Assignment 1

Abstract and Introduction Summary

- Alignment algorithms have been in the spotlight for the last few decades, due to a vast genomic data explosion.
- We need a lot of computational resources and time to process biological data, both of which are expensive for us, so we'd like to process the data in the shortest time possible with the fewest resources available.
- The structural and functional relationships between genes are determined largely through the comparison of genomic sequences. This distinction is made by comparing genomic sequences for similarities, variations, and mutations. This enables researchers to investigate and examine species' genetic and evolutionary relationships.
- Comparison of DNA sequences is done through software based on alignment algorithms that give results in the form of scores and percentages of similarities and identities.
- The main mission of this research area is to analyze and interpret deoxyribonucleic acid (DNA) sequences in central databases, accessible worldwide, to enable scientists to present and search biological information.
- Concerning sequence alignment, three types of alignment of the DNA sequences can be distinguished: Global alignment, Semi-Global alignment, and Local alignment.
- This study will present a new DNA sequence alignment algorithm called Discrete-To-Continuous (DTC) that ensures three forms of alignment. The proposed methodology was compared against other existing methods, which are largely based on the concept of string matching.
- The DTC algorithm delivers supremely efficient alignment with a reduced response time.
- Unlike string matching algorithms, which try to find a point-to-point correspondence of the chains, the DTC approach solves this problem in its entirety by superimposing the discrete representation of the test points on the continuous representation of the reference points.
- DTC is based on polynomial interpolation of data and uses dynamic programming.

Assignment 2

Related Work Summary

- Several studies and works have been published due to the importance of string matching algorithms in evaluating the functional and structural relationships of biological sequences.
- String matching algorithms are divided into two groups that play a key role in the analysis of biological sequences: 1) The "exact string matching”, is used to find the exact substring match; 2) the "approximate match," which tries to find strings that match to a given pattern as closely as possible. Rabin Karp and, Brute Force, and Fuzzy String are often used in this area of matching.

- Smith-Waterman algorithm: This algorithm is often used in DNA sequence alignment, especially for gene prediction, phylogeny, or function prediction. Its operating principle is to give an alignment corresponding to the best matching score between the nucleotides of the subject sequences. It relies on dynamic programming using similarity matrices or substitution matrices. Alignment is accomplished by inserting "gaps" or "INDELs" into the reference sequence or subject sequence in order to increase the number of matching characters between the two sequences.
- Needleman–Wunsch algorithm: This algorithm is often used in the maximum global alignment of two-character chains, especially a protein or DNA sequence. The algorithm looks for the maximum score alignment. This was the first application of dynamic programming for the comparison of biological sequences.
- Boyer-Moore algorithm: It is considered one of the most commonly used string matching algorithms. The operating principle of this approach is based on the analysis of the characters of the text from right to left starting with the rightmost. If a complete match is detected, it deploys two pre-computed functions to shift the window to the right, known by the matching shift and occurrence shift.
- Karp-Rabin algorithm: The Rabin-Karp algorithm calculates a numeric value (hash) for the pattern p and for each substring of m characters from text. Then, it confronts numerical values instead of confronting the real symbols. At the moment when a match is detected, the pattern is compared to the substring by a naive approach. If not, it goes to the next substring of the sequence to compare with p.
- Knuth Morris-Pratt algorithm: This algorithm was developed by Morris and Pratt as the first linear time match algorithm based on the analysis of the naive algorithm. The Knuth-Morris-Pratt algorithm preserves the information that the naive approach has consumed during the text analysis period. The use of this algorithm is effective because it minimizes the total number of comparisons of the pattern with the input string.
- Comparative Study of the String Matching Algorithms:

TABLE I. COMPARATIVE STUDY OF STRING MATCHING ALGORITHMS

Algorithm	complexity	Accuracy	Execution Time
Brute Force	O(mn)	66.7%	≈ 85ms
Deterministic Finite Automaton	O(n)	72%	≈ 65ms
Rabin-Karp	O(mn)	70%	≈ 72ms
Morris-Pratt	O($n + m$)	65%	≈ 68ms
Colussi	O(n)	74%	≈ 58ms
Boyer-Moore	O(mn)	83%	≈ 84ms
Turbo-BM	O(mn)	82.52%	≈ 86ms
Tuned Boyer-Moore	O(mn)	82.1%	≈ 88ms
Reverse Colussi	O(n)	79%	≈ 57ms
Apostolico-Giancarlo	O(n)	74%	≈ 61ms
Smith-Waterman	O(mn)	71.4%	≈ 81ms
Needleman–Wunsch	O(mn)	60%	≈ 85ms
Raita	O(mn)	76%	≈ 82ms
Reverse Factor	O(mn)	75.4%	≈ 82ms
Berry-Ravindran	O($m + n$)	77%	≈ 74ms
Aho–Corasick	O($m + n$)	79.7%	≈ 70ms
Alpha Skip Search	O(mn)	78.5%	≈ 83ms