Sepinoud Azimi | Tero Harju | Miika Langille | Ion Petre | Vladimir Rogojin

Directed overlap-inclusion graphs as representations of ciliate genes

Turku Centre *for* Computer Science

TUCS Technical Report
No 1001, February 2011

Directed overlap-inclusion graphs as representations of ciliate genes

Sepinoud Azimi
Åbo Akademi University, Department of Information Technologies
Joukahaisenkatu 3-5 A, Turku 20520 Finland
sepinoud.azimi@abo.fi
Tero Harju
University of Turku, Department of Mathematics
tero.harju@utu.fi
Miika Langille
Åbo Akademi University, Department of Information Technologies
Joukahaisenkatu 3-5 A, Turku 20520 Finland
miika.langille@abo.fi
Ion Petre
Åbo Akademi University, Department of Information Technologies
Joukahaisenkatu 3-5 A, Turku 20520 Finland
ipetre@abo.fi
Vladimir Rogojin
University of Helsinki, Computational Systems Biology Laboratory
vladimir.rogojin@helsinki.fi

TUCS Technical Report
No 1001, February 2011

Abstract

The simple intramolecular model for gene assembly in ciliates consists of three molecular operations based on local DNA manipulations. It was shown to predict correctly the assembly of all currently known ciliate gene patterns. Mathematical models in terms of signed permutations and signed strings proved limited in capturing some of the combinatorial details of the simple gene assembly process. A different formalization in terms of overlap-inclusion graphs, recently introduced by Brijder and Hoogeboom, proved well-suited to describe two of the three operations of the model and their combinatorial properties. We introduce in this paper an extension of the framework of Brijder and Hoogeboom in terms of directed overlap-inclusion graphs where more of the linear structure of the ciliate genes is described. We investigate a number of combinatorial properties of these graphs, including a necessary property in terms of forbidden induced subgraphs.

Keywords: Directed overlap-inclusion graphs, gene assembly in Ciliates, simple operations

TUCS Laboratory
Computational Biomodeling Laboratory

1 Introduction

Ciliates form a large and old group of unicellular eukaryotes. One of their characteristics is that each ciliate contains two types of functionally different nuclei: the germline nuclei (micronuclei) and the somatic nuclei (the macronuclei), each having multiple copies. The genes are differently organized in the two types of nuclei: micronuclear genes are split into blocks (called MDSs), which are separated by noncoding blocks. The MDSs come in a shuffled order, some of them being also inverted. Each MDS M ends with a short sequence of nucleotides (called *pointer*) that has a second occurrence in the beginning of the MDS that should follow M in the orthodox order. Macronuclear genes have all the MDSs spliced together (or *assembled*) on their common pointers. During sexual reproduction, all macronuclei are destroyed and new macronuclei are formed starting from a copy of a micronucleus. During this process, micronuclear genes get transformed into macronuclear genes by having excised all noncoding blocks and assembling the MDSs in the orthodox order. The process is called *gene assembly* and has been subject to intense combinatorial and computational research in the last decade. We refer for details to [6], [1], and [23] and references therein.

Several molecular models were considered for the gene assembly process, see [1]. Among them is the simple intramolecular model introduced in [8]. Unlike the other models, the simple intramolecular model postulates that gene assembly takes place as a result of local interactions, where only neighboring MDSs are able to interact with each other. The model was shown in [15] to predict correctly the assembly of all currently known gene patterns, see the database discussed in [5] for an up-to-date list. The simple model was modeled mathematically as a sorting of signed permutations in [16], and as a string rewriting system in [3, 4]. Both formal frameworks turned out to be limited in capturing the details of the local interactions postulated by the simple model and made it difficult to characterize, e.g., all gene patterns that can be assembled through simple operations. A similar difficulty in the case of the general intramolecular model was overcome by extending the model to signed overlap graphs, see [6]. In the case of simple operations, signed overlap graphs seem however insufficient to capture unambiguously the information about the distance among various pointers and MDSs, a crucial ingredient in the very definition of the simple model. A partial solution was introduced in [2] where genes were modeled as signed overlap-inclusion graphs. However, only two of the three operations of the simple model could be modeled in this context.

In this paper we extend the graph framework of [2] and introduce *directed* signed overlap-inclusion graphs as a model for ciliate genes. We explore some of their basic properties in connection to the other modeling frameworks for ciliate genes: strings, overlap graphs, and overlap-inclusion graphs. We also prove a number of combinatorial results about the directed signed overlap-inclusion graphs such as a necessary property for these graphs in terms of forbidden induced subgraphs. Even though the difference with respect to the framework of [2] may seem relatively minor, in modeling the overlap relationship among pointer intervals as directed rather than undirected edges, the properties of the directed overlap-inclusion graphs are remarkably different. In particular, they are able to support defining all three operations of the simple intramolecular model, which was not possible in the framework of [2]. Due to lack of space, we only briefly discuss the modeling of the simple model operations in this new framework and rather focus in this paper on its combinatorial properties.

2 Preliminaries

We recall in this section some of the basic definitions we need throughout the paper. For more details we refer to [6].

2.1 Legal strings

For an alphabet Σ and two strings u, v over Σ, we say that v is a *scattered subsequence* of u if $u = a_1 a_2 \ldots a_n$ and $v = a_{i_1} a_{i_2} \ldots a_{i_k}$, for some $0 \leq k \leq n$, $1 \leq i_1 < \ldots < i_k \leq n$, and $a_j \in \Sigma$, for all $1 \leq j \leq n$.

Let $\Delta_k = \{2, 3, \ldots, k\}$ be an alphabet of *pointers*, $M = \{b, e\}$ a set of markers and $\Sigma_k = \Delta_k \cup M$, for some $k \geq 1$. Without risk of confusion, we will often omit the subscript k and simply write Σ instead of Σ_k. We denote by $\overline{\Sigma}_k = \{\overline{2}, \ldots, \overline{k}, \overline{b}, \overline{e}\}$ a signed copy of Σ_k and let $\Sigma_k^{\maltese} = (\Sigma_k \cup \overline{\Sigma}_k)^*$.

We say that a string u in $\Sigma_k^{\maltese}$ is *legal* if for any $a \in \Delta_k$, u contains either 0, or 2 occurrences from the set $\{a, \overline{a}\}$ and moreover, u contains exactly one occurrence from the set $\{b, \overline{b}\}$ and one occurrence from the set $\{e, \overline{e}\}$. If u contains occurrences from the set $\{a, \overline{a}\}$, for some $a \in \Sigma_k$, then we say that a occurs in u and denote it $a \in u$. We define the *domain* of u as $\mathrm{dom}(u) = \{a \in \Sigma_k \mid a \in u\}$.

Let $p \in \Sigma \cup \overline{\Sigma}$ and let $u \in \Sigma^{\maltese}$ be a legal string. If u contains both substrings p and $\overline{p}$ then p is said to be *positive* in u; otherwise, it is said to be *negative*. If $u = u_1 p' u_2 p'' u_3$, with $p', p'' \in \{p, \overline{p}\}$, then the *$p$-interval* of u is the substring u_2.

For any distinct $p, q \in u$, p and q have one of the following relations:

- p and q *overlap* if exactly one occurrence from $\{p, \overline{p}\}$ can be found in the q-interval of u. We denote the overlapping relation by $p \Rightarrow_u q$, if the first occurrence from $\{p, \overline{p}\}$ occurs in u before the first occurrence from $\{q, \overline{q}\}$ and we denote it by $q \Rightarrow_u p$ otherwise;
- q is *included* in p if the two occurrences from $\{q, \overline{q}\}$ are found within the p-interval. This relation is denoted by $p \rightarrow_u q$;
- p and q are *disjoint* if they do not overlap and neither is included in the other in u.

(a) (b)

Figure 1: (a) The overlap graph corresponding to actin I gene in Sterkiella nova, (b) The overlap-inclusion graph corresponding to actin I gene in Sterkiella nova.

For details on how to associate a legal string to a ciliate gene we refer to [6]. For example, the legal string corresponding to actin I gene in Sterkiella nova is $34456756789e\bar{3}\,\bar{2}b289$, see [6].

2.2 Overlap graphs

The overlap relationships of the pointers of a legal string can be presented through an *overlap graph* (also known as *interlacement graphs*, see, e.g., [11]. The overlap-graph based pointer reduction system was introduced in [10, 7]) to model gene assembly in ciliates through rewriting of overlap graphs. For a legal string u, its overlap graph $G = (V, \sigma, E)$ was defined as follows: $V = \text{dom}(u)$, $\sigma : V \to \{+, -\}$ is the signing of vertices from V (i.e., if $p \in V$ is positive in the corresponding string u, then $\sigma(p) = +$, otherwise $\sigma(p) = -$) and $E = \{\{p, q\} | p \Rightarrow_u q$ or $q \Rightarrow_u p\}$.

Example 1. The overlap graph corresponding to actin I gene in Sterkiella nova is shown in Figure 1(a).

2.3 Overlap-inclusion graphs

The overlap and the inclusion relations between the pointers of a legal string can be captured through *overlap-inclusion graphs* as defined in [2]. For a legal string u its overlap-inclusion graph G was defined as follows: $V = \text{dom}(u)$ and $E = \{\{p, q\} | p \Rightarrow_u q$ or $q \Rightarrow_u p)\} \bigcup \{(p, q) | p \to_u q\}$. In this way, for any pair of overlapping pointers $\{p, q\}$ in u there is an undirected edge in G between p and q, and for any pointer p whose interval includes in the interval of some pointer q, G has the edge $p \to_G q$ from p to q. Note that in [2], the authors used the reverse orientation for the inclusion edges.

Example 2. The overlap-inclusion graph corresponding to actin I gene in Sterkiella nova is shown in Figure 1(b).

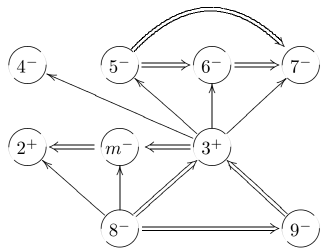


Figure 2: The directed overlap-inclusion graph corresponding to actin I gene in Sterkiella nova.

3 Directed overlap-inclusion graphs

We introduce in this section a new type of graph to represent the overlap and the inclusion relations among the pointers of a legal string. We extend the overlap-inclusion graph representation of micronuclear gene patterns introduced in [2]. The change we introduce is minimal: we substitute undirected edges which represent overlap relation between pointers with directed edges. This change is however enough to be able to define all three simple operations for gene assembly in ciliates on the level of graphs, a problem left (partially) open in [2]. Due to lack of space, we only focus in this paper on the properties of the directed overlap-inclusion graphs and only briefly discuss the graph-based modeling of the simple gene assembly operations.

3.1 Definitions and basic results

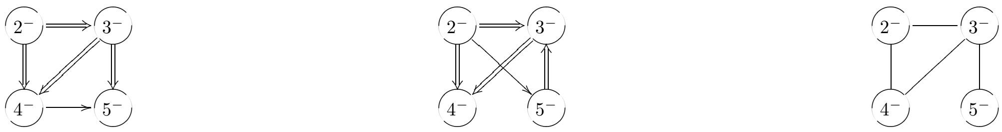
We define the directed overlap-inclusion graphs as follows.

Definition 1. Let u be a legal string. The directed overlap-inclusion (in short *DOI*) graph $G_u = (V, E_o, E_i, \sigma)$ corresponding to u is defined as follows: $V = \mathrm{dom}(u)$ is the set of vertices, $\sigma : V \to \{+, -\}$ is the signing of its vertices such that for each $p \in V$, $\sigma(p) = +$ if p is a positive pointer in u and $\sigma(p) = -$ otherwise. E_o and E_i are sets of its directed edges, $E_o = \{(p, q) | p \Rightarrow_u q\}$ and $E_i = \{(p, q) | p \rightarrow_u q\}$. For a *DOI* graph G and any string u such that $G = G_u$ we say that u corresponds to G.

Example 3. The *DOI* graph corresponding to actin I gene in Sterkiella nova is shown in Figure 2.

Example 4. Note that more than one string may correspond to a *DOI* graph, for example $u = 6224335546$ and $v = 6224553346$ have the same *DOI* graph.

Definition 2. Let G be a directed labeled graph with $\{+, -\}$ as vertex labels and $\{'overlap', 'inclusion'\}$ as edge labels. The *underlying digraph* of G is the graph



(G_1) (G_2) (G)

Figure 3: The *DOI* graphs G_1 and G_2 (corresponding to $u_1 = 23425354$ and $u_2 = 25354234$, resp.) have the same underlying overlap graph G.

obtained be removing edge labels, and the *underlying graph* of G is the graph obtained be removing edge labels and orientations.

We prove now several basic results about *DOI* graphs. The following result gives the connection between the directed overlap-inclusion graph of a string and its overlap graph. The result is straightforward to prove based on Definition 1.

Lemma 1. *Let G_u be the DOI graph corresponding to string u and G'_u the graph constructed from G_u by removing its inclusion edges, and replacing its directed overlap edges with undirected ones. Then G'_u is the overlap graph corresponding to u.*

Example 5. There are distinct *DOI* graphs having the same underlying overlap graph, see Figure 3.

Lemma 2. *Every induced subgraph of a DOI graph is a DOI graph.*

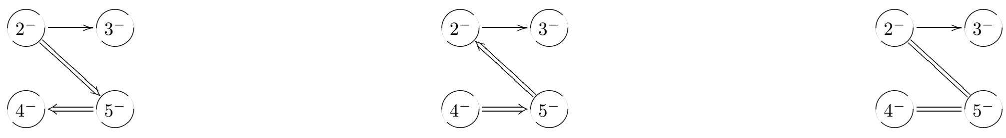
Proof. Let $G = (V, E_o, E_i, \sigma)$ be the *DOI* graph corresponding to u. Let G' be its induced subgraph on vertices $V' = \{v_1, v_2, ..., v_k\} \subseteq V$. Let u' be the string obtained from u by removing all occurrences of every $v \in V \setminus V'$. We claim that the *DOI* graph corresponding to u' is G'. Let p, q be two overlapping pointers in u' and $p \Rightarrow_{u'} q$. It follows that $p \Rightarrow_u q$ and so, $p \Rightarrow_G q$. Thus, since $p, q \in V'$, we have $p \Rightarrow_{G'} q$. Take now an overlap edge of G', $r \Rightarrow_{G'} s$. It follows that $r \Rightarrow_G s$ and, $r \Rightarrow_u s$. Since $r, s \in V'$, we obtain that $r \Rightarrow_{u'} s$ $\square$

Lemma 3. *Let $G_u = (V, E_o, E_i, \sigma)$ be the DOI graph corresponding to legal string u. Let E'_o be the set of undirected edges over V defined as follows:*

$$E'_o = \{\{p, q\} \mid p, q \in V \text{ and either } (p, q) \in E_o, \text{ or } (q, p) \in E_o\}.$$

Then the graph $H = (V, E'_o, E_i, \sigma)$ is the overlap-inclusion graph corresponding to u.

Example 6. There are distinct *DOI* graphs having the same underlying overlap-inclusion graph, see Figure 4.



(G_1) (G_2) (G)

Figure 4: The two *DOI* graphs G_1 and G_2 (corresponding to $u_1 = 23352454$ and $u_2 = 45425332$, resp.) have the same underlying overlap-inclusion graph G.

Theorem 4. *Any DOI graph G is a directed acyclic graph.*

Proof. Since the direction of an edge is always determined by the order in which the elements occur in the double occurrence string u, the *DOI* graph, G_u, corresponding to u is acyclic, i.e., there are are no directed cycles in G_u. $\square$

Corollary 5. *Any connected component of a DOI graph G, is* rooted*, i.e., the underlying digraph is acyclic and there exists exactly one vertex, called the* root *of G, of indegree zero.*

It turns out that the directed overlap relation between pointers establishes their order in any corresponding string.

Lemma 6. *Let G be a DOI graph that contains the following path* $s_1 \Rightarrow_G s_2 \Rightarrow_G \cdots \Rightarrow_G s_n$, $s_i \neq s_j$ *for* $i \neq j$*. Then the first occurrences of the pointers in string* u *corresponding to G appear in the order* $s_1 s_2 \cdots s_n$*. The same holds also for the sequence of their second occurrences.*

Proof. According to the definition of the overlap relation, if $s_i \Rightarrow_G s_{i+1}$, then in all strings u corresponding to G, $s_i s_{i+1} s_i s_{i+1}$ is a scattered subsequence of u, for all $1 \leq i \leq n-1$.

$\square$

Lemma 7. *Let G be a DOI graph. If* $s_1 \Rightarrow_G s_n$, $s_1 \Rightarrow_G s_2 \Rightarrow_G \ldots \Rightarrow_G s_n$*, then any legal string* u *corresponding to G has the following (scattered) subsequence:*

$$s_1 s_2 \cdots s_n s_1 s_2 \cdots s_n.$$

Proof. Let G be a *DOI* graph with edges $s_1 \Rightarrow_G s_n$, $s_1 \Rightarrow_G s_2 \Rightarrow_G \ldots \Rightarrow_G s_n$. Since $s_1 \Rightarrow_G s_n$, pointers s_1 and s_n occur in order $s_1 s_n s_1 s_n$ in any string corresponding to G. Since we have $s_1 \Rightarrow_G s_2 \Rightarrow_G \ldots \Rightarrow_G s_n$, by Lemma 6 pointers $s_1, s_2, \ldots, s_n$ occur in order $s_1 s_2 s_n s_1 s_2 s_n$. $\square$

Lemma 7 has the following additional implication.

Corollary 8. *Let G be a DOI graph. If* $s_1 \Rightarrow_G s_n$, $s_1 \Rightarrow_G s_2 \Rightarrow_G \ldots \Rightarrow_G s_n$*, then* $s_i \Rightarrow_G s_j$*, for all* $2 \leq i < j \leq n$.

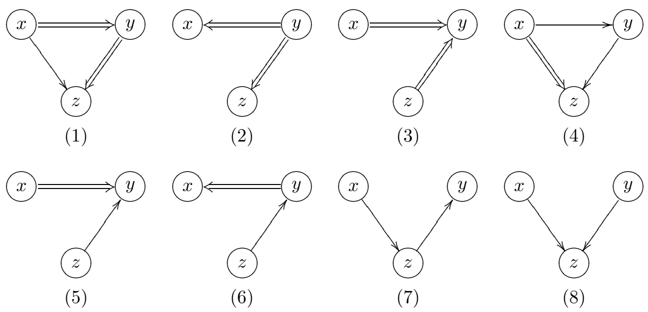


Figure 5: All 3-vertex, acyclic, forbidden graphs. Inclusion edges are illustrated as simple arrows and overlap edges as double arrows.

Proof. By Lemma 7 there is a scattered sequence of pointers

$$s_1 s_2 \cdots s_n s_1 s_2 \cdots s_n$$

in any string corresponding to G. In this way, for any i and j, $1 \leq i < j \leq n$ pointers s_i and s_j occur in order $s_i s_j s_i s_j$ in any string corresponding to G. Then $s_i \Rightarrow_G s_j$. $\square$

The following three results results correspond Lemma 6, Lemma 7 and Corollary 8 for inclusion edges.

Lemma 9. *Let G be a DOI graph that contains the following path* $s_1 \rightarrow_G s_2 \rightarrow_G \cdots \rightarrow_G s_n$, $s_i \neq s_j$ *for* $i \neq j$. *Then the first occurrences of the pointers in string* u *corresponding to G appear in the order* $s_1 s_2 \cdots s_n$. *the second occurrences of the pointers in string* u *corresponding to G appear in the order* $s_n s_{n-1} \cdots s_1$.

Lemma 10. *Let G be a DOI graph. If* $s_1 \rightarrow_G s_n$, $s_1 \rightarrow_G s_2 \rightarrow_G \ldots \rightarrow_G s_n$, *then any legal string* u *corresponding to G has the following (scattered) subsequence:*

$$s_1 s_2 \cdots s_n s_n s_{n-1} \cdots s_1.$$

Corollary 11. *Let G be a DOI graph.* $s_1 \rightarrow_G s_n$, $s_1 \rightarrow_G s_2 \rightarrow_G \ldots \rightarrow_G s_n$, *then* $s_i \rightarrow_G s_j$, *is an inclusion edge for all* $2 \leq i < j \leq n$.

3.2 Forbidden Subgraphs

In this section we introduce the concept of forbidden subgraphs of directed overlap-inclusion graphs.

Definition 3. Let G be a directed, vertex- and edge-labeled graph. We say that G is forbidden if there is no string u such that G is the *DOI* graph corresponding to u.

Definition 4. Let u be a legal string. If $u = a_1 a_2 \dots a_n$, then $u^R = a_n \dots a_2 a_1$ is the reversal of string u. If G is the *DOI* graph corresponding to legal string u, then G^R is the graph corresponding to u^R.

The following result is straightforward.

Lemma 12. *A minimal (in number of vertices) forbidden directed, vertex- and edge-labeled graph is connected, i.e., its underlying graphs is connected.*

Lemma 13. *For any DOI graph G, G^R is also a DOI graph.*

Theorem 14. *Let G be a directed labeled graph with $\{+,-\}$ as vertex labels and $\{'overlap','inclusion'\}$ as edge labels. If G is a 3-vertex, acyclic graph, then G is forbidden if and only if it is isomorphic to one of the graphs in Figure 5.*

Proof. Depending on the type of edges that G consists of, we consider the following cases:

i. three overlap edges;

ii. two overlap edges and one inclusion edge;

iii. two overlap edges;

iv. one overlap edge and two inclusion edges;

v. one overlap edge and one inclusion edge;

vi. one overlap edge;

vii. three inclusion edges;

viii. two inclusion edges;

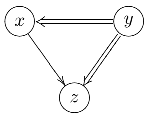
ix. one inclusion edge;

x. no edges.

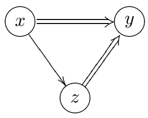
We discuss each case separately. Let $\{x, y, z\}$ be the vertices of G.

i. All acyclic graphs with three vertices and three overlap edges are isomorphic to the graph with $x \Rightarrow_G y$, $z \Rightarrow_G y$ and $z \Rightarrow_G x$. The string $zxyzxy$ corresponds to G, therefore, G is not forbidden.

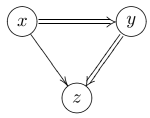
ii. It is straightforward to see that an acyclic graph with two overlap edges and one inclusion edge is isomorphic to one of the graphs in Figure 6. Graphs H_1 and H_2 are not forbidden: strings $yxzyzx$ and $xzyzxy$ correspond to them, respectively. In the case of H_3, let u be a string corresponding to it. Then $xyxy$ and $yzyz$ are scattered subsequences of u. Thus, $u = xyzxyz$ or $u = xyxzyz$. In neither of these strings does the x-interval include the z-interval.



(H_1)

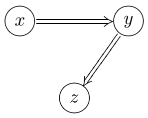


(H_2)

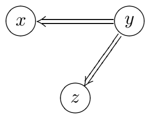


(H_3)

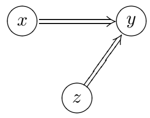
Figure 6: All graphs with two overlap edges and one inclusion edge of the type considered in Theorem 15.



(H_4)



(H_5)



(H_6)

Figure 7: All graphs with two overlap edges of the type considered in Theorem 15.

iii. The graph can only be isomorphic to one of the graphs in Figure 7. String $xyxzyz$ corresponds to H_4, so it is not forbidden.

Any string u corresponding to H_5 has $yxyx$ and $yzyz$ as scattered subsequences. Thus, there should be either an overlap, or inclusion relation between x and z. This is a contradiction.

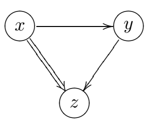
Any string u corresponding to H_6 has $xyxy$ and $zyzy$ as scattered subsequences. Thus, there should be either an overlap, or inclusion relation between x and z. This is a contradiction.

iv. The graph can only be isomorphic to one of the graphs in Figure 8. String corresponding to H_7 has $xyyx$ and $yzzy$ as scattered subsequences. Therefore, $u = xyzzyx$, contradicting $x \Rightarrow_G z$. Thus, H_7 is forbidden.

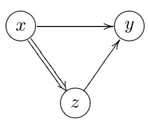
Strings $xzyyxz$ and $yxzxzy$ correspond to H_8 and H_9, respectively.

v. The graph can only be isomorphic to one of the graphs in Figure 9. The corresponding strings to H_{11} and H_{12} are $xyxzzy$ and $yzzxyx$ respectively, so they are not forbidden.

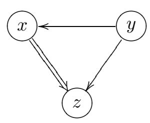
In the case of H_{10} and H_{13} we have $zyyz$ as a scattered subsequence of any corresponding string. Also x should have an occurrence in the y-interval. Consequently there must be an edge(of some kind and orientation) between x and z, a contradiction. Thus, H_{10} and H_{13} are forbidden.



(H_7)



(H_8)

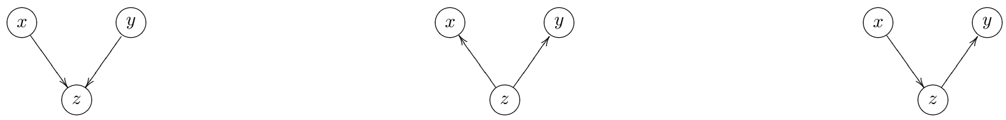


(H_9)

Figure 8: All graphs with one overlap edge and two inclusion edges of the type considered in Theorem 15.

(H_{10}) (H_{11}) (H_{12}) (H_{13})

Figure 9: All graphs with one overlap edge and one inclusion edge of the type considered in Theorem 15.



(H_{14}) (H_{15}) (H_{16})

Figure 10: All graphs with two inclusion edges of the type considered in Theorem 15.

vi. The graph is isomorphic to the *DOI* graph corresponding to $xyxyzz$ and thus, it is a *DOI* graph.

vii. All acyclic graphs with three vertices and three inclusion edges are isomorphic to the graph with $x \to_G y$, $y \to_G z$ and $x \to_G z$. The string $xyzzyx$ corresponds to G, therefore, G is not forbidden.

viii. The graph can only be isomorphic to one of the graphs in Figure 10. String $zxxyyz$ corresponds to H_{15}, so it is not forbidden.

For graph H_{14} we have scattered subsequences $xzzx$ and $yzzy$. On the other hand, the x-interval and the y-interval of u are disjoint. This is a contradiction so, H_{14} is forbidden.

For graph H_{16} we have scattered subsequences $xzzx$ and $zyyz$. On the other hand, the x-interval and the y-interval of u are disjoint. This is a contradiction so, H_{16} is forbidden.

ix. The graph is isomorphic to the *DOI* graph corresponding to $xyyxzz$ and thus, it is a *DOI* graph.

x. The graph is isomorphic to the *DOI* graph corresponding to $xxyyzz$ and thus, it is a *DOI* graph. □

Corollary 15. *A graph G with an induced 3-vertex subgraph isomorphic to one the graphs in Figure 5 is forbidden.*

Proof. This follows easily from Lemma 2 and Theorem 15. □

Example 7. The opposite direction of Corollary 16 is not generally true as it can be seen in Figure 11. By Theorem 15, none of the induced subgraphs of G is forbidden. On the other hand, G is forbidden. To prove it, assume that there is a string u corresponding to G. Then we have $xwzxwz$ as a scattered substring of u. Since $w \to_G y$, both occurrence of y come in the $x - interval$ of u. It

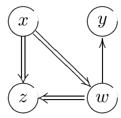
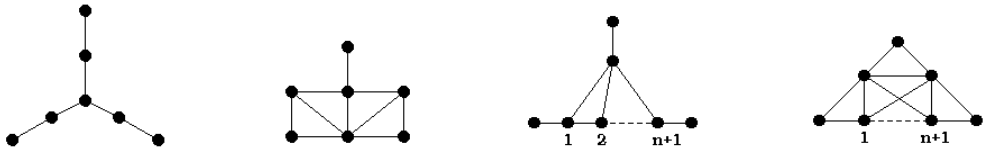


Figure 11: DF^4, an example of a forbidden 4-vertex *DOI* graph that has no forbidden 3-vertex induced subgraph.



(T_2) (X_{31}) (XF_2^{n+1}) (XF_3^n)

Figure 12: Four forbidden interval graphs.

follows now that there should exist edges of some kind between $\{y, z\}$ and $\{y, x\}$, a contradiction. We denote the graph in Figure 11 by DF^4.

Definition 5. An undirected graph G is called an *interval graph* if its vertices can be put into one-to-one correspondence with a set of intervals I of a linearly ordered set (like the real line) such that two vertices are connected by an edge of G if and only if their corresponding intervals have nonempty intersection [12].

It is straightforward to conclude the following lemma from the Definition 6.

Lemma 16. *Interval graphs are exactly the underlying graphs of DOI graphs. In other words, the DOI graphs are* edge-colored orientations *of interval graphs.*

The following result is a characterization of [22] of interval graphs in term of forbidden graphs.

Lemma 17. *Let G be a DOI graph. If the underlying graph of G has an induced subgraph isomorphic to either C_{n+4} (a directed cycle with $n \geq 0$) or one of the graphs in Figure 12, then G is forbidden.*

Let G be *DOI* graph and p, q two vertices of G. If $p \Rightarrow_G q$ or $p \rightarrow_G q$, then we write $p \rightsquigarrow_G q$.

Lemma 18. *A forbidden graph G of four or more vertices is rooted (with a unique root) or it contains a directed cycle C_n for some $n \geq 3$.*

Proof. Suppose that G is acyclic. Then the underlying digraph is acyclic, and it contains one or more vertices with indegree 0. Suppose there are two such vertices p and q. By Theorem 15, p and q do not have a common neighbour. Let t be a vertex such that there is a directed path from p to t and from q to t such that the sum of the lengths is minimal. Then there are vertices t_p and t_q with $t_p \rightsquigarrow_G t$ and $t_q \rightsquigarrow_G t$ such that t_p is on the path from p and t_q is on the path from q. By the forbidden triplets we must have also $t_p \rightsquigarrow_G t_q$ or $t_q \rightsquigarrow_G t_p$. However, this contradicts the minimality assumption, since now it should be $t = t_q$ or $t = t_p$. $\square$

Lemma 19. *Let G be a forbidden graph, and let p be its root. Then the digraph $G - p$ is connected, and there is a vertex q such that $p \Rightarrow_G q$.*

Proof. Let p be the unique root of G provided by Lemma 19, and let $A_1, \cdots, A_k$ be the connected components of the underlying graph of the *DOI* graph $G - p$. Suppose $k \geq 2$. By Lemma 22, the degree of p is at least two (and it has no incoming edges). Hence the subgraphs G_i induced by $A_i \cup \{p\}$ are *DOI* graphs, and thus $G_i = G(w_i)$ for some double occurrence string w_i.

If for all neighbors $q \in A_i$ of p, we have $p \rightarrow q$, then clearly we must have $w_i = pv_ip$, since the digraph induced by A_i is connected, and in this case $p \rightarrow q$ for all A_i. Moreover, if this holds for all i, then $w = pv_1v_2 \cdots v_kp$ is a double occurrence string such that $G = G(w)$; a contradiction on the choice of G. Hence there must be one component, say A_1, such that $p \Rightarrow_G q$ for some $q \in A_1$. Now $G_1 = G(w_1)$, where $w_1 = pu_1qu_2pu_3qu_4$ for some strings u_1, u_2, u_3, u_4. By the forbidden triplets, the index 1 is the only one with this property. Hence $p \rightarrow t$ for all $t \in \cup_{i=2}^k A_i$. Now $w = pv_2 \cdots v_ku_1qu_2pu_3qu_4$ satisfies $G = G(w)$; again a contradiction. $\square$

Lemma 20. *Let DF_m be a graph of order $m+4$ for $m \geq 1$ consisting of vertices p, q, s, t and $t_1, t_2, \ldots, t_m$ such that $t \Rightarrow q \Rightarrow s$, $t \Rightarrow t_1 \Rightarrow \ldots \Rightarrow t_m \Rightarrow s$, $q \rightarrow p$, $q \rightarrow t_i$ for each $i = 1, 2, \ldots, m$. DF_m is forbidden.*

Proof. Assume that there is a string u corresponding to DF_m. Then we have $tqt_1tt_2t_1t_3t_2t_3 \cdots t_{m-1}t_mt_{m-1}st_mqs$ as a scattered string of u. Since $q \rightarrow p$, both occurrences of p come within the q-interval, therefore there should be an edge of some kind between p and t, which is a contradiction. Thus, DF_m is forbidden. $\square$

Lemma 21. *Let G be a minimal forbidden graph with a vertex of degree one. Then G has one of the graphs from Figure 5, DF^4 or DF_m for some $m \geq 1$ as an induced subgraph.*

Proof. The graph G is connected, suppose $\deg(p) = 1$ and let q be the unique neighbor of p. Consider the graph $G - p$ where the vertex p is removed. By assumption of minimality, $G - p$ is a *DOI* graph, and hence there exists a double occurrence string $w = -q - q-$ such that $G - p$ is the *DOI* graph corresponding to w .

(1) If $p \rightarrow_G q$ is the only outgoing edge of p, then, let t be a neighbor of q different from p. Now, $\{p, q, t\}$ forms a forbidden triple as can be seen from Theorem 15.

(2) Let $p \leftarrow_G q$. Now, add pp after the first occurrence of q to obtain $w^{(0)} = -qpp - q-$. Since G is not a ***DOI*** graph, $G \neq G_{(w^{(0)})}$, and hence there must exist a vertex t in G such that pp belongs to the t-interval in $w^{(0)}$. This can happen only if $t \rightarrow_G q$ or $t \Rightarrow_G q$. The first option gives a forbidden nontransitive triple $t \rightarrow_G q \rightarrow_G p$ in G. Hence $t \Rightarrow_G q$.

Choose t such that its second occurrence is the last one with $t \Rightarrow_G q$. (It is in the q-interval.)

Then add pp in the original w after the second occurrence of t to obtain $w^{(1)} = -t - q - tpp - q-$. Again, since G is forbidden and therefore not corresponding to $w^{(1)}$, there exists a vertex t_1 in G such that pp belongs to the t_1-interval in $w^{(1)}$. If the second occurrence of t_1 is not in the q-interval, then we necessarily have the forbidden subgraph DF^4. Hence the second t_1 occurs between pp and the second q, and thus $q \rightarrow t_1$ and $t \Rightarrow t_1$ hold. Choose t_1 to be the last element with this property.

Consider the original w and replace the last t_1 by t_1pp. Once again, there exists a t_2 such that pp occurs in the t_2-interval. Now the t-interval and the t_2-interval must be disjoint by the choice of t and t_1. Hence the elements t_2 have two choices:

$$\begin{aligned}
&- t - q - t_1 - t - t_2 - t_1pp - t_2 - q- \,, \\
&- t - q - t_1 - t - t_2 - t_1pp - q - t_2 - \ .
\end{aligned}$$

The second one creates a forbidden DF_5 in G. In the first case, let t_2 be the last one with $t_1 \Rightarrow t_2$ and $q \rightarrow t_2$.

We proceed inductively. Let m be the first index for which $t_{m-1} \Rightarrow t_m$ and $q \rightarrow t_m$, and there exists an element an element s the second occurrence of s does not belong to the q-interval, and $t_m \Rightarrow s$ holds. (This s is obtained by considering the word $w^{(m)}$ obtained from w replacing the last occurrence of t_m by t_mpp.) Now $q \Rightarrow s$, since $s \rightarrow q$ would result to the forbidden induced subgraph $s \rightarrow q \rightarrow p$.The word w is now of the form

$$w = -t - q - t_1 - t - t_2 - t_1 - \cdots - s - t_m - q - s- \,,$$

which is the forbidden DF_m.

(3) Assume $p \Rightarrow_G q$. Then the forbidden triples yield that the indegree of q is one. Replace the first occurrence of q in w by pqp. Since G is not a ***DOI*** graph, there is a vertex t such that $t \Rightarrow_G q$ or $t \rightarrow_G q$. However, this not possible by the indegree of q.

(4) Assume $p \Leftarrow_G q$. This case is a dual case of (3), i.e., it reduces to (3) by considering the reverse string w^R of w. $\square$

4 Discussion

In this paper we proposed a new type of graph, directed overlap-inclusion graph, as a model for the pointer structure of ciliate genes. The main goal of introducing the model was to be able to investigate the combinatorial properties of the simple intramolecular model for gene assembly (such as characterizing the gene patterns that can be assemble through applications of simple operations), which was not possible in terms of permutations or string, and only partially possible in terms of overlap-inclusion graphs. In particular, all three operations of the simple model can easily be defined in terms of signed directed overlap-inclusion graphs as follows.

Let G be a *DOI* graph and p an arbitrary node of G. We denote by $inSet_i(p)$ the set of vertices with an inclusion edge ending in p. Moreover, $inDeg_i(p)$ is the number of vertices in $inSet_i(p)$. Similarly, we use $outSet_i(p)$ and $outDeg_i(p)$ to denote the set and number of vertices with an inclusion edge starting from p. We use the notation $inSet_o(p)$, $inDeg_o(p)$, $outSet_o(p)$, and $outDeg_o(p)$, resp. to denote the corresponding notions for overlap edges. adjacent to p.

For a *DOI* graph $G = (V, E_o, E_i, \sigma)$ and vertices $p, q \in V$ we define the following operations:

i. *The simple graph negative rule* sgn_p:

- sgn_p can be applied to G if $\sigma(p) = -$ and $inDeg_o(p) + outDeg_o(p) + outDeg_i(p) = 0$;
- If $G' = \mathsf{sgn}_p(G)$, then $V' = V \setminus \{p\}$, $\sigma'(r) = \sigma(r)$ for all $r \in V'$, $E'_o = E_o$ and $E'_i = E_i \setminus \{(q,p) | q \in inSet_i(p)\}$;

ii. *The simple graph positive rule* sgp_p:

- sgp_p can be applied to G if $\sigma(p) = +$, $inDeg_o(p) + outDeg_o(p) = 1$, and $outDeg_i(p) = 0$;
- If $G' = \mathsf{sgp}_p(G)$, then $V' = V \setminus \{p\}$, $\sigma'(r) = \sigma(r)$ for all $r \in V' \setminus \{q\}$, $\sigma'(q) = -\sigma(q)$, $E'_o = E_o \setminus \{(p,q), (q,p)\}$ and $E'_i = E_i \setminus \{(r,p) | r \in inDeg_i(p)\}$, where $inSet_o(p) \cup outSet_o(p) = \{q\}$;

iii. *The simple graph double rule* $\mathsf{sgd}_{p,q}$:

- $\mathsf{sgd}_{p,q}$ can be applied to G if $\sigma(p) = \sigma(q) = -$, $q \in outSet_o(p)$ and $inSet_o(p) \cup p = inSet_o(q)$, $outSet_o(p) = outSet_o(q) \cup q$, $inSet_i(p) = inSet_i(q)$ and $outSet_i(p) = outSet_i(q)$;
- If $G' = \mathsf{sgd}_{p,q}(G)$, then $V' = V \setminus \{p,q\}$, $\sigma'(r) = \sigma(r)$ for all $r \in V'$, $E'_o = E_o \setminus \{(p,q)\} \cup \{(p,s),(s,p),(t,q),(q,t) | s \in inSet_o(p) \cup outSet_o(p), t \in inSet_o(q) \cup outSet_o(q)\}$, $E'_i = E_i \setminus \{(p,s),(s,p),(t,q),(q,t) | s \in inSet_i(p) \cup outSet_i(p), t \in inSet_i(q) \cup outSet_i(q)\}$,

Example 8. Consider the *DOI* graph G corresponding to string $u = b234566e\bar{3}\,\bar{2}45$. Its *DOI* graph-based simple assembly is illustrated in Figure 13.

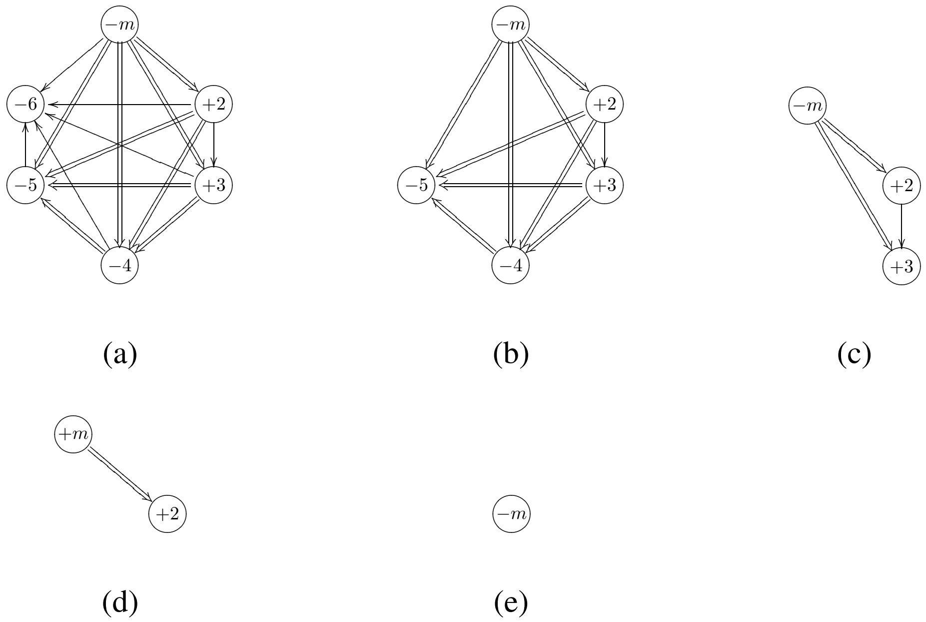


Figure 13: (a) G is the *DOI* graph corresponding to $u = b234566e\bar{3}\,\bar{2}45$, (b) $G' = \mathsf{sgn}_6(G)$ corresponds to $u' = b2345e\bar{3}\,\bar{2}45$, (c) $G'' = \mathsf{sgd}_{4,5}(G')$ corresponds to $u'' = b23e\bar{3}\,\bar{2}$, (d) $G''' = \mathsf{sgp}_3(G'')$ corresponds to $u''' = b2\bar{e}\bar{2}$, (e) $G^{iv} = \mathsf{sgp}_2(G''')$ corresponds to $u^{iv} = be$.

Due to lack of space we do not investigate in this paper the computational and combinatorial properties of the *DOI*-based model for simple gene assembly.

We proved in this paper that distinct signed double occurrence strings may have the same corresponding *DOI* graph. Characterizing all such strings corresponding to a given *DOI* graph remains however an open problem.

References

[1] Brijder, R., Harju, T., Jonoska, N., Petre, I., and Rozenberg, G., Gene assembly in ciliates. In: G. Rozenberg, T.H.W. Bck, J.N. Kok (Eds.): *Handbook of Natural Computing*, Springer, 2011, to appear.

[2] Brijder, R., and Hoogeboom, H.J., Combining overlap and containment for gene assembly in ciliates. *Theoretical Computer Science*, **411**(6), pp. 897–905. *doi:10.1016/j.tcs.2009.07.047*

[3] Brijder, R., Langille, M., Petre, I., A String-Based Model for Simple Gene Assembly. In: E. Csuhaj-Varju and Z. Ésik (Eds.): Proceedings of FCT 2007, Springer, Lecture Notes in Computer Science 4639, 161-172, 2007.

[4] Brijder, R., Langille, M., and Petre, I., Extended strings and graphs for simple gene assembly. Theoret Comp Sci **411**, 730-738, 2010.

[5] Cavalcanti, A., Clarke, T.H., Landweber, L., MDS_IES_DB: a database of macronuclear and micronuclear genes in spirotrichous ciliates. *Nucleic Acids Research* **33** (2005) D396–D398.

[6] Ehrenfeucht, A., Harju, T., Petre, I., Prescott, D. M., and Rozenberg, G., *Computation in Living Cells: Gene Assembly in Ciliates*, Springer (2003).

[7] Ehrenfeucht, A., Petre, I., Prescott, D. M., and Rozenberg, G., String and graph reduction systems for gene assembly in ciliates. *Math. Structures Comput. Sci.*, **12**, (2001), pp. 113–134.

[8] Ehrenfeucht, A., Petre, I., Prescott, D. M., and Rozenberg, G., Universal and simple operations for gene assembly in ciliates. In: V. Mitrana and C. Martin-Vide (eds.) *Words, Sequences, Languages: Where Computer Science, Biology and Linguistics Meet*, Kluwer Academic, Dortrecht, (2001) pp. 329–342.

[9] Ehrenfeucht, A., Prescott, D. M., and Rozenberg, G., Computational aspects of gene (un)scrambling in ciliates. In: L. F. Landweber, E. Winfree (eds.) *Evolution as Computation*, Springer, Berlin, Heidelberg, New York (2001) pp. 216–256.

[10] Ehrenfeucht, A., Harju, T., Petre, I., Prescott, D. M., and Rozenberg, G., Formal systems for gene assembly in ciliates. *Theoret. Comput. Sci.* **292** (2003) 199–219.

[11] de Fraysseix, H., and Ossona de Menzdez, P., A short proof of a Gauss problem. *Lecture Notes in Computer Science*, **1353**, (1997), pp. 230–235.

[12] Golumbic, M. C., Algorithmic Graph Theory and Perfect Graphs, *Academic Press*,(1980), ISBN 0-12-289260-7.

[13] Harju, T., Li, C, Petre, I., and Rozenberg, G., Complexity Measures for Gene Assembly In: K. Tuyls (Eds.), Proceedings of the Knowledge Discovery and Emergent Complexity in Bioinformatics workshop, Springer, Lecture Notes in Bioinformatics 4366, 42-60, 2007.

[14] Harju, T., Petre, I., and Rozenberg, G., Formal properties of gene assembly: Equivalence problem for overlap graphs. *Lecture Notes in Comput. Sci,* **2950** (2004) 202–212.

[15] Harju, T., Li, C, Petre, I., and Rozenberg, G., Modelling simple operations for gene assembly. In: Junghuei Chen, Natasha Jonoska, Grzegorz Rozenberg (Eds), Nanotechnology: Science and Computation, 361-376, Springer, 2006.

[16] Harju, T., Petre, I., Rogojin, V. and Rozenberg, G., Patterns of Simple Gene Assembly in Ciliates, *Discrete Applied Mathematics*, **156**(14), Elsevier, (2008), pp. 2581–2597.

[17] Harju, T., and Rozenberg, G., Computational processes in living cells: gene assembly in ciliates. *Lecure Notes in Comput. Sci.* **2450** (2003) 1–20.

[18] Landweber, L. F., and Kari, L., The evolution of cellular computing: Nature's solution to a computational problem. In: *Proceedings of the 4th DIMACS Meeting on DNA-Based Computers*, Philadelphia, PA (1998) pp. 3–15.

[19] Landweber, L. F., and Kari, L., Universal molecular computation in ciliates. In: L. F. Landweber and E. Winfree (eds.) *Evolution as Computation*, Springer, Berlin Heidelberg New York (2002).

[20] Langille, M., Petre, I., Simple gene assembly is deterministic. *Fundamenta Informaticae* **72** (2006) 1–12, IOS Press.

[21] Langille, M., Petre, I., Rogojin, V., Three models for gene assembly in ciliates: a comparison. Computer Science Journal of Moldova, **18** (1), 1-26, 2010.

[22] Lekkerkerker, C., Boland, J., Representation of a finite graph by a set of intervals on the real line, *Fundam. Math.* **51** (1962), 45–64

[23] Petre, I. and Rozenberg, G., Gene assembly in ciliates. Scholarpedia **5** (1), 9269, 2010.

[24] Prescott, D. M., Ehrenfeucht, A., and Rozenberg, G., Molecular operations for DNA processing in hypotrichous ciliates. *Europ. J. Protistology* **37** (2001) 241–260.

Turku Centre *for* Computer Science

Joukahaisenkatu 3-5 B, 20520 Turku, Finland | www.tucs.fi

University of Turku
- Department of Information Technology
- Department of Mathematics

Åbo Akademi University
- Department of Information Technologies

Turku School of Economics
- Institute of Information Systems Sciences

ISBN 978-952-12-2563-5
ISSN 1239-1891