BMEG 3105 Lecture 4

Assembly & Mapping

September 12, 2025 (Friday)

DP (Dynamic Programming):

Dynamic Programming divides the original problem into smaller problems and solves the original problem recursively.

Original problem: Dynamic Programming

Smaller problem: Different alignment arrangement

What is the Dynamic Programming table?

- Stores answer of sub problem.
- Preserve the path (or alignment arrangement) information
- DP is efficient because hopeless alignments won't be calculated in the DP table
- Alignment score is the sum of the score for each pair in the alignment
- The scoring matrix is required to draw the DP table and calculate the alignment score.

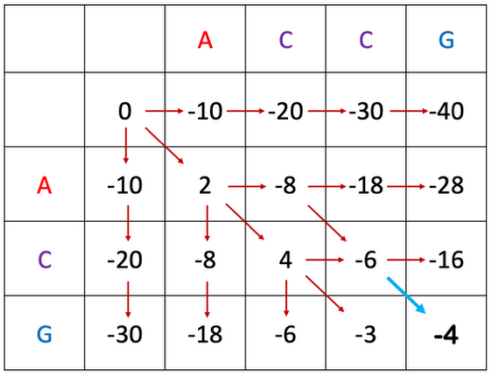
Scoring matrix:

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

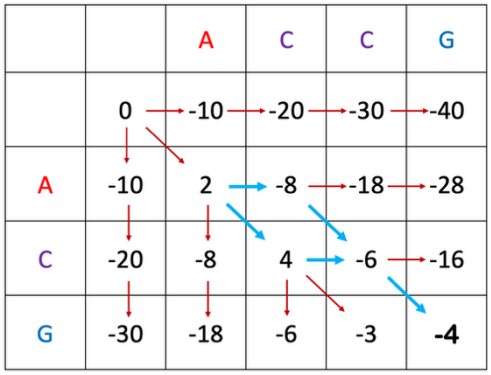
Gap penalty = -10

How to trace back the optimal alignments?

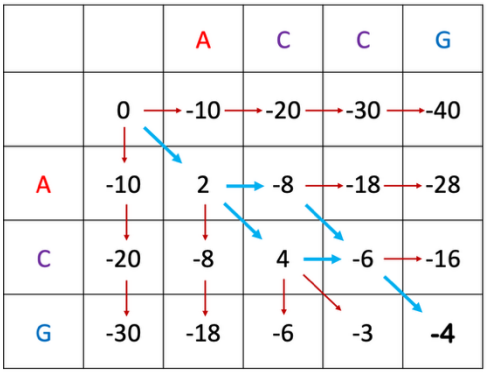
- Starting from the bottom right box (-4 for this example)



- Traceback along the arrows in the DP table (the blue arrows)



- Trace back until the box with number 0 at the top left corner.



*From the above example, 2 blue arrow pathways can be observed. It means that there are 2 optimal alignments.

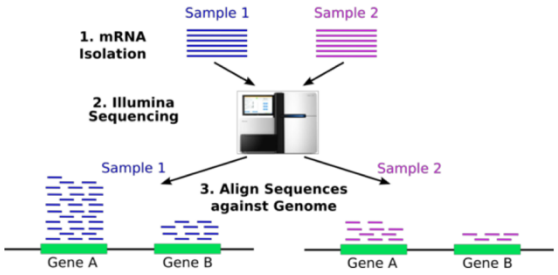
Sequence	ACCG ACCG
Optimal alignment 1	ACCG A_CG
Optimal alignment 2	ACCG AC_G

Gene Expression:

- The human Genome is very similar in all people (genetic variation ~0.001%)
- May account for the Phenotype (Genotype + Environment) difference sequence data, so we need to learn gene expression.

How do we get the gene expression matrix?

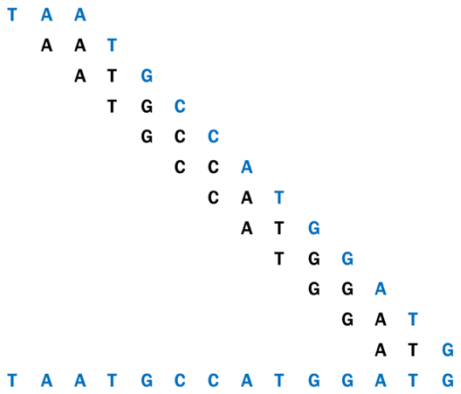
- Transition from sequence data to data matrix
  1. RNA sequencing: Cut long DNA into short RNA pieces
  2. Map the short read to the genome
  3. Count the number of reads
  4. Build a gene expression matrix



Genome assembly:

- Two 200bp short reads can have overlap regions
- The genome is assembled based on the short reads and overlap regions

1. Genome assembly seems quite simple just to line up, why do scientists cost a lot of time to do research on Human Genome Project?
   - Many factors, e.g. mutation, conflict, repeat sequence, etc.
   - The complicate alignment will slow down the process.
2. How to make the algorithm faster?
   - Implement optimized algorithms and improve data structure
   - Using de Bruijn Graphs (breaks all the reads down into even smaller, fixed-length sequences called k-mers)



- This is a simple genome-assembly example. It uses overlapping reads to assemble the genome.

Disclaimer: all figures are adapted from Prof. Li BMEG3105 Lecture Notes