Vol.1 Issue-10, JUNE 2021 (e-ISSN: 2582-8223)

JUST AGRICULTURE
multidisciplinary e-Newsletter

A modern approach to pairwise genome comparison using synteny mapping

Nalinikanta Choudhury*, Bulbul Ahmed*
*Deptt. Of Bioinformatics
IARI, Library Avenue, PUSA, New Delhi-110012

ARTICLE ID: 55

Comparative genomics is an arena of biological research in which the genomic features of different organisms are compared by using different techniques. Pairwise genome comparisons are made by matching two significant unit pair of whole genome sequences. The comparison of outdated pairwise sequence algorithms are used to process much larger datasets than they were eventually designed for the process of genome comparison. With the limitation of the computational resources, the performance of the traditional algorithms decreases with the increase in sequence size that leads to starvation of computational memory. These cause constrictions in processing in inadequate capabilities of software and thus restrict full use of available resources of hardware. To overcome this obstacle that limits the efficiency of computational analysis of large scale sequence datasets of biological origin by constituting existing algorithms or by constructing novel application that solves a major huddles for the Bioinformatics.

A brief discussion of steps involve in genome comparison:

To get rid of the limitations of traditional sequence comparison methods, first an application specific reduction of main memory and also computational space usage, and secondly modularizing the process by using standard software engineering concepts which is the main concern of modern Bioinformatics Community. Reduction in the memory usage by adopting an out-of-core strategy that designed to manage data structures that are too large to fit into core memory at one time. Therefore, there is need of virtual memory concept to deal with the flood of data. For large scale sequence analysis, poorer data locality can result in degradation of performance in memory concentrated applications. A modular application design is used to identify collections of HSPs (High Scoring Segment Pairs) by pairwise genome comparison techniques that can be used to obtain gapped fragments called k-mers. The strategies applied to the architecture of genome comparison are:

(a) Dictionary calculation

JUST AGRICULTURE
multidisciplinary e-Newsletter

(b) Hits determination

(c) HSP detection

(d) HSP post-processing

1. **Dictionary Calculation:**

The calculation of dictionary is based on the binary tree technique. Each tree node contains a word called as key and its list of occurrences called as values. In a binary tree, left hand side nodes of a particular tree come in a lexicographical order before nodes on the right hand side. Memory consumption problems are avoided that caused by the huge number of possible words i.e. a maximum theoretical of **4K** different words, without repetitions. So splitting the calculation in to **P** steps (with **P** being a multiple of 4), thus reducing the amount of memory used by the program by a factor of **P**. The dictionary is splitted up and arranged in a lexicographical order, a prefix of length **log4 P** is used. The strategy requires iterating p times over the whole sequence, using a different lexicographically-organized prefix each time to preserve word order. Memory allocation is avoided by requesting for each node, a single memory pool is reserved at the beginning of the process and new memory pools are then only reserved once the currently reserved memory is used up. The result is obtained by traverse the tree in a proper order, storing the word pertained in the node together with the list of their occurrences. Thus the memory consumption issues are reduced by Software Engineering approaches.

2. **Hits determination:**

The second section of the workflow starts with the identification of the starting points or seeds points for the local alignment. If a word say *"wi"* appears **n** times in the first sequence at positions **Pj** *(j* = 1,2,...,*n)*; and particularly, the same word *"wi"* appears **m** times in the second sequence at positions **Pk** *(k* = 1,2...,*m)*, then a hit will occur in all **(Pj, Pk)** coordinates producing the following set of hits i.e. $h = \{(1, 1), ...(1,m), (2, 1), ...(2,m),\ldots,(n, 1), ...(n, m)\}$. These hits are all then considered as starting points for possible local alignments. The number of resulting hits could be very high which depend on how similar the sequences are and also on the K value (Word size) used. It is highly recommended to disguise low complexity regions in order to reduce the hits produced by repetitive sequences. Further reduce in the number of hits, a proximity approach is

JUST AGRICULTURE
multidisciplinary e-Newsletter

applied and by the use of proximity approach, those hits which appear on the same diagonal, defined as **d = (Pj − Pk)**, and at a predefined distance are combined. This can be achieved very quickly and easily by sorting the hits by diagonal and offset, these are performed using the help ofa threaded version of the quick sort algorithm and then joining all the hits that are present within the distance parameter value. This produces a set of ungapped HSPs that conform to a local alignment.

3. HSP detection:

After formation of a set of ungapped HSPs, the score is calculated either by adding or subtracting a given weight value on the basis of DNA identity, depending on if a match or mismatch is found, respectively. The fragment usually starts from a hit with a positive score and is stretched along the sequence, modifying the overall HSP score until and unless, it becomes negative or the end of one of the sequences is reached. Fragment boundaries are the positions that give the highest added score at both ends as HSPs are extended in both directions along the sequence i.e. in both forward and backward direction. Thus the algorithm continues searching for HSPs within the next hit in diagonal or the first one of the next diagonal. If the subsequent hit in the same diagonal has been concealed by addition of the previous HSP, thus it would not be used since it will result in a redundant sub-HSP within the previous one. It results a set of recognized HSPs that are defined as starting and ending coordinates in both the sequences, together with HSP length, score and the identity levels.

4. HSP Post –processing:

All the existing methods provide a graphical way of representation of local alignments after computation. The process can also incorporates a visualization technique that can generates a PNG file so as to show the ability to output its analyses in formats that can be processed by the visualization techniques included with an existing analysis programs. In addition to that, it also includes post-processing applications that enable the tasks such as application of additional filters to HSP collections or to generate gapped alignment constructions based on ungapped ones.

The basic problems of sequence comparison are –

I. Reduction of main memory and computational space usage

II. Modularizing the process using classical software engineering concepts