BMEG 3105
Data Analytics for Personalized Genomics and Precision Medicine

Lecture 3
The Foundation of Modern Biology and Genomics: Sequence Data

Lecturer: Professor Yu LI *Scribe: JIA Zihan*

Course Outline of Lecture 3:
- Sequence Data
- Sequence Comparison and Alignment Score
- Dynamic Programming

I. Sequence Data

1.1 Why Considering Sequence Data?

(1) Central Dogma: The central dogma of molecular biology describes the flow of genetic information from DNA to RNA to protein.

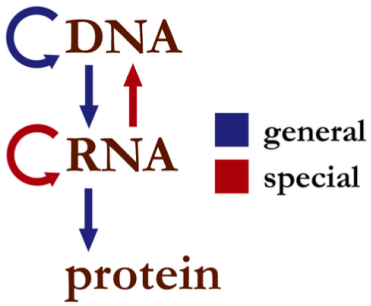

Figure 1 Central Dogma

(2) The genetic information is hidden in **DNA sequences.**

(3) Phenotype (How we look like) = Genotype (Determined by sequences) + Environment. In Prof Li's example, two twins with the same genotype may have different traits (personalities, interests etc.) due to their living environment.

1.2 What are the Sequence Data? (DNA, RNA, Protein)

(1) DNA Sequence
- Composed of base A, T, C, G
- Complementary Double Strand
- Approximately **3 billion** of these base pairs

(2) RNA Sequence
- Composed of A, U, C, G

(3) Protein Sequence

- Usually composed of 20 amino acids
- Multiple Sequent Alignment

1.3 How to Get the Sequences

(1) DNA/RNA sequencing is still **under active development**, and the entire development timeline is from short reads to long reads, as illustrated below:

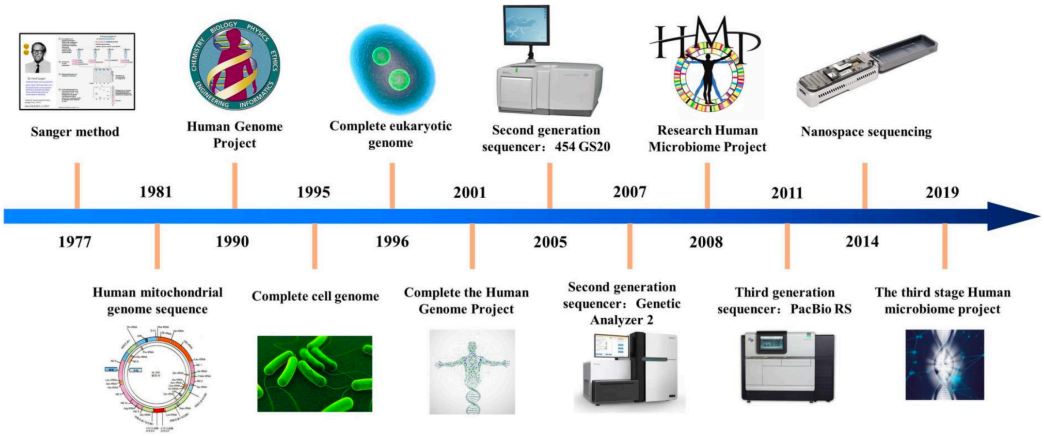

Figure 2 Development of DNA/RNA Sequencing Method

We mainly describe **nanopore sequencing:**

- DNA goes through a chemical pore, and the different bases will cause difference in electrical current change.
- Scientists get the sequences by detecting these current changes, as the picture show below.

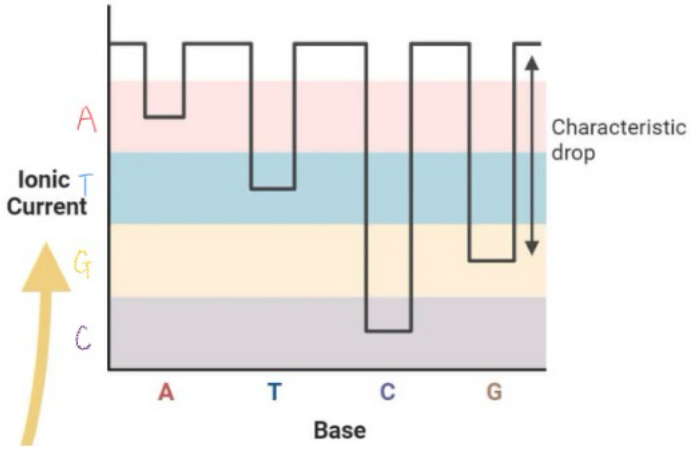

Figure 3 Current Indicating Different Bases

- By using traditional ways like the generation sequencer, it usually requires the process to break large pieces of DNA into smaller ones, but the nanopore sequencing method doesn't. It directly uses very long (up to 400Mb VS 1000bp) DNA samples.

- Relatively high error rate (5% VS 0.001%).

(2) Protein Sequencing: Mostly based on **mass spectrometry (MS).**

- Break the long sequence into short pieces, and each piece can be determined by MS. (Mainly determined by the **weight** of the piece.)
- Assemble the short pieces into the raw sequences.
- Digestion→HPLC→Mass spectrometry→Computational sequence assembly

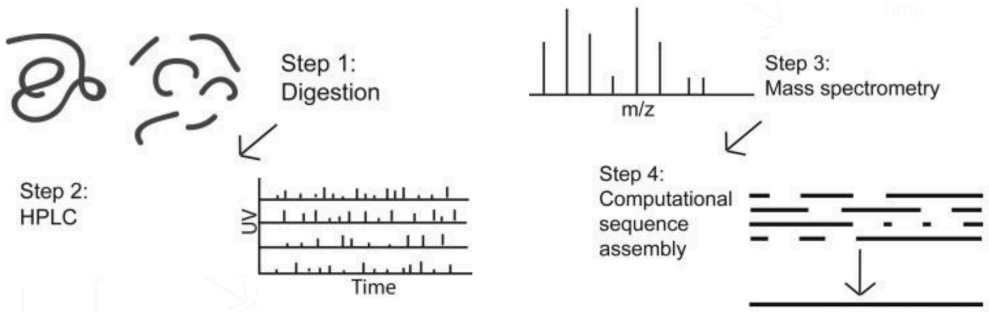

Figure 4 Protein Sequencing Procedures

1.4 What are the Raw Data and What do We Do to Them?
*** Sequence Comparison is the Key!**

(1) DNA Sequences
- Quality Control
- Map Reads to Reference Genome
- Variant Calling
- Phenotype Associated Variants
- Related to Bioinformatics

(2) Protein Sequences
- Multiple Sequence Alignment
- Similar sequence indicates **similar structure**, further indicates **similar function**.
- Similar sequence can also suggest common **ancestors.** (Homology)

II. Sequence Comparison and Alignment Score

2.1 The Importance of Sequence Alignment

(1) Similar sequence → Similar structure → Similar Function: Can be used for **biomolecular function and property prediction.**
(2) Similar sequence → Common ancestor: **Evolution, identifying conservative region, investigating mechanism.**
(3) …

2.2. Method: Pairwise Sequence Alignment

(1) To arrange two sequences to maximize the similarity between them, and for each sequence we can insert gaps in these sequences. Rules:

- Match: A<->A
- Mismatch (Substitution): G<->T
- Gap (Insertion or Deletion): C<->_

(2) 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

Figure 5 Scoring Matrix

If two bases match with each other (i.e., A<->A, T<->T, C<->C, G<->G), the score is +2; If two bases mismatch with each other, in case i (i.e., A<->T, A<->C, C<->G, G<->T) the score is -7 and in case ii (i.e., A<->G, C<->T) the score is -5; If there is a gap (i.e., A<->_, T<->_, C<->_, G<->_) the score is -10. The total score is the summation of that of each base. Here shows two examples for comprehension:

A G G C C G
A T G C _ G

Alignment score 1 = 2 + (-7) + 2 + 2 + (-10) + 2 = -9

A G G C C G
A T G C G _

Alignment score2 = 2 + (-7) + 2 + 2 + (-7) + (-10) = -18

Figure 6 Examples for Alignment score

2.3 How to Find the Best Pairwise Alignment?

(1) Straightforward Solution: Enumeration

Enumerate all the possible alignments between two sequences and calculate the scores for all alignments. Select one with the highest score so that that kind of alignment

indicates the highest similarity between two sequences.

(2) Problem

- Too many possible alignments
- Number of Possible solutions: $\binom{2n}{n} = \frac{(2n)!}{(n!)^2}$
- If n = 300, this number is 7×10^{88}, too large
- But if we use dynamic programming, we can reduce it to $300 \times 300 = 90000$.

III. Dynamic Programming

3.1 A Simple Example: Cheapest Ticket from KAUST to CUHK

There is no direct flight and need to transfer. To ensure the final cost is the lowest, we have to make sure that the single journey to the transfer point must be the cheapest.

$$\text{Cheapest}(\text{KAUST, CUHK}) = \text{Cheapest} \begin{cases} \text{Cheapest}(\text{KAUST, Qatar}) + \text{Cheapest}(\text{Qatar, CUHK}) \\ \text{Cheapest}(\text{KAUST, Dubai}) + \text{Cheapest}(\text{Dubai, CUHK}) \\ \text{Cheapest}(\text{KAUST, GZ}) + \text{Cheapest}(\text{GZ, CUHK}) \end{cases}$$

Core Idea:
(1) Break the problem into **smaller sub-problems;**
(2) Solve these sub-problems **optimally and recursively;**
(3) Use these optimal solutions to construct the optimal solution for the original problem.

3.2 An Example for Dynamic Programming

Find the Optimal Alignment Score for F (ACCG, ACG)

ACCG
ACG

(1) General Idea: we aim to break the large problem into many sub-problems. Here are 7 bases in total, so we firstly align the last position of these sequences:

$$F(\text{ACCG, ACG}) = \text{Best} \begin{cases} F(\text{ACC, ACG}) + F(\text{G, \_}) \\ F(\text{ACCG, AC}) + F(\text{\_, G}) \\ F(\text{ACC, AC}) + S(\text{G, G}) \end{cases}$$

-> maximum 6 bases

In the first case we add a "_" at the end of the second sequence, in the second case we

add a "_" at the end of the first sequence, in the last case we align two "G" together. Under such circumstances we can reduce the problem size by 1 or 2 bases each time.

For each of these cases, we can further divide them gradually to reduce the number of the bases until all of them fulfill boundary cases, i.e. F(X, X). Here illustrate an example of the last case, F(ACC, AC) + F(G,G):

$$F(ACC, AC) = \text{Best} \begin{cases} F(AC, AC) + F(C, \_) \\ F(ACC, A) + F(\_, C) \\ F(AC, A) + S(C, C) \end{cases}$$

(2) * Table Representation

Use the horizontal and vertical axes of the table to represent two base sequences respectively, so that it will be easier to calculate the total score and compare.

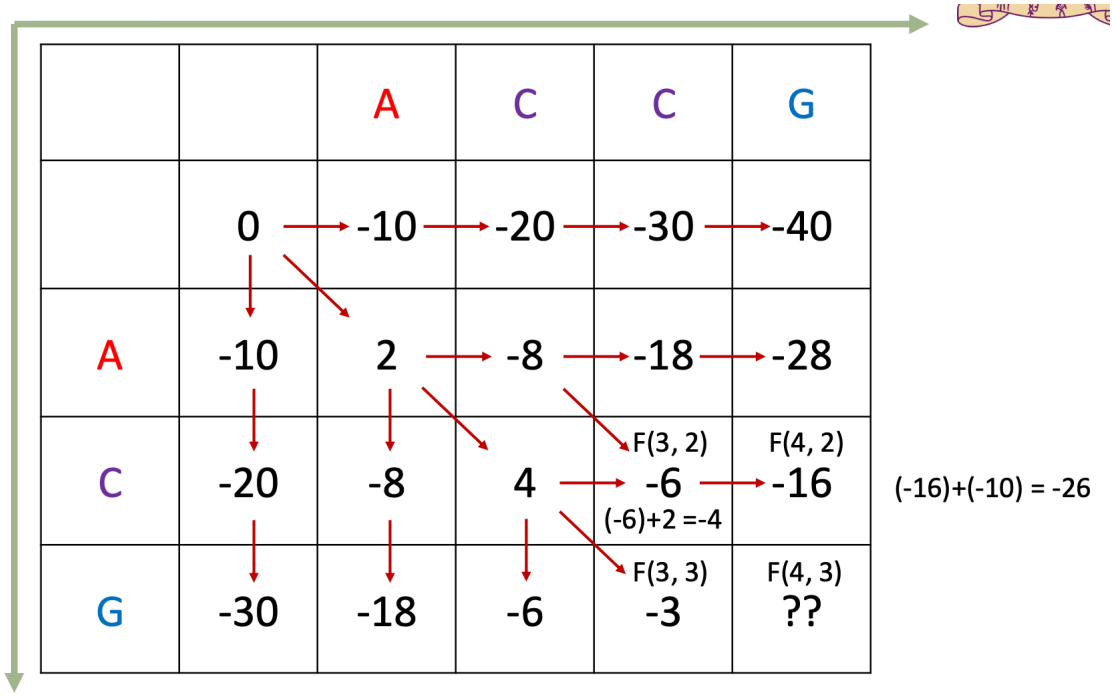

Figure 7 Calculation Using Tables

How to fill the table:

- F (4,3) is what we want to get.
- If the arrow come from upper rows or left columns, it means that the current base is paired up with a "_", so the total score should minus 10 based on the formal result.
- If the arrow come from diagonal, it means the horizontal and the vertical coordinate pair up with each other. According to scoring matrix (Figure 5), calculate the score of the paired bases.
- The value of one specific cell is not determined by one direction's calculation. After comparing 3 direction's result (the upper, left, and diagonal ones) and fill

the cell with the maximum value of those three.

- The scores in the score matrix is crucial for the choices of the paths.

So the ?? value in the cell should be -4 (-26<-13<-4), which is also the final result. This method can not only be used to calculate the alignment score of different alignment methods, but also we can justify the best path from the beginning to the end by tracing back the arrows in the table:

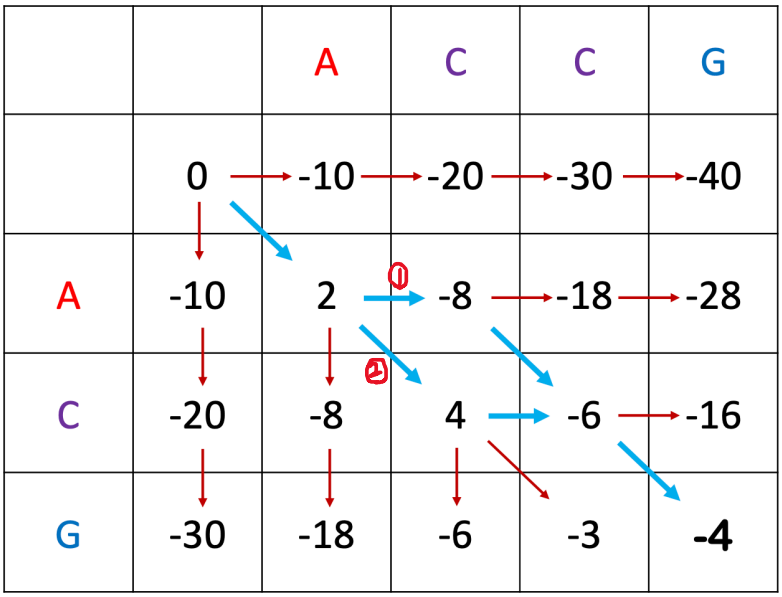

Figure 8 Tracing the Best Alignment Arrows

And the optimal alignment is:

Optimal alignment 1
ACCG
A_CG

Optimal alignment 2
ACCG
AC_G

IV. Useful Websites and resources

- Webserver for sequence alignment: https://www.ebi.ac.uk/Tools/psa/emboss_needle/
- Biopython: https://biopython.org
- Bioinformatics: Sequence and Genome Analysis---Chapter 2&3
- Time complexity and space complexity analysis
- Local alignment
- Multiple sequence alignment
- Affine gap penalty
- Sequence database search: BLAST