BMEG3105 - Data analytics for personalized genomics & precision medicine
Professor: Yu Li
Scribing L2 – Dynamic programming
Fri 13-09-2024

The foundation of modern biology and genomics: Sequence data

1 What are the sequence data?

- DNA sequence
  - Complementary double strand
  - Approximately 3 billion base pairs → A, T, C, G
- RNA sequence
  - Single strand
  - Base pairs → A, U, C, G
- Protein sequence
  - Composed of 20 amino acids
  - Sequences can be aligned via multiple sequence alignment

2 How to find the best pairwise alignment?

- Start from:
  - 2 sequences
  - Scoring matrix
- Intuitively enumeration as solution
  - Enumerate all possible alignments between both sequences
  - Calculate the score for all possible alignments
  - Select the alignment with the highest score, here is the similarity between the sequences the best
- However:
  - There are too many possible alignments to calculate them all
  - Number of possible alignments: $\binom{2n}{n} = \frac{(2n)!}{(n!)^2}$

3 Dynamic programming

- *Compare with cheapest flight problem*
  - *You want to fly from KAUST to CUHK and need to transfer in GZ, Dubai or Qatar*
  - *In order to find the cheapest flight you need to check every sub flight and take the sum*
  - *Cheapest ticket = optimal alignment score*
    - *Total ticket price is sum of price for each flight segment*
    - *Finite choice for each base*
      - *Align to another base → match or mismatch*
      - *Align to a gap*

BMEG3105 - Data analytics for personalized genomics & precision medicine
Professor: Yu Li
Scribing L2 – Dynamic programming
Fri 13-09-2024

3.1 Dynamic programming

- Break the problem into smaller sub-problems
- Solve the sub-problems optimally & recursively
- Use optimal solutions to construct the optimal solution for the original problem

3.2 Mechanism for how to solve the sequence alignment

- Start from
    - Scoring matrix

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

    - Input sequences
    - Gap penalty
- Questions you need to solve
    - What is the optimal alignment score? → F(sequence 1, sequence 2)
    - What is the optimal alignment?
- **Method 1:**
  →principle: F(XXX, XXX) will be reduced to F(X,X) in the scoring matrix or F(X, _) which is the boundary case
    - Write down the possible alignment options for the last pair of the alignment
        - E.g. for ACCG and ACG: $\begin{matrix} G \\ \_ \end{matrix}$ $\begin{matrix} \_ \\ G \end{matrix}$ $\begin{matrix} G \\ G \end{matrix}$
    - Determine which alignment is the best by making use of the gap penalty and scoring matrix
        - E.g. for ACCG and ACG:
            - F(ACC, ACG) + F(G, _) → -10 for the gap
            - F(ACCG, AC) + F(_, G) → -10 for the gap
            - F(ACC, AC) + S(G, G) → +2 for match between G & G
    - Take the best alignment and repeat the process of aligning the last pair of the alignment
        - E.g. ACC and AC:
            - F(AC, AC) + F(C, _) → -10 for the gap
            - F(ACC, A) + F(_, C) → -10 for the gap
            - F(AC, A) + S(C, C) → +2 for the match
    - Determine again which alignment is the best and repeat the process for the last time
        - E.g. for AC and A:
            - F(AC, _) + F(_, A) → -20 because 2 bases paired with gap -10 = -30
            - F(A, A) + F(C, _) → +2 for match – 10 for gap = -8
            - F(A, _) + S(C, A) → -10 for gap -7 for mismatch = -17

- **Method 2:**
  →table representation
  →principle: fill in the table (including arrows!!) to find the value in the last cell which represents the best alignment score, tracing back the path to get to this point will give the optimal alignment
  - Make a n+1, m+1 table (with n and m being the lengths of the sequences) containing the first sequence on the X-axis and the second sequence on the Y-axis

	Gap	A	C	C	G
Gap					
A					
C					
G					

  - First fill in the gap penalties
    - The example gap penalty is -10
    - For each box distract 10 from the number in the box left/above the box you want to fill in

	Gap	A	C	C	G
Gap	0	→ -10	→ -20	→ -30	→ -40
A	↓ -10				
C	↓ -20				
G	↓ -30				

  - Then fill in the rest of the table starting in the upper left box

	Gap	A	C	C	G
Gap	0	-10	-20	-30	-40
A	-10				
C	-20				
G	-30				

- There are 3 options to fill the box
  - MATCH between row 2 and column 2 coming from row 1, column 1
    - +2 (according to the example scoring matrix)
    - Diagonal arrow

	Gap	A	C	C	G
Gap	0	-10	-20	-30	-40
A	-10	↘ +2			
C	-20				
G	-30				

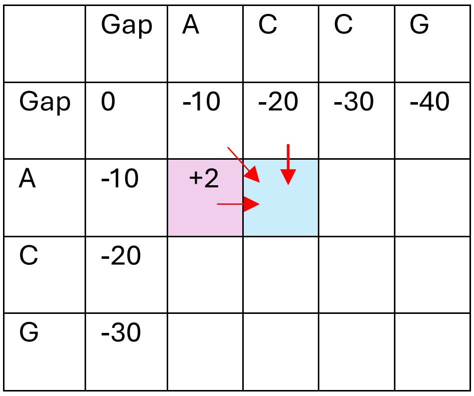
  - GAP between row 2 and gap coming from row 2, column 1
    - -10 (according to the example gap penalty)
    - Horizontal arrow

	Gap	A	C	C	G
Gap	0	-10	-20	-30	-40
A	-10	→ -20			
C	-20				
G	-30				

  - GAP between gap and column 2 coming from row 1, column 2
    - -10 (according to the example gap penalty)
    - Vertical arrow

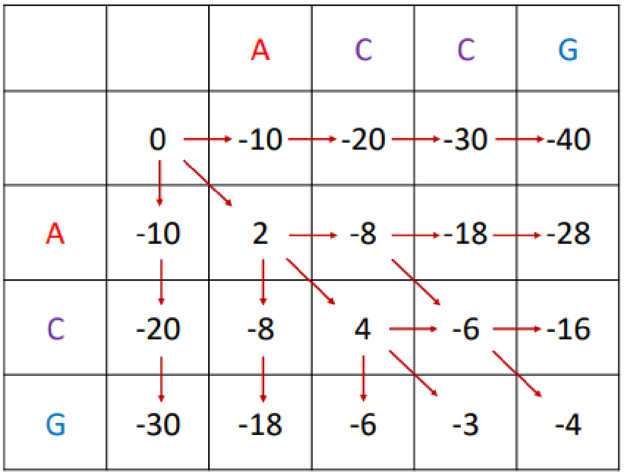
	Gap	A	C	C	G
Gap	0	-10	-20	-30	-40
A	-10	-20 ↓			
C	-20				
G	-30				

- We opt for the best solution being the match between A and A
- Then you move to the box on the right hand side (blue) of the pink box and again you calculate the 3 options



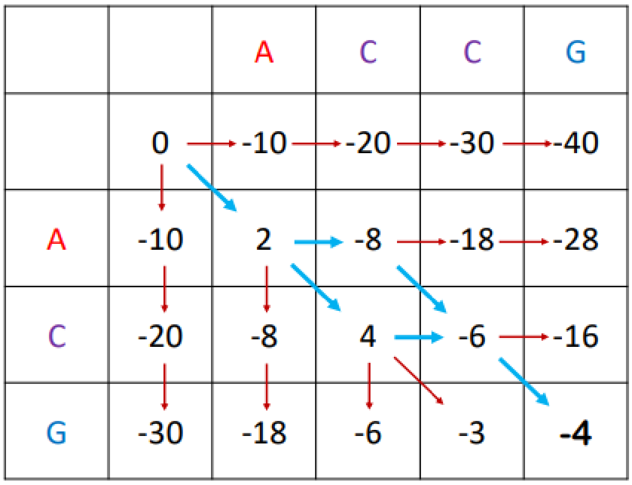
	Gap	A	C	C	G
Gap	0	-10	-20	-30	-40
A	-10	+2			
C	-20				
G	-30				

- Repeat this process until the table is filled in completely



		A	C	C	G
	0	-10	-20	-30	-40
A	-10	2	-8	-18	-28
C	-20	-8	4	-6	-16
G	-30	-18	-6	-3	-4

- Preserve the path/alignment information!!
  - Trace back by following the arrows

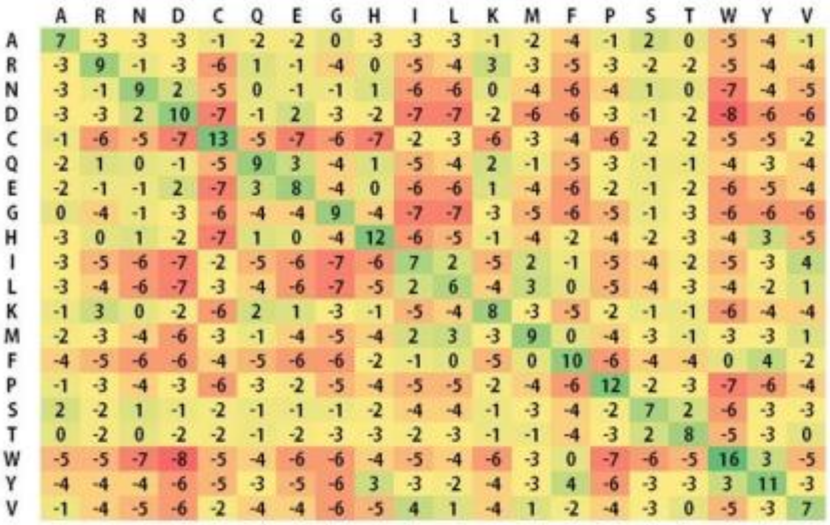


		A	C	C	G
	0	-10	-20	-30	-40
A	-10	2	-8	-18	-28
C	-20	-8	4	-6	-16
G	-30	-18	-6	-3	**-4**

  - In this example there are 2 optimal alignments
    - Option 1: ACCG
A_CG
    - Option 2: ACCG
AC_G

4 Score matrix

- Numbers differ depending on the database that is used or on the needs of the alignment
- E.g. BLOSUM: BLOck SUbstitution Matrix → used for alignment of proteins



	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	7	-3	-3	-3	-1	-2	-2	0	-3	-3	-3	-1	-2	-4	-1	2	0	-5	-4	-1
R	-3	9	-1	-3	-6	1	-1	-4	0	-5	-4	3	-3	-5	-3	-2	-2	-5	-4	-4
N	-3	-1	9	2	-5	0	-1	-1	1	-6	-6	0	-4	-6	-4	1	0	-7	-4	-5
D	-3	-3	2	10	-7	-1	2	-3	-2	-7	-7	-2	-6	-6	-3	-1	-2	-8	-6	-6
C	-1	-6	-5	-7	13	-5	-7	-6	-7	-2	-3	-6	-3	-4	-6	-2	-2	-5	-5	-2
Q	-2	1	0	-1	-5	9	3	-4	1	-5	-4	2	-1	-5	-3	-1	-1	-4	-3	-4
E	-2	-1	-1	2	-7	3	8	-4	0	-6	-6	1	-4	-6	-2	-1	-2	-6	-5	-4
G	0	-4	-1	-3	-6	-4	-4	9	-4	-7	-7	-3	-5	-6	-5	-1	-3	-6	-6	-6
H	-3	0	1	-2	-7	1	0	-4	12	-6	-5	-1	-4	-2	-4	-2	-3	-4	3	-5
I	-3	-5	-6	-7	-2	-5	-6	-7	-6	7	2	-5	2	-1	-5	-4	-2	-5	-3	4
L	-3	-4	-6	-7	-3	-4	-6	-7	-5	2	6	-4	3	0	-5	-4	-3	-4	-2	1
K	-1	3	0	-2	-6	2	1	-3	-1	-5	-4	8	-3	-5	-2	-1	-1	-6	-4	-4
M	-2	-3	-4	-6	-3	-1	-4	-5	-4	2	3	-3	9	0	-4	-3	-1	-3	-3	1
F	-4	-5	-6	-6	-4	-5	-6	-6	-2	-1	0	-5	0	10	-6	-4	-4	0	4	-2
P	-1	-3	-4	-3	-6	-3	-2	-5	-4	-5	-5	-2	-4	-6	12	-2	-3	-7	-6	-4
S	2	-2	1	-1	-2	-1	-1	-1	-2	-4	-4	-1	-3	-4	-2	7	2	-6	-3	-3
T	0	-2	0	-2	-2	-1	-2	-3	-3	-2	-3	-1	-1	-4	-3	2	8	-5	-3	0
W	-5	-5	-7	-8	-5	-4	-6	-6	-4	-5	-4	-6	-3	0	-7	-6	-5	16	3	-5
Y	-4	-4	-4	-6	-5	-3	-5	-6	3	-3	-2	-4	-3	4	-6	-3	-3	3	11	-3
V	-1	-4	-5	-6	-2	-4	-4	-6	-5	4	1	-4	1	-2	-4	-3	0	-5	-3	7

5 Online solution of sequence alignment

- https://www.ebi.ac.uk/Tools/psa/emboss_needle/
  - The webserver Emboss needle/water can perform global and local alignments for you
  - You just need to fill in the protein sequences you want to align and then the tool will calculate the rest for you
- https://biopython.org/
  - Python also has a tool to determine alignments → Bio import pairwise2
  - How to align via python
    - Alignments = pairwise2.align.globalxx("ACCGT", "ACG")
    - From Bio.pairwise2 import format_alignment
    - Print(format_alignment(*alignments[0]))