Searching for Sequences in Databases

In this class, we will discuss the need for searching for similar sequences in large scale. While pairwise sequence alignment is an accurate of measuring sequence similarity it is not feasible for large scale search. Hashing and word-based search implementations will be discussed. The BLAST algorithms are presented and simple implementation is provided and extended.

Learning Objectives

- Measuring sequence similarity at scale.
- Optimizing sequence search based on k-mers and extension approaches.

1. Review the second part of the slides “Searching for Similar Sequences in Databases”.

Quiz

Task 1 – Search Sequences with Blast

- *Go to NCBI Blast website https://blast.ncbi.nlm.nih.gov/Blast.cgi*
- *For the amino-acid sequences of the proteins HBA Human (P69905) and HBB Human (P68871), searching independently the most similar sequences in a database of non-redundant protein sequences database (nr).*
- *Select the most appropriate program for the effect.*
- *What is the E-value of the Top-hit?*
- *In what species the sequence has an hit?*

Task 1 – Complete the code

1. Revise the code from the slides and complete the methods for the MyBlast.py class.

2. Describe the meaning for each of the parameters in the function *best_alignment*?

3. Write a test function for the query sequences in query1.fasta and query2.fasta that finds the most similar sequence in seqBlast.txt (use as identifier the number of the sequence in the file). Print the respective score of the best alignment.

4. Develop a similar test function as in 3) with the *glyco_sequences*, using the query and the *db* for the search of the most similar sequence. For the most similar sequence perform the global and local alignment and provide as output of the function.

Task 2 – Code extension

5. Consider the function *get_hits* provided in the slides. Create a variant of this function that given an additional parameter mismatch *mismatch*, allows at most mismatch characters to be different between the query and the sequence words. Test that the mismatch is smaller than the length of the word.

6. Adapt the appropriate functions to allow returning a ranking of the best alignments and not only the sequence that scores highest.