Pairwise Sequence Alignment

In this class, we will describe how to compare sequences in terms of similarity. We will develop methods to visualize the alignment between two sequences. How the sequence alignment can be formulated as an optimization problem. The Global Alignment Algorithms with Dynamic Programming the Needleman-Wunsch algorithm and Local Alignment Algorithms with Dynamic Programming the Smith-Waterman algorithm will be discussed and their implementation will be analysed.

Learning Objectives

- Understand the problem of sequence alignment between two sequences.
- Identify the similarity between sequences through dotplots.
- Approach the sequence alignment as an optimization problem with its biological components, Namely substitution matrices and gap penalties.
- Use Dynamic Programming to implement Global and Local alignments

1. Review the second part of the slides "Pairwise Sequence Alignment".

Quiz

Task 1 – Retrieve Sequences from Uniprot

- *Go to Uniprot.org and retrieve the amino-acid sequence of the proteins HBA Human (P69905) and HBB Human (P68871).*

- *Save in Fasta format in two separate files.*

- *Run the dotmatcher program with the two above sequences:*
  https://www.bioinformatics.nl/cgi-bin/emboss/dotmatcher

Highlight sequence similarity with dotplots. Develop python programs for the different exercises.

Task 2 – Fill the dotplot functionalities

- *Write a function that given two sequences creates a dot plot by filling the matrix with 1s where there is a match and 0s where there is a mismatch. Fill the function dotplot() in dotplots.py.*
- *Write a function that given the dotplot matrix and the two respective input sequences does the pretty printing of the matrix. Fill the function print_dotplot() in dotplots.py.*
- *Test the respective functions above with two sequences HBA and HBB.*

Auxiliary Functions. Develop auxiliary functions for your code.

Task 3 – Auxiliary Functions

- *Write a function max_mat() that receives as input a matrix with numbers and returns the indices where the maximum value is found. Complete in sequence_alignments.py.*

Sequence Alignment. Develop functions to run sequence alignments between biological sequences.

Task 4 – Sequence Alignment

- *Write a function identity() that given two sequences calculates their identity distance.*
- *Write a function score_align that given two sequences calculates the overall score. Use the function score_pos() to score every pair of symbols from the two sequences. Sum these scores and return its value. Assume sequences have equal length. Complete function score_align() in sequence_alignments.py.*
- *Create a test function to implement and print the global of two proteins, HBA and HBB using the Blosum62 as the similarity matrix.*