylogeny problem
  - NP-hard in general
  - In P given state trees
- Large maximum parsimony problem
  - NP-hard
  - Heuristic using local search