Tree Loop Graphs ⋆

Liliana Alcón [a] Márcia R. Cerioli [b] Celina M. H. de Figueiredo [b]
Marisa Gutierrez [a] João Meidanis [c]

[a] *Departamento de Matemática, Universidad Nacional de La Plata, CC 172, (1900) La Plata, Argentina*

[b] *Instituto de Matemática and COPPE/Sistemas, Universidade Federal do Rio de Janeiro, Caixa Postal 68511, 21945-970, Rio de Janeiro, RJ, Brazil*

[c] *Scylla Bioinformatics and Instituto de Computação, Universidade Estadual de Campinas, Caixa Postal 6176, 13084-971, Campinas, SP, Brazil*

Abstract

Many problems involving DNA can be modeled by families of intervals. However, traditional interval graphs do not take into account the repeat structure of a DNA molecule. In the simplest case, one repeat with two copies, the underlying line can be seen as folded into a loop. We propose a new definition that respects repeats and define loop graphs as the intersection graphs of arcs of a loop. The class of loop graphs contains the class of interval graphs and the class of circular-arc graphs. Every loop graph has interval number 2. We characterize the trees that are loop graphs. The characterization yields a polynomial-time algorithm which given a tree decides whether it is a loop graph and, in the affirmative case, produces a loop representation for the tree.

Key words: interval graphs, interval number, computational molecular biology, DNA fragment assembly problem, DNA physical mapping.

⋆ Partially supported by Prosul/CNPq Proc. 690136/2003-0.
Email addresses: `liliana@mate.unlp.edu.ar` (Liliana Alcón), `cerioli@cos.ufrj.br` (Márcia R. Cerioli), `celina@cos.ufrj.br` (Celina M. H. de Figueiredo), `marisa@mate.unlp.edu.ar` (Marisa Gutierrez), `meidanis@ic.unicamp.br` (João Meidanis).

1 Introduction

Interval graphs are graphs that represent the intersections among a set of intervals in the real line. In many circumstances, a DNA molecule can be viewed as part of a real line, and contiguous fragments of the molecule can be seen as intervals in this line. As a result, many problems involving DNA can be modeled by interval graphs. For instance, problems related to fragment assembly and problems related to physical mapping of DNA are both amenable to modeling through interval graphs [10].

However, a DNA molecule is in fact a sequence. One consequence of this fact is the existence of *repeats*, which are long contiguous sections identical or almost identical to other sections in the same molecule. Repeats often bring additional challenges in many DNA-related problems. For instance, a direct repeat with three or more copies, or an inverted repeat, if long enough, introduce ambiguities in fragment assembly, so that extra information is needed to assemble a DNA stretch correctly [10]. In physical mapping, when probes are used to help find the relative positioning of long clones, there are efficient algorithms if the probes are unique. But if a probe falls into distinct copies of a repeat, it is not unique, and more sophisticated algorithms have to be used [10].

The repeat structure of a DNA sequence can be very complicated. For instance, take the 73 kilonucleotide sequence identified as HUMHBB stored in the public database GenBank [1] under accession number U01317.1 (or GI:455025). This sequence corresponds to a region on human chromosome 11 coding for several proteins involved with hemoglobin, the oxygen-carrier molecule of our blood red cells. The repeat structure of HUMHBB is shown in Figure 1.

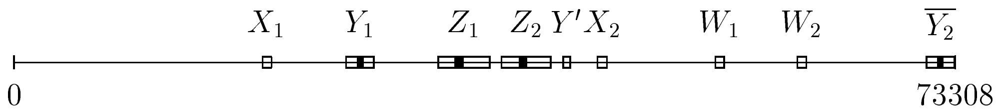

Fig. 1. Map of repeats for HUMHBB. Regions X_1 and X_2 share a 70% similarity. Regions Z_1 and Z_2 are 87% similar. Moreover, X_1 and X_2 are similar, at levels ranging from 68–73%, to a block occurring inside both Z_1 and Z_2, marked as a black box in the figure. The two occurrences of this block are almost identical (>99% similarity). The similarity between W_1 and W_2 is 89%. All these are direct repeats. There is an inverted repeat, represented here by Y_1 and Y_2, which share an 87% similarity. The region Y' is similar to blocks inside Y_1 and Y_2, with similarities around 80% among Y' and the two blocks, and between the blocks. The figure is drawn to scale. The entire sequence is 73308 base pairs long, and the sizes of the regions in base pairs are: region $X_1 = 606$, region $Z_1 = 3931$, region $Z_2 = 3766$, region $X_2 = 678$, region $W_1 = 593$, region $W_2 = 602$, region $Y_1 = 2096$, region $Y' = 462$, and region $Y_2 = 2101$.

Most DNA problems come in the following form: find an interval model compatible with the adjacencies revealed by some kind of sequence comparison or lab experiment. In general, repeats cause problems because they introduce additional adjacencies, which make interval graphs look like noninterval graphs. However, DNA molecules with repeats can be modeled by families of intervals if we modify the "adjacency" definition for interval graphs. Meidanis and Takaki [8] adjust the adjacency definition to suit DNA assembly. Here we adopt an alternative definition, more suitable for physical mapping problems.

We begin our investigation in this line of research by attacking first the simplest case: one repeat with exactly two copies. We looked into the recognition problem for this new class of graphs, which are intersection graphs of families of intervals, but where the adjacency relation takes the repeats into account. Since the intervals can be seen as continuous lines over a topological entity that resembles a loop, we call them *loop graphs*. However, we were not able to solve this recognition problem fully. The class of loop graphs properly contains circular-arc graphs and is properly contained in the class of two-interval graphs (intersection graphs of sets that are unions of at most two intervals). The characterization of circular-arc graphs by a family of forbidden induced subgraphs is a known open problem in intersection graph theory. The recognition of two-interval graphs is a difficult problem. In this paper, we solve the recognition problem for *tree loop graphs*, that is, loop graphs that are trees. The solution is based on a characterization of tree loop graphs that yields both its family of forbidden minimal induced subgraphs and a polynomial-time recognition algorithm.

Section 2 contains definitions of graph classes and inclusion relations that motivate the proposed characterization of tree loop graphs. Section 3 contains some general results about loop graphs that will be useful in the proof of the main theorem. Section 4 contains the statement and the proof of the main result. In Section 5 we present our final remarks concerning the corresponding recognition algorithm for tree loop graphs.

2 Definitions

A graph G is called an *interval graph* if to each vertex u of G there is a closed interval I_u of the real line, so that distinct vertices u, v of G are adjacent if and only if $I_u \cap I_v \neq \emptyset$. Structural characterizations of interval graphs have been provided by Lekkerkerker and Boland [6] in terms of forbidden subgraphs, by Gilmore and Hoffman [4] in terms of transitive orientations, and by Fulkerson and Gross [3] in terms of matrices. The class of interval graphs has been much studied and has been generalized in many ways motivated by scheduling and allocation problems that arise when a graph is used to model constraints on interactions between components of a large scale system.

One generalization of interval graphs are *circular-arc graphs*, obtained by replacing the real line by a circle, and intervals by arcs on the circle. Another generalization is obtained by considering the *interval number* of G, denoted $i(G)$, as the smallest positive integer t such that for each vertex u of G there exists a subset S_u of the real line $\mathbb{R}$ which is the union of t (not necessarily disjoint) closed intervals of $\mathbb{R}$ and distinct vertices u, v of G are adjacent if and only if $S_u \cap S_v \neq \emptyset$. The family $\{S_v\}_{v \in V(G)}$ is called a *t-representation* of G. Thus interval graphs are precisely the graphs having interval number 1. Every graph G with n vertices has interval number $i(G) \leq n-1$, and thus $i(G)$ is well defined. Trotter and Harary [11] proved that every tree T satisfies $i(T) \leq 2$; Scheinerman and West [9] proved that every planar graph P satisfies $i(P) \leq 3$. In both cases, the bound was shown to be best possible.

A *loop* is a pair (A, B) of two closed intervals $A = [a_1, a_2]$ and $B = [b_1, b_2]$ of the real line such that $a_1 \leq a_2 < b_1 \leq b_2$ and $b_2 - b_1 = a_2 - a_1$. Denote $b_1 - a_1$ by ℓ. Given an interval $C = [c_1, c_2]$ we define $C + \ell = [c_1 + \ell, c_2 + \ell]$ and $C - \ell = [c_1 - \ell, c_2 - \ell]$.

A *loop representation* of a graph G consists of a loop together with a family of closed intervals of the real line $(A, B, \{I_v\}_{v \in V(G)})$ such that distinct vertices u, v of G are adjacent if and only if (i) $I_u \cap I_v \neq \emptyset$; or (ii) $((I_u \cap A) + \ell) \cap I_v \neq \emptyset$ or $((I_u \cap B) - \ell) \cap I_v \neq \emptyset$. A *loop graph* is a graph that admits a loop representation. Hence every induced subgraph of a loop graph is also a loop graph.

Figure 2 shows on the left a graph T that is not an interval graph, and on the right a loop representation of T. We represent the loop (A, B) by two vertical strips. Note that adjacent vertices 6 and 3 correspond to intervals I_6 and I_3 satisfying $((I_6 \cap A) + \ell) \cap I_3 \neq \emptyset$.

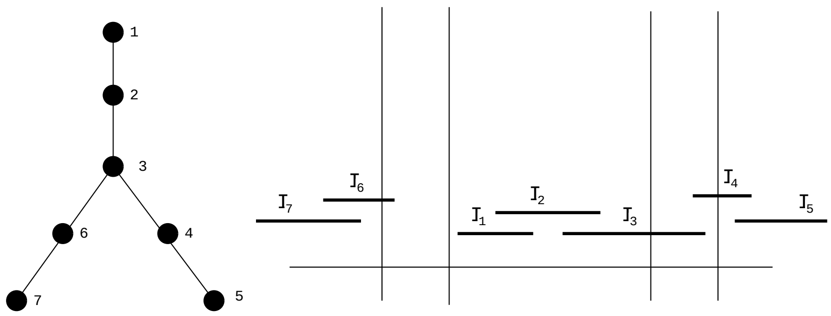

Fig. 2. A loop graph and a loop representation of it.

The next two results show that loop graphs are a limited generalization of interval graphs.

Lemma 1 *Every circular-arc graph is a loop graph.*

PROOF. Let G be a circular-arc graph and consider a representation of G in an oriented circle of radius 1, $\mathcal{F} = \{A_v\}_{v\in V(G)}$, where each arc A_v is given by its initial point a_v, $0 \le a_v < 2\pi$, and its length l_v, $0 < l_v < 2\pi$. Notice that if $a_v \le a_w$, then v and w are adjacent if and only if $a_v + l_v \ge a_w$ or $a_w + l_w - 2\pi \ge a_v$. We construct a loop representation for G as follows. Let $h = \max\{l_v : v \in V(G)\}$. Consider the loop with $A = [0, h]$ and $B = [2\pi, 2\pi + h]$, and the family of intervals $\{[a_v, a_v + l_v]\}_{v\in V(G)}$. In this loop representation, if v and w are adjacent, with $a_v \le a_w$, then $a_v + l_v \ge a_w$ or $a_w + l_w - 2\pi \ge a_v$. Conversely, if $a_v + l_v \ge a_w$ then, clearly, v and w are adjacent. And in the other case, if $a_v + l_v < a_w$ and $a_w + l_w - 2\pi \ge a_v$, then v and w are also adjacent because, in this situation, $2\pi \le a_w + l_w \le 2\pi + h$, since $l_w \le h$. $\square$

Lemma 2 *Every loop graph admits a 2-representation.*

PROOF. Let $(A, B, \mathcal{F})$ be a loop representation of G. Define a 2-representation of G as follows: Let $I_v = [x_v, y_v]$ be the interval of $\mathcal{F}$ corresponding to vertex v. If $I_v \cap (A \cup B) = \emptyset$ or, $I_v \cap A \ne \emptyset$ and $I_v \cap B \ne \emptyset$, then $S_v = I_v$. If $I_v \cap A \ne \emptyset$ but $I_v \cap B = \emptyset$, then $S_v = I_v \cup ((I_v \cap A) + \ell)$. If $I_v \cap B \ne \emptyset$ but $I_v \cap A = \emptyset$, then $S_v = I_v \cup ((I_v \cap B) - \ell)$. Now uv is an edge of G if and only if $S_u \cap S_v \ne \emptyset$. $\square$

West and Shmoys [13] showed that for a fixed value of $t \ge 2$ it is NP-complete to determine whether the graph has interval number at most t. Every tree [11] and, by Lemma 2, every loop graph admit a 2-representation. This motivates the study of tree loop graphs. Lemma 1 proves that loop graphs generalize circular-arc graphs, which can be recognized in polynomial time by a classical algorithm of Tucker [12], and by more recent algorithms of Eschen and Spinrad [2], and of McConnell [7]. The recognition problem for circular-arc graphs is harder than the recognition problem for interval graphs and other classes of intersection graphs. One reason is the absence of the Helly property in the families of arcs of a circle; this property is essential to construct a canonical representation. In other words, the consecutive clique arrangements of interval graphs do not generalize to circular clique arrangements. The difficulty of circular-arc graph recognition and the fact that no characterization of circular-arc graphs by a forbidden family is known motivate the study of recognition algorithms and forbidden families for related classes of graphs such as loop graphs.

3 More definitions and basic results

In what follows, we assume without loss of generality that the endpoints of any interval used in a loop representation are not a_1, a_2, b_1, or b_2 — the interval can always be lengthened to avoid this.

Definition 3 *Let (A, B) be a loop and let $I = [x, y]$ be an interval with* end-points *x and y. The* virtual part *I^* of I is the smallest union of closed intervals containing $((I \cap A) + \ell) \cup ((I \cap B) - \ell)) - I$.*

Definition 4 *Let (A, B) be a loop, I an interval, and $z \in \mathbb{R}$. The interval I* covers *the point z if $z \in I \cup I^*$.*

Clearly if I contains z then I covers z. Notice that distinct vertices u and v of a loop graph are adjacent if and only if I_u and I_v cover a common point of the real line, i.e., there exists $z \in \mathbb{R}$ such that $z \in (I_u \cup I_u^*) \cap (I_v \cup I_v^*)$.

Figure 3 illustrates all possible different positions of an interval with respect to a loop. For each interval I the virtual part I^* is represented. Notice that I^* consists of zero, one, or two closed intervals; we will refer to these intervals as the intervals of I^*. If $I \cap I^* \neq \emptyset$, then $I \cap I^*$ is $\{x\}$, $\{y\}$, or $\{x, y\}$, and the latter case implies that I^* consists of two intervals. In addition, if $I \cap I^* \neq \emptyset$ then $I \cup I^*$ is an interval of the real line containing the interval $[a_1, b_2]$. In Lemma 5 below, we show that the cases where $I \cap I^* \neq \emptyset$ (cases 4, 9 and 10 of Figure 3) may be omitted.

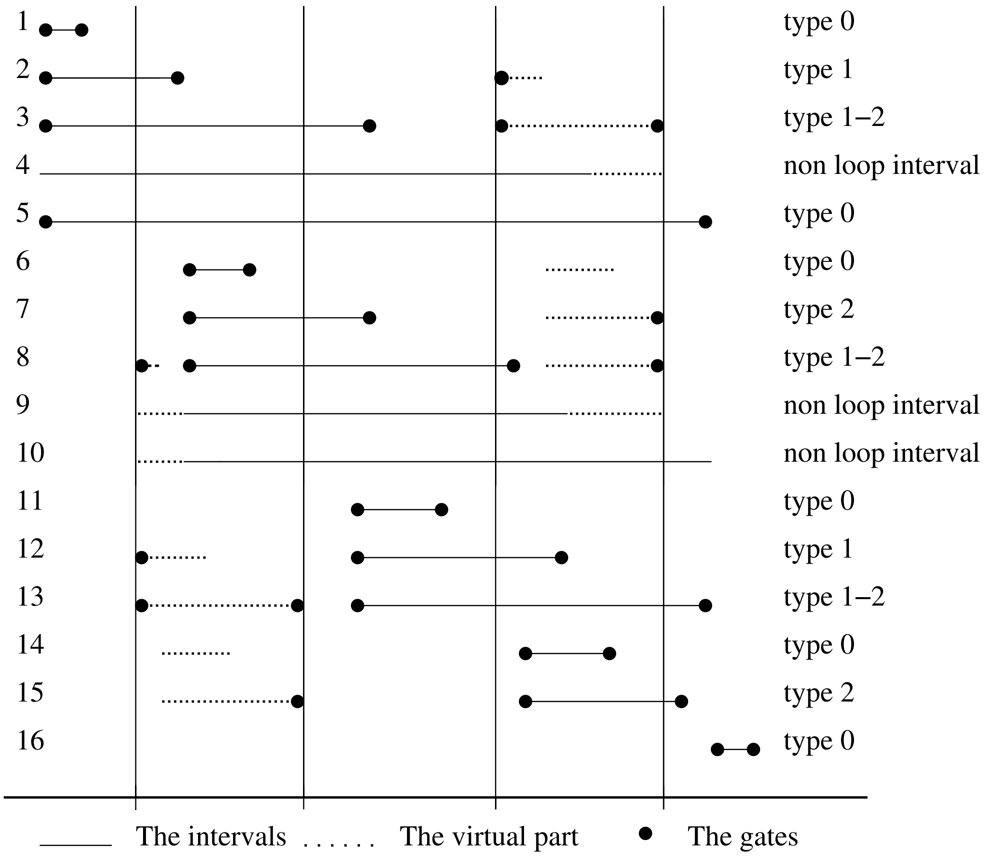

Fig. 3. Intervals in a loop.

Lemma 5 *Every loop graph admits a loop representation $(A, B, \mathcal{F})$ where every $I \in \mathcal{F}$ satisfies $I \cap I^* = \emptyset$.*

PROOF. Let I be an interval in $\mathcal{F}$ such that $I \cap I^* \neq \emptyset$. Define $J = I \cup I^*$. Since $I \cap I^* \neq \emptyset$, J is an interval containing $[a_1, b_2]$ and we have $J^* = \emptyset$ and $J \cap J^* = \emptyset$. Take $\mathcal{F}'$ consisting of the intervals in $\mathcal{F}$ except for I, which has been replaced by J. It follows that $(A, B, \mathcal{F}')$ is also a loop representation of G. Repeat the process until there is no interval I such that $I \cap I^* \neq \emptyset$. $\square$

Definition 6 *Let (A, B) be a loop. An interval $I = [x, y]$ is a* loop interval *if $x, y \notin \{a_1, a_2, b_1, b_2\}$ and $I \cap I^* = \emptyset$.*

In what follows, we assume without loss of generality that the intervals used in any loop representation are loop intervals. Notice that for such intervals $I^* = \emptyset$ (cases 1, 5, 11 and 16 of Figure 3) or $I^* = ((I \cap A) + \ell) \cup ((I \cap B) - \ell))$ (any other case of Figure 3).

Definition 7 *Let $I = [x, y]$ be a loop interval. The endpoints x and y of I are also called the* real gates *of I. The* virtual gates *of I are the points belonging to $I^* \cap \{a_1, a_2, b_1, b_2\}$. Refer to a point that is a real gate or a virtual gate of I simply as a* gate *of I.*

Definition 8 *A loop interval I is* type 0 *if I has no virtual gate. I is* type 1 *if I has exactly one virtual gate and it is a_1 or b_1. I is* type 2 *if I has exactly one virtual gate and it is a_2 or b_2. I is* type 1-2 *if I has exactly two virtual gates.*

Notice that if I is type 1-2, the virtual gates of I must be a_1 and a_2, or b_1 and b_2, or a_1 and b_2. Figure 3 also illustrates all the cases according to this classification of loop intervals into types.

The proof of the next three facts about gates are trivial. See Figure 3.

Fact 9 *Let I be a loop interval. The total number of gates of I is two, three or four.*

Fact 10 *Let I be a loop interval. If an interval J contains any endpoint of an interval of I^*, then J covers a gate of I. If J contains both endpoints of an interval of I^*, then J covers two different gates of I.*

Fact 11 *Let I be a loop interval not of type 0 and $x = a_1, a_2, b_1,$ or b_2. If $x \in I \cap \{a_1, a_2\}$, then $x + \ell$ is a gate of I; if $x \in I \cap \{b_1, b_2\}$, then $x - \ell$ is a gate of I.*

Definition 12 *Let v and u be adjacent vertices of a graph. Vertex v is* dominated by *vertex u if every vertex adjacent to v is also adjacent to u.*

In an interval representation, if two intervals I_u and I_v of the real line intersect, then one contains the two endpoints of the other, or each one contains one endpoint of the other. In addition, if I_v does not contain an endpoint of I_u, then v is dominated by u. In the sequel, we prove that in a loop representation analogous results are true for loop intervals, by replacing "contain" by "cover", and "endpoint" by "gate".

Lemma 13 *If two loop intervals cover a common point then one of the loop intervals covers two gates of the other, or each loop interval covers one gate of the other.*

PROOF. Let I and J be loop intervals covering a common point z. Hence, $z \in (I \cup I^*) \cap (J \cup J^*)$, and consider the following four possible cases:

Case i: If $z \in I \cap J$, then I contains the two endpoints of J, or J contains the two endpoints of I, or each loop interval contains one endpoint of the other. Thus the result follows because the endpoints of the intervals are gates, and *containing a point* implies *covering the point.*

Case ii: If $z \in I \cap J^*$, then there exists an interval J' of J^* such that $I \cap J' \neq \emptyset$. As in the previous case, J' contains the two endpoints of I, or I contains the two endpoints of J', or each interval contains one endpoint of the other. In all three cases the result follows because the endpoints of I are gates of I and by Fact 10.

Case iii: The case $z \in I^* \cap J$ is analogous to the previous case.

Case iv: Finally, if $z \in I^* \cap J^*$, then $z \in A$ (so $z + \ell \in I \cap J$) or $z \in B$ (so $z - \ell \in I \cap J$), and the result follows as in the first case. □

Corollary 14 *Let I and J be two loop intervals not of type 0. If I and J are both of the same type, or if one of them is type 1-2, then one of the loop intervals covers two gates of the other.*

Lemma 15 *Let u and v be adjacent vertices in a loop graph. If I_v does not cover a gate of I_u, then v is dominated by u.*

PROOF. Assume that

$$I_v \text{ covers no gate of } I_u. \tag{1}$$

We are going to prove that v is dominated by u. Since u and v are adjacent, let $z \in (I_u \cup I_u^*) \cap (I_v \cup I_v^*)$, and consider, as we have considered in the proof of Lemma 13, the following four possible cases:

Case i: $z \in I_u \cap I_v$. By hypothesis (1), we must have $I_v \subseteq I_u$, and so clearly v is dominated by u.

Case ii: $z \in I_u \cap J$, J an interval of I_v^*. By hypothesis (1) and Fact 10, we must have $J \subseteq I_u$. We consider two possibilities: $J \cap \{a_1, a_2, b_1, b_2\} = \emptyset$ or $J \cap \{a_1, a_2, b_1, b_2\} \neq \emptyset$. If $J \cap \{a_1, a_2, b_1, b_2\} = \emptyset$ (cases 6 or 14 of Figure 3), then $J = I_v^* \subseteq I_u$ and $I_v \subseteq I_u^*$, and so clearly v is dominated by u. If $J \cap \{a_1, a_2, b_1, b_2\} \neq \emptyset$, assume for instance that $a_1 \in J$. Then $a_1 \in I_u$. By Fact 11, I_u is of type 0 or $a_1 + \ell$ is a gate of I_u, but in this latter case I_v covers a gate of I_u, a contradiction. We conclude that I_u is type 0, and since it contains a virtual part, I_u must be as in case 5 of Figure 3. Now, clearly, $I_v \cup I_v^* \subseteq I_u$, and so v is dominated by u.

Case iii: $z \in J \cap I_v$, J an interval of I_u^*. By hypothesis (1), we must have $I_v \subseteq J \subseteq I_u^*$, which implies $I_v^* \subseteq I_u$, and so v is dominated by u.

Case iv: $z \in J' \cap J$, $J' \subseteq I_u^*$, and $J \subseteq I_v^*$. By hypothesis (1), we must have $J \subseteq J' \subseteq I_u^*$. As in case ii), by Fact 11, if $J \cap \{a_1, a_2, b_1, b_2\} \neq \emptyset$, then I_v covers a gate of I_u, a contradiction. Thus $J \cap \{a_1, a_2, b_1, b_2\} = \emptyset$, which implies $J = I_v^* \subseteq I_u^*$ and $I_v \subseteq I_u$, and so v is dominated by u. $\square$

A path $(v_1, \ldots, v_n)$ is an *induced* P_n in a graph G if $v_i \in V(G)$, and the only edges $v_i v_j \in E(G)$ are $v_i v_{i+1} \in E(G)$, for $1 \leq i < n$. We say that P_n has *length* $n - 1$. If $n = 2k - 1$, we say that v_k is the *central vertex* of P_n.

Corollary 16 *Let (u, v, w) be an induced P_3 in a loop graph G. Then I_v covers a gate of I_u in every loop representation of G.*

Definition 17 *Let G be an interval graph and v a vertex of G. Vertex v is* external *in G if there exists an interval representation D of G having $I_v = [x_v, y_v]$ such that x_v or y_v is not contained in another interval of D. In this case, interval I_v* has a free endpoint.

It is clear that if there is an interval representation of G having x_v as a free endpoint, then there exists another representation having y_v as a free endpoint.

A *tree* is a graph where each pair of vertices is connected by precisely one path. A *trivial* tree contains just one vertex. The following definition of ramification vertex is motivated by the forbidden induced tree for interval graphs, depicted in Figure 2. The trees that are interval graphs are precisely the trees with no ramification vertex [6].

Definition 18 *A vertex v in a tree T is a* ramification vertex *if its removal gives a graph $T - v$ containing at least three nontrivial trees. The* ramification degree *of a ramification vertex v is the number of nontrivial trees in $T - v$.*

Lemma 19 *Let T be a tree and an interval graph. A vertex of T is external if and only if*
(1) it is not the central vertex of an induced P_5 in T, and
(2) it is not a vertex dominated by the central vertex of an induced P_5.

PROOF. Note that an interval representation of T is also a loop representation. Let (u, v, w, v', u') be an induced P_5 in T. By Corollary 16, in any interval representation of T both I_v and $I_{v'}$ contain an endpoint of I_w. Since v and v' are not adjacent, each one of I_v and $I_{v'}$ contains a different endpoint of I_w. It follows that w is not an external vertex, which establishes (1). Now let z be a vertex dominated by w. Since T is a tree, z is neither adjacent to v nor to v'. Since we have just proved that in any interval representation of T, I_v contains one endpoint of I_w and $I_{v'}$ contains the other, it follows that in any interval representation of T, I_z must be contained in I_w. Hence I_z does not have a free endpoint, which establishes (2).

We prove the converse by induction on n, the number of vertices in T. The case $n = 1$ is trivial. Let T be a tree and an interval graph with $n > 1$ vertices, and v a vertex of T satisfying (1) and (2). Consider two cases:

Case a: Vertex v has a neighbor u of degree 1. Let T' be the tree and an interval graph $T - u$, obtained from T by the removal of vertex u. It is clear that v satisfies (1) with respect to T'. Vertex v satisfies (2) with respect to T', because if v is dominated by the central vertex of a P_5 in T', then in T there is a ramification vertex, which is a contradiction. It follows by the induction hypothesis that there exists an interval representation D' of T' such that the interval I_v has a free endpoint. An interval representation of T is obtained from D' by adding an interval I_u included in I_v.

Case b: Vertex v does not have a neighbor of degree 1. Since v satisfies (1), the degree of v is 1. Let u be the unique neighbor of v, and note that u dominates v. Let T' be the tree and an interval graph $T - v$. Now u must satisfy (1) and (2) with respect to T', and so by the induction hypothesis there exists an interval representation D' of T', such that the interval I_u has a free endpoint. An interval representation of T is obtained from D' by adding an interval I_v overlapping I_u, i.e., I_v and I_u intersect without either containing the other. It is clear that I_v has one free endpoint. $\square$

4 Trees that are loop graphs

In the present paper, the recognition of loop graphs is left as a problem. We solve the particular case of trees, characterizing the trees that are loop graphs: they must not have many ramification vertices and those vertices must not have large ramification degree.

The characterization leads to a polynomial-time algorithm that decides whether a given tree is a loop graph and, in the affirmative case, also gives a loop representation. Our result also leads to a characterization by forbidden induced subgraphs.

Theorem 20 *A tree T is a loop graph if and only if one of the following conditions holds:*

(i) T has no ramification vertex.
(ii) T has exactly one ramification vertex and it has ramification degree 3 or 4.
(iii) T has exactly two ramification vertices and they have ramification degree 3.

PROOF. The necessity is proved by the following three claims:

Claim 21 *If T has a vertex u with ramification degree 5 or more, then T is not a loop graph.*

PROOF. Let v_i, $i = 1, \ldots, 5$, be five vertices adjacent to u, such that each v_i belongs to a nontrivial tree of $T - u$. By Corollary 16, each I_{v_i} covers a gate of I_u, and these gates must be all different because the vertices v_i are not adjacent. Thus I_u must have at least five gates, which contradicts Fact 9. $\square$

Claim 22 *If T has two ramification vertices and one of them has ramification degree 4, then T is not a loop graph.*

PROOF. Let u and v be two ramification vertices, and assume u has ramification degree 4. Thus, by Corollary 16, I_u is type 1-2 and I_v is type 1, type 2, or type 1-2. Clearly u and v are adjacent and, by Corollary 14, I_u covers two gates of I_v or I_v covers two gates of I_u. Label by 1, 2, 3 the three neighbors of u that are distinct from and non adjacent to v, have degree at least 2, and are mutually non adjacent. Label by 4 and 5 two neighbors of v that are distinct from and non adjacent to u, have degree at least 2, and are mutually non adjacent.

Each of I_v, I_1, I_2, I_3 covers a distinct gate of I_u and so I_v covers a_1 or a_2 but not both; assume I_v covers a_2 (so it is type 2) and one of the intervals I_1, I_2, or I_3 covers a_1. Now, by Corollary 14, either I_v covers two different gates of I_u, which contradicts v not being adjacent to 1, 2, 3, or I_u covers two gates of I_v, and these two gates of I_v are distinct from the two gates covered by I_4, I_5, which says I_v is type 1-2, again a contradiction. $\square$

Claim 23 *If T has three ramification vertices with ramification degree 3, then T is not a loop graph.*

PROOF. Let u, v, w be three ramification vertices with ramification degree 3. Since these three vertices cannot be mutually adjacent, assume that u and w are not adjacent, and with no loss of generality assume further that I_u is type 1 and I_w is type 2. Suppose now that I_v is type 1. Then u and v are adjacent and Corollary 14 says that I_u covers two gates of I_v or I_v covers two gates of I_u. In either case we get a contradiction, since u and v are ramification vertices and their intervals I_u, I_v are assumed to have each one only three gates. We conclude that I_v must be type 1-2. Now Corollary 14 says that I_v covers two gates of I_u, which contradicts u being a ramification vertex and its interval I_u having only three gates, or I_u covers two gates of I_v. We conclude that I_u covers two gates of I_v, and I_w covers two gates of I_v. Note that those are the four distinct gates of the type 1-2 loop interval I_v as u and w are not adjacent. Finally, I_v has its four gates covered by I_u and I_w, which contradicts v being a ramification vertex. $\square$

The sufficiency is proved by the following two claims and by the fact that a tree without a ramification vertex is an interval graph and hence a loop graph.

Claim 24 *A tree with only one ramification vertex u and such that u has ramification degree 3 or 4 is a loop graph.*

PROOF. Without loss of generality, assume u is a vertex of ramification degree 4 in a tree T. Let T_i, $1 \leq i \leq 4$, be the nontrivial trees in $T - u$. Let v_i be the vertex of T_i that is adjacent to u in T. Since there is no other ramification vertex, T_i is an interval graph. In addition, v_i and T_i satisfy conditions (1) and (2) of Lemma 19. Hence there exists an interval representation D_i of T_i with I_{v_i} having a free endpoint. Define a loop representation $(A, B, \mathcal{F})$ of T as follows. Take a loop (A, B) with $A = [a_1, a_2]$ and $B = [b_1, b_2]$. Define $I_u = [x, y]$ with $x < a_1$ and $a_2 < y < b_1$. Since I_u is type 1-2, it has four gates which may be covered by each I_{v_i}, such that no other intersection occurs, as shown in Figure 4. The vertices of degree 1 adjacent to u in T (if they exist) can be

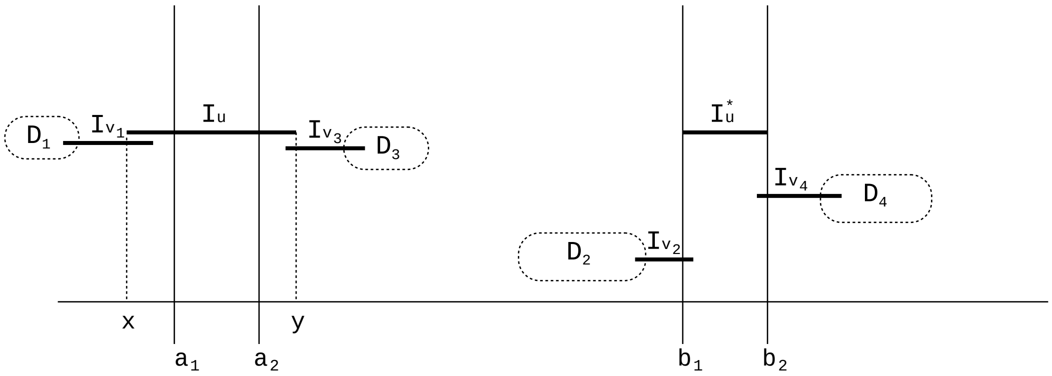

Fig. 4. A loop representation of a tree with a vertex of ramification degree4.

represented by a family of pairwise disjoint intervals appropriately included in $I_u \cap A$. $\square$

Claim 25 *A tree with only two ramification vertices v and w both having ramification degree 3 is a loop graph.*

PROOF. Let $(v = v_0, v_1, \dots, v_{p-1}, v_p = w)$ be the path between v and w in T. Let T' be the nontrivial tree obtained by identifying v and w into vertex u in the graph $T - \{v_1, \dots, v_{p-1}\}$. Let $(A, B, \mathcal{F})$ be the loop representation of T' obtained as in Claim 24. To obtain a loop representation of T it is enough to break the interval I_u into $p+1$ intervals I_{v_i}, $i = 0, \dots, p$, corresponding to an interval representation of the path $(v_0, v_1, \dots, v_p)$. The vertices of degree 1 adjacent to some v_i in this path can be represented by a family of pairwise disjoint intervals included in the respective I_{v_i}. $\square$

This concludes the proof of Theorem 20. $\square$

The following corollary is obtained by examining the adjacencies of the ramification vertices in a tree that is not a loop graph.

Corollary 26 *A tree is a loop graph if and only if it does not contain an induced subgraph isomorphic to the graph of Figure 5.*

5 Final remarks

Both the characterization in terms of ramification vertices given in Theorem 20, and the characterization in terms of an infinite family of forbidden induced subgraphs given in Corollary 26 lead to polynomial-time recognition

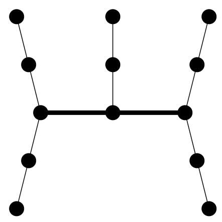

Fig. 5. The forbidden configuration for a tree loop graph. Bold lines are paths of any length.

algorithms for tree loop graphs. Given a tree, we can either look at its ramification vertices and their corresponding ramification degrees, or we can look for the forbidden configuration presented in Figure 5. Both tasks can be done in polynomial time. Moreover, when the tree is a loop graph, the proof of Theorem 20 constructs a corresponding loop representation in polynomial time.

Acknowledgements

We would like to express deep gratitude to the anonymous referees for their careful reading, and for the following two observations: Tree loop graphs are exactly trees with asteroidal number at most four [5]; Tree loop graphs can be recognized in linear time by testing whether the input graph is a tree, then by removing its leaves, and finally by checking whether the resulting tree has at most four leaves.

References

[1] D.A. Benson, I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, B.A. Rapp, and D.L. Wheeler. GenBank. *Nucleic Acids Res.*, 30:17–20, 2002.

[2] E.M. Eschen and J.P. Spinrad. An $O(n^2)$ algorithm for circular-arc graph recognition. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (Austin, TX, 1993)*, pages 128–137, New York, 1993. ACM.

[3] D.R. Fulkerson and O.A. Gross. Incidence matrices and interval graphs. *Pacific J. Math.*, 15:835–855, 1965.

[4] P.C. Gilmore and A.J. Hoffman. A characterization of comparability graphs and of interval graphs. *Canad. J. Math.*, 16:539–548, 1964.

[5] T. Kloks, D. Kratsch, and H. Müller. On the structure of graphs with bounded asteroidal number. *Graphs and Combinatorics*, 17:295–306, 2001.

[6] C.G. Lekkerkerker and J.C. Boland. Representation of a finite graph by a set of intervals on the real line. *Fund. Math.*, 51:45–64, 1962.

[7] R. McConnell. Linear-time recognition of circular-arc graphs. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, (Las Vegas, NV, 2001)*, pages 386– 395, New York, 2001. IEEE.

[8] J. Meidanis and P. Takaki. Interval graphs with repeats and the DNA fragment assembly problem. Technical Report 02-14, IC - Unicamp, 2002.

[9] E.R. Scheinerman and D.B. West. The interval number of a planar graph: three intervals suffice. *J. Combin. Theory Ser. B.*, 35:224–239, 1983.

[10] J.C. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology.* PWS, Boston, 1997.

[11] W.T. Trotter Jr and F. Harary. On double and multiple interval graphs. *J. Graph Theory*, 3:205–211, 1979.

[12] A. Tucker. An efficient test for circular-arc graphs. *SIAM J. Comput.*, 9:1–24, 1980.

[13] D.B. West and D.B. Shmoys. Recognizing graphs with fixed interval number is NP-complete. *Discrete Appl. Math.*, 8:295–305, 1984.