MIDTERM

COM SCI CM122/222
CHEM CM160B/260B
BIOINFO M222

01 May 2019
Instructor : Eleazar Eskin

Name (Print):
UID :

Under [redacted] □

Guidelines for midterm.

- Please write down your name, UID, and whether you are un undergraduate or a graduate student.
- Closed book and no calculator.
- Your answers should not require much more space than is provided. Be concise.
- There is an extra blank sheet of paper at the end of exam. Please use it if you need more space for your answer.

Question	Score
1	10/10
2	39/40
3	18 /20
4	19 /20
5	10/10
Total	96

1 Coverage (10 pts)

(a) Assume we want to sequence a 1 billion base pair (bp) genome with a sequencer that generates 50 bp reads. How many reads do we need to achieve 15x coverage. (5 pts)

Cover= $\# \text{ reads} \cdot \frac{\text{read len}}{\text{ref len}}$

1 billion = 10^9

$15 = X \cdot \frac{50}{1,000,000,000}$

$X = \frac{15 \cdot 1 \text{ billion}}{50} = \frac{15 \cdot 10^9}{50} = \boxed{3 \cdot 10^8 \text{ reads}}$

5

(b) If the sequencing error rate is ϵ, what's the probability that exactly k out of n total reads at a nucleotide are correct? It suffices to provide the equation. (5 pts)

$P(k \text{ of } n \text{ reads is correct}) = \boxed{\binom{n}{k}(1-\epsilon)^k(\epsilon)^{n-k}}$

5

$p\left(\begin{matrix}\text{1 read is}\\ \text{correct}\end{matrix}\right) = 1 - \epsilon$

$p(\text{1 read is wrong}) = \epsilon$

2 Alignment (40 pts)

Genomes (4 pts)

7500000000

(a) What is the approximate length of the human genome? If we use the minimum number of bits to represent nucleotide bases, approximately how much space in bytes would be required to store one person's entire genome. Show your work.

4

3 billion bases long

4 bases = 2 bits to represent a base

$3 \cdot 10^9 \text{ bases} \cdot \frac{2 \text{ bits}}{1 \text{ base}} \cdot \frac{1 \text{ byte}}{8 \text{ bits}} = \boxed{0.75 \cdot 10^9 \text{ bytes}}$

750,000,000 bytes

Trivial Alignment (12 pts)

(a) Write (pseudo-)code that aligns a single read **read** to a reference genome **ref** using the trivial alignment algorithm. You should return the index in **ref** where the alignment with least mismatches begins. If there are multiple alignments that result in the minimum number of mismatches, return a list containing all such indices. (8 pts)

8

```
loop through each base in ref:
    loop through read's bases:
        count # mismatches to ref genome when read aligned starting at current base.
    if # mismatches counted is less than current minimum # mismatches:
        .store the new cur # mismatches
        .store the location of new min mismatch (keep a list of all indexes that have this
            min # of mismatches for the read).
    else if # mismatches counted is equal to current min # mismatches:
        .add to list of indexes this location

after looping, return list of indexes in ref that read aligns to
    with best fit (all produce min # mismatches).
```

(b) Assume that the only operation that takes time in the trivial aligner is comparisons between the read and the reference. If the computer takes t seconds to make one comparison, approximately how much time (in seconds) will be required to align N reads of length k to a genome of length L? (4 pts)

4

$\frac{t \text{ seconds}}{1 \text{ comparison}}$ N reads $\frac{k \text{ bases}}{\text{read}}$ L bases

LkN comparisons

$\frac{t \text{ sec}}{\text{comparison}}$

$\boxed{\cancel{\frac{L \cdot k \cdot N}{t}}}$

$\boxed{L \cdot k \cdot N \cdot t}$

Alignment by Hashing (12 pts)

(a) List one advantage and one disadvantage of using a hash table to store our reference genome (3 pts) ✓

2.5

Advantage: time complexity is faster in referencing ref genome

Disadvantage: space complexity is much bigger & inefficient

-.5 clarify

most cases: $K < N$

(b) For an N base-pair-long reference genome, if we use a hash table to store the starting positions of each k-mer, on average, how many starting positions will each hash table entry contain? (express your answer in terms of N and k) (3 pts)

3

4 bases, k mer reads ⇒ 4^K possible k mers

N total bases $\longrightarrow \boxed{\frac{N}{4^K}}$ ✓

(c) Assume you are given the following hash table index, where '-' represents an empty entry. The entries represent perfect matches of the sequences in the reference genome. (6 pts)

Sequence	Positions
AAA	-
ACG	10
AGA	2019
AGG	7218
GAA	42, 609
GAT	25, 200, 128
GCA	16, 529
CAA	456
CAT	1919
CCC	1; 93
CGG	-
CTG	32
TTT	-

Answer the questions below assuming that the reads given can contain up to two mismatches relative to the reference genome indexed above.

i. List the most likely starting position in the reference genome where ACGTTTGCA can match.

3

Possible matches: 10, –, $\underbrace{16, 529}_{-6}$ $\longrightarrow$ 10, 10, 523 $\boxed{\text{Position 10}}$ ✓

ii. List ALL possible starting positions in the reference genome where AGAAGGCCC can match.

3

All possible matches:

		Shifted
AGA: 2019	$\xrightarrow{-0 \text{ shift}}$ ~~20~~	2019
AGG: 7218	$\xrightarrow{-3}$	7215
CCC: 1, 93	$\xrightarrow{-6}$	-5, 87

$\boxed{\text{possible: } 2019, 7215, 87}$

(-5 is impossible starting location)

Pileup (12 pts)

(a) What is the consensus sequence generated by these reads? Break ties in favor of the reference sequence. From the consensus sequence and reference sequence, how many single nucleotide polymorphisms (SNPs) can you find? Mark the positions of the SNPs you find on the consensus sequence. (4 pts)

```
 Ref:ACGAGTCCGTTGACCTACGT
Read:...AGTCCGATGACCTCCCT
Read:.......CGATGACCTCCGT
Read:....GTCCGATGACCTCCGT
Read:ACGAGTCCGCTGACCTC...
Read:TCGAGTCCGATG........
```

4

Consensus: ACGAGTCCGATGACCTCCGT

$\overset{\text{A}}{*}$ ... $\overset{\text{C}}{*}$

2 SNPs

* = SNP

✓

(b) Write (pseudo-)code that generates a consensus sequence based on a list of reads, **reads**, and a reference sequence, **ref**. You may assume that all reads are padded with the "." character to make them all the same length as **ref** (in the same fashion as above). Break ties in favor of the reference sequence. (8 pts)

7.5

- Loop through all ref's bases : (position i) ✓
  - Loop through each read: ✓
    - count # of nucleotides at each read's base at position i (same position as the current ref base). Ignore if it's a period. ✓
  - Look at counts of all 4 bases at position i:
    - if one of the 4 base types (A, C, T, G) is clear winner across all reads (counts is biggest & no ties), consensus string at position i is the majority base. ✓
    - Else if there's a tie, default to the ref genome's base at position i to break the tie for the consensus string ✓
- After building the consensus string outlined above, return it

clarity finding clear winner -.5

3 Dynamic Programming (20 pts)

↘ = match or transform

→ = insert into (A)

↓ = delete into (B) (Put in (B))

Edit Distance (10 pts)

Edit distance is the minimum number of operations needed to convert one string to another. Assume each operation (substitution, insertion, deletion) counts as one edit (i.e. has a cost of 1), find the edit distance between the sequence GCATCGT and the sequence GGATCGGCT. Identify all possible alignments that can result in the edit distance by highlighting the path in the grid and writing down the aligned sequences. You must show your work. *Hint: The grid is larger than you need it to be.*

	$	G	G	A	T	C	G	G	C	T	
$	0	←1	←2	←3	←4	←5	←6	←7	←8	←9	
G	1	0	1	←2	←3	←4	5	←6	←7	←8	
C	2	1	1	←2	←3	3	←4	←5	6	←7	
A	3	2	2	1	←2	←3	←4	←5	←6	←7	
T	4	3	3	2	1	←2	←3	←4	←5	6	
C	5	4	4	3	2	1	←2	←3	4	←5	
G	6	5	4	4	3	2	1	←2	←3	←4	
T	7	6	5	5	4	3	2	2	←3	3	

Edit distance = 3 ✓

-2 missing alignment

(1) $ G G A T C G G C T
$ G C A T C - G - T

$ G G A T C G G C T
$ G C A T C G - - T

(8)

→Global align but don't let go below 0.
base case = 0

Local Alignment (10 pts)

indel

Assume the following scoring scheme: gap=−2, mismatch=−1, match=1. Find the optimal local alignment between the sequence ACATGCGC and the sequence CATCATCGC. If you obtain multiple equal optimal alignments, you may choose any of them. Identify your alignment by highlighting the path in the grid and writing down the aligned sequences. You must show your work. You may assume that reaching a 0 ends the local alignment. *Hint: The grid is larger than you need it to be.*

	$	A	C	A	T	G	C	G	C		
$	0	0	0	0	0	0	0	0	0		
C	0	0	1	0	0	0	1	0	1		
A	0	1	0	2	0	0	0	0	0		
T	0	0	0	0	3	1	0	0	0		
C	0	0	1	0	~~0~~ 1	~~0~~ 2	2	0	1		
A	0	1	0	2	0	0	~~0~~	~~0~~	0		
T	0	0	0	0	3	1	0	0	0		
C	0	0	1	0	1	2	2	0	1		
G	0	0	0	0	0	2	1	3	1		
C	0	0	1	0	0	0	3	4	4		

local edit dist = 4 (score) ✓

$A-CATGCGC ✓
$CATCAT-CGC

10

4 Burrows-Wheeler Transform (20 pts)

(a) What is the main advantage of backward search algorithm over indexing (hash) algorithm for read alignment? (2 pts)

Space efficiency

+2

(b) What is the Burrows-Wheeler Transform of the string "TORNADO"? *Hint: Don't forget to include the dollar sign* (8 pts)

Shifts:

TORNADO$
X ORNADO$T
X RNADO$TO
X NADO$TOR
X ADO$TORN
X DO$TORNA
X O$TORNAD
X $TORNADO

sort →

Sorted!

$TORNADO
ADO$TORN
DO$TORNA
NADO$TOR
O$TORNAD
ORNADO$T
RNADO$TO
TORNADO$

ONARDTO$

+8

(c) Un-permute the string represented by the BWT: SLCSBA$AAA (10 pts)

First	Last
$	S1
A1	L1
A2	C1
A3	S2
A4	B1
B1	A1
C1	$
L1	A2
S1	A3
S2	A4

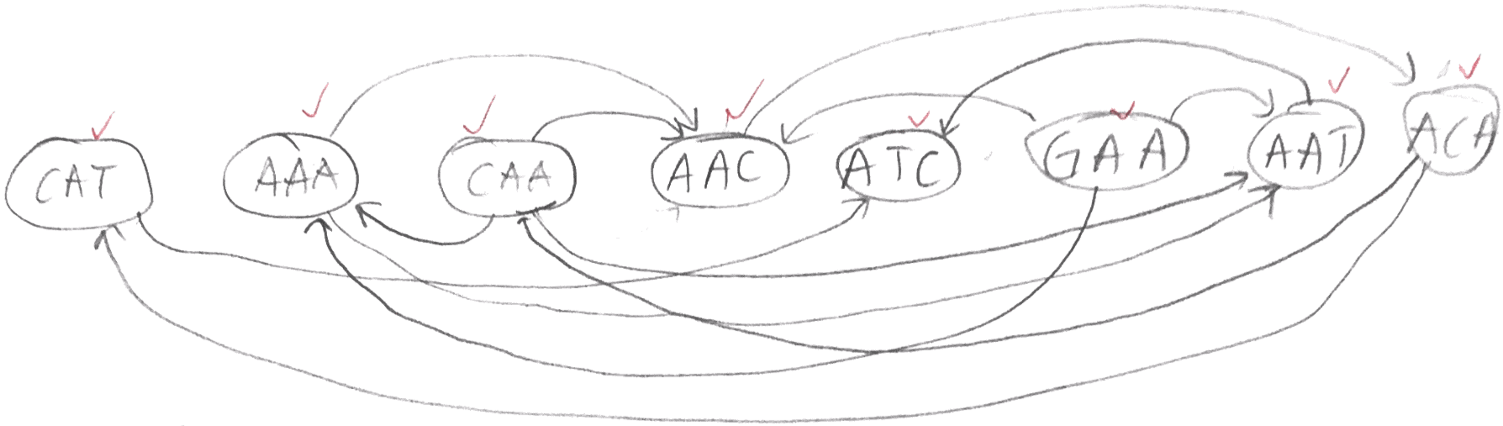
C A L A B A S A S $

$C_1\ A_2\ L_1\ A_1\ B_1\ A_4\ S_2\ A_3\ S_1\ \$$

+9

5 Assembly (10 pts)

(a) Create an overlap graph from the following k-mers and determine if there is a Hamiltonian path through the resulting graph. Assume we define overlap as an overlap of $k-1$ nucleotides of the suffix of one k-mer and the prefix of another k-mer. (4 pts)
{CAT, AAA, CAA, AAC, ATC, GAA, AAT, ACA}

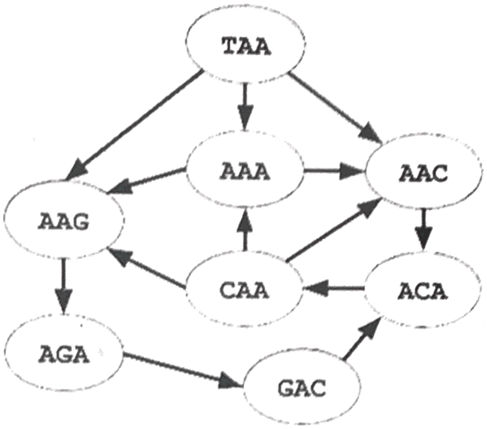


H-path is possible:

GAACAAATC

+4

(b) Given the following overlap graph, list all the sequences that can be assembled by a Hamiltonian path through the graph. (4 pts)



1. TAAGACAAAC ✓

2. TAAAGACAAC ✓

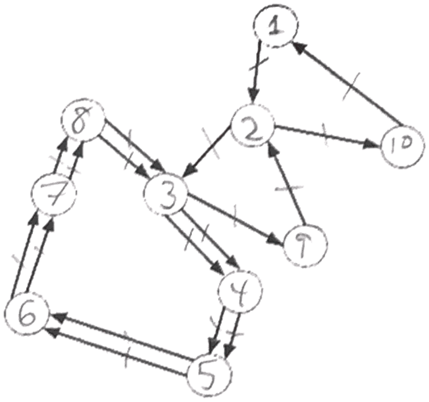
3. TAACAAAGAC ✓

4, TAAAGACAAC

✓

+4

(c) Given the following graph, is there a valid Eulerian cycle through the graph? (2 pts)



Yes there is, ~~path~~ cycle's path: +2

1, 2, 3, 4, 5, 6, 7, 8, 3, 4, 5, 6, 7, 8, 3, 9, 2, 10, 1