

## Abstract:

- The increase in the size of biological data and the need to search for patterns.
- The need to decrease the running time of the algorithm.
- The paper presented three pattern matching algorithms for decreasing the time of the finding pattern in large DNA sequences.
- Used word processing instead of (character processing used in other algorithms) and searching for the least frequent word of the pattern in the sequence.

## Introduction:

- Pattern matching is looking for all positions of appearance of the pattern in the text.
- The problem with pattern matching and its applications in important fields such as text processing, search engines, question answer systems, and especially, in different scopes in computational bioinformatics is increasing, such as
  - finding the positions of pattern(s) which may be remarkable for amino acid(s) or specific gene(s) or detecting disease.
  - Sequences Comparison to detect the similarity between the sequences.
- The problem is the increasing size of biological data and many existing algorithms are not suitable for large DNA sequences due to a long running time.
- The paper deals with exact pattern matching and proposes three algorithms to reduce the drawbacks of the previous algorithms.
- The operation of the algorithms is divided into:
  - The preprocessing phase: the potential intervals of the text to be matched with the pattern are called candidate windows.
  - The matching phase: compare candidate windows with the pattern.
  - And aim to decrease the number of the windows.
- The first algorithm for finding windows by searching the text for the first and the last character of the pattern at the same time instead of separately as in some other algorithms.
- Due processors can handle 4 or 8 bytes of data (4 or 8 characters) and are referred to as a word processor which can compared to another word.  
The second algorithm for word-based comparisons by processing the word using the benefit of the processing power of the processor instead of character-based algorithms to decrease the running time and enhance the performance.
- The third algorithm looks for the least frequently occurring word of the pattern in the text