

BMEG3105

Fall 2023

Data Analytics for Personalized Genomics and Precision Medicine

Lecture 4: Dynamic Programming

Lecturer: Yu LI (李煜) from CSE

Liyu95.com, liyu@cse.cuhk.edu.hk

15 September 2023

Lecture outline:

- Recap from last lecture
- Dynamic Programming

Part 1. Recap from last lecture

Sequence Data:

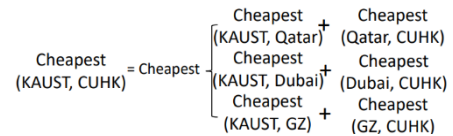
- DNA sequence
 - Composed of A, T, C, G bases
 - Consists of complementary double strands
- RNA sequence:
 - Composed of A, U, C, G bases
- Protein sequence:
 - Composed of 20 amino acids
 - Multiple sequence alignment
- To find the best pairwise alignment
 - With two sequences and scoring matrix known
 - Enumeration as a solution
 - Enumerate all the possible alignments between two sequences
 - Calculate scores for all alignments
 - Choose the alignment with the highest score
 - Problem with this solution is that there are too many possible alignments

Part 2. Dynamic Programming (DP):

- Break down a problem into smaller sub-problems
- Solve these sub-problems optimally and recursively
- Use these optimal solutions to construct the optimal solution for the original problem

Flight problem example:

- Direct flight not always cheap
- Finite pre-destination
- Break the flight into several flight trips
- Compare the sum of flight trips for each option and choose the cheapest option



Similarly with Optimal alignment score:

- Each base either aligns to a gap or another base
- Alignment score equals to the sum of score for each alignment pair

Sequence alignment with DP:

- Consider the possibilities of the last pair of the alignment
- Calculate alignment score and choose the best option
- Break down the best option and repeat the last step

ACCG and ACG alignment example:

1. Consider the last base pair. (_ represent blank)

$$F(\text{ACCG}, \text{ACG}) = \text{Best} \left\{ \begin{array}{l} F(\text{ACC}, \text{ACG}) + F(\text{G}, _) \\ F(\text{ACCG}, \text{AC}) + F(_, \text{G}) \\ F(\text{ACC}, \text{AC}) + S(\text{G}, \text{G}) \end{array} \right.$$

$$F(\text{ACCG}, \text{ACG}) = \text{Best} \left\{ \begin{array}{l} F(\text{ACC}, \text{ACG}) - 10 \\ F(\text{ACCG}, \text{AC}) - 10 \\ F(\text{ACC}, \text{AC}) + 2 \end{array} \right.$$

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

Gap penalty = -10

➔ Choose the option with highest last base pair score, i.e. $F(\text{ACC}, \text{AC}) + 2$

2. Break down unknown part the option chosen

$$F(\text{ACC}, \text{AC}) = \text{Best} \left\{ \begin{array}{l} F(\text{AC}, \text{AC}) + F(\text{C}, _) \\ F(\text{ACC}, \text{A}) + F(_, \text{C}) \\ F(\text{AC}, \text{A}) + S(\text{C}, \text{C}) \end{array} \right. = \text{Best} \left\{ \begin{array}{l} F(\text{AC}, \text{AC}) - 10 \\ F(\text{ACC}, \text{A}) - 10 \\ F(\text{AC}, \text{A}) + 2 \end{array} \right.$$

The Formula is reduced to boundary case

$$\begin{aligned}
 F(\text{AC}, \text{A}) = \text{Best} & \begin{cases} F(\text{AC}, _) + F(_, \text{A}) \\ F(\text{A}, _) + S(\text{C}, \text{A}) \\ \mathbf{F(\text{C}, _) + S(\text{A}, \text{A})} \end{cases} \\
 = \text{Best} & \begin{cases} -20 - 10 = -30 \\ -10 - 7 = -17 \\ 2 - 10 = -8 \end{cases}
 \end{aligned}$$

The optimal alignment is **ACCG** **AC_G** or **ACCG** **A_CG**

Table representation of the method:

	Gap	A	C	C	G
Gap	0	-10	-20	-30	-40
A	-10	2 $F(\text{A}, \text{A})$	-8	-18	-28
C	-20	-8	4 $F(\text{A}, \text{A}) + F(\text{C}, \text{C})$	-6 $F(\text{AC}, \text{AC}) + F(\text{C}, _) /$ $F(\text{AC}, \text{A}) + F(\text{C}, \text{C})$	-16
G	-30	-18	-6	-3	-4 $F(\text{ACC}, \text{AC}) + F(\text{G}, \text{G})$

Optimal alignment 1:

ACCG

A_CG

Optimal alignment 2:

ACCG

AC_G

- The arrows preserve the path information
- Arrows in blue highlight the direction that gives the best alignment score
- Trace back the arrows to get the optimal alignment