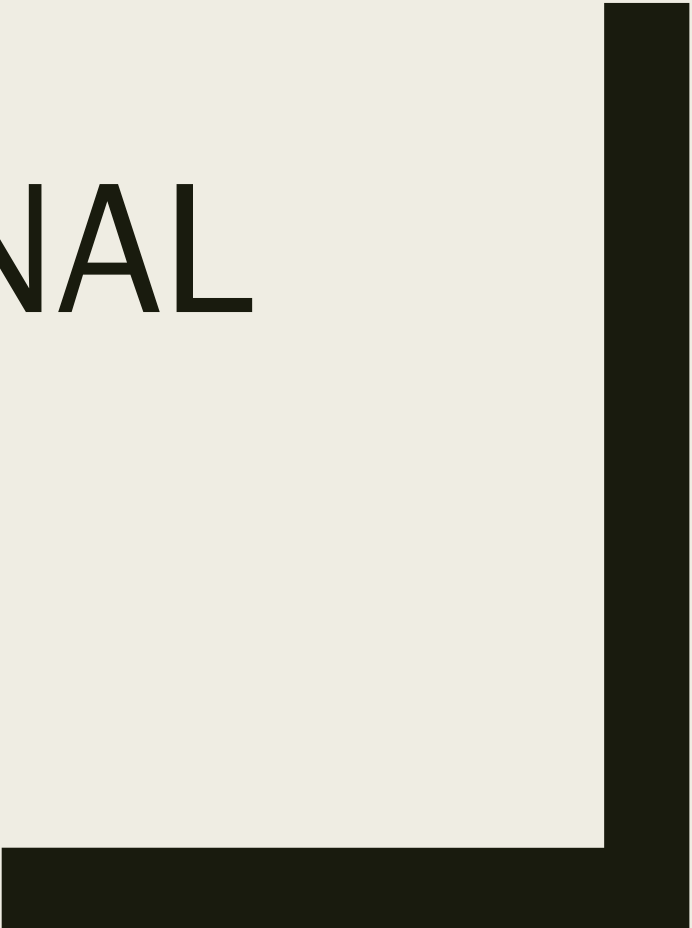


CS 364  
COMPUTATIONAL  
BIOLOGY

Sara Mathieson  
Haverford College



# Outline

- Midterm review



# Topics for Midterm 1

1) String search

2) BWT and Read Mapping

3) Genome Assembly

4) Pairwise Sequence Alignment

5) Multiple Sequence Alignment and Phylogenetics

## (2) BWT and Read Mapping

- Input: previously assembled reference sequence and millions-billions of reads from a new individual of the same species
- Output: the location(s) where each read maps (+ where the mismatches are)
- Pairwise sequence alignment is too slow
- What is the runtime of constructing the BWT and FM-Index? After that, what is the runtime of pattern matching? (see Lab 2)

# BWT and FM-Index runtime

- Building the FM-Index: dominated by sorting the rotations (cyclic permutations). There are actually linear time algorithms for this, but we will assume a standard sorting algorithm so  $O(n \log n)$  where  $n$  is the length of the reference.
- Creating  $M$ ,  $occ$ , and  $A$  are all linear in  $n$
- Read mapping after FM-Index has been created:
  - *Linear in the length of the pattern ( $m$ )*
  - *Linear in the number of patterns/reads ( $R$ )*
  - *Constant in the length of the genome ( $n$ )*

# (3) Genome Assembly

- Often the first step in studying the genetics of a new species
- Input: millions-billions of reads (used to be “long” reads, now are “short”)
- Output: contigs (ideally long and accurate, making up as much of the original genome as possible)
- Overlap graph assembly (Overlap Layout Consensus: OLC). Accurate but very slow
- De Bruijn graph (DBG) assembly. Fast but sometimes not as accurate
- What are the runtimes of these assembly algorithms in terms of  $n$ ,  $m$ ,  $R$ ?

# (4) Pairwise Sequence Alignment

- Used for studying the relationship between homologous sequences (often genes or regions from different species)
- Could be run after assembling two very different species
- Could be run on repetitive but diverged regions from the same individual
- We are giving up runtime by allowing gaps and mismatches
- Input: two sequences  $x$  and  $y$ , typically of similar length but not always. We also need a substitution matrix and gap penalty
- Output: optimal alignment(s) between  $x$  and  $y$ , AND an alignment score (higher is more similar, negative is usually not biologically meaningful)
- Two dynamic programming variations: global sequence alignment (align entire  $x$  with entire  $y$ ) and local alignment (align highly similar regions in  $x$  and  $y$ )

# How to study

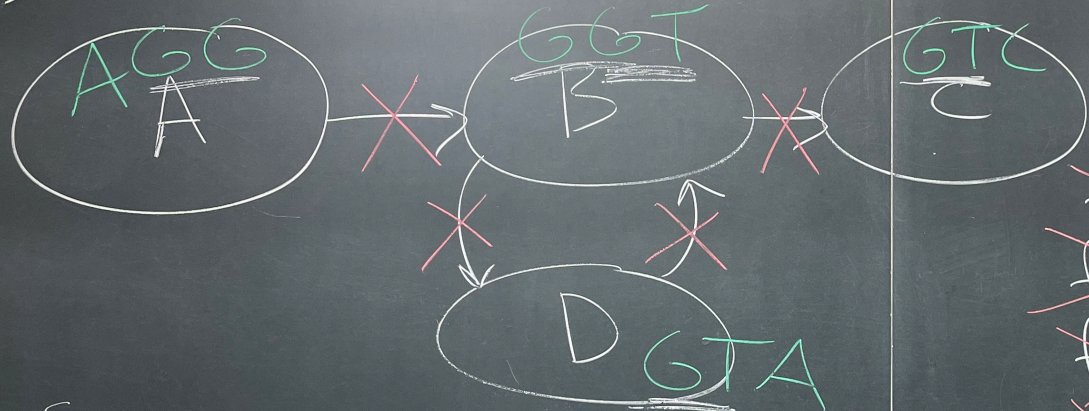
- Go over all slides and readings => create study sheet (handwritten)
- Redo all handouts and questions/problems during class
  - *Including runtime*
- Come to office hours and lab next week to ask questions! (and/or Piazza)

Requested Topics:  
DBG runtime, Fleury's, non-linear gaps



DBG

$(k-1)$ -mers  $\Rightarrow$  nodes



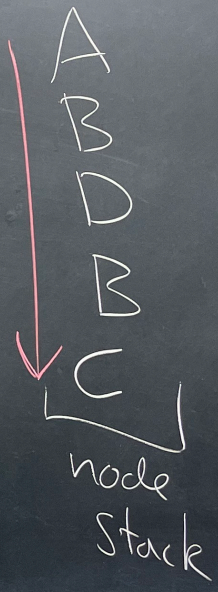
Semi-balanced: A & C  
 balanced: B & D  
 Eulerian? Yes

Fleury's

$k? = 4$

- ~~f(A)~~
- ~~f(D)~~
- ~~f(C)~~
- ~~f(B)~~
- ~~f(A)~~

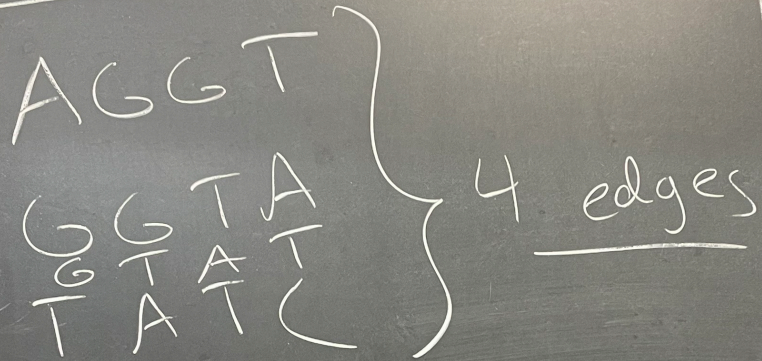
function call stack



- AGG
- GGT
- GTA
- GGT
- GTC



k-mers



$k \approx \frac{1}{3}$  read length  
odd  $m$

contig

AGGTATC

hopefully same as original genome.

$T = \text{min overlap}$   
for overlap graphs

$\approx \frac{1}{3} m$



read  
length  
odd  $m$

in overlap

or  
overlap  
graphs

$\frac{1}{3} m$

$R = \# \text{ reads}$   
 $m = \text{len of each read}$   
 $n = \text{len of orig genome}$



DBG runtime?

making  $k$ -mers?  $O(mR)$

# edges (weighted):  $O(n)$

# nodes:  $O(n)$

traversal  $\Rightarrow O(n)$

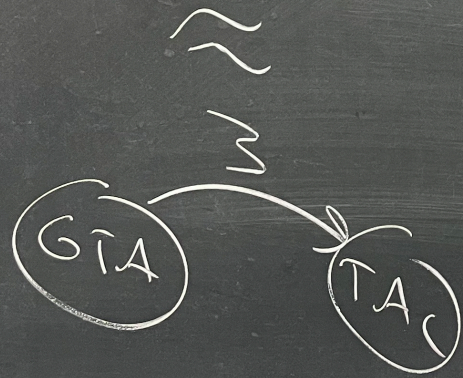
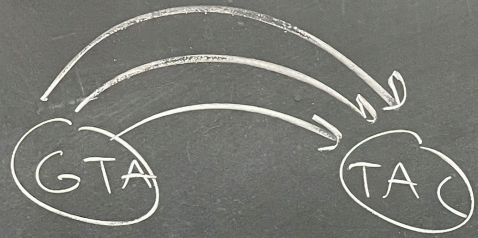


Overlap graph

nodes:  $O(R)$

find edges:  $O(R^2 m^2)$

way too slow!



Part 1 }  
4 } first 3  
Part 2 } pages



non-linear gaps

X = G C T A G C T  
Y = G C - - - C T

                  ↑  ↑  ↑  
                 -3 -1 -1

} -3 + (-1)(2)  
= -5

$$\delta(l) = g + e(l-1)$$

          ↑                  ↑                  ↑  
          gap          gap          gap  
          length      open      extend



# Practice Exam



(a)  
bad char

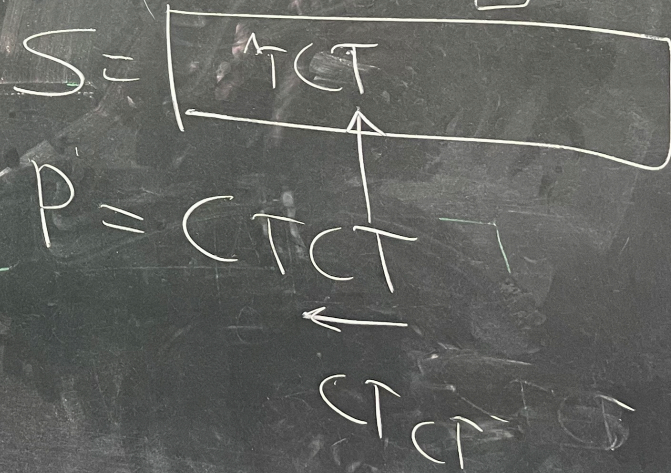
|   |   |   |   |   |
|---|---|---|---|---|
|   | C | T | C | T |
| A | 1 | 2 | 3 | 4 |
| C | 0 | 1 | 0 | 1 |
| T | 1 | 0 | 1 | 0 |

good suffix

|   |   |   |   |
|---|---|---|---|
| C | T | C | T |
| 2 | 2 | 4 | 1 |

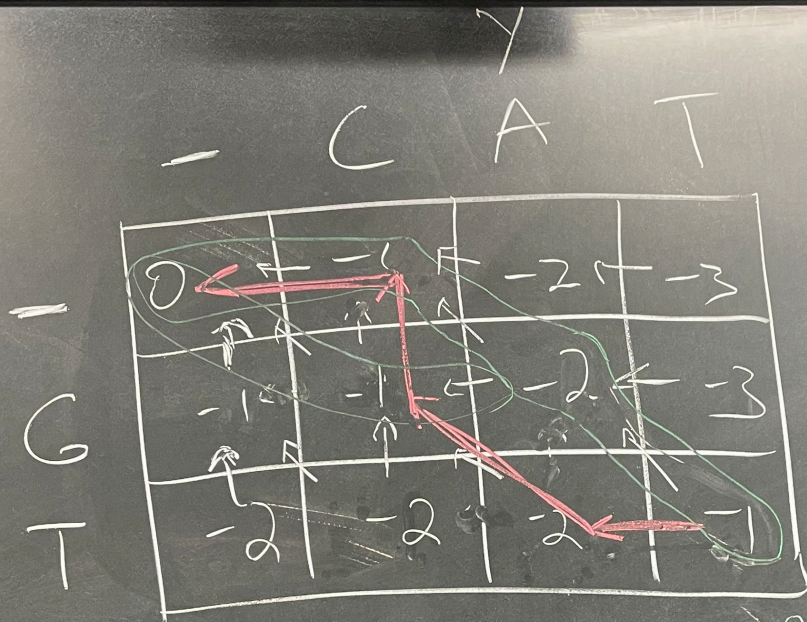
(b) # compare  
 1+4+1  
 ↑  
 match  
 return 5  
 (or 4) if it starts @ c

CTAA CTCTA  
 CTCT |||| (4,1)  
 CTCT  
 CTCT





①



X = - G T - } → -4  
 Y = C - A T

②

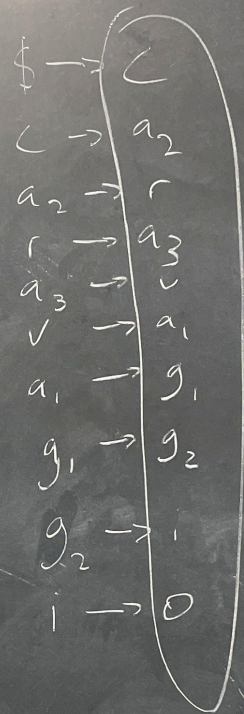
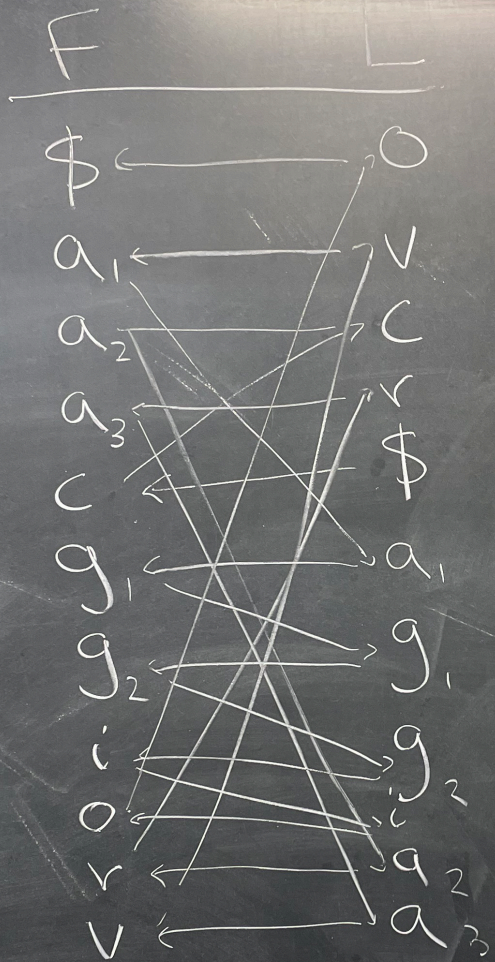
X = - G T      G - T  
 Y = C A T      C A T

high road  
 Score = -1

Score = -1



(e)



(f)

sum = 4150  
 midpoint = 2075

NSO = 700

{ 2000, 700, 700, ... }

2900      3400

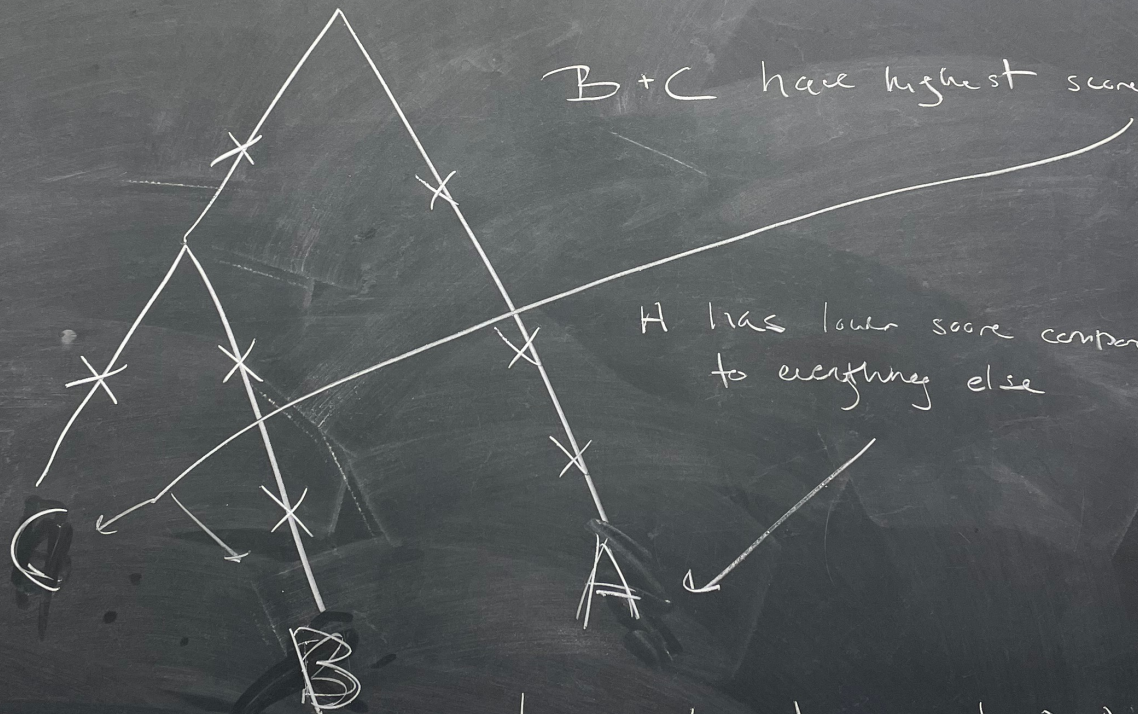


(9)

$$\text{Average} = \frac{m \cdot k}{n}$$

$$= \frac{100,000 \cdot 50}{10,000} = 500$$

(10)

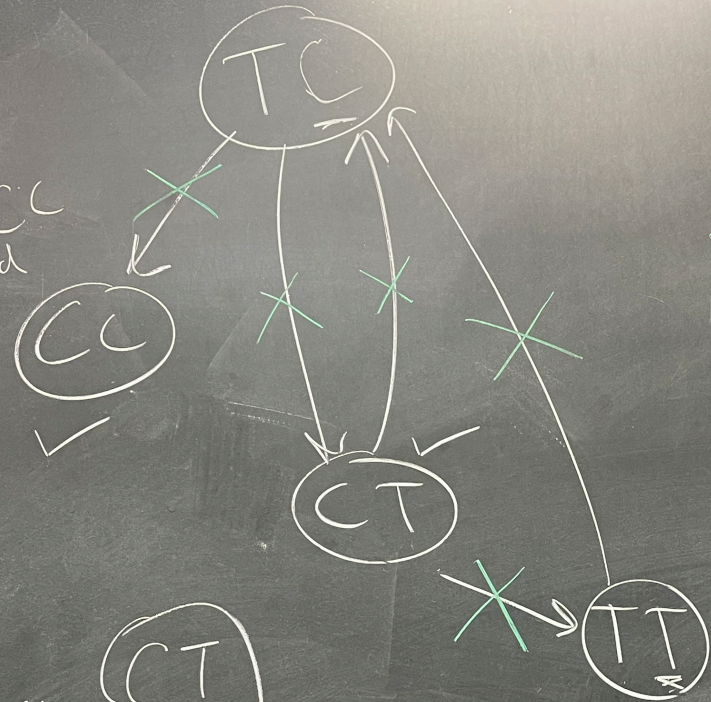


lowest alignment  $\Rightarrow A+B$



Part 2

(a) Yes,  
CT and CC  
are S.B<sup>d</sup>



(c) CTCTCC  
There is more  
than 1 traversal

(b)

|               |    |
|---------------|----|
| <del>CC</del> | CT |
| <del>TC</del> | TT |
| <del>CT</del> | TC |
| <del>TC</del> | CT |
| <del>TC</del> | TC |
| <del>TT</del> | TC |
| <del>CT</del> | CC |
| ft            | s  |

$k=3$   
repeat of  $k-1$

CTCTTC



Part 3

base case

$$\text{minCoins}(0) = 0$$

$$\text{minCoins}(x) = 1 \quad \text{if } x = 1, 3, 5$$

$k = \#$  denominations  
 $n =$

recursion

$$\text{minCoins}(N) = 1 + \min$$

$$\left\{ \begin{array}{l} \text{mc}(N-1) \\ \text{mc}(N-3) \\ \text{mc}(N-5) \end{array} \right.$$

$$\text{minChange} = \text{mc}$$

$$\text{minChange}(n) = 1 + \min(\text{mc}(n-1), \text{mc}(n-3), \text{mc}(n-5))$$

Base case

$$\text{mc}(1) = 1$$

$$\text{mc}(3) = 1$$

$$\text{mc}(5) = 1$$

$O(kn)$

|   |   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|---|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | 1 | 2 | 1 | 2 | 1 | 2 | 3 | 2 | 3 | 2  |