MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **82** (2019) 581-598

ISSN 0340 - 6253

Outerplanar Graph Data Structure: A New Computational Analysis Model of Genome Rearrangements

Nafiseh Jafarzadeh, Ali Iranmanesh*

Department of Mathematics, Faculty of Mathematical Sciences,

Tarbiat Modares University, P.O. Box: 14115-137,

Tehran, Iran

(Received February 8, 2019)

Abstract

The computational study of genome rearrangements is one of the most important research area in computational biology and bioinformatics. In this paper, we define a novel graph data structure as a rearrangement model for whole genome alignment in large scales. This model is capable of realizing non-collinear changes as well as collinear changes. Also we apply our rearrangement graphical model to present a dynamic programing method for alignment of an arbitrary sequence to a pan-genome reference which is encoded as an outerplanar graph. In this method, a gapped alignment is considered where the gaps could be affine, linear or constant.

1. Introduction

The analysis of genome rearrangements has started from 1983, when Dobzhansky and Sturtevant [5] observed that the evolution of certain Drosophila species could be explained using a sequence of reversals. In 1988, Jeffrey Palmer [25] observed some interesting patterns in the evolution of plant organelles and he compared the mitochondrial genomes of cabbages and turnips. About 99.9% of the genes were identical in both the genomes. However, it was

*Corresponding author.

Email address: iranmanesh@modares.ac.ir

noted that the gene orders of both these vegetables were considerably different. These discoveries along with similar findings suggested that genome rearrangements might play an important role in genome evolution [22]. Up until 1990s, evolution was traditionally explained through nucleotide-level changes in the DNA sequence. The novel investigation of approaches based on comparison of gene sequences, were pioneered by David Sankoff [37]. Genome rearrangements in comparison to point mutations, are rare events. However, they can accumulate over time, prompting a clear distinction between the gene orders of the original and evolved genomes [36]. As a result, the similarity between the gene orders of two species can reveal their proximity to each other. Thus, genome rearrangements act as good phylogenetic markers. Definitely, combinatorial problems posed by genome rearrangements have attracted significant interest over the years.

There are several biological problems that can be treated with mathematical methods. In [2,4,12-15,20,24,26,30,33,40,42-49] you can see some mathematical methods as graphical and numerical representations for similarity analysis of DNA sequences. Recent advances in rapid, low-cost sequencing have opened up the opportunity to study complete genome sequences. The computational approach of multiple genome alignment allows investigation of evolutionarily related genomes in an integrated fashion, providing a basis for downstream analyses such as rearrangement studies and phylogenetic inference [19]. As an effective modeling, analysis and computational tool, graph theory is widely used in biological mathematics to deal with various biology problems like sequence comparison. Multiple genome sequence alignment is an indispensable tool for comparing genomes and finding their shared histories. In bioinformatics, a sequence alignment is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. Sequence alignment try to uncover homologies by assigning sequence positions to each other. A breakpoint is defined as a dissimilar region or point where one or more sequences have altered from the other sequences. The genome rearrangement describes the one or more breakpoints which make up a structural and evolutional variant. Evolutionary events are often classified into small changes and large structural changes. Small changes work on only one or few sequence positions which include substitutions, insertions, and deletions. They do not influence the order of sequence positions, and thus can be captured by collinear alignment. Structural changes involve longer genomic segments, thereby working on the structure and order of genomic sequences. They include non-collinear changes like inversions, translocations and duplications in addition to insertions and

deletions of longer segments. The aim of genome rearrangement is to investigate the order of homologous segments and infer genomic distances based on the number of breakpoints or predict scenarios of evolutionary changes.

These investigations often employ graphs such as breakpoint graphs [1,3,18] that resemble graph data structures used for genome alignment. Graphs can assist in improving genome comparison through multiple alignments and analysis of rearrangements. In addition, graphs provide an intuitive representation of similarities and changes between genomes, and so visualize alignment structures. In comparison to tabular alignments, genome alignment graphs are more versatile insofar that it is possible to model collinear and non-collinear changes without the need of choosing a reference genome [19]. The earliest graph is the alignment graph which has been proposed by Kececioglu in 1993 [17]. The alignment graph defined for collinear multiple alignment and this graph contains a vertex for each sequence character and edges for aligned characters. The alignment graph has been used in various versions [8,34,35]. In all versions, a collinear alignment can be obtained from the alignment graph by solving the maximum weight trace problem. In 2004, Pevzner *et al*. [31] introduced A-Bruijn graphs as a generalization of de Bruijn graphs [32] which often use for genome sequencing and fragment assembly. The structure of A-Bruijn graphs revisits an idea briefly mentioned by Kececioglu [17], the idea of merging aligned vertices. A-Bruijn graphs have one vertex for sets of aligned positions, and edges represent sequence adjacencies. In 2008, another graph has been presented which named the Enredo graph [29]. Enredo graphs which applied for collinear alignments of segments, have two vertices per set of aligned segments, a head and a tail vertex, resembling breakpoint graphs from rearrangement studies. The Enredo method iteratively eliminates various substructures from the Enredo graph before deriving a final genome segmentation. Also, Paten *et al.* [27,28] introduced a cactus graph model structure as a dissimilar graph which has vertices for adjacencies and edges for genome segments. Their structure has two valuable properties. The cactus property subdivides the graph (and genomes) into independent units by ensuring that any edge is part of at most one simple cycle. The second property is the existence of an Eulerian circuit. This circuit traverses all genome segments exactly once, even duplicated segments, conveniently providing a consensus genome.

In this paper, we present a new graph-based genome alignment approach using concept of outerplanar graph and properties of this graph. Comparing with traditional alignment matrix or partial order alignment graph, in common with A-Bruijn and Cactus graphs, our model is flexible by classification non-collinear structural changes like inversion, translocations and

duplications as well as collinear changes like insertion and deletion. But in addition, our model provides a unique circular visualization to simplify the study of evolutionary relationships between aligned genomes. Also, in the line graph approach we can get a unique Eulerian path in our representation. This rearrangement model can be use in computational analysis of cancer genomic data and other chromosomal aberrations. Then inspiring by V-align algorithm [16], we present a dynamic programing method for a gapped local alignment of an arbitrary sequence to a pan-genome reference which is encoded as our graphical model structure. This alignment can be used for finding a special pattern in a pan-genome reference.

2. Genome alignment with outerplanar graph structure

In this section, at first, we give a brief review about the definition and properties of outerplanar graphs. Outerplanar graphs occur for the first time in the literature in Harary's classical book [9]. In graph theory, a graph is outerplanar if it can be embedded in the plane such that all vertices lie on the outerface boundary. An edge of an outerplane graph is called chord, if it is not incident with the outerface. A maximal connected subgraph of a graph G is called a component of G and a cutvertex of a component is a vertex such that the component without this vertex is not connected. A graph G is called biconnected if $|G| > 2$ and $G - \{u\}$ be connected for every vertex $u \in G$. The outerplanar graphs are a subset of the planar graphs, the subgraphs of series-parallel graphs, and the circle graphs. The maximal outerplanar graphs, those to which no more edges can be added while preserving outerplanarity, are also chordal graphs and visibility graphs.

In the following, we bring some useful theorems related to outerplanar graph which we need them in this paper.

Theorem 2.1. [9] A graph G is outerplanar if and only if it contains no induced subgraph isomorphic to K_4 or $K_{2,3}$.

Theorem 2.2. [9] A biconnected outerplanar graph contains a unique Hamiltonian cycle.

Theorem 2.3. [9] Every maximal outerplanar graph of order at least 3 is biconnected.

Now, we describe the construction of outerplanar graph model structure for genomic data.

Let S be the set of input whole genome sequences, we can assume that the input sequences be either linear or circular sequences. Mathematically, a sequence is just a string (likely circular) of symbols taken from an alphabet set.

The whole genome base pairs alphabet set is $\{A/T, T/A, C/G, G/C\}$ and using sings according to oriented reverse complement, we have: $A/T = -T/A$ and $C/G = -G/C$.

Our graph-based model will cover both types of rearrangements which we explain in the following:

1. *Balanced rearrangements*: This case of rearrangements changes the chromosomal gene order but does not remove or duplicate any of the DNA of the chromosomes. The two simple classes of balanced rearrangements are inversions and translocations. An inversion is a rearrangement in which an internal segment of a chromosome has been broken twice, flipped 180 degrees, and rejoined. A translocation is a rearrangement in which acentric fragments of two non-homologous chromosomes trade places. Note that, for both inversions and translocations, no chromosomal material is gained or lost. There is simply a change in the relative locations of genes on the rearranged chromosomes.

2. *Imbalanced rearrangements*: This case of rearrangements changes the gene dosage of a part of the affected chromosomes, such as the loss of one copy or the addition of an extra copy of a segment of a chromosome which can disrupt normal gene balance. The two simple classes of imbalanced rearrangements are duplications and deletions. A duplication is a repetition of a segment of a chromosome and the loss of part of chromosome is called deletion. See Figure 1.

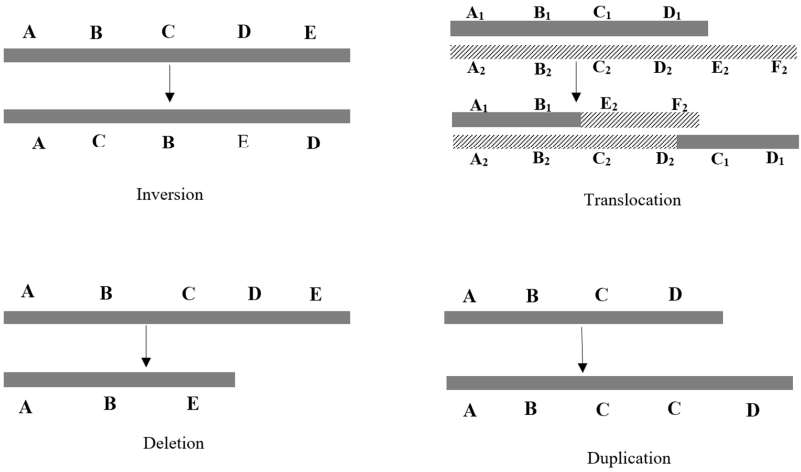

Figure 1. Different aberrations in structure of chromosome

To represent the common structure between homologous segments in a set of whole genome sequences, we define the concept of "Alignment-set" as follows:

"Alignment-set" is a set of maximal homologous segments with maximal length and denoted by *A*-set. The size of "*A*-set" is the number of aligned segments.

Note that each *A*-set may contain multiple segments of the same genome when there is some duplication in a genome. Also one *A*-set has two equivalent representations, in the first representation, some segments are in the forward orientation and some may be in the reverse complemented orientation. In the second representation all segments which are in the forward orientation in the first representation are in the reverse complemented orientation and all segments which are in the reverse complemented orientation in the first representation are in the forward orientation. The essential information about possible inversions is the orientation of segments with respect to each other and not the orientation of the *A*-set representation. In Figure 2, an example of an *A*-set in two representations is shown.

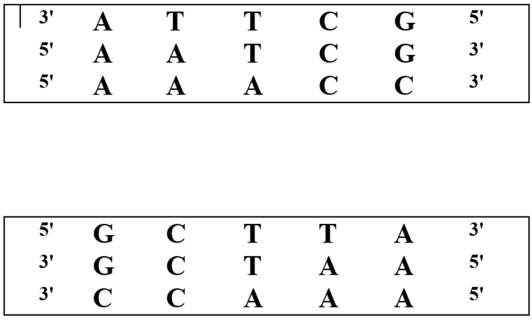

Figure 2. Two equivalent representations of A-set

We denote all of the A-sets of genome S by Σ_S which is the input for building a genome alignment graph. Note that Σ_S includes all A-sets of size one and all pairs of A-sets have to be non-overlapping.

Two A-sets $\sigma_1, \sigma_2 \in \Sigma_S$ are adjacent if there exist two segments $s_1 \in \sigma_1$, and $s_2 \in \sigma_2$ which are adjacent. The adjacency is defined by the set of positions.

In fact, assume that $s_1 = (p_1, q_1)$ and $s_2 = (p_2, q_2)$ where p_1, q_1 and p_2, q_2 are referred to the first and end positions of segments s_1 *and* s_2, with $q_1 = p_2$, then s_1 *and* s_2 are adjacent.

Since all A-sets have two orientations, there may be up to four different adjacencies between two A-sets. It means the head/tail of σ_1 can be adjacent to the head/tail of σ_2. Each of the four adjacencies is defined by a set of adjacency positions between segments from the two A-sets.

An adjacency of two A-sets $\sigma_1, \sigma_2 \in \Sigma_S$ is called a "breakpoint" if they are adjacent in at least two segments but not in all their segments. A breakpoint is defined as a region or point where the sample sequence has altered from the reference sequence. The genome rearrangement describes the one or more breakpoints which make up a structural variant. More formally, let $s_1 \in \sigma_1$, and $s_2 \in \sigma_2$ be two adjacent segments with $s_1 = (p_1, q_1)$ and $s_2 = (p_2, q_2)$ and let $p_2 = q_1$, then, s_1 *and* s_2 define a breakpoint if there is a segment $s_1' = (p_1', q_1') \in \sigma_1$ with for which no segment $s_2' = (p_2', q_2') \in \sigma_2$ with exists where $q_1' = p_2'$. So in a breakpoint adjacency,

the set of adjacency positions is smaller than the size of the A-set. In Figure 3 is shown an example of 4 breakpoints in multiple alignment of 3 sequences.

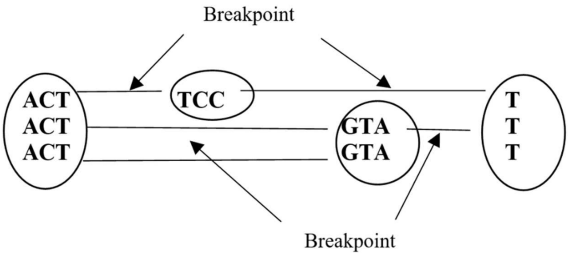

Figure 3. Breakpoints in multiple alignment

The biconnected outerplanar graph G_1 is built in fourth steps:

1. At first we construct a graph G which every A-set is a vertex of G and there is an edge between two A-sets if there adjacency between two segments of them. In fact, the graph G is an adjacency graph. As an example. Since one A-set may contain more than one segment from the same genome, each A-set can be adjacent to itself and also there may be multiple edges between two vertices.

2. Using pDFS $Algorithm$ [6], we compute biconnected components of G. Since in [9] has shown that a graph is outerplanar if and only if every one of its biconnected components is outerplanar, we restrict the outerplanarity to biconnected subgraphs. In [23] a conceptually simple algorithm is presented to determine if a graph is a maximal outerplanar or outerplanar graph. The algorithm is linear in the number of vertices. It relies on the fact that a maximal outerplanar graph has a unique Hamiltonian cycle which forms the outer face, the remainder of the graph is a triangulation of this cycle. So we apply MOP-TEST $Algorithm$ [23] to recognize outerplanar and non-outerplanar subgraphs of . If all of the connected components of G are outerplanar graphs, then we do not need to third step and we can skip that. But if there is one or more non-outerplanar components, they contain some minors isomorphic to K_4 or $K_{2,3}$.

3. By merging two adjacency-connected vertices of K_4 minors and two non-adjacency vertices of $K_{2,3}$ minors, we form graph G to an outerplanar graph G_1. It is shown in Figure 5 and 6.

4. Finally, to make our graph biconnected, we need to make it bridgeless. In the second step, if there is any bridge as a biconnected component, we easily merge vertices of that bridge just like you see in Figure 7.

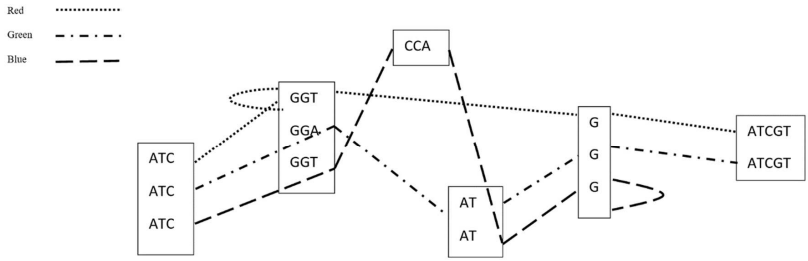

Figure 4. The graph G according to 3 sequences ATCGGTTGGGATCGT (Red), ATCAGGATGATCGT (Green) and ATCTGGCCATAGG (Blue)

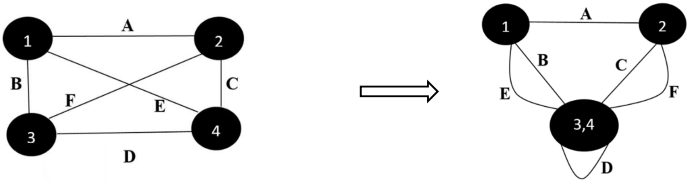

Figure 5. The original graph K_4 (left) and after merge two vertices (right)

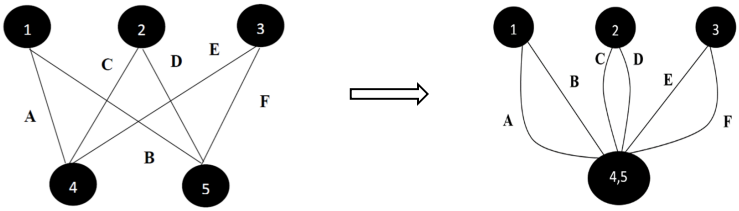

Figure 6. The original graph $K_{2,3}$ (left) and after merge two adjacent vertices (right)

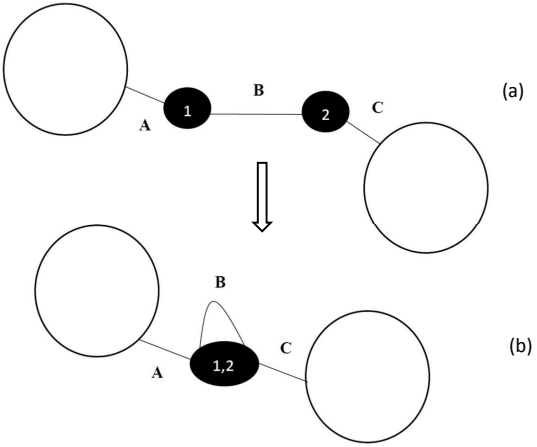

Figure 7. A non-biconnected graph (a) and after merge the bridge (b)

To show a circular visualization of this model, we can use $Algorithm\ 1$ in [38]. Input should be a biconnected outerplanar graph and output would be a circular drawing Γ of G_1 such that each node in V lies on the periphery of a single embedding circle. The time complexity of $Algorithm\ 1$ [38] is $O(m)$, where m is the number of edges in G_1.

If the biconnected graph given to $Algorithm\ 1$ [38] is outerplanar then the result will be a circular visualization such that no two edges cross. This technique has been inspired by the

algorithm for recognizing outerplanar graphs presented in [7]. By the definition of outerplanar graphs, we know that there exists a plane circular drawing for any outerplanar graph. Also, by that same definition we know that a graph which is not outerplanar does not admit a plane circular drawing. In fact, the set of biconnected graphs which may be drawn in a circular fashion without any crossings is exactly the set of biconnected outerplanar graphs. The requirement of placing all nodes on the periphery of some embedding circle is equivalent to placing all nodes on a single face of some embedding. Furthermore, if a zero-crossing visualization exists for a biconnected graph, then that drawing can be found by $Algorithm\ 1$ [38].

As we mentioned in the previous section, one of important properties of this method is that any biconnected outerplanar graph contains a unique Hamiltonian cycle which can be found in linear time. We close this section by another interesting property of this method which show that there is a unique permutation of vertices which give us a cross-free circular visualization of our outerplanar graph data structure model.

Theorem 2.4. [38] There exists only one clockwise ordering of the nodes in a biconnected outerplanar graph G such that the drawing of G with the nodes in that order around the embedding circle is plane.

3. Line outerplanar graph structure

In this section, we would like to present another approach to describe genome rearrangements by concept of line outerplanar graph and finally will compare this method with the previous one. In Graph theory, the Line graph $L(G)$ of undirected graph G is another graph that represents the adjacencies between the edges of G. The Line graph is defined as follows:

The Line graph [11] of G denoted by $L(G)$ is the intersection graph of the edges of G, representing each edge by the set of its two end vertices. In other words, $L(G)$ is a graph such that each vertex of $L(G)$ represents an edge of G andTwo vertices of $L(G)$ are adjacent if their corresponding edges share a common end point in G.

In this method, we start with graph G_1 which we have mentioned it is a biconnected outerplanar graph with a Hamiltonian cycle which passes every A-set just once. So we are going to construct a new graph G_2 which the Line graph of this graph, is isomorphism to G_1. This

approach is inspired by Pevsner's approach [32] for fragment assembly using De-bruijn graphs. We need the following theorems to prove that G_2 is Eulerian.

Theorem 3.1. [41] Let G be a non-outerplanar graph, then $L(G)$ is also non-outerplanar.

Theorem 3.2. [10] Let H be the line graph of G, then there is an Eulerian path/circuit in G if and only if there is a Hamiltonian path/circuit in H.

Our new graph G_2 is built in the following steps:

1. Let C be the set of all adjacencies of segments. The vertices V of G_2 will be a pairwisely disjoint subset of C and the edges E of G_2 represent A-sets. It's like block edges in Enredo graph [6]. We consider each A-set, as an undirected edge $e = u, v$ which the endpoint $u \in V$ of e represents a subset of adjacencies in C that contains all adjacencies at one end of e, and the other endpoint $v \in V$ contains all adjacencies at the other end of e. It is possible that $u = v$.
   Easily one can see that each maximal component in G which includes vertices connected only by adjacency edges is considered as a vertex in this new graph by ignoring all the A-sets which were defined as edges in G_2.
2. According to the collapsed vertices of G_1, we do similar collapses for some edges in graph G_2 corresponded to those vertices.

In 2015, Liu [21] presented a new and efficient algorithm, "ILIGRA", for inverse line graph construction. Given a line graph H, ILIGRA constructs its root graph G with the time complexity being linear in the number of nodes in H. using ILIGRA $Algorithm$ and considering G_1 as input, easily we can compute the graph G_2 which $L(G_2) = G_1$.

Corollary 3.1. The graph G_2 is an Eulerian Outerplanar graph.

Proof. By ILIGRA $Algorithm$, it is known that $L(G_2) = G_1$. If G_2 is a non-outerplanar graph, according to Theorem 3.1, $L(G_2)$ should be non-outerplanar too, but G_1 is an outerplanar graph, then G_2 is outerplanar too. Also G_1 has a unique Hamiltonian cycle and using Theorem 3.2, we find that G_2 is an Eulerian graph with a unique Eulerian cycle. ■

Comparing with pervious approach, here we do not miss any A-set, it means that we consider all of aligned segment in our path of circuit.

4. Gapped alignment on outerplanar graph structure

In this section, as an application of outerplanar model structure, we present a gapped local alignment model which is shifting from focusing on a single reference genome to using a 'pan-genome', that is, a representation of all genomic content in a certain species or phylogenetic clade.

Let's consider the alignment of an arbitrary sequence to a pan-genome reference which is encoded as an outerplanar graph. Our goal is to compute an alignment of the input sequence to a path in the outerplanar graph having maximum alignment score among all paths in the graph. We consider gapped alignments where the gaps could be affine, linear or constant.

In the following inspiring by the method of V-Align Algorithm [16], we define a dynamic programing formulation that would allow us to find optimal alignments.

According to substitution matrix for local alignment [39], for each base pairs a and b in the alphabet set of $\{A/T, T/A, C/G, G/C\}$, let $s(a, b)$ denote the substitution score between a and b. Let $\Delta(k)$ denote the penalty for a k length gap, it can be an affine, linear, or constant gap and $\Delta(0) = 0$ by definition. We assume that G_1 is the biconnected outerplanar graph which the vertices are A-sets of Σ and we define graph G'_1 derived from G_1 as follows:

For each vertex σ in G_1, since $\sigma \epsilon \Sigma$ is an A-set, we can show the vertex σ by $\sigma := (v_1, \dots v_{l(\sigma)})$, which each v_i is an aligned base pairs, for $1 \leq i \leq l(\sigma)$ and $l(\sigma)$ is the length of A-set σ. We denote $\Sigma' = \bigcup_{\sigma \epsilon \Sigma} \{v_1, \dots, v_{l(\sigma)}\}$ as the set of vertices of G'_1 and Clearly $|\Sigma'| = \sum_{\sigma \epsilon \Sigma}(l(\sigma))$. In G'_1, there exists an edge between v_i and v_j if $j = i + 1$, for all $\sigma \epsilon \Sigma$ and $1 \leq i \leq l(\sigma)$. Also, if there is an edge between σ and σ' in G'_1, then there is an edge between the last aligned base pairs of σ and the first aligned base pairs of σ'.

For two vertices σ'_1, σ'_2 in G'_1, let $d'_1(\sigma'_1, \sigma'_2)$ denote the minimum number of edges on any path from σ'_1 to σ'_2 in G'_1.

Let $x = (x_1, \dots, x_{l(x)})$ be the input sequence of length $l(x)$, and M denote the scoring matrix of size $|\Sigma'| \times (l(x) + 1)$, where $M(\sigma', j)$ is the entry for $\sigma' \epsilon \Sigma'$ and $1 \leq i \leq l(x)$Then we define scoring matrix M, using the following recurrence relation on $M(\sigma', j)$ for all $1 \leq j \leq m$ and $\sigma' \epsilon \Sigma'$.

$$M(\sigma', j) = Max \begin{cases} M(\sigma', j-k) - \Delta(k), & for\ all\ 1 \le k \le j \\ M(\sigma'_1, j-1) + s(\sigma'_2, x_j) - \Delta(d'_1(\sigma'_2, \sigma')), & for\ all\ (\sigma'_1, \sigma'_2) \epsilon E(G'_1) \\ 0 \end{cases}$$

The entry $M(\sigma', j)$ stores the maximum score for aligning the subsequence $(x_1, \ldots, x_j)$ from the sequence $x = (x_1, \ldots, x_l)$ to any path ending at vertex σ' in G'_1. The first term of the above formula corresponds to an alignment having k gaps in the end due to the deletion of the last k elements of $(x_1, \ldots, x_j)$. The second term corresponds to aligning x_j to an intermediate σ'_2 in the path followed by gaps due to the deletion of the remaining path, whose length is no more than $d'_1(\sigma'_2, \sigma')$ for an optimal alignment. And the third term, always is considered for local alignment.

In the same manner, we know that G_2 is the biconnected outerplanar graph which the edges are A-sets (section 3) and we define graph G'_2 derived from G_2 as follows:

For each edge σ in G_2, since $\sigma \epsilon \Sigma$ is an A-set, we can show the edge σ by $\sigma := (e_1, \ldots e_{l(\sigma)})$, which each e_i is an aligned base pairs, for $1 \le i \le l(\sigma)$ and $l(\sigma)$ is the length of A-set σ. We denote $\Sigma' = \cup_{\sigma \epsilon \Sigma} \{e_1, \ldots, e_{l(\sigma)}\}$ as the set of edges of G'_2 and Clearly $|\Sigma'| = \sum_{\sigma \epsilon \Sigma} (l(\sigma))$. In G'_2, there exists a vertex between e_i and e_j if $j = i + 1$, for all $\sigma \epsilon \Sigma$ and $1 \le i \le l(\sigma)$. Also, if there is a vertex between edges σ and σ' in G'_2, then there is a vertex between the edges corresponding to last aligned base pairs of σ and the first aligned base pairs of σ'.

For two edges σ'_1, σ'_2 in G'_2, let $d'_2(\sigma'_1, \sigma'_2)$ denote the minimum number of vertices between edge σ'_1 and edge σ'_2 in G'_2.

Now, let $x = (x_1, \ldots, x_{l(x)})$ be the input sequence of length $l(x)$, and $M'$is denoted as scoring matrix of size $|\Sigma'| \times (l(x) + 1)$, where $M'(\sigma', j)$ is the entry for $\sigma' \epsilon \Sigma'$ and $1 \le i \le l(x)$. Then we define scoring matrix M', using the following recurrence relation on $M'(\sigma', j)$ for $1 \le j \le m$ and $\sigma' \epsilon \Sigma'$.

$$M'(\sigma', j) = Max \begin{cases} M'(\sigma', j-k) - \Delta(k), & for\ all\ 1 \le k \le j \\ M'(\sigma'_1, j-1) + s(\sigma'_2, x_j) - \Delta(d'_2(\sigma'_2, \sigma')), & for\ all\ (\sigma'_1, \sigma'_2) \epsilon V(G'_2) \\ 0 \end{cases}$$

Similarly, the entry $M'(\sigma', j)$ stores the maximum score to align the subsequence $(x_1, \ldots, x_j)$ from the sequence $x = (x_1, \ldots, x_l)$ to any path ending at edge σ' in G'_2. The first term of the mentioned max expression corresponds to an alignment having k gaps in the end due to the

deletion of the last k elements of $(x_1, \ldots, x_j)$. The second term corresponds to aligning x_j to an intermediate ${\sigma'}_2$ in the path followed by gaps due to the deletion of the remaining path, whose length is no more than $d'_2({\sigma'}_2, \sigma')$ for an optimal alignment.

As we mentioned above, each matrix was presented as a scoring matrix for local alignment of an input sequence on an outerplanar graph as a graphical model of a pan-genome reference to find most similar regions between a sequence and the pan-genome reference or to match an input query with a pan-genome reference which is encoded as an outerplanar graph. Easily, with removing the third term in the above matrices, one can perform an end-to-end alignment or so-called global alignment to compare an input sequence with a pan-genome reference which encoded as an outerplanar graph, along their entire length and to get a direct mapping between all positions in the sequence and the pan-genome reference.

5. Conclusion

Genome rearrangements problem consists of finding the evolution between genomes by solving a combinatorial puzzle to find the shortest sequence of rearrangements that can transform each genome into another. In this problem, as input a set of genomes is provided where each genome is defined by the order of genes along the chromosomes. In this paper, we described a new graph theoretical data structure as a genome rearrangement model which represents and analyses repeat segments in a set of related genomes. Our method determines an outer planar graph model for multiple alignment of whole genome sequences. Also, this graph representation provides a circular visualization to simplify the study of evolutionary relationships between aligned genomes. Comparing with traditional alignment matrix or partial order alignment graph, our model is more flexible by classification non-collinear structural changes like inversion, translocations and duplications as well as collinear changes like insertion and deletion. This rearrangement model can be use in computational analysis of cancer genomic data and other chromosomal aberrations. Also, we presented a dynamic programing method for gapped local alignment of an arbitrary sequence to a pan-genome reference which is encoded as our graphical model structure. This alignment can be used for finding a special pattern in a pan-genome reference.

Acknowledgment: The authors would like to thank the referee for the valuable comments. This research was supported by Research Core: "Bio-Mathematics with computational approach" of Tarbiat Modares University, with grant number "IG-39706". Also, the first author was supported in part by Iran National Science Foundation (INSF) (Grant No. 96012405).

References

[1] M. Alekseyev, P. A. Pevzner, Breakpoint graphs and ancestral genome reconstructions, *Genome Res.* **19** (2009) 943–957.

[2] V. Aram, A. Iranmanesh, 3D-dynamic representation of DNA sequences, *MATCH Commun. Math. Comput. Chem.* **67** (2012) 809–816.

[3] V. Bafna, P. A. Pevzner, Genome rearrangements and sorting by reversals, *SIAM J. Comput.* **25** (1996) 272–289.

[4] W. Chen, B. Liao, Y. Liu, W. Zhu, Z. Su, A numerical representation of DNA sequences and its applications, *MATCH Commun. Math. Comput. Chem*. **60** (2008) 291–300.

[5] T. Dobzhansky, A. H. Sturtevant, Inversions in the chromosomes of Drosophila pseudoobscura, *Genetics*. **23** (1938) 28–64.

[6] J. A. Edwards, U. Vishkin, Brief announcement: speedups for parallel graph triconnectivity, in: M Guo, Z. Huang (Eds.), *Proceedings of the 2012 International Workshop on Programming Models and Applications for Multicores and Manycores*, ACM, New York, 2012, pp. 103–114.

[7] C. Esposito, Graph graphics: Theory and practice, *Comput. Math. Appl*. **15** (1988) 247–253.

[8] J. Fostier, S. Proost, B. Dhoedt, Y. Saeys, P. Demeester, Y. Van de Peer, K. Vandepoele, A greedy graph-based algorithm for the alignment of multiple homologous gene lists, *Bioinformatics* **27** (2011) 749–756.

[9] F. Harary, *Graph Theory*, Addison–Wesley, Boston, 1969.

[10] F. Harary, C. S. J. Nash-Williams, On eulerian and hamiltonian graphs and line graphs, *Canadian Math. Bull*. **8** (1965) 701–709.

[11] F. Harary, R. Z. Norman, Some properties of line digraphs, *Rendiconti del Circolo Matematico di Palermo* **9** (1960) 161–168.

[12] N. Jafarzadeh, A. Iranmanesh, A novel graphical and numerical representation for analyzing DNA sequences based on codons, *MATCH Commun. Math. Comput. Chem*. **68** (2012) 611–620.

[13] N. Jafarzadeh, A. Iranmanesh, C- curve: A novel 3D graphical representation of DNA sequence based on codons, *Math. Biosci*. **241** (2013) 217–224.

[14] N. Jafarzadeh, A. Iranmanesh, A new graph theoretical method for analyzing DNA sequences based on genetic codes, *MATCH Commun. Math. Comput. Chem*. **75** (2016) 731–742.

[15] N. Jafarzadeh, A. Iranmanesh, A new measure for pairwise comparison of protein sequences, *MATCH Commun. Math. Comput. Chem*. **74** (2015) 563–574.

[16] V. N. S. Kavya, K. Tayal, R. Srinivasan, N. Sivadasan, Sequence alignment on directed graphs, *J. Comput. Biol*. **26** (2019) 53–67.

[17] J. Kececioglu, D. Sankoff, Efficient bounds for oriented chromosome inversion distance, in: M. Crochemore, D. Gusfield (Eds.), *Combinatorial Pattern Matching*, Springer, Berlin, 1994, pp. 307–325.

[18] J. Kececioglu, The maximum weight trace problem in multiple sequence alignment, in: A. Apostolico, M. Crochemore, Z. Galil, U. Manber (Eds.), *Combinatorial Pattern Matching*, Springer, Berlin, 1993, pp. 106–119.

[19] B. Kehr, K. Trappe, M. Holtgrewe, K. Reinert, Genome alignment with graph data structures: a comparison, *BMC Bioinf.* **15** (2014) 99–110.

[20] B. Liao, C. Zeng, F. Q. Li, Y. Tang, Analysis of similarity/dissimilarity of DNA sequences based on dual nucleotides, *MATCH Commun. Math. Comput. Chem*. **56** (2006) 209–216.

[21] D. Liu, S. Trajanovski, P. Van Mieghem, ILIGRA: an efficient inverse line graph algorithm, *J. Math. Modell. Alg. Oper. Res*. **14** (2015) 13–33.

[22] C. A. Makaroff, J. D. Palmer, Mitochondrial DNA rearrangements and transcriptional alterations in the male-sterile cytoplasm of Ogura radish, *Mol. Cell. Biol*. **8** (1988)1474–1480.

[23] S. L. Mitchell, Linear algorithms to recognize outerplanar and maximal outerplanar graphs, *Infor. Process. Lett*. **9** (1979) 229–232.

[24] Z. Mu, G. Li, H. Wu, X. Qi, 3D-PAF curve: A novel graphical representation of protein sequences for similarity analysis, *MATCH Commun. Math. Comput. Chem.* **75** (2016) 447–462.

[25] J. D. Palmer, L. A. Herbon, Plant mitochondrial DNA evolved rapidly in structure, but slowly in sequence, *J. Mol. Evol*. **28** (1988) 87–97.

[26] D. Panas, P. Wąż, D. Bielinska–Wąż, A. Nandy, S. C. Basak, 2D–dynamic representation of DNA/RNA sequences as a characterization tool of the zika virus genome, *MATCH Commun. Math. Comput. Chem.* **77** (2017) 321–332.

[27] B. Paten, D. Earl, N. Nguyen. M. Diekhans, D. Zerbino, D. Haussler, Cactus: Algorithms for genome multiple sequence alignment, *Genome Res*. **9** (2011) 1512–1528.

[28] B. Paten, D. Earl, N. Nguyen. M. Diekhans, D. Zerbino, D. Haussler, Cactus graphs for genome comparisons, *J. Comput. Biol*. **18** (2011) 469–481.

[29] B. Paten, J. Herrero, K. Beal, S. Fitzgerald, E. Birney, Enredo and Pecan: genome-wide mammalian consistency-based multiple alignment with paralogs, *Genome Res*. **18** (2008) 1814–1828.

[30] J. Pesek, A. Žerovnik, Numerical characterization of modified Hamori curve representation of DNA sequences, *MATCH Commun. Math. Comput. Chem*. **60** (2008) 301–312.

[31] P. A. Pevzner, H. Tang, G. Tesler, De novo repeat classification and fragment assembly, *Genome research*. **14** (2004) 1786–1796.

[32] P. A. Pevzner, H. Tang, M. S. Waterman, An Eulerian path approach to DNA fragment assembly, *Proc. Nat. Acad. Sci*. **98** (2001) 9748–9753.

[33] Z. H. Qi, M. Z. Jin, An intuitive graphical method for visualizing protein sequences based on linear regression and physicochemical properties, *MATCH Commun. Math. Comput. Chem.* **75** (2016) 463–480.

[34] T. Rausch, A. Emde, D. Weese, A. Doring, N. Reinert. Segment-based multiple sequence alignment, *Bioinformatics* **24** (2008) 187–192.

[35] K. Reinert, H. P. Lenhof, P. Mutzel, K. Mehlhorn, J. D. Kececioglu, A branch-and-cut algorithm for multiple sequence alignment, in: M. Waterman (Ed.), *Proceedings of the First Annual International Conference on Computational Molecular Biology (RECOMB)*, ACM, New York, 1997, pp. 241–250.

[36] A. Rokas, P. W. Holland, Rare genomic changes as a tool for phylogenetics, *Trends Ecol. Evol*. **15** (2000) 454–459.

[37] D. Sankoff, R. Cedergren, Y. Abel, Genomic divergence through gene rearrangement, *Meth. Enzym*. **3** (1990) 428–438.

[38] J. M. Six, I. G. Tollis, Circular Drawings of biconnected graphs, *Proc. Alenex*. **99** (1999) 57–73.

[39] T. F. Smith, M. S. Waterman, Comparison of biosequences, *Adv. Appl. Math*. **2** (1981) 482–489.

[40] D. Sun, C. Xu, Y. Zhang: A novel method of 2D graphical representation for proteins and its application, *MATCH Commun. Math. Comput. Chem.* **75** (2016) 431–446.

[41] M. M. Syslo, Characterizations of outerplanar graphs, *Discr. Math.* **26** (1979) 47–53.

[42] S. Wang, J. Yuan, DNA computing of directed line-graphs, *MATCH Commun. Math. Comput. Chem.* **56** (2006) 479–484.

[43] R. Wu, Q. Hu, R. Li, G. Yue, A novel composition coding method of DNA sequence and its application, *MATCH Commun. Math. Comput. Chem.* **67** (2012) 269–276.

[44] R. Wu, R. Li, B. Liao, G. Yue, A novel method for visualizing and analyzing DNA sequences, *MATCH Commun. Math. Comput. Chem.* **63** (2010) 679–690.

[45] J. F. Yu, J. H. Wang, X. Sun, Analysis of similarities/dissimilarities of DNA sequences based on a novel graphical representation, *MATCH Commun. Math. Comput. Chem.* **63** (2010) 493–512.

[46] Q. Zhang, B. Wang, On the bounds of DNA coding with H-distance, *MATCH Commun. Math. Comput. Chem.* **66** (2011) 371–380.

[47] Y. Zhang, W. Chen, A new measure for similarity searching in DNA sequences, *MATCH Commun. Math. Comput. Chem.* **65** (2011) 477–488.

[48] Y. Zhang, W. Chen, New invariant of DNA sequences, *MATCH Commun. Math. Comput. Chem.* **58** (2007) 197–208.

[49] X. Zhou, K. Li, M. Goodman, A. Sallam, A novel approach for the classical Ramsey number problem on DNA-based supercomputing, *MATCH Commun. Math. Comput. Chem.* **66** (2011) 347–370.