CS 466
Introduction to Bioinformatics
Lecture 17

Mohammed El-Kebir

October 23, 2020

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

Characters

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

matrix A

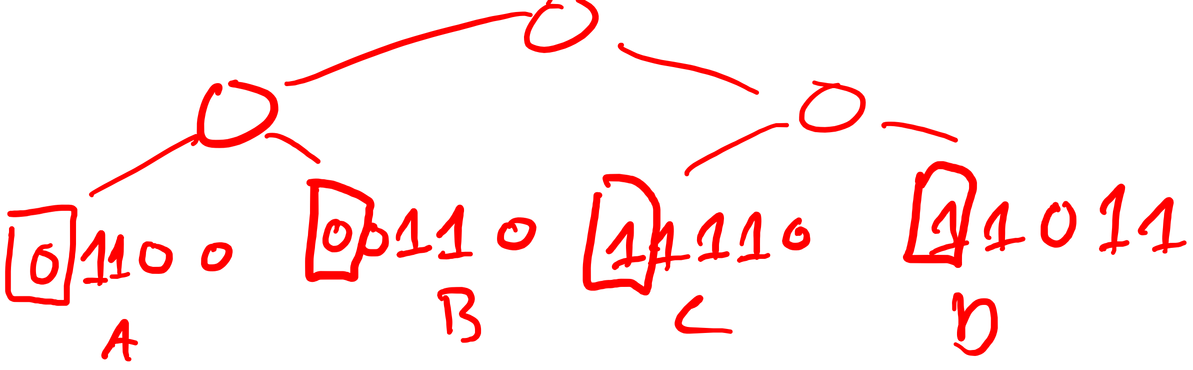
Characters only have two possible states

Possible Encoding:
0 : not-mutated
1 : mutated

wild type

Possible Encoding:
0 : no wings
1 : wings

Binary Characters



Characters

Species	1	2	3	4	5
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

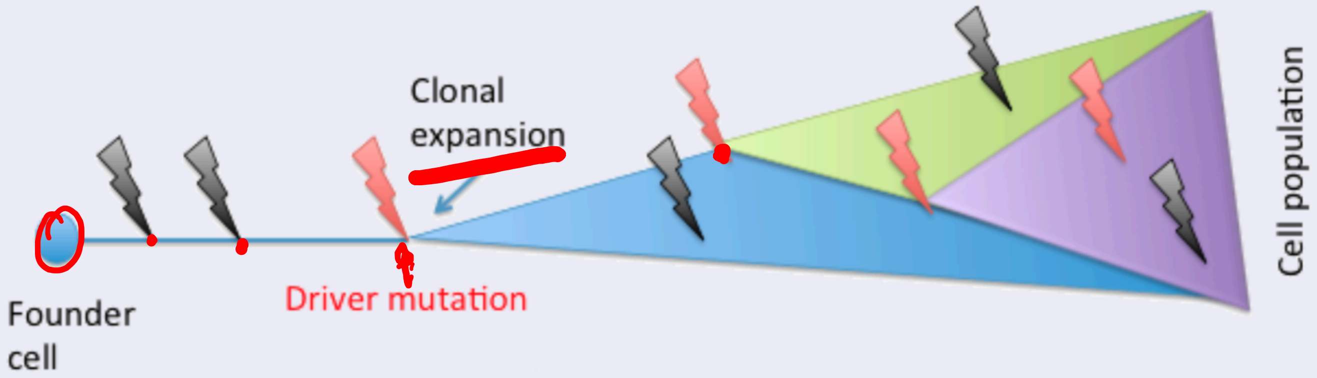
Possible Encoding:
0 : no wings
1 : wings

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

Somatic Mutations and Cancer

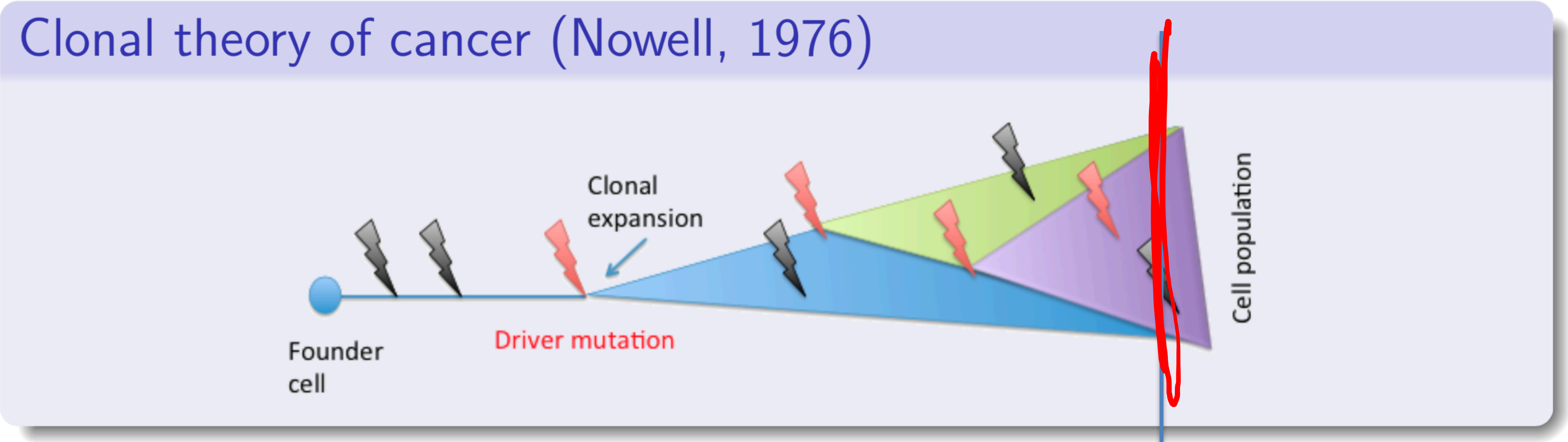
↳ mutations in normal cells (NOT inherited)

Clonal theory of cancer (Nowell, 1976)



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

Somatic Mutations and Cancer



Clonal theory of cancer (Nowell, 1976)

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

The Cancer Genome Atlas

International Cancer Genome Consortium

Sequence genome

Progression of Somatic Mutations

Single nucleotide mutation

ref ... CGTAATTAG ...

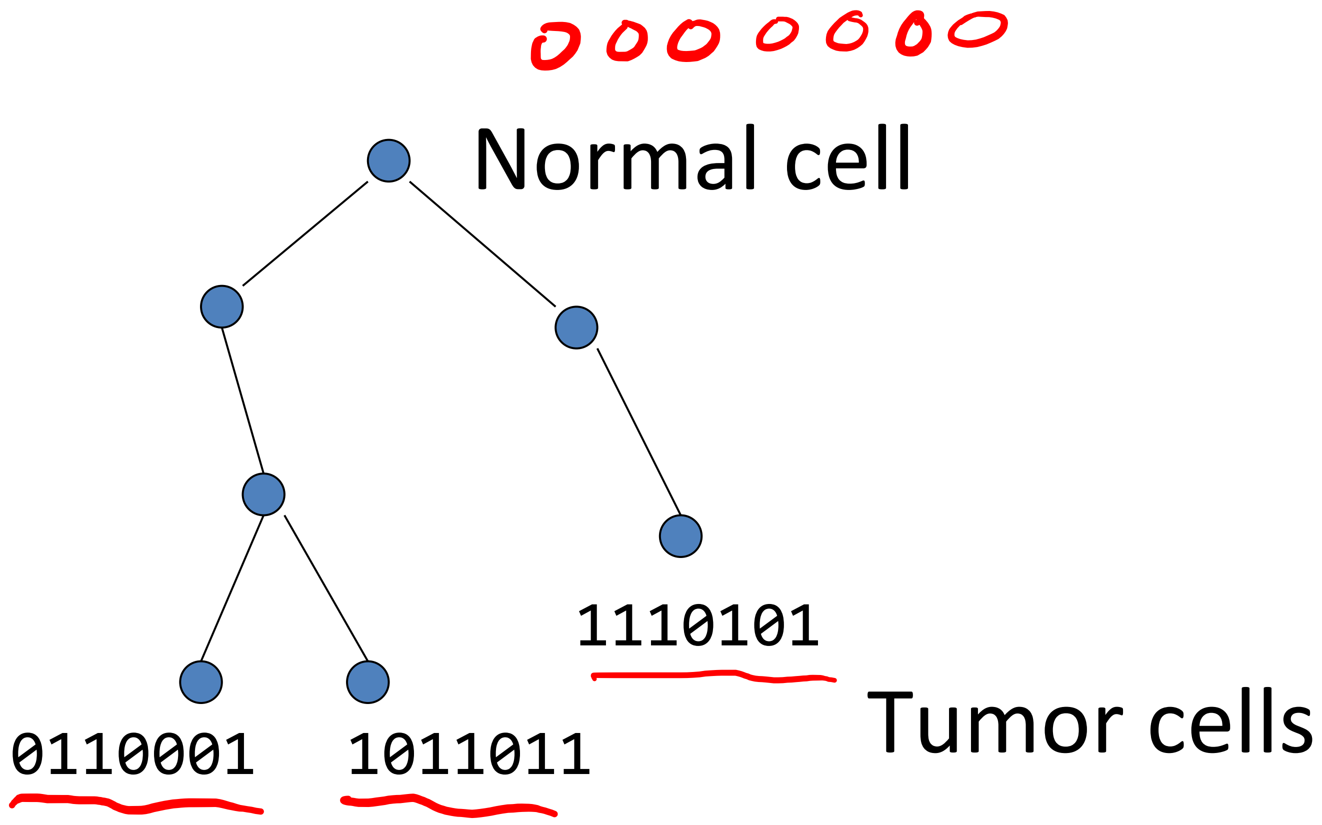
0

tumor cell ... CGTCATTAG ...

1

0 = normal

1 = mutated



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

Progression of Somatic Mutations

Single nucleotide mutation

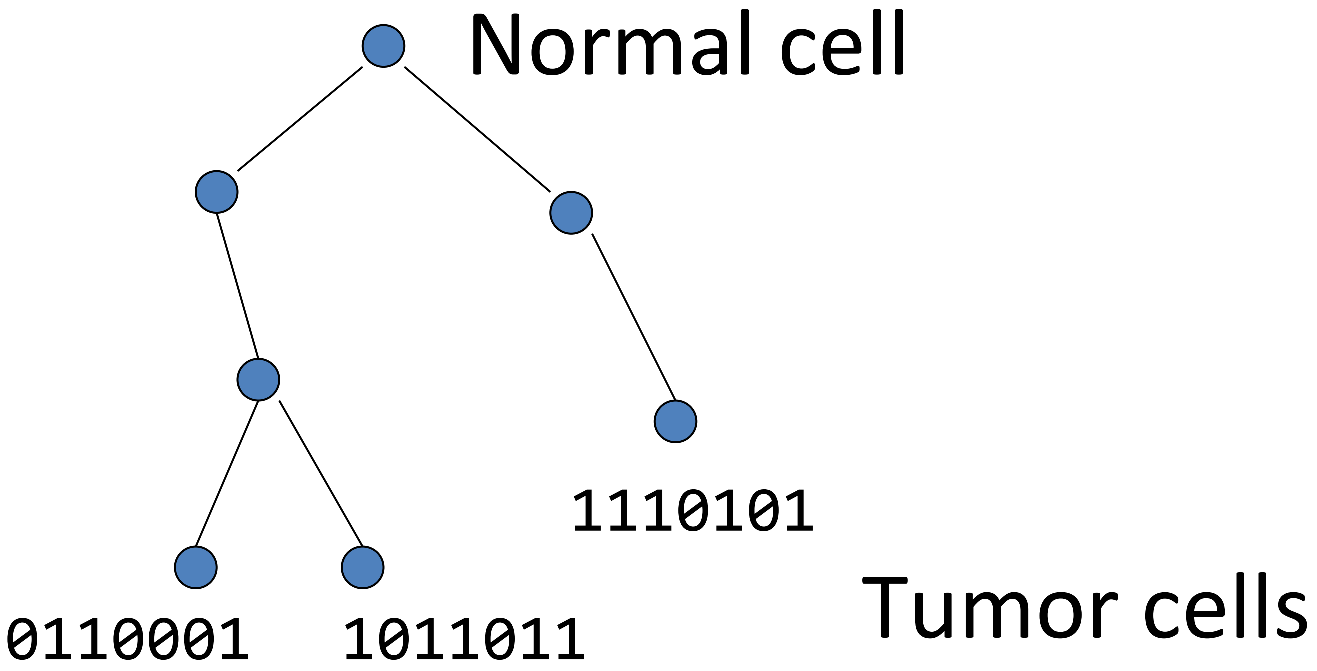
… CGTAATTAG …

↓

… CGTCATTAG …

0 = normal

1 = mutated

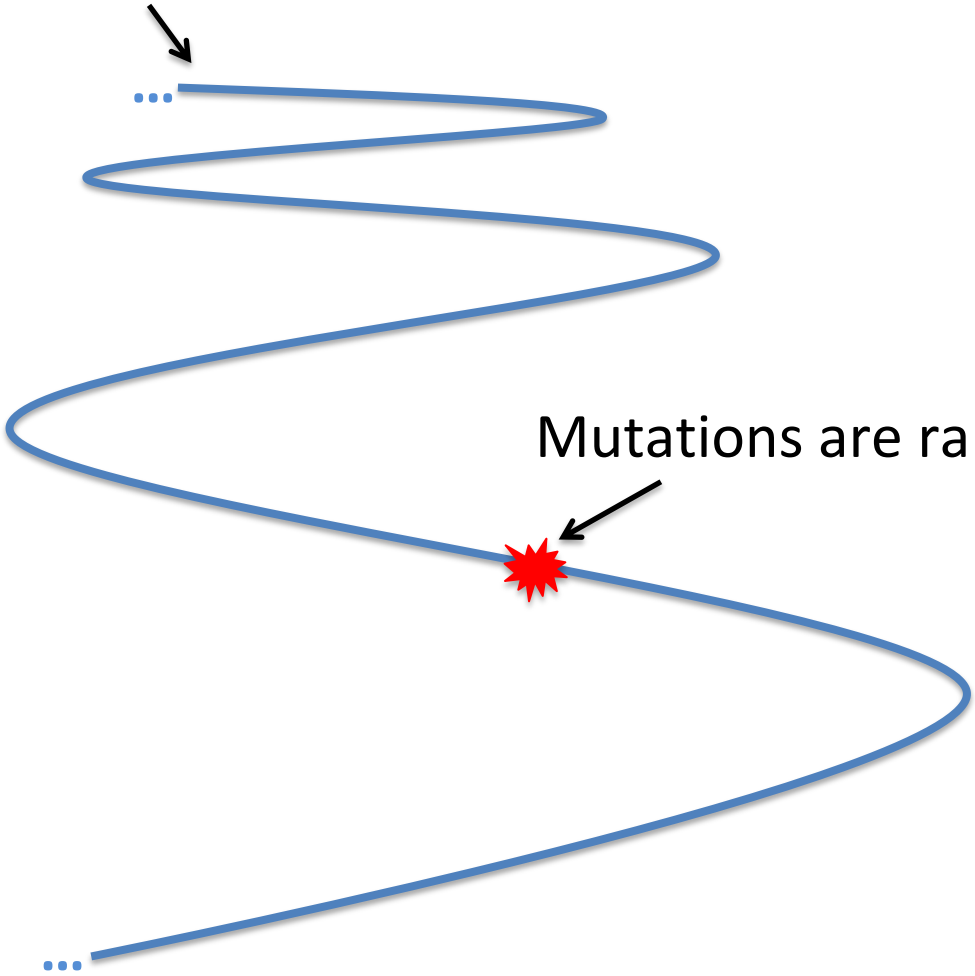


Root is the normal, founder cell and leaves are cells in tumor.

Infinite sites assumption: each locus mutates only once.

Infinite Sites Model = Two-state Perfect Phylogeny

The genome is large



Mutations are rare

[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

characters

taxa

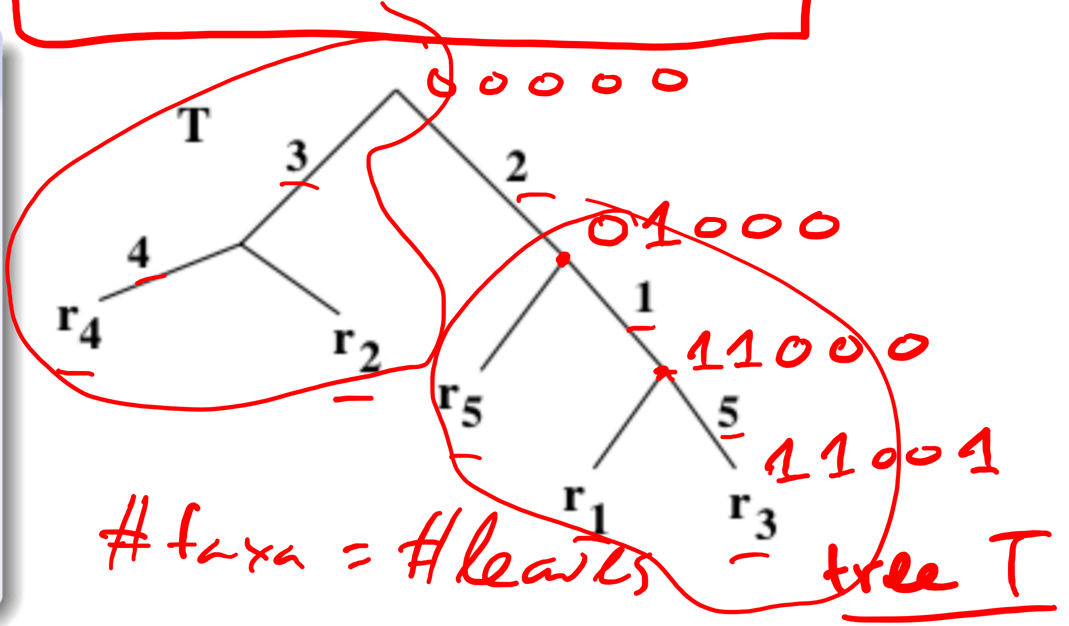
	c_1	c_2	c_3	c_4	c_5	
r_1	1	1	0	0	0	c_1 c_2
r_2	0	0	1	0	0	c_3
r_3	1	1	0	0	1	c_1 c_2 c_5
r_4	0	0	1	1	0	c_3 c_4
r_5	0	1	0	0	0	c_2

input

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. Characters possessed by a taxon are on unique path to root



#taxa = #leaves

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? If so, construct this tree?

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?

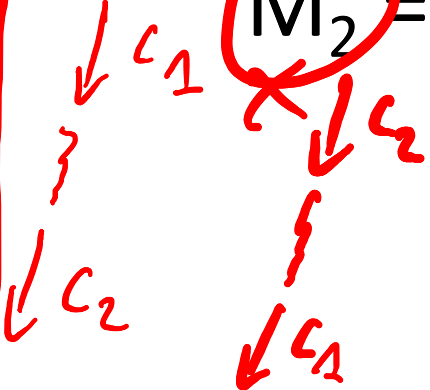
Characters

$$\checkmark\ M_1 = \text{Species}\quad \begin{array}{c|ccccc} & C_1 & C_2 & C_3 & C_4 & C_5 \\ \hline A & 0 & 1 & 0 & 0 & 0 \\ B & 0 & 0 & 1 & 0 & 0 \\ C & 1 & 1 & 0 & 0 & 0 \\ D & 0 & 0 & 1 & 1 & 0 \\ E & 1 & 1 & 0 & 0 & 1 \end{array}$$

$I(C_1) = \{\cancel{A}\, C, E\}$

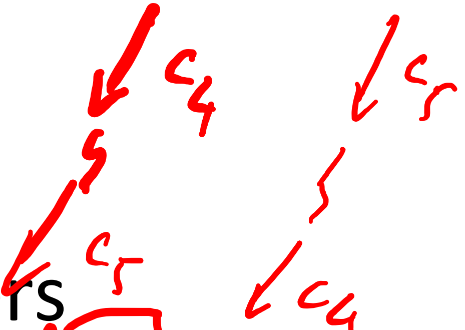
$I(C_2) = \{A, C, E\}$

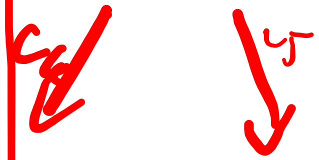
$I(C_2) \supseteq I(C_1)$



m Characters

$$M_2 = \text{Species}\ n\quad \begin{array}{c|ccccc} & C_1 & C_2 & C_3 & C_4 & C_5 \\ \hline A & 0 & 0 & 1 & 1 & 0 \\ B & 0 & 0 & 1 & 0 & 1 \\ C & 1 & 1 & 0 & 0 & 1 \\ D & 1 & 1 & 0 & 0 & 0 \\ E & 0 & 1 & 0 & 0 & 1 \end{array}$$





$O(n^3 m^2)$

$O(m^2 n)$

Problem. Does a given binary matrix $B \in \{0,1\}^{n\times m}$ have a two-state p.p. T? If so, construct T.

necessary & sufficient

Goal: $O(mn)$ time ~~decides~~ that solves problem.
linear time algorithm

1) Let $\bar{B} \in \{0,1\}^{n\times m}$ obtained from B by sorting colums of B in descending order by the number of ones they contain. Matrix B has a two-state p.p. if and only if matrix $\bar{B}$ has a two-state p.p.

② $B' \in \{0,1\}^{n\times m'}$ from $B^{\in\{0,1\}^{n\times m}}$ s.t. B' does not contain any repeated columns in B.

$$m' \leq m.$$

Def 2. Binary matrix $B \in \{0,1\}^{n\times m}$ B conflict free if no pair of columns c and d contain the three pairs $(0,1)$ $(1,0)$ $(1,1)$.

$O(mn)$ time $\cancel{nm^2}$

Thm. Matrix B has a two-state p.p. **if and only if** B is conflict free.

Two naive algorithms for checking conflict free property

① look at all 3×2 submatrices

$O(n^3 m^2)$ time

② look at all pairs of columns $\rightarrow O(m^2)$

$\rightarrow$ for each pair scan through rows $O(n)$

$O(nm^2)$

③ $O(mn)$ time

Lemma (Shared prefix property).

(Pre: B is 'sorted')

Let d be the rightmost column in B possessed by two taxa f and g.

Then, if no pair of columns of B conflict, f and g must be identical from column 1 to d.

					d
→	f	0	1	.	1
→	g	0	?	.	1

The Perfect Phylogeny Problem – Preliminaries

O(mn) time alg.

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.

c5 c3

c4 c1 c2

0	1	2	3	4	5

O(n)

m character

	2	3	2	1	1
	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

n rows

$\Rightarrow$

	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
	3	2	2	1	1

O(mn) time

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

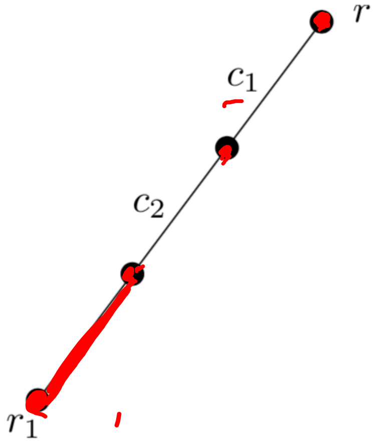
n log m time

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 $\sigma(1)$

$c < d$ iff $|I(c)| \geq |I(d)|$

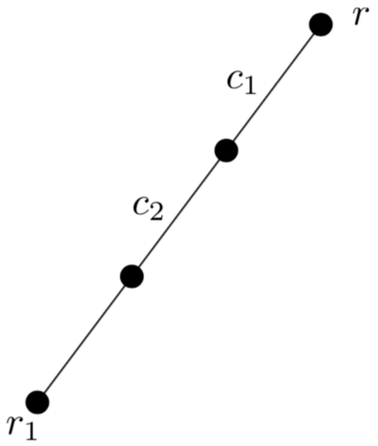
	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	1	0
r_4	0	0	1	0	1
r_5	1	0	0	0	0



$c < d$ iff $|I(c)| \geq |I(d)|$

- 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 $\tau(i+1)$

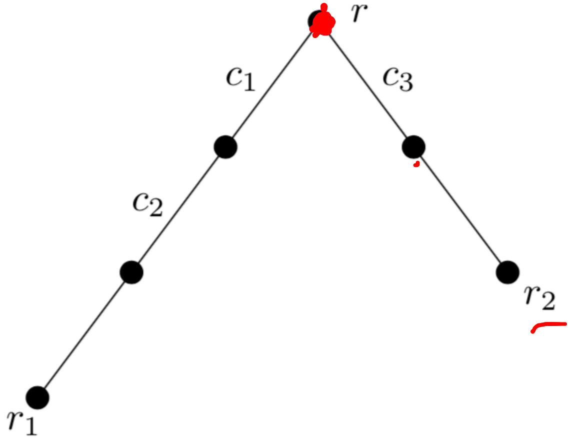
	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	1	0
r_4	0	0	1	0	1
r_5	1	0	0	0	0



- 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 $\tau(i+1)$

$c < d$ 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



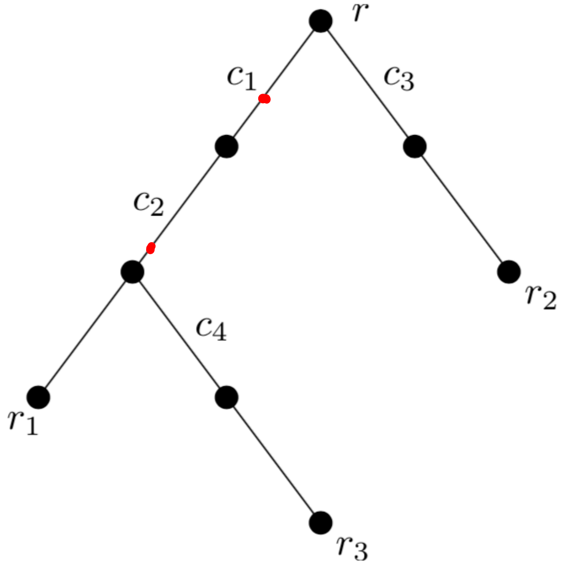
- Consider rows of M iteratively
  - T_i is tree of first i rows of M
- T_1 is a path graph
  - Terminal nodes r and 1
  - $|\sigma(1)| + 1$ edges labeled by $\sigma(1)$
- T_{i+1} is a supertree of T_i
  - Let v be last node on walk from r matching characters $\sigma(i+1)$
    - Character d is the last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

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

n

n

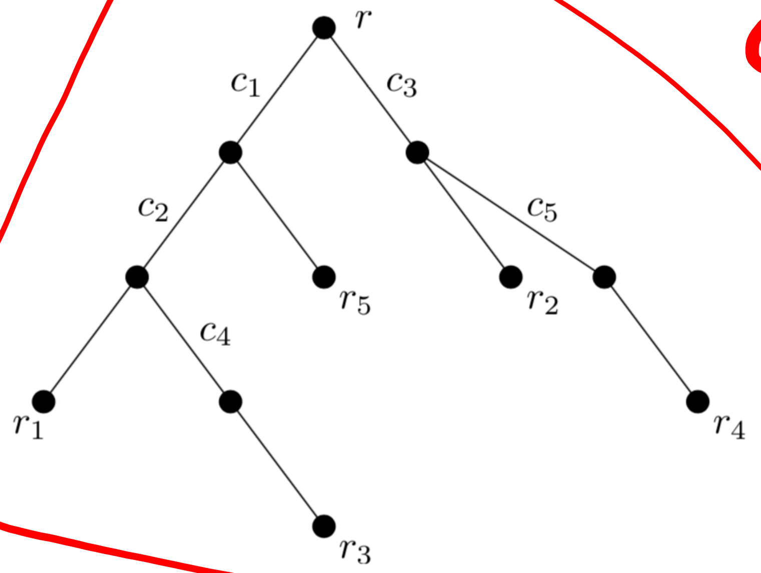
	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	1	0
r_4	0	0	1	0	1
r_5	1	0	0	0	0



- 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



O(mn) time

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

binary characters $k=2$

↳ n

n Characters

m Species	1	2	3	4	5
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

$n \cdot \underline{(k-1)}$

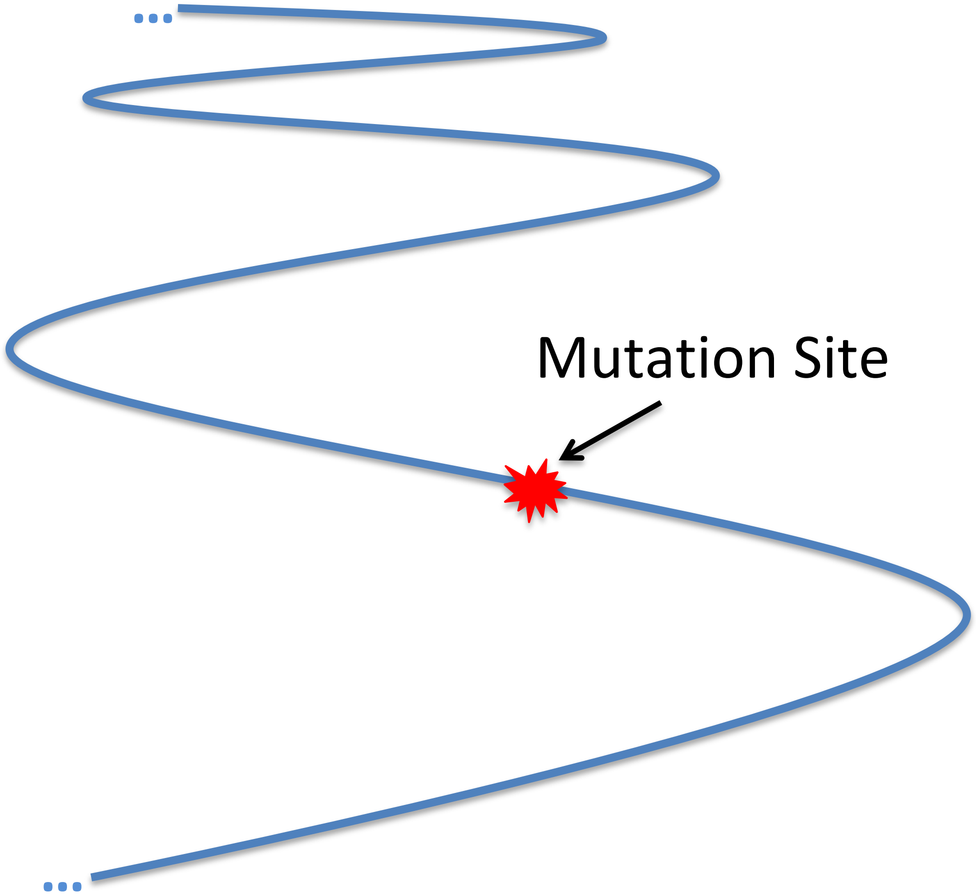
$k=2 : n(2-1) = n$

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

$k \geq 2$

Infinite Alleles Model = Multi-state Perfect Phylogeny

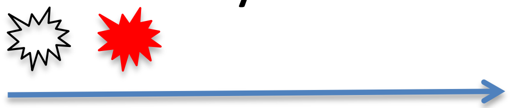
Infinite vs. Sites Model $(k=2)$



Infinite alleles model:

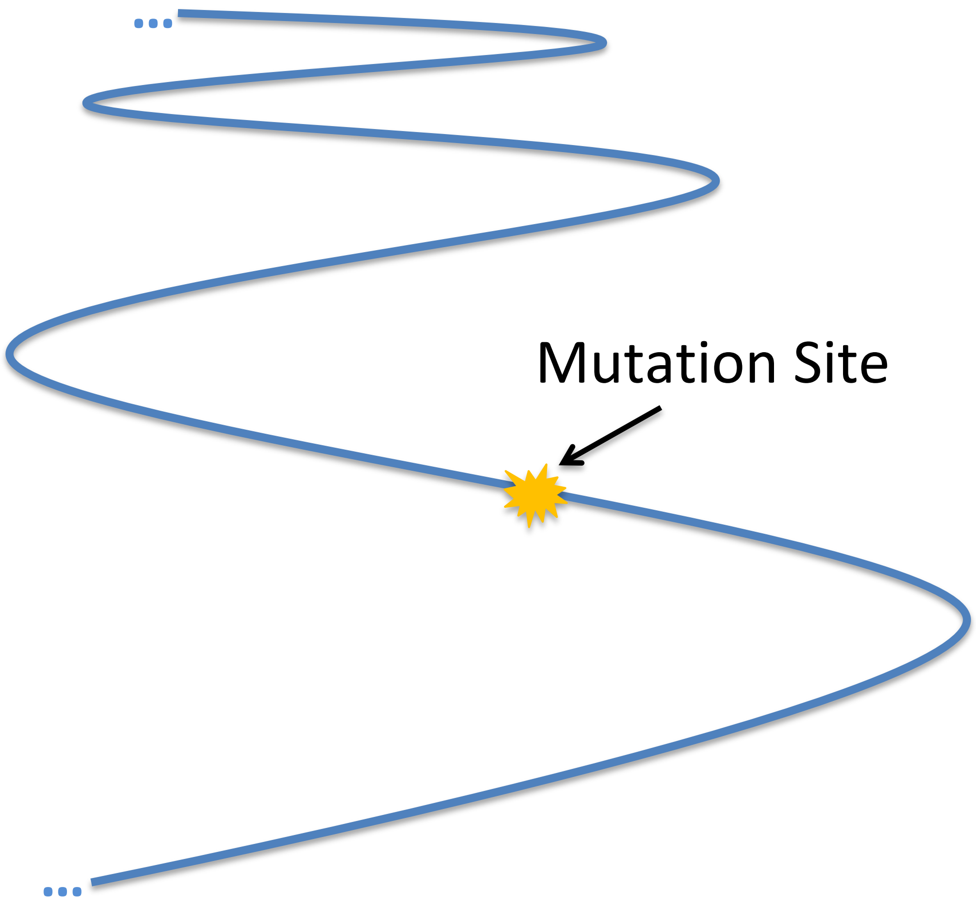
- For any mutation, there are an infinite number of possibilities of what mutation looks like (states).
- So, the same position can be mutated multiple times, but it never mutates to the same “allele” or state.

Site History:



Characters have integer states

Infinite Alleles Model = Multi-state Perfect Phylogeny

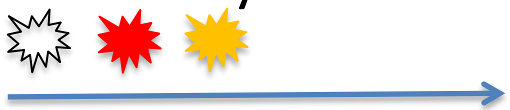


Mutation Site

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.

Site History:

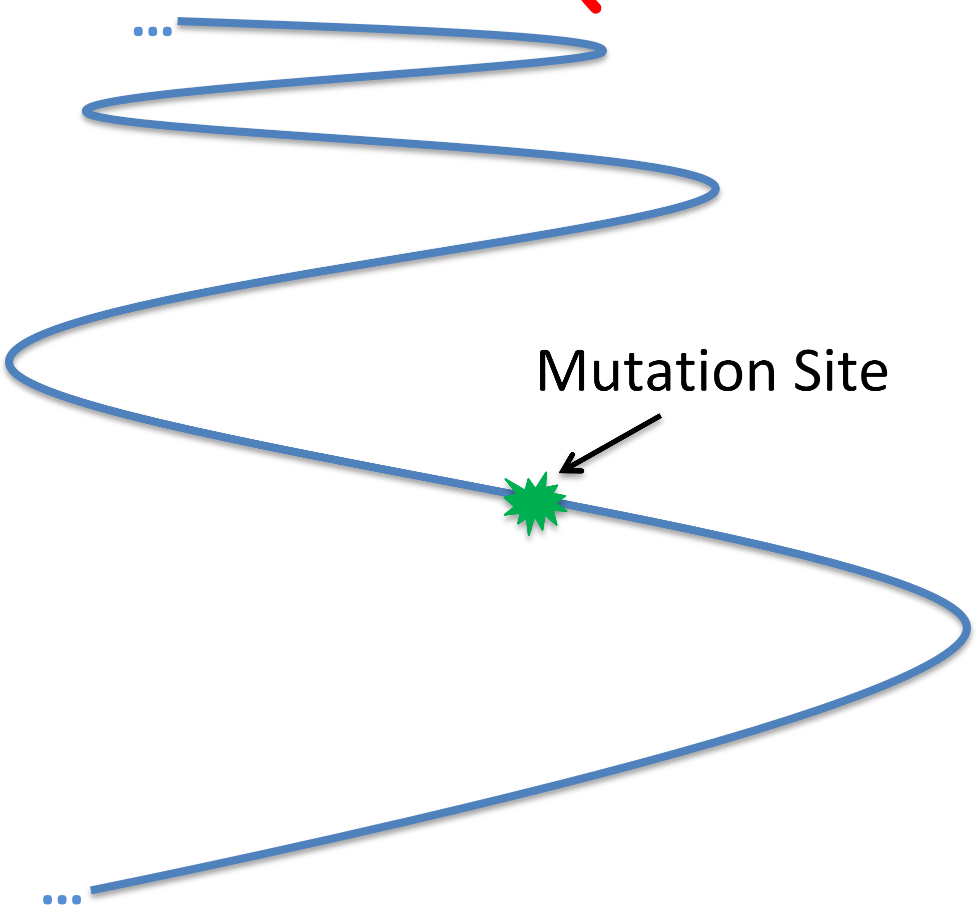


Time

Characters have integer states

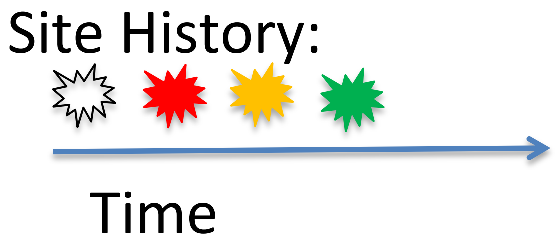
Infinite Alleles Model = Multi-state Perfect Phylogeny

$(k-1)\, n$



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

m characters

n taxa

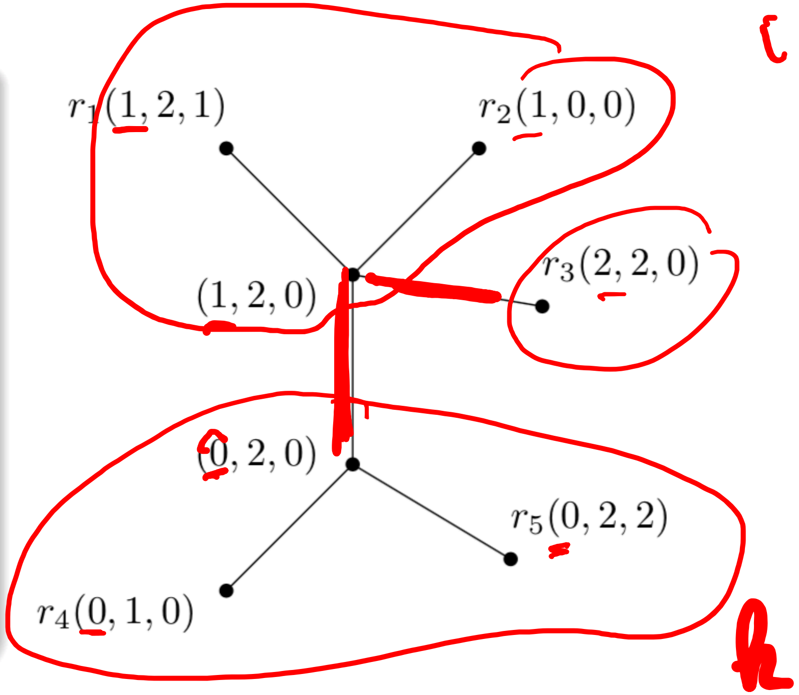
	c_1	c_2	c_3
r_1	1	2	1
r_2	1	0	0
r_3	2	2	0
r_4	0	1	0
r_5	0	2	2

c = 1
c = 0

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 $\{0, \dots, k-1\}^m$
3. Nodes labeled with state i for character c form a connected subtree $T_c(i)$



k = O(n)
k = O(n)

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 $T_c(i)$

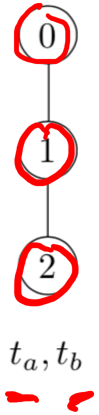
A cladistic character c has a state tree t_c on its states

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

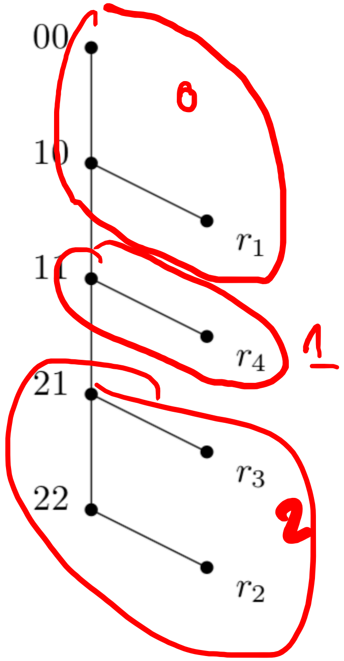
characters

taxa

	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:

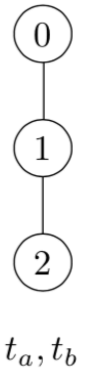
1. Each taxon labels exactly one leaf
2. Each node is labeled by $\{0, \ldots, k-1\}^m$
3. Nodes with state i for character c form a connected subtree $T_c(i)$

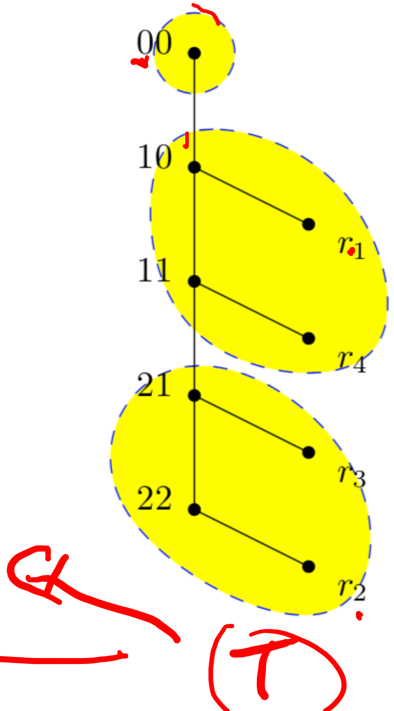
A cladistic character c has a state tree t_c on its states

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

R





Cladistic vs. Qualitative Characters

Definition

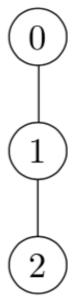
A multi-state perfect phylogeny for M is a tree T with n leaves such that:

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 $T_c(i)$

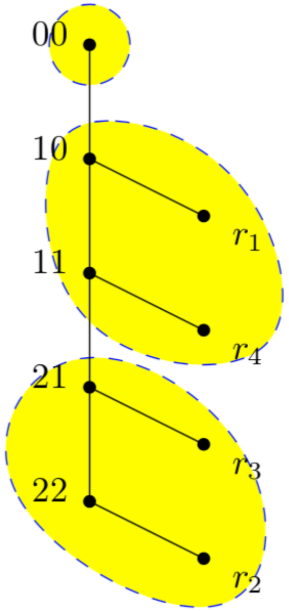
A cladistic character c has a state tree t_c on its states

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

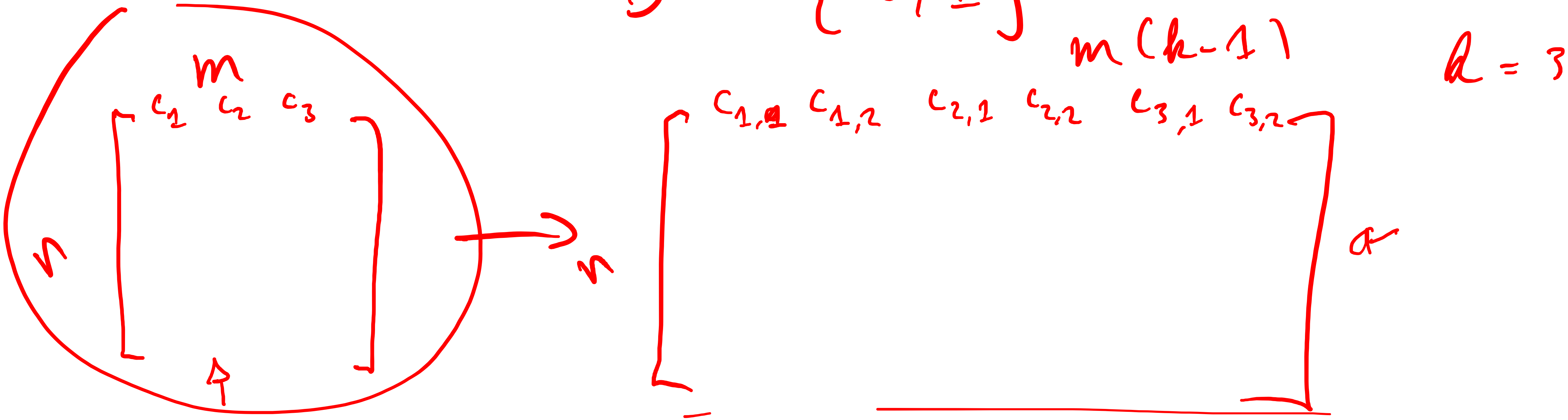


Multi-state Cladistic Perfect Phylogeny

$n \times m$ matrix $A \in \{0, \dots, k-1\}^{n \times m}$

$\Big\downarrow$ Use $\underline{t_1, \dots, t_m}$

$B \in \{0,1\}^{n \times (k-1)m}$



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.

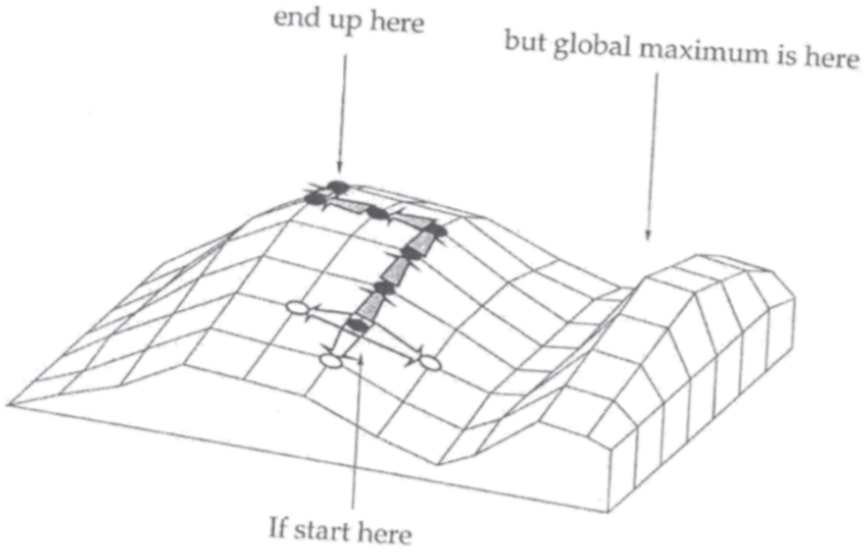
General Large Maximum Parsimony Phylogeny

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

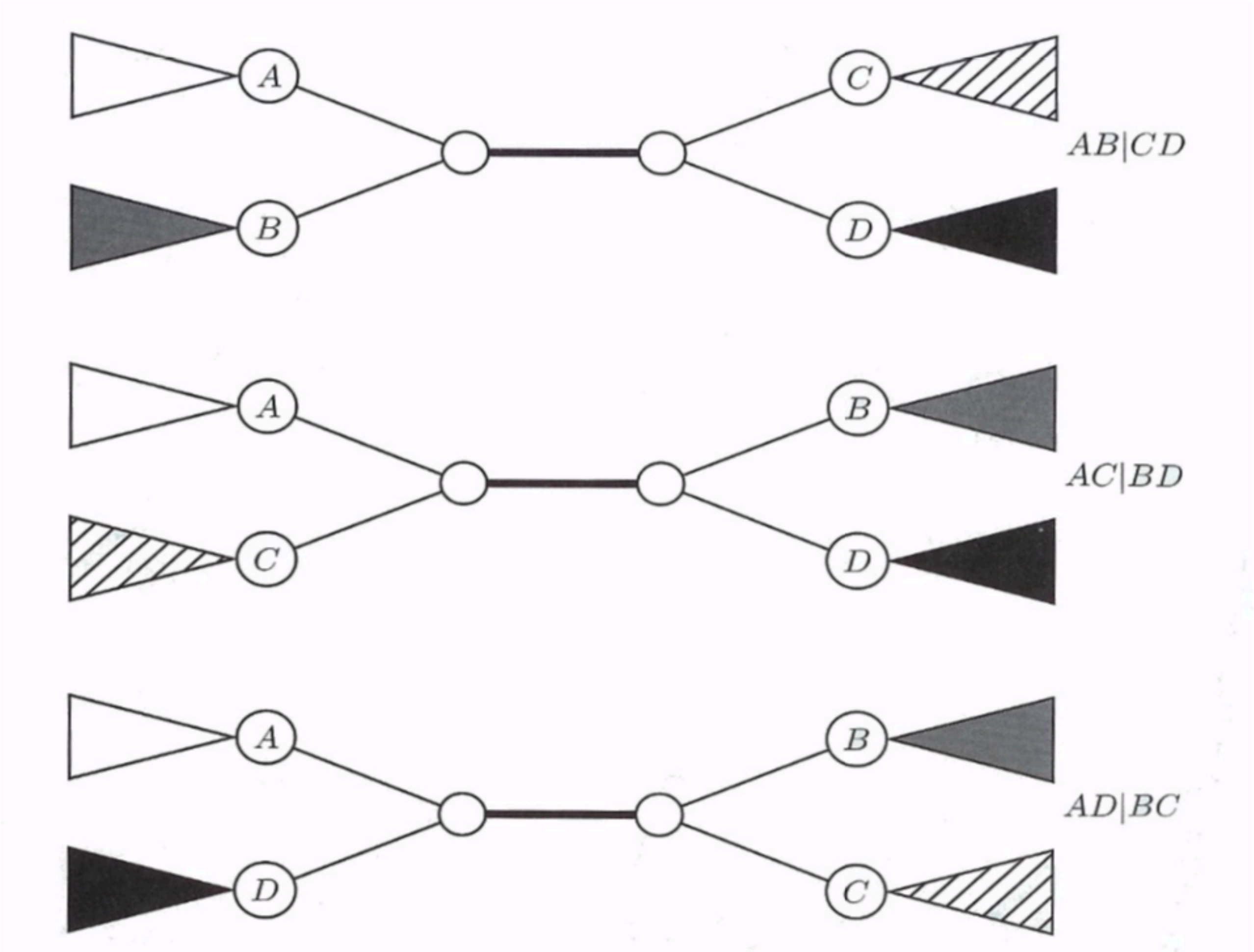
1. Start with an arbitrary tree *T.*
2. Check "neighbors" of *T*.
3. Move to a neighbor if it provides the best improvement in parsimony/likelihood score.

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

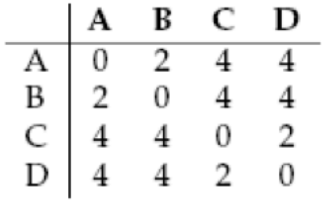
- Two-State Perfect Phylogeny
- Multi-State Perfect Phylogeny
- Large Maximum Parsimony Phylogeny Problem
- Summary

Reading:

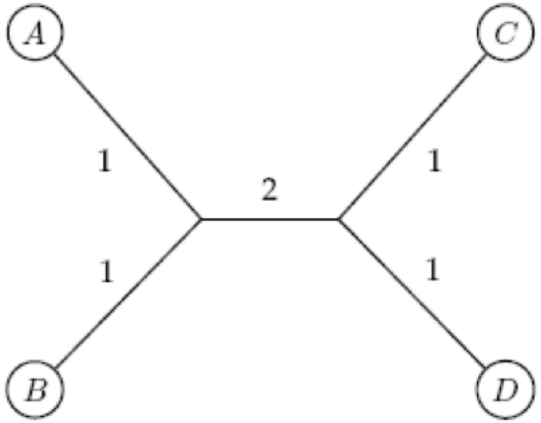
- Lecture notes

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