Abstract and Introduction summary.

Abstract:

- Nowadays bioinformatics is considered as an active research field that acts as an interface between the computer science and biology.
- We have two sequences, A subsequence is a sequence that appears in the same relative order, but not necessarily contiguous.
- The longest common subsequence (LCS) is one of the essential issue to be solved viably in computational science.
- In this paper, we will discuss new calculation for LCS of two DNA sequences, will find all common subsequences in the Dna Sequences and determine their location and length then will determine the length and location of longest subsequence present between all sequences and the time complexity will be computed too.
- To achieve this, we store DNA sequences in an array then start to compare them using matching algorithm.
- nowadays DNA sequence comparison is an important and upcoming area in computational molecular biology

Intro:

- Bioinformatics helps us to understand and organize the information associated with biological molecules and to build up a guide in the examination of organic information .
- Cell is the basic unit of all living tissue, each Cell performs several operations like molecule transportation, energy conversion and reproduction.
- Each cell contains the genetic information as a form of (DNA) sequence. DNA contains the instructions for making proteins(genes).
- Human genome contains approximately 3 billion base pairs and about 20,000 genes.
- DNA sequence analysis and gene analysis are used for finding forensics related problems and discovering changes in human DNA sequence or genes and biotechnology related problems .
- To discover changes in DNA sequence we compare the affected DNA sequence with the normal one, and these kind of comparisons would take long time for processing because human genome contains three million nucleotide bases.
- biological data are big and very complex, and we need efficient ways to analyze it
- Computational methods can be useful to reduce the time and space complexity.
- Subsequences are used to determine how similar the two strands of DNA are , so it will help in discovering diseases.
- Effective subsequence match still remains challenging.

Assignment 2: Related works

- **Tripathi and Pandey (2016) :** using two similarity algorithms Rabin-Karp and (MCS) maximum common sub stream to find LCS. They found that Rabin-Karp approach is better than MCS calculation.
- **Rubi and Arockiam (2012) :** proposed Positional LCS for decreasing the time complexity.
- **Alsmadi and Nuser (2012):** evaluated two algorithms longest common subsequences (LCSS) and longest common substring (LCS) for comparing DNA sequences. Performance evaluation has been done
  by using different kind of code .
- **Wang et al. (2011):** used the dominant point approach(dominant points are computed using divide and conquer technique) for finding LCS of any number of strings.
- **Beal et al. (2015):** using suffix tree method for recognizing Common substrings in two sequences.
  then by using the direct acyclic graph(DAG) longest path of the string is determined.
- **Lavanya and Murugan (2013):** first approach for finding Multidimensional Shortest Common Subsequences (MSCS) and Multidimensional Longest Common Subsequences (MLCS) is using support vector ,second one using Positional Weight Matrix .for testing they used DNA sequences of length (100 to 25,000).
- **Peng and Wang (2017):** using graph-based model(levelled-DAG) for finding MLCS .this way takes out the hubs that cannot add to the development of MLCS. last graph have a single hub which incorporates all the required MLCS. Tests tells that levelled-DAG needs less amount of space and time.
- **Chen et al. (2017):** patients reports may contain mistakes but using LCS based on semantic similarity will help patients to get accurate reports .Two dimensional matrices are used to store the LCS and recursive relation is used for semantic similarity process.

Assignment 3: Methodology, Results and discussion

- in two DNA sequences all common subsequences will be found then the lcs among them.
- DNA sequences are stored in an array and the matching process is performed. matched subsequences are obtained then the longest subsequence is found and the time complexity is computed for different lengths of DNA sequences
- The algorithm was simulated, implemented and tested using MATLAB R2012b.
- overall flow of the methodology: Got two DNA sequences from (NCBI) database these two are given as an input to the algorithm.. they are selected and downloaded in the FASTA format. these two have different lengths.
- The length of the two sequences is calculated (100, …, 500.).
- These two sequences are represented in X and Y array. comparison is repeatedly performed until reach the last character in X and Y array.
- Length and location of the matched subsequences are stored then the greater length of common sub sequences are found.

- Dataset: total length of DNA sequence 1 is 4,412,379 bp. total length of DNA sequence 2 is 16,966 bp
- In Algorithm1 matching algorithm: each character of DNA sequence 1 and 2 are compared. If the character is matched then it is stored and next character is checked .then length and location off all matched subsequence are stored
- The algorithm gets the length of DNA 1 and DNA2. The indexes are I for DNA1and J for the DNA2
- at first it check whether I and J are less than the length of the DNA sequences.
- If two characters are matched, then perform find_string match(calculate the length and location of the matched substring). Then repeat the process .
- Algorithm2 explains find_stringmatch :at first set length to 0
- If two characters are matched, the length is increased by one ,Then read next character and repeat the process ,finally length is returned.
- Algorithm3 Information_retrieving algorithm: matching algorithm returns many subsequence. length and location of each match between two DNA sequences are stored. The maximum length can be identified from the obtained lengths of the sequences. From that the information will be retrieved and the LCS can be located.
- Algorithm: If the index is less than the Length of DNA it begins reading the arrays. matched subsequences and their location are retrieved from the array and displayed, Then the maximum length is identified and displays the longest common subsequence with the location.

- **Results and discussion:**

- *Comparison of DNA sequences:* In this algorithm, each character of two DNA sequences are compared and the length and location of matched subsequences are stored in the array.

- After finding all subsequences, the array is traversed for finding the subsequence with maximum length. Then the location and length of that subsequence is retrieved from the array and displayed

- *Performance analysis:* DNA sequences with Five different lengths are given as input to the process

- Five iterations are carried out.
- using different lengths of the DNA sequences in each iteration and results are computed
- find the LCS with their location and length for five different lengths of input DNA sequences and the total time of processing is computed.
- the retrieval process takes longer than The matching process
- total processing time increases when using longer DNA sequences