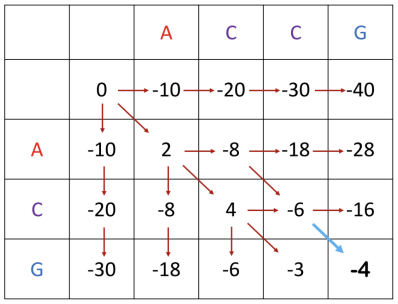
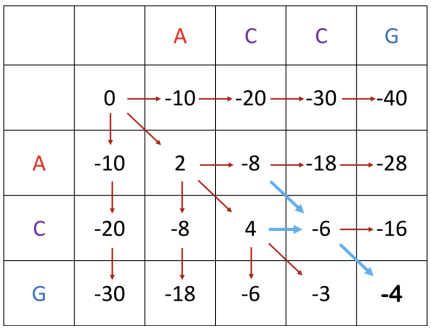
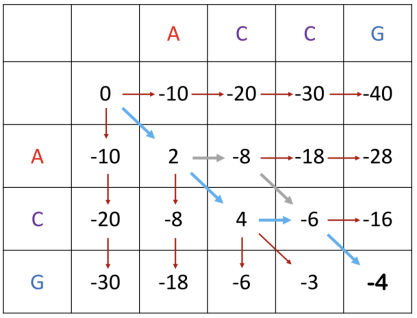
To find the optimal alignment

- Traceback:
    1. Starting from the bottom right box (-4 in this example)



    2. Traceback along the arrows (blue arrow)
    3. When there is more than one arrow, traceback to the larger number





    4. Until the 0 box (top left box with the first number)

Summary

- Sequence alignment identifies sequence similarity
- Given an alignment and the scoring matrix, calculate the alignment score
- Dynamic programming to find the optimal alignment by dividing the original problem into smaller problems and solving them recursively.
- DP table stores answers to sub-problems and construction path

Local alignment: highest number of the whole table
Global alignment: only care about the bottom right cell
DP is efficient because hopeless alignments won't be calculated in the DP table
Scoring matrix: define similarity
Biopython: Another plug-in function for alignment
BLOSUM (BLOcks SUbstitution Matrix): substitution matrix for sequence alignment of proteins

- Mismatch because of mutation
- Gap because of insertion, deletion, gene duplication
- Different scoring matrices used because built from different database

Sequence Data

- Assumption of the central dogma
- Genetic information is in DNA/ RNA sequences
- Phenotype = Genotype (determined by sequence) + environment
- Gene expression different = phenotype different, because human genomes are similar

Gene Expression Matrix

- Row: different genes
- Column: Conditions / Sample cells

Transit sequence data into a matrix

1. RNA sequencing: cut long DNA into small RNA pieces (reads)
2. Mapping: find the location of reads in the genome
3. Count the number of reads in each gene
4. Build a gene expression matrix

Genome sequence: by assembling