Journal of Theoretical Biology 494 (2020) 110215

Contents lists available at ScienceDirect

Journal of Theoretical Biology

journal homepage: www.elsevier.com/locate/jtb

Graph based analysis for gene segment organization In a scrambled genome

Mustafa Hajij[a], Nataša Jonoska[b,1,*], Denys Kukushkin[b], Masahico Saito[b,2]

[a] Department of Computer Science, Ohio State University, Columbus, OH 43210, USA
[b] Department of Mathematics and Statistics, University of South Florida, Tampa, FL 33612, USA

ARTICLE INFO

Article history:
Received 23 February 2018
Revised 23 February 2020
Accepted 25 February 2020
Available online 26 February 2020

Keywords:
Scrambled genome
Gene segment organizations
Point cloud from graph properties
Hierarchical cluster analysis (HCA)
Oxytricha trifallax

ABSTRACT

DNA recombinant processes can involve gene segments that overlap or interleave with gene segments of another gene. Such gene segment appearances relative to each other are called here *gene segment organization*. We use graphs to represent the gene segment organization in a chromosome locus. Vertices of the graph represent contigs resulting after the recombination and the edges represent the gene segment organization prior to rearrangement. To each graph we associate a vector whose entries correspond to graph properties, and consider this vector as a point in a higher dimensional Euclidean space such that cluster formations and analysis can be performed with a hierarchical clustering method. The analysis is applied to a recently sequenced model organism *Oxytricha trifallax*, a species of ciliate with highly scrambled genome that undergoes massive rearrangement process after conjugation. The analysis shows some emerging star-like graph structures indicating that segments of a single gene can interleave, or even contain all of the segments from fifteen or more other genes in between its segments. We also observe that as many as six genes can have their segments mutually interleaving or overlapping.

1. Introduction

It has long been observed that genome combining processes on an evolutionary scale can lead to speciation (Dobzhansky, 1933), while on developmental scale they often involve DNA deletions (Beermann, 1977; Shibata et al., 2012) as well as wholescale programmed rearrangements (Smith et al., 2012; Prescott, 1994). For example, the highly diverse collection of antibodies often is attributed to somatic DNA recombination (Tonegawa, 1983), and rearrangements on a chromosomal levels can be observed during homologous recombination (Rieseberg, 2001). In recent years there are numerous observations of alternative splicing where rearranging patterns of exons and introns of a single gene can produce different protein variants from a single mRNA (e.g. Haussmann et al. (2016)). Rearranging segments of nucleotide sequences can be organized in a variety of arrangements on the locus, for example, they can be overlapping or interleaving (Braun et al., 2018). *Oxytricha trifallax* is a single cell organism that is often taken as a model organism to study DNA rearrangement processes. This, and similar species of ciliates undergo massive genetic restructuring of a germline micronucleus (MIC) during the development of a somatic macronucleus (MAC) specializing in gene expression Within this process, over 16,000 macronuclear nanochromosomes assemble through DNA processing events involving global deletion of 90-95% of the germline DNA, effectively eliminating nearly all so-called "junk" DNA, including intervening DNA segments (internally eliminated sequences, IESs). Because these IES segments interrupt the coding regions of the precursor macronuclear gene loci in the micronucleus, each macronuclear gene may appear as several nonconsecutive segments (macronuclear destined sequences, MDSs) in the micronucleus. Moreover, the precursor order of these MDS segments for thousands of genes can be permuted or inverted in the micronucleus such that during the macronuclear development, all IESs are deleted and the MDSs are rearranged to form thousands of gene-sized chromosomes.

In Chen et al. (2014) and later in Braun et al. (2018), it was observed that an IES between consecutive MDSs of one gene can contain MDS segments from other genes, and that this process can be nested. Furthermore MDSs from different MAC genes can overlap or one MDS can be a subsegment of an MDS of another gene.

* Corresponding author.
E-mail address: jonoska@mail.usf.edu (N. Jonoska).

[1] NJ was partially supported by the grants NSF DMS-1800443/1764366 and the Southeast Center for Mathematics and Biology, an NSF-Simons Research Center for Mathematics of Complex Biological Systems, under National Science Foundation grant no. DMS-1764406 and Simons Foundation grant no. 594594.
[2] MS was partially supported by the grants NSF DMS-1800443/1764366

https://doi.org/10.1016/j.jtbi.2020.110215
0022-5193/

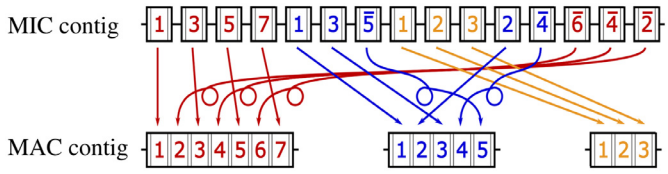

Fig. 1. An example of a rearrangement of interleaving gene segments in *Oxytricha trifallax*. A MIC contig containing MDSs of three MAC contigs indicated in red, blue and yellow. The MDSs are indicated with colored boxes with numbers corresponding to their respective order in the corresponding MAC contig, while the IESs are short line segments in the precursor MIC contig connecting the MDSs. The different colors indicate different genes. Barred numbers represent segments that are inverted with the respect to the other segments in the MAC contig, their rearrangement requires inversions which are indicated with loops (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

An example of a type of interleaving situation is schematically depicted in Fig. 1. The MDSs are represented by colored boxes with numbers. This illustrative example shows a MIC contig[3] that contains MDSs of three MAC contigs indicated with red, blue and yellow. Each MDS segment is represented by a box while the numbers within the boxes correlate with the order of the MDSs in the corresponding MAC contigs. The barred numbers indicate MDSs in a reverse orientation (inverted) in the MIC contig relative to the ordering of the MDSs in the corresponding MAC contig. The short line segments between the boxes indicate IESs. Recent sequencing and annotation of the whole *O. trifallax* genome allows genome level studies (Chen et al., 2014; Burns et al., 2015). Scrambling patterns within thousands of genes were observed revealing hidden structures among the scrambled gene/nanochromosome segments that explain over 95% of the scrambled genome (Burns et al., 2016). While those studies were focused on scrambled recurrent patterns within a single gene, in this paper we study inter-gene segment arrangements.

The goal of this paper is to expand the studies of gene segment arrangement and identify prevalent patterns in inter-gene segment organization in *O. trifallax*. DNA rearrangement processes involve MIC segments of the same MAC chromosome that can overlap or interleave with segments of other MAC chromosomes. Such overlapping or interleaving gene segment appearances are called here *gene segment organization*. Prevalent scrambling patterns of DNA segments from the same gene were used in evolutionary analysis of several ciliate species (Chang et al., 2005). Detecting patterns in inter-gene segment organization may provide another additional method for species comparison. To our knowledge, sequencing of the whole genome has not been reported for another ciliate species within the subclass of *Stichotrichia*, where scrambled genetic structure is most pronounced. While the method that we present in this paper is used on *O. trifalax*, it can be applied to other species as their genomes are readily available. Our initial findings of this study were a basis for more systematic description of certain segment organizations in *O. trifallax* that resembled 'Russian dolls' (Braun et al., 2018).

We use graphs to represent segment organizations in a chromosome locus. We represent a micronuclear locus with the interleaving and overlapping gene segments by a directed graph. Such obtained graph data is then converted to a set of points (point cloud) in a Euclidean space and we apply hierarchical clustering technique to the whole genome data of *Oxytricha trifallax* (Chen et al., 2014; Burns et al., 2015). We identified a set of star-like graphs that show several situations where segments of a single gene interleave with segments of up to 15 other genes.

[3] contig is a term used for an adjoining length of a DNA sequence obtained by assembly process after sequencing

Due to advances in bioscience and biotechnology, the growth of biomolecular data has exploded and many data analysis algorithms have been developed aiming to better understand the generated data (Cheng et al., 2006; Fernandez-Lozano et al., 2014; Meinicke, 2015). Data analysis using topological methods has proven to be useful in showing general patterns that were difficult to observe with other techniques. In the last decade, Topological Data Analysis (TDA) has shown to be another tool for data analysis and data mining that can be used to extract topological information from various types of data (Carlsson, 2009). TDA in dimension 0 can be used for hierarchical analysis, and this is the approach we use in this paper (Section 3). More details on TDA and persistence homology can be found in Carlsson (2009); Edelsbrunner and Harer (2008); Ghrist (2008a).

Our data set consists of a set of graphs $\mathcal{G}$ representing inter-gene segment organizations of the *O. trifallax* scrambled genome. This set is converted to a set of points in a Euclidean space, a point cloud $S_{\mathcal{G}}$ (Section 2). For the hierarchical clustering analysis we applied TDA at dimension zero to the obtained point cloud in $\mathbb{R}^n$. This process is described in Section 3. Section 4 describes the results of our findings. We end our exposition with a short discussion (Section 5).

2. Graphs associated with gene segment organization

2.1. Sequencing data used

The MDS interrelationship analyzed below uses the genome sequencing data in Chen et al. (2014), and can be downloaded from the Supplemental Information in Chen et al. (2014) and also in Burns et al. (2016). In this paper, the data used for analysis is the same data used in Burns et al. (2016) where it was suitably processed. No further processing for our study was performed. We refer to this data as data $\mathcal{D}$ and it is available at http://knot.math.usf.edu/data/scrambled_patterns/processed_annotation_of_oxy_tri.gff.

2.2. Graphs corresponding to MIC contings

As MDS from different MAC contigs can overlap or interleave in a MIC contig we define the following types of relationships between MAC contigs located on a single MIC contig.

- (Type 1 : Overlapping) If an MDS of a MAC contig g_1 overlaps with an MDS of another, distinct MAC contig g_2, then it is said that g_1 and g_2 *overlap*, or they are *overlapping*. We also say that g_1 has type 1 interaction with g_2, or g_1 has interaction of type 1 with g_2.
  Two MAC contigs are considered to be overlapping if they have at least one pair of MDSs that overlap with at least 20bp in common. This is because two consecutive MDSs of the same MAC contig usually share sequences at their ends (pointers) that guide the rearrangement process (Prescott, 1994), and two MDSs from distinct MAC contigs can share the same pointer sequence. The length of these pointer sequences usually ranges between 2 to 20 nucleotides.
  The overlapping relation is symmetric in the sense that, if g_1 overlaps with g_2, then g_2 overlaps with g_1. The situation is depicted in Fig. 2 (Type 1). In the figure, MDSs of g_1 and g_2 are represented by blue and red rectangles, respectively. This case excludes the case when one MDS is completely included in another, even though being a subsequence is a particular type of "overlapping". Such situation is included in Type 2 case (below).
- (Type 2 : Containment) If an MDS of a MAC contig g_1 (the blue segments in Fig. 2) is contained in (is a subsegment of) an MDS of another distinct MAC contig g_2 (red in Fig. 2), then it is said that an MDS of g_1 is contained in an MDS of g_2, and we say g_1 has type 2 interaction with g_2.

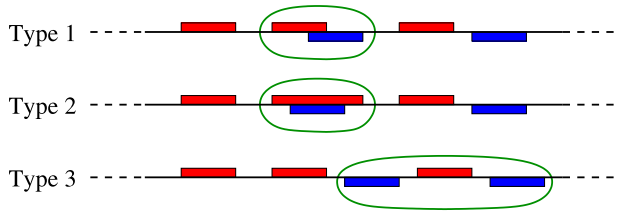

Fig. 2. Three types of MDS segment organization of different MAC contigs. MDSs of the same MAC contig are colored the same. The overlapping, subsegment and interleaving relationships of segments of distinct genes are indicated by circles in the figure, respectively. Uncircled segments indicate that there may be other gene segments that may or may not participate in gene organizations in question. The blue segments correspond to gene g_1 and the red segments correspond to gene g_2 (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

For this interaction when an MDS M of g_2 contains an MDS M' of g_1, we require that both ends of M have at least 5 bps that are not in common with M'. That is, we require a complete inclusion such that there are no pointer sequences in common. In Fig. 2 (Type 2), MDSs of g_1 are depicted in blue, and those for g_2 in red. This relation is not symmetric. We distinguish this situation from the one in Type 1 because the unscrambling of an MDS that is next to at least one IES (Type 1) may use a different biological process involving Piwi-interacting RNA (Fang et al., 2012) rather than the one that does not neighbor an IES (g_1 in Type 2).

- (Type 3 : Interleaving) If an IES of a MAC contig g_1 (blue in Fig. 2) contains an MDS of another, distinct MAC contig g_2 (red in Fig. 2), then it is said that an MDS of g_1 interleaves (or is interleaving) with g_2, or g_1 has Type 3 interaction with g_2. This relationship may not be symmetric. We allow that the ends of an interleaving MDS of g_1 and the MDSs of g_2 to intersect (overlap) up to (including) 5 bases. This requirement distinguishes type 3 case from the 'overlapping' case where the requirement is at least 20 bases.

We consider pairs (g_1, g_2) of MAC contigs g_1 and g_2 that belong to the same MIC contig. To each pair of MAC contigs (g_1, g_2) we associate a triple $c(g_1, g_2) = (b_1, b_2, b_3)$ where each entry b_i $(i = 1, 2, 3)$ indicates whether g_1 is in relationship of Type i with g_2. The value of b_i is either 0 (there is no relationship of Type i) or 1 (g_1 is related to g_2 with Type i).

To investigate the situations of these three types of interactions of MDSs, we associate an edge-labeled graph with directed edges to each MIC contig in data $\mathcal{D}$ as follows.

Each graph $G = G_M = (V(G_M), E(G_M))$, which may be disconnected and have multiple connected components, corresponds to a MIC contig M. Each vertex $g \in V(G_M)$ corresponds to a MAC contig g whose MDSs are segments of the MIC contig M.

A labeled directed edge from g_1 to g_2 with label $c(g_1, g_2$ is in $E(G_M)$ if $c(g_1, g_2) \neq (0, 0, 0)$.

In the figures below we use colors on the edges to indicate the labels of the edges: red$= (1, 1, 1)$, green$= (1, 1, 0)$, blue$= (1, 0, 1)$, orange$= (0, 1, 1)$, purple$= (1, 0, 0)$, cyan$= (0, 1, 0)$, and black$= (0, 0, 1)$. All graphs can be found at http://knot.math.usf.edu/data/Colored_Graphs/index.html.

Fig. 3 shows a locus of the MIC contig ctg7180000069854 containing MDSs of three MAC contigs 5027.0 (purple), 21621.0 (black), and 4739.0 (cyan). Figures 15 and 28 depict some other examples of graphs for MIC contigs in the data.

The set of graphs corresponding to the data $\mathcal{D}$ that is analyzed here is denoted $\mathcal{G}_\mathcal{D}$ such that $\mathcal{G}_\mathcal{D} = \{G_M \mid M$ is a MIC contig in $\mathcal{D}\}$. There are 629 distinct colored graph isomorphism classes and 288 isomorphism classes of colored connected components of the graphs in $\mathcal{D}$, and they can be found at: http://knot.math.usf.edu/data/Colored_Components/index.html.

2.3. Graph features selection

We describe a method of converting graph data set to a set of points in the Euclidean space $\mathbb{R}^n$, i.e., the point cloud.

To each (directed and colored) graph G in the data we associate a vector $P(G) \in \mathbb{R}^n$ obtained by using relevant numerical graph invariants. This vector is then considered as a point in $\mathbb{R}^n$ (Fig. 4).

The vector $P(G)$ is obtained by using local, vertex specific, and global, graph specific, features of G. In this first global analysis of the genome we cluster the data according to general graph structure properties, therefore for each $G \in \mathcal{G}_\mathcal{D}$ we also consider a corresponding undirected graph $U(G)$. The undirected, uncolored graph $U(G)$ is obtained from G by replacing each pair of parallel edges with opposite directions in G with an undirected edge, and by ignoring the direction and the colors of the edges as shown in Fig. 5.

Global Vector. A vector $P_{gl}(G)$ with three entries, called the *global vector*, is associated to each graph $G \in \mathcal{G}_\mathcal{D}$. This vector $P_{gl}(G)$ consists of three features $\langle |V(G)|, |E(G)|, CN(G)\rangle$ where $|V(G)|$ and $|E(G)|$ are the numbers of vertices and edges in G, respectively, and $CN(G)$ is the size of the largest clique in $U(G)$. The isolated vertices are not counted in $|V(G)|$ as they represent MAC contigs that have no interrelation with any other MAC contig present in the MIC contig represented by the graph. In the analyzed data the maximum number of vertices is 43, the maximum number of edges is 74, and the largest clique size is 6 (appears twice in the data).

Local Vector. Vectors that use local properties of the vertices are associated to each $G \in \mathcal{G}_\mathcal{D}$. For each vertex v_i we consider two numbers, its valency $val(v_i)$, and the clique number $cq(v_i)$. The valency $val(v_i)$ is a summation of its out-degree and its in-degree (including the parallel oppositely oriented edges). The clique number $cq(v_i)$ is the number of cliques (induced subgraphs of $U(G)$ isomorphic to the complete graph K_k for some k) that contain vertex v_i.

The vertices in G, $v_1, v_2, \ldots, v_{|V(G)|}$ are ordered such that their valences are non-increasing such that $v_{|V(G)|}$ is minimal. Vertices that have the same valency are further ordered such that their clique numbers are non-increasing. This order remains fixed for the graph G. The *valency vector*, denoted $P_{val}(G)$, consists of a list of valencies of the preordered vertices $P_{val}(G) = \langle \text{valence}(v_i)\rangle_{v_i \in V(G)}$ of the graph G. The maximum valency of a vertex in the analyzed data is 29 achieved by contig 67157 with 25 outgoing edges and 4 incoming edges. Out of those 29 edges, 23 of the outgoing edges have label (0,0,1), one edge has label (0,1,0) and the other is labeled (0,1,1). The maximum outgoing valency is 25 achieved by the contig 67157 (see Figure 25 in the SI). The maximum incoming valency is 6 and it is achieved by contig 67223 (see Figure 26 in SI).

The vertex order of the clique vector follows the same predetermined order of vertices for G. We denote this vector by $P_{cq}(G) = \langle cq(v_i)\rangle_{v_i \in V(G)}$. An example of construction of $P_{cq}(G)$ is depicted in Fig. 7.

The Graph Vector. The *graph feature vector* $P(G)$ is defined by concatenating the vectors $P_{gl}(G)$, $P_{val}(G)$ and $P_{cq}(G)$. For a graph G, the number of entries of the vectors $P_{val}(G)$ and $P_{cq}(G)$ are the same and therefore $P(G)$ is a vector in $\mathbb{R}^{2|V(G)|+3}$. We denote the set of vectors associated to $\mathcal{G}_\mathcal{D}$ with $S_\mathcal{D}$, or simply S.

The Point Cloud. Observe that the number of entries of the vectors in S is not uniform, because this number depends on the number of vertices in the corresponding graph. In order to work in the common Euclidean space, we expand some of the vectors (by appending 0's) to obtain a consistent number of entries in all vectors. This modification is obtained in the following way.

Let $d = \max\{\ |V(G)|\ \mid G \in \mathcal{G}_\mathcal{D}\ \}$. If the valence vector of G is

$$P_{val}(G) = \langle v_1, v_2, \ldots, v_{|V(G)|}\rangle,$$

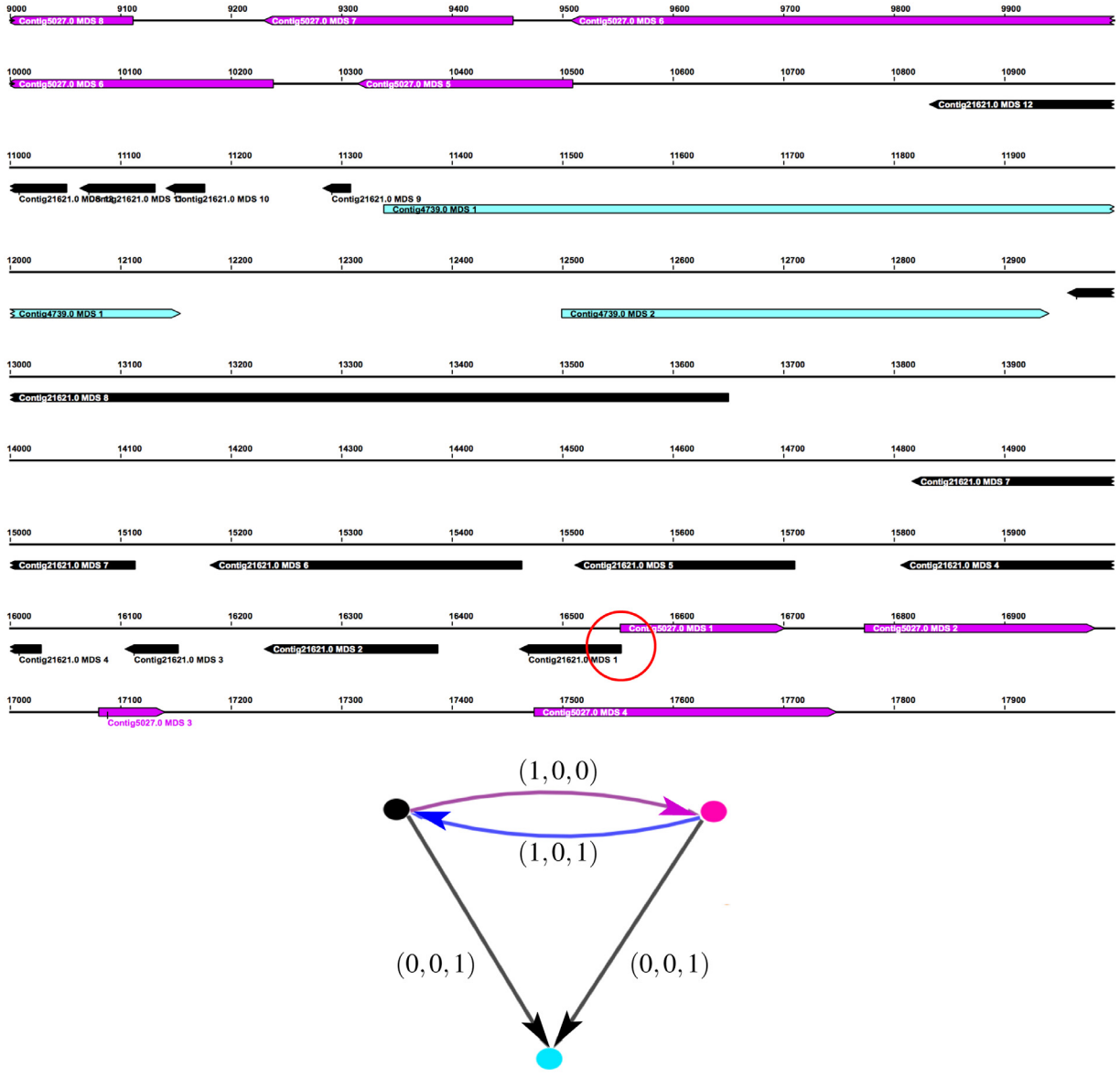

Fig. 3. A segment of MIC contig ctg7180000069854 (top) and its corresponding graph (bottom). There is an overlap of MDSs of the black and purple contigs (as circled in the top figure) and the purple contig interleaves with the black (hence there is a blue edge indicating label (1,0,1) from the purple to the black vertex) but there is no MDS segment of the black purple contig that interleaves with the purple contig, so there is a purple edge indicating (1,0,0) in the opposite direction. IESs of both black and purple contigs contain MDSs of the cyan contig, so there are two black edges indicating labels (0,0,1) ending at the cyan vertex.

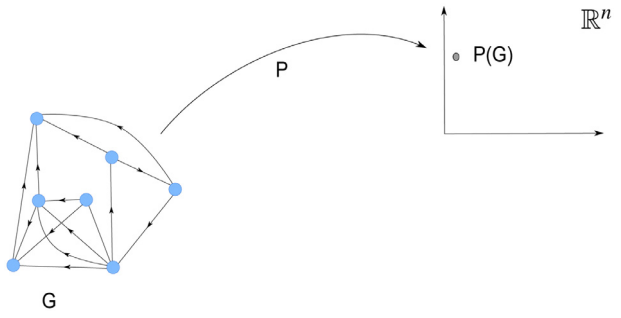

Fig. 4. Every graph G is associated to a feature vector $P(G)$, a point in the Euclidean space $\mathbb{R}^n$.

then we construct an auxiliary valence vector for G with

$$\hat{P}_{val}(G) = \langle \nu_1, \nu_2, \ldots, \nu_{|V(G)|}, 0, \ldots, 0 \rangle$$

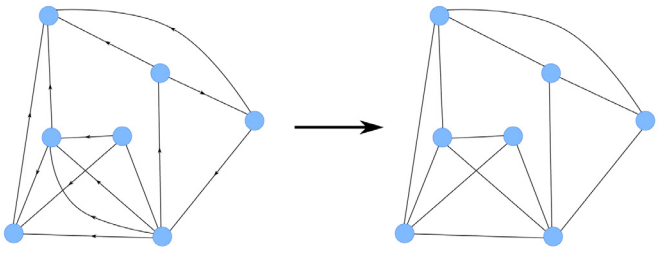

Fig. 5. An undirected graph to the right, associated to a directed one to the left.

increasing the number of entries of $P_{val}(G)$ to d such that $d - |V(G)|$ entries of zeros are added at the end. Similarly we construct auxiliary clique vector $\hat{P}_{cq}(G)$ by adding $d - |V(G)|$ zeros at the end of $P_{cq}(G)$. The graph vector $\hat{P}(G)$ is redefined with the concatenation $\langle P_{gl}(G), \hat{P}_{val}(G), \hat{P}_{cq}(G) \rangle$.

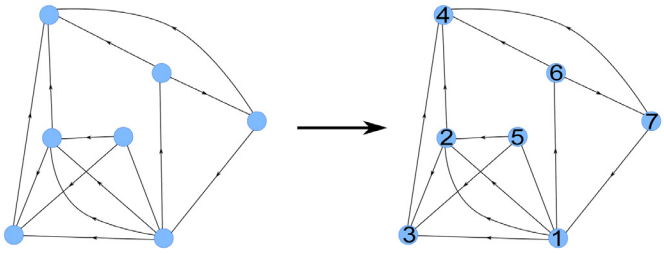

Fig. 6. Vertices of a graph G are ordered by their valances. Here, $P_{val}(G) = \langle 6, 5, 4, 4, 3, 3, 3\rangle$.

There are several reasons for adding 0s to the vector entries where necessary. The points of the point cloud must be in the same dimension in order to apply hierarchical clustering. The additional 0s can be seen as a situation where there are other MIC contigs, that are further away from the MIC contig in question, that do not have any relative interactions. Furthermore the entries of valency and clique vectors are arranged in non-increasing orders so that the entries to the right are smaller numbers therefore adding 0s to the right would not make significant differences representing graph features we analyze. Lastly, vector entries representing the local features (valencies and cliques) must compare with entries of the same type of local features, i.e., we cannot compare valencies with cliques.

For the analyzed graph data $\mathcal{G}_{\mathcal{D}}$, the maximum number of vertices in a graph is $d = 43$. We abuse the notation and use $P(G)$ instead of $\hat{P}(G)$ to refer to the zero-augmented feature vector associated with a graph G. The final point cloud set $S = \{P(G) | G \in \mathcal{G}_{\mathcal{D}}\}$ forms a subset of $\mathbb{R}^{2d+3} = \mathbb{R}^{89}$.

For comparison, we also consider the point cloud S_{gl} from the global features vector $P_{gl}(G)$. The point cloud S_{gl} is in $\mathbb{R}^3$. The data produced 273 vectors obtained with the above construction.

It is important to notice that the entries of the vectors P, and P_{gl} for a graph G are graph isomorphism invariants.

Lemma 2.1. *If graphs G and G' are graph isomorphic (but not necessarily edge label preserving) then $P(G) = P(G')$.*

Proof. Let ϕ: $G \to G'$ be a graph isomorphism. Then G and G' have the same number of vertices, edges and the size of the maximal cliques in their undirected versions $U(G)$ and $U(G')$. Therefore $P_{gl}(G) = P_{gl}(G')$ and the first three entries of $P(G)$ and $P(G')$ are the same. Also the number of non-zero entries in $P(G)$ and $P(G')$ are the same. A graph isomorphism maps vertices of G to vertices of G' with the same number of outgoing and incoming edges. Similarly, the number of cliques incident to a vertex in $U(G)$ is the same to the number of cliques of the corresponding vertex in $U(G')$. Let $V_1, \ldots, V_s$ be a partition of $V(G)$ such that

i. for all $v, w \in V_i$, for all $i = 1, \ldots, s$ $val(v) = val(w)$ and $cq(v) = cq(w)$, and
ii. for all $v \in V_i$ and $w \in V_j$ with $i < j$ $(i, j = 1, \ldots, s)$, either $val(v) > val(w)$ or $val(v) = val(w)$ and $cq(v) > cq(w)$.

Then $\{\phi(V_1), \ldots, \phi(V_s)\}$ is a partition of the vertices of G' satisfying the properties [i] and [ii]. Any order of vertices of $V(G)$ (resp. $V(G')$) that has non-increasing valencies and non-increasing clique numbers must list vertices of V_i (resp. $\phi(V_i)$) before vertices of V_j (resp. $V(G')$) whenever $i < j$. Therefore it must be that $P_{val}(G) = P_{val}(G')$. □

The construction of the vectors induced 273 distinct vectors for the 629 isomorphism clases in data $\mathcal{D}$. There are three reasons that produced this size reduction. First, many of the graphs obtainded from $\mathcal{D}$ are isomorphic if the edge color is ignored, and second, distinct directed graphs often correspond to isomorphic undirected graphs. Lastly, of course, there are graphs G and G' that are non isomorphic but $P(G) = P(G')$. Consider attaching two edges to a 4-cycle to obtain a 6-vertex graph. They can be attached to neighboring or to diagonally opposite vertices of the cycle, producing non-isomorphic graphs. However, in both cases the associated feature vectors will be the same.

3. Clustering analysis with TDA

For a data set $S \subset \mathbb{R}^n$, in our case corresponding to a set of directed graphs, a TDA analysis at dimension 0 gives rise to a hierarchy of connected components of (clustered) graphs as described below.

To understand the distribution of the points of S in $\mathbb{R}^n$ we use the notion of the neighborhood graph, as defined below, and construct a hierarchy of undirected graphs whose vertices are S. The neighborhood graph of S depends on a chosen distance function. In our case the distance d is the Euclidean distance between two points, that is, for $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ the distance $d(x, y)$ is $d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$.

Definition 3.1. Let S be a set of points in $\mathbb{R}^n$ and let $\epsilon \geq 0$ be a non-negative number. The *ϵ-neighborhood graph* is an undirected graph $\Gamma_\epsilon(S)$, where $\Gamma_\epsilon(S) = (S, E(\Gamma_\epsilon)$ and $E(\Gamma_\epsilon) = \{[u, v] \mid d(u, v) \leq \epsilon, u, v \in S, u \neq v\}$.

The clustering analysis is done by considering a sequence of neighborhood graphs $\Gamma_{\epsilon_1}(S), \Gamma_{\epsilon_2}(S), \ldots$ for $S \subset \mathbb{R}^n$ obtained by a sequence of incrementally increasing values $\epsilon_1 < \epsilon_2 < \cdots$.

Definition 3.2. A *cluster* of S at level ϵ is a connected component in the neighborhood graph $\Gamma_\epsilon(S)$.

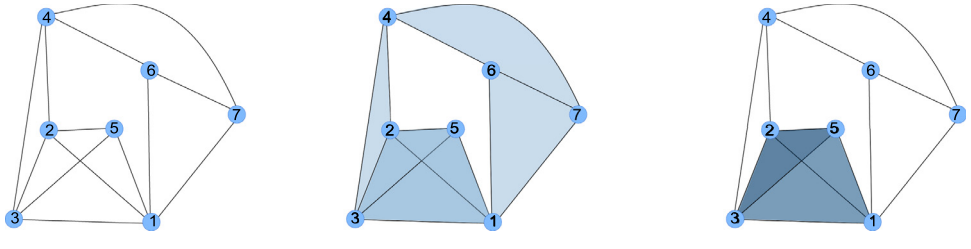

Fig. 7. The number of cliques associated with the vertex v_i, vertices ordered as in Fig. 6. Left: vector entries for cliques $k = 1$ (vertices) and $k = 2$ (incident edges) for each vertex. Middle: vector entries for clique $k = 3$ (three cycles). Right: vector entries for clique $k = 4$ (complete graph on four vertices, K_4). There is only one clique K_4, therefore only four vertices have entries 1 in the vector. This graph has no cliques of size higher than 4. The clique vector in this example is $P_{cq}(G) = \langle 11, 10, 10, 7, 8, 6, 6\rangle$ which is the sum of the four described vectors.

We observe some facts about the graph vectors $P(G)$ and $P_{gl}(G)$. Suppose $\mathcal{G}$ is a family of graphs and $S = S(\mathcal{G})$ and $S_{gl} = S_{gl}(\mathcal{G})$ are points in $\mathbb{R}^n$ obtained as described above. The vectors of the sets S and S_{gl} are part of the integer lattice of $\mathbb{R}^n$ and $\mathbb{R}^3$ respectively, therefore any distance between two distinct vectors is at least 1. The observations below indicate that small changes in the graphs can induce relatively large distances of the corresponding vectors in S.

Lemma 3.3. *Let $G, G' \in \mathcal{G}$. Then the following hold.*

(a) *If G' is obtained from G by addition of one vertex and one edge incident to that vertex. Then $d(P(G), P(G')) \geq 3$ and $d(P_{gl}(G), P_{gl}(G')) \geq \sqrt{2}$.*
(b) *If G' is obtained from G by addition of one directed edge without changing the total number of vertices, nor the number of cliques, then $d(P(G), P(G')) \geq \sqrt{3}$.*
(c) *If G' is obtained from G by addition of one edge that adds a clique to the graph $U(G')$ without changing the number of vertices, then $d(P(G), P(G')) \geq \sqrt{5}$.*
(d) *If G' is obtained from G by changing the target of one edge from vertex v to vertex v' without changing the number of the cliques, either $d(P(G), P(G')) \geq \sqrt{2}$ or $d(P(G), P(G')) = 0$.*

Proof. (a) The addition of a vertex in G' changes the number of non-zero entries in $P(G')$ in two places, once at $P_{val}(G')$ and again at $P_{cq}(G')$. Let w be the new vertex in G' added to $V(G)$ and let $[v, w]$ be the new edge in G' connecting $v \in V(G)$ with the new vertex w. Then w can be taken to be the last vertex in $V(G')$ in the order of the vertices, while the order of v in $V(G')$ might be either the same as its order in $V(G)$ or different. In both cases the entries in $P(G')$ corresponding to $|V(G')|$, $|E(G')|$, $val(v)$, $cq(v)$, $val(w)$ are at least one more than the corresponding entries in $P(G)$ and the entry of $cq(w)$ is at least two more (a 1-clique vertex w and a 2-clique the new edge) than the corresponding entry in $P(G)$ which is 0. So $d(P(G), P(G')) = \sqrt{\sum_i (x_i - y_i)^2} \geq \sqrt{5 + 2^2} \geq 3$, and $d(P_{gl}(G), P_{gl}(G')) \geq \sqrt{2}$.

The proofs of (b) and (c) follow a similar argument. Note that in case of (b), if the new directed edge is incident to vertices v and w, then because the number of cliques in $U(G')$ is not changed from the number of cliques in $U(G)$, there is an edge in G incident to v and w in opposite direction. So $P(G')$ has at least one more in the entries $|E(G)|$, $val(v)$ and $val(w)$. Observe that this may imply a change in the order of the vertices, in which case there may be a difference in the entries corresponding to the $cq(v)$ and $cq(w)$ which would increase the distance between the vectors. Therefore $d(P(G), P(G')) \geq \sqrt{3}$.

For the case of (c), the entries of $|E(G')|$, $val(v)$, $val(w)$, $cq(v)$, $cq(w)$ in vector $P(G')$ have a change of at least one and therefore the distance $d(P(G), P(G')) \geq \sqrt{5}$ and $d(P_{gl}(G), P_{gl}(G')) \geq \sqrt{1} = 1$. The case (d) follows the argument of (b) if the valencies change or, if valencies don't change, the graphs are represented by the same vectors and the distance is 0. □

3.1. Analyzing the data using neighborhood graphs

A *filtration of a graph* Γ is a sequence of nested graphs $\Gamma_1 \subseteq \Gamma_2 \subseteq \cdots \subseteq \Gamma_k = \Gamma$ where each Γ_i is a subgraph of Γ_{i+1}. The definition of the neighborhood graph for a point cloud S naturally induces a filtration for a connected graph with vertices S. Namely, given a point cloud $S \in \mathbb{R}^n$ and a finite sequence of non-negative numbers $0 = \epsilon_1 < \epsilon_2 < \cdots < \epsilon_k$, we obtain a filtration $\Gamma_{\epsilon_1}(S) \subseteq \Gamma_{\epsilon_2}(S) \subseteq \cdots \subseteq \Gamma_{\epsilon_k}(S)$. We assume that $\epsilon_1 = 0$, which implies that $E(\Gamma_{\epsilon_1}) = \emptyset$. This filtration also helps to extract the connected components (clusters) of S at various spatial resolutions. For a given ϵ, each connected component of $\Gamma_\epsilon(S)$ corresponds to a cluster of graphs whose corresponding points in $\mathbb{R}^n$ are connected by edges that are of lengths less than ϵ. This means that each graph associated to a vector in a cluster is at most ϵ apart from some other graphs within the same cluster, i.e., the corresponding graphs within the cluster have similar graph properties represented by in the vectors. To have a better information about the topological properties encoded in a filtration one usually considers *the persistence diagram* of the filtration. For our purpose, the persistence diagram describes a way the connected components of the neighborhood graph merge together as we increase the value of ϵ. The persistence diagram is also equivalently described by the persistence *barcode* (Ghrist, 2008b). The barcode construction is described as follows.

Let $S = S_{\mathcal{D}} \subset \mathbb{R}^n$, where $n = 2d + 3$ (in our case $n = 89$). In Figs. 8 and 12, the vertical axis enumerates points of S, and ϵ-values are listed on the horizontal axis. At $\epsilon_1 = 0$, $E(\Gamma_{\epsilon_1}) = \emptyset$, and each point of $S \subset \mathbb{R}^n$ forms a single connected component. There are $|S|$ connected components, and hence the number of bars in the barcode at value 0 is equal to the number of data points in S corresponding to the "birth" of all connected components. With appropriate increments of ϵ new edges are added to the neighborhood graph and the connected components start joining each other forming larger clusters. The merging event of connected components is represented by a termination of all but one of the corresponding bars of the barcode. The choice of the bar that does not terminate in a merge of components is arbitrary, and we use the established convention (see Ghrist (2008b)) where bars are vertically ordered by their length from the shortest at the bottom of the diagram to the longest on the top.

The number of connected components of the graph Γ_ϵ is the number of horizontal bars intersecting the vertical line at distance equal to ϵ. For instance, from Fig. 8 we deduce that the number of connected components in $\Gamma_\epsilon(S)$ is 2 for $\epsilon = 15$ indicating two clusters at that distance. Typically, the filtration ends with a neighborhood graph that has a single connected component. That is, the sequence of ϵ values increase from 0 to the value that gives rise to a single component graph. In the case of data $\mathcal{D}$ for the set of global vectors and the point clouds S and S_{gl}, the ϵ values range from 0 to 22 and 0 to 15 respectively.

3.2. Tree diagrams representing merging components

The merging events of connected components described in the persistence diagram can be encoded using a tree diagram called a *dendrogram* (Murtagh, 1983). The bottom points of the tree diagram correspond to the points of S (resp. S_{gl}), that also correspond to the connected components of $G_0(S)$. The vertical direction of the tree diagram represents values of ϵ.

At each level ϵ the connected components (clusters) are enumerated and each vertex in the tree is labeled by (i, ϵ) where i is an index that corresponds to the ith cluster of the graph at level ϵ. At each level ϵ, the number of nodes corresponds to the number of clusters of $\Gamma_\epsilon(S)$. For a node (vertex) v at level ϵ_k, the children of v correspond to the clusters at level ϵ_{k-1} (i.e., the connected components pf graph $\Gamma_{\epsilon_{k-1}}(S)$) that have joined to a single connected component represented by v in Γ_{ϵ_k}.

For a large enough value of ϵ, $\Gamma_\epsilon(S)$ is connected, and it corresponds to the single node (root) of the tree. The dendrograms corresponding to the persistent diagrams for S and S_{gl} are shown in Figs. 9 and 13 in the supplementary documentation, respectively.

3.3. Implementation

The point cloud generated from the data $\mathcal{D}$ was computed using a custom Python script. The persistence diagrams were generated using Javaplex (Tausz et al., 2014) and the dendrogram tree diagrams were generated using Mathematica

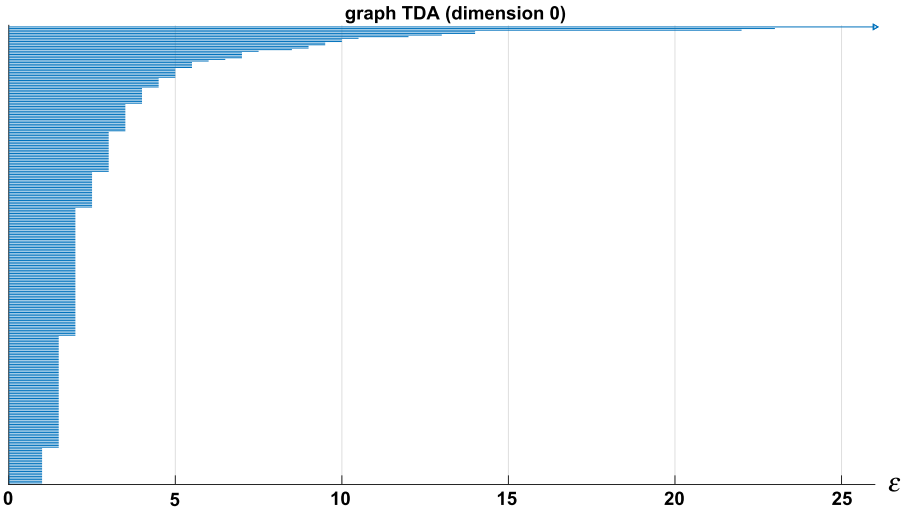

Fig. 8. The barcode diagram describing the birth and death of the connected components of the neighborhood graph of the dataset S. The horizontal axis represents increasing values of ϵ. Each horizontal bar represents a point in the point cloud. The horizontal line stops at the ϵ value when the corresponding point joins a connected component of Γ_ϵ. The number of horizontal bars intersecting a vertical line at a given ϵ value indicate the number of components in Γ_ϵ. The short bars at the bottom indicate that the corresponding points are merged into the same cluster at a small (< 2) value of ϵ. The vertical line at $\epsilon = 15$ intersects with two horizontal bars, indicating that there are two clusters at $\epsilon = 15$.

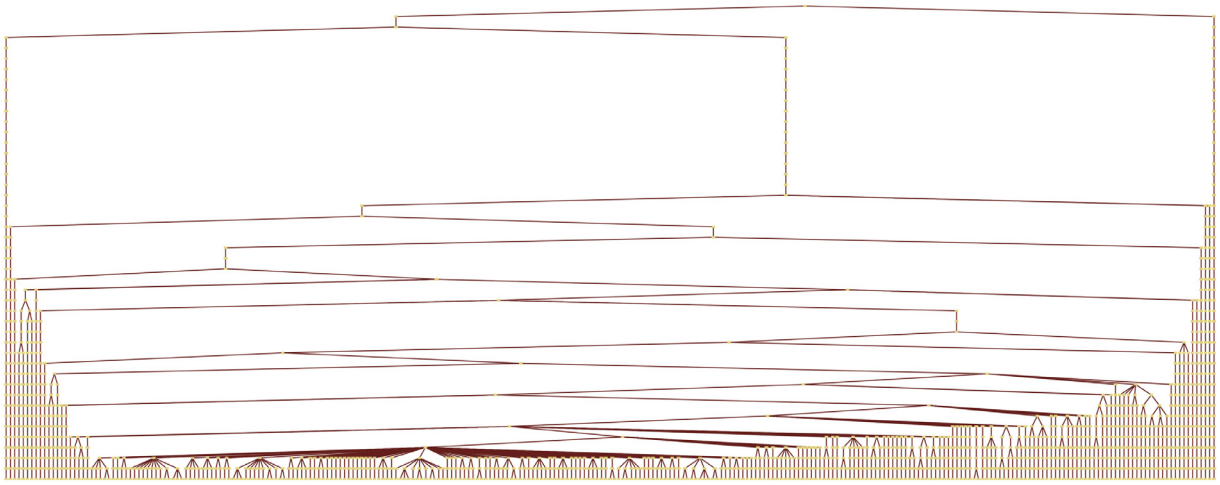

Fig. 9. The dendrogram clustering tree of dataset S.

(Wolfram Research, Inc., 2017). The sequence data, the graph data and the scripts are available at http://knot.math.usf.edu/data/GeneSegmentInteractions/dna_graph_study/.

4. Results

The analyzed data $\mathcal{D}$ consists of processed (Burns et al., 2016) micronuclear contigs obtained after sequencing of *O. trifallax* (Chen et al., 2014) as used in Burns et al. (2016). The directed graphs that correspond to the contigs in $\mathcal{D}$ can be found at http://knot.math.usf.edu/data/Colored_Graphs/index.html.

As mentioned, the data $\mathcal{D}$ produced 273 distinct vectors corresponding to $\mathcal{G} = \mathcal{G}_\mathcal{D}$ that raise to the same number of isomorphism classes of graphs ranging from 2 to 43 vertices. Each MIC contig corresponds to a vector in $S = S_\mathcal{D}$ while the MAC contigs whose MDS segments do not have any of the types 1, 2 or 3 interactions with MDSs of other contigs represent isolated vertices in the graphs and are not taken in consideration for the construction of $S_\mathcal{D}$.

We constructed filtration with ϵ increments of.5 in order to detect small neighborhood changes in the neighborhood graph, these sometimes are reflected by reorienting a directed edge.

4.1. Output of hierarchal clustering

The bar code diagram and the dendrogram for the filtration and clustering of the neighborhood graph of S are depicted in Fig. 8 and Fig. 9. As expected by Lemma 3.3, the neighborhood graph consists of isolated vertices for $\epsilon \leq 1$ and the first edges appear at $\epsilon = 1.5$ when there are 14 two point and 4 three point clusters. The two or three graphs joined at this distance differ from each other by small changes such as a single directed edge addition that does not change the cliques.

At $\epsilon = 2$, as noted in Lemma 3.3, most points remain distant from each other and only those representing graphs with small changes in their structure are joined by en edge. In addition two, three and in one instance four of the previously formed clusters join in (also with some additional points) to form new clusters, and there are 25 new small two or three point clusters. Most of the points in S remain as isolated vertices. At $\epsilon = 2.5$ a dramatic change occurs and one large cluster of 155 elements is formed with a second cluster of 5 points, and several small (two or three point) clusters. All other points stay as isolated vertices. At this point the feature of the point-cloud becomes clear, it consists of a single large cluster, singletons, and some small two or three el-

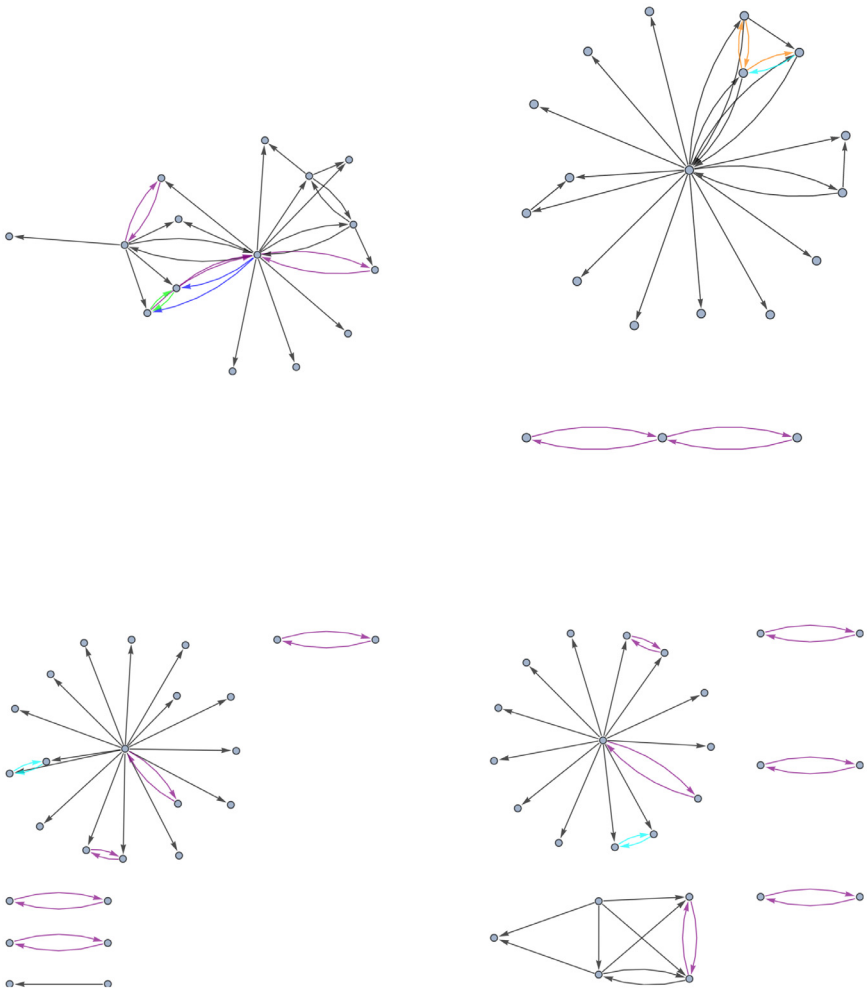

Fig. 10. Graphs for contigs ctg7180000088928 and ctg7180000088096 (top) and graphs for contigs ctg7180000067742 and ctg7180000067187 (down).

ement components. From Lemma 3.3 we can conclude that each graph in the 155 member cluster has a neighbor in the cluster that differs only by a vertex or an edge.

At $\epsilon = 9.5$, there is one large cluster of 269 points while the second largest cluster is of 4 elements, and there are 10 isolated points.

In the last 5 digits of contig numbers (see notation of the contigs in Chen et al. (2014)), the second largest cluster consists of contigs

88928, 88096, 67742, 67187.

Fig. 10 shows the graphs that are contained in this cluster.

All four of these graphs contain a 'star' vertex that is of high valency having multiple black (label (0,0,1)) outgoing edges. Such a 'star' vertex represents a MAC contig whose MDSs interleave with MDSs of many other MDS contigs. We observed that for most of the 'star' vertices, one IES contains all (or most of) MDSs of other contigs. This feature was further investigated in Braun et al. (2018), where the interleaving depth was considered.

The isolated points belong to 10 contigs

67761, 87162, 87484, 67363, 67280, 67243, 67157, 67223, 67417, 67411.

These are depicted in the corresponding figures in Supplementary Material Section. We note that some of these graphs have multiple 'star' vertices, or the component that contains a 'star' vertex also has additional cycles and cliques. In particular, the two graphs with 6-cliques (contigs 67243 and 67223) and the one with a 5-clique (contig 67411) are part of these isolated points. Furthermore, the graph with the longest path of 5 vertices (contig 87484) also featured in Braun et al. (2018) as one of the most in-depth embedding of genes within a single IES is also on this list. In all these cases we observe that the majority of the edges are black and purple, meaning that the prevailing inter-gene MDS organization is interleaving.

As ϵ increases, the four-element cluster becomes part of the large cluster at $\epsilon = 10.5$ and the isolated singleton points join the large cluster one or two at the time until $\epsilon = 14.5$ when the two, most distant contigs 67517 and 67223 remain isolated until $\epsilon = 22$ and $\epsilon = 23$ respectively.

The pattern of clusters for S_{gl} is similar to that of S. A large single cluster is formed at value $\epsilon = 1.5$, with 2 clusters of 5 elements, 3 clusters of 2 elements, and 23 singleton clusters.

At $\epsilon = 4.5$, the clusters consist of a large single cluster, the second largest of 9 elements, two clusters of two elements, and 5 singleton clusters. The size two clusters are {67417, 67243} and {67187, 67228}. The elements of the former cluster appear as isolated points in the neighborhood $\epsilon = 9.5$ of S, while 67187 of the latter cluster, appears in the 4-elements cluster of S, and 67228 is in the largest cluster of S.

The isolated points for $\epsilon = 4.5$ are 67223, 67363, 67157, 67280, 87484. We note that all these con-

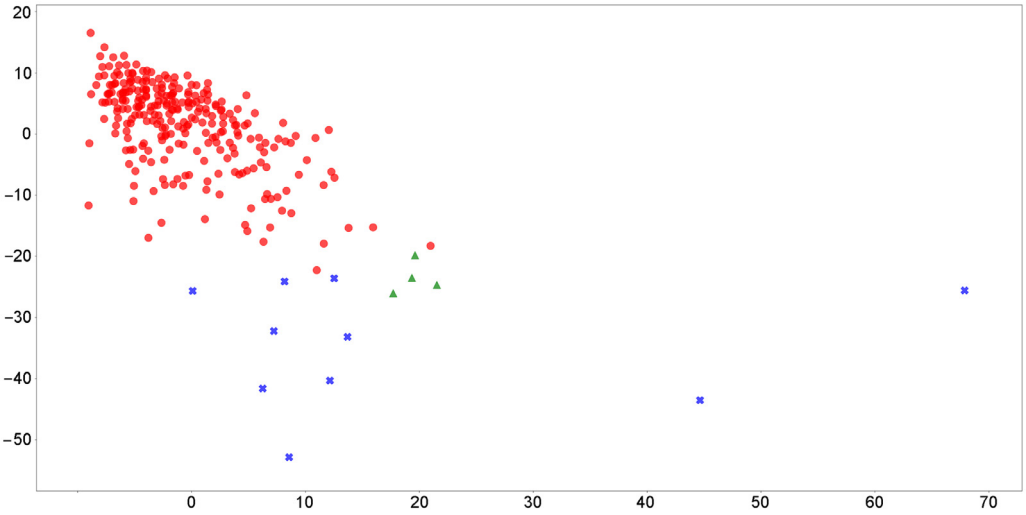

Fig. 11. A 2d multidimensional scaling projection for S as a visual depiction of the point cloud S. The points of S are colored according to clustering at $\epsilon = 9.5$. At this level we have 12 clusters, the largest cluster is colored red in the figure, the second largest cluster consists of 4 elements and is colored green and the singletons are all colored blue. To generate the 2d multidimensional scaling projection, we used the software implementation available in the Scikit-Learn Python Library (Pedregosa et al., 2011). Here and in figure 14 in SI there are no particular interpretations of coordinate axes (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

tigs also appear as isolated points of the neighborhood graph of S for $\epsilon = 9.5$.

In the case of S_{gl}, as in the case of S, the two most distant graphs correspond to the contigs 67517 and 67223 that join the large cluster at $\epsilon = 14.5$ and $\epsilon = 18.5$ respectfully.

Figs. 11 and 14 represent the 2d multidimensional scaling (MDS) projections (Kruskal and Wish, 1978) of the point clouds S and S_{gl}, respectively. Multidimensional scaling projections are commonly used to visualize a higher dimensional point cloud as points in the plane. The already developed technique (available in MathLab with toolbox *cmdscale*) projects the points of the point cloud in a plane such that mutual distances between the points are more or less preserved. Since our point cloud S is in $\mathbb{R}^{89}$, we used 2d multidimensional scaling projections to produce Figs. 11 and 14.

5. Discussion

In this paper we initiated a mathematical method of representing and analyzing inter-gene segment organization in a scrambled genome of *Oxytricha trifallax*. Although the whole genome was sequenced, such genome wide study for inter-gene segment arrangement has not been done before. The segment arrangements are represented by graphs representing their mutual relationship, such as overlapping and interleaving sequences. We analyzed the graph data by converting these graphs to a point cloud in a higher dimensional Euclidean space. In order to identify patterns in the graph structures, we applied hierarchical clustering methods borrowed from topological data analysis.

The big majority of segment organization within a single MIC contig are represented with small graphs up to five vertices (corresponding to the large cluster at $\epsilon = 2.5$) and one can 'move' from one graph to another by small vertex/edge changes. These small graphs constitute a major cluster. There are clusters consisting of singletons (represented by isolated points farther away from major clusters in Figs. 11 and 14) that correspond to MAC contigs with complex interaction patterns, and their patterns are often unique and rare. Most of inter-gene organizations involve only two or three MAC contigs, and interactions appear between two gene segments.

The most prevalent multi-gene segment organization in the *Oxytricha*'s genome are interleaving and often this appears as one gene interleaving with multiple other MAC contigs (as observed as the 'star' like vertices). For most of the 'star' vertices, one IES contains all (or most of) MDSs of other contigs. In the cluster consisting of four graphs, a single IES of the 'star' MAC contig interleaves with multiple MAC contigs. All star contigs are scrambled, which follows the analysis in Braun et al. (2018) where it was observed that contigs whose IESs interleave with other MAC contigs are mostly scrambled.

The graph representation of the inter-gene segment relationship introduced here is novel. We hope that a similar approach can be used in studies of scrambled genomes of other species, in particular those somewhat closely related to *O. trifallax*, such as *Tetmemena*. Comparisons among orthologous genes in other species with scrambled genomes may reveal whether patterns in these graph structures are conserved or abolished over evolutionary time. Furthermore, if genes with interleaved gene segments are co-expressed may indicate whether the rearrangement of these MAC segments are in parallel or sequential. We suggest that models that study gene rearrangement should also focus on operations that can be applied to these frequent interleaving gene segments, which in some cases resemble the odd-even patterns detected within scrambled genes (Burns et al., 2016).

The representation of the graph data into a point cloud in this paper is by a vector whose entries are common graph invariant properties, such as the number of vertices, edges and cliques. We used two vectors, one that had more local vertex properties and the other in $\mathbb{R}^3$ which included only the number of vertices, edges and the maximal clique. It is interesting that in both cases the isolated points are the same, and much distant from the rest of the points. The rearrangement process of the MIC contigs corresponding to these isolated points may indicate specific biological process that include multiple genes simultaneously. Studies that isolate intermediate DNA produced during the rearrangement may reveal the process in which they recombine. The graphs with large cliques (5 and 6) imply that segments of up to 5 or 6 genes mutually interleave and we suggest further experimentation to analyze rearrangement processes for these situations. In our study we did not consider the length of overlapping segments, nor the number of interleaving gene segments. Further methods can include edge weights on the graphs indicating size of overlaps and number of interleaving segments to give more detailed analysis.

Although this paper is focused on analysis of a specific data of interleaving/overlapping gene segments, the method that we propose for converting a graph data to a point cloud data is novel and general, and can be applied to analyze similarities in various graph data. We represented of graphs via a feature vectors in $\mathbb{R}^n$. Similar attempts in this direction have been made for detecting "graph similarities". In Gibert et al. (2012); Riesen and Bunke (2010); Hajij et al. (2018) the focus is on undirected graphs and the local properties that we used here are not considered. There are other avenues for developing a similarity measure between graphs (Bunke and Riesen, 2011; Papadimitriou et al., 2010), or graph kernels (Gärtner et al., 2003; Baur and Benkert, 2005), that we have not explored here. These methods often rely on the structural properties of the graph sometimes identified through topological methods. Such an approach may reveal other properties in the genome. For example, such methods have been successfully applied in protein function prediction (Borgwardt et al., 2005). Comparison of such graph analysis methods is subject of another study.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.jtbi.2020.110215.

CRediT authorship contribution statement

Mustafa Hajij: Formal analysis. **Nataša Jonoska:** Project administration, Formal analysis. **Denys Kukushkin:** Data curation. **Masahico Saito:** Project administration, Formal analysis.

References

Baur, M., Benkert, M., 2005. Network comparison. In: Network Analysis. Springer, pp. 318–340.

Beermann, S., 1977. The diminution of heterochromatic chromosomal segments in cyclops (crustacea, copepoda). Chromosoma 60 (4), 297–344.

Borgwardt, K.M., Ong, C.S., Schönauer, S., Vishwanathan, S., Smola, A.J., Kriegel, H.-P., 2005. Protein function prediction via graph kernels. Bioinformatics 21 (suppl 1), i47–i56.

Braun, J., Nabergall, L., Neme, R., Landweber, L.F., Saito, M., Jonoska, N., 2018. Russian doll genes and complex chromosome rearrangements in oxytricha trifallax. G3:Genes-Genomes-Genetics 8 (5), 1669–1674.

Bunke, H., Riesen, K., 2011. Recent advances in graph-based pattern recognition with applications in document analysis. Patt. Recognit. 44 (5), 1057–1067.

Burns, J., Kukushkin, D., Chen, X., Landweber, L.F., Saito, M., Jonoska, N., 2016. Recurring patterns among scrambled genes in the encrypted genome of the ciliate oxytricha trifallax. J. Theor. Biol. 410, 171–180.

Burns, J., Kukushkin, D., Lindblad, K., Chen, X., Jonoska, N., Landweber, L.F., 2015. <mds_ies_db>: a database of ciliate genome rearrangements. Nucl. Acid. Res. 44 (D1), D703–D709.

Carlsson, G., 2009. Topology and data. Bull. Am. Math. Soc. 46 (2), 255–308.

Chang, W.-J., Bryson, P.D., Liang, H., Shin, M.K., Landweber, L.F., 2005. The evolutionary origin of a complex scrambled gene. PNAS 102, 15149–15154.

Chen, X., Bracht, J.R., Goldman, A.D., Dolzhenko, E., Clay, D.M., Swart, E.C., Perlman, D.H., Doak, T.G., Stuart, A., Amemiya, C.T., Sebra, R.P., Landweber, L.F., 2014. The architecture of a scrambled genome reveals massive levels of genomic rearrangement during development. Cell 158 (5), 1187–1198.

Cheng, J., Sweredoski, M.J., Baldi, P., 2006. Dompro: protein domain prediction using profiles, secondary structure, relative solvent accessibility, and recursive neural networks. Data Min. Knowl. Discov. 13 (1), 1–10.

Dobzhansky, T., 1933. On the sterility of the interracial hybrids in drosophila pseudoobscura. Proceed. Natl. Acad. Sci. 19 (4), 397–403.

Edelsbrunner, H., Harer, J., 2008. Persistent homology – a survey. Contemp. Math. 453, 257–282.

Fang, W., Wang, X., Bracht, J.R., Nowacki, M., Landweber, L.F., 2012. Piwi-interacting RNAs protect DNA against loss during Oxytricha genome rearrangement. Cell 151 (6), 1243–1255.

Fernandez-Lozano, C., Fernández-Blanco, E., Dave, K., Pedreira, N., Gestal, M., Dorado, J., Munteanu, C.R., 2014. Improving enzyme regulatory protein classification by means of SVM-RFE feature selection. Mol. Biosyst. 10 (5), 1063–1071.

Gärtner, T., Flach, P., Wrobel, S., 2003. On graph kernels: Hardness results and efficient alternatives. In: Learning Theory and Kernel Machines. Springer, pp. 129–143.

Ghrist, R., 2008. Barcodes: the persistent topology of data. Bull. Am. Math. Soc. 45 (1), 61–75.

Ghrist, R., 2008. Barcodes: The persistent topology of data. Bull. Am. Math. Soc. 45, 61–75.

Gibert, J., Valveny, E., Bunke, H., 2012. Graph embedding in vector spaces by node attribute statistics. Patt. Recognit. 45 (9), 3072–3083.

Hajij, M., Wang, B., Scheidegger, C., Rosen, P., 2018. Visual detection of structural changes in time-varying graphs using persistent homology. IEEE Pacific Visual. Symp. (PacificVis) 125–134.

Haussmann, I.U., Bodi, Z., Sanchez-Moran, E., Mongan, N.P., Archer, N., Fray, R.G., Soller, M., 2016. m6a potentiates Sxl alternative pre-mrna splicing for robust drosophila sex determination. Nature 540 (7632), 301–304.

Wolfram Research, Inc., 2017. Mathematica, Version 11.2 Wolfram Research, Inc., Champaign, Illinois https://support.wolfram.com/41360.

Kruskal, J.B., Wish, M., 1978. Multidimensional scaling, 11. Sage.

Meinicke, P., 2015. Uproc: tools for ultra-fast protein domain classification. Bioinformatics 31 (9), 1382–1388.

Murtagh, F., 1983. A survey of recent advances in hierarchical clustering algorithms. Comput. J. 26 (4), 354–359.

Papadimitriou, P., Dasdan, A., Garcia-Molina, H., 2010. Web graph similarity for anomaly detection. J. Internet Serv. Applic. 1 (1), 19–30.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Prescott, D.M., 1994. The dna of ciliated protozoa. Microbiol. Rev. 58 (2), 233–267.

Rieseberg, L.H., 2001. Chromosomal rearrangements and speciation. Trend Ecol. Evol. 16 (7), 351–358.

Riesen, K., Bunke, H., 2010. Graph Classification and Clustering based on Vector Space Embedding. World Scientific.

Shibata, Y., Kumar, P., Layer, R., Willcox, S., Gagan, J.R., Griffith, J.D., Dutta, A., 2012. Extrachromosomal microDNAs and chromosomal microdeletions in normal tissues. Science 336 (6077), 82–86.

Smith, J.J., Baker, C., Eichler, E.E., Amemiya, C.T., 2012. Genetic consequences of programmed genome rearrangement. Curr. Biol. 22 (16), 1524–1529.

Tausz, A., Vejdemo-Johansson, M., Adams, H., 2014. JavaPlex: A research software package for persistent (co)homology. In: Hong, H., Yap, C. (Eds.), Proceedings of ICMS 2014, pp. 129–136. Software available at http://appliedtopology.github.io/javaplex/.

Tonegawa, S., 1983. Somatic generation of antibody diversity. Nature 302 (5909), 575–581.