Matemática Contemporânea, Vol 39, 169-178
©2010, Sociedade Brasileira de Matemática

Gene clusters as intersections of powers of paths*

Vítor Costa   Simone Dantas
David Sankoff   Ximing Xu

Abstract

There are various definitions of a gene cluster determined by two genomes and methods for finding these clusters. However, there is little work on characterizing configurations of genes that are eligible to be a cluster according to a given definition. For example, given a set of genes in a genome is it always possible to find two genomes such that their intersection is exactly this cluster?

In one version of this problem, we make use of the graph theory to reformulated it as follows: Given a graph G, does there exist two θ-powers of paths $G_S = (V_S, E_S)$ and $G_T = (V_T, E_T)$, such that $G_S \cap G_T$ contains G as an induced subgraph? In this work, we show an $\mathcal{O}(n^2)$ time algorithm that generates the smallest θ-powers of paths G_S and G_T (with respect to θ and the number of vertices n of G), when G a unit interval graph.

1 Introduction

Due to recent research on genetic mapping, a large amount of information is available and stored in databases of various research centers in the world. Processing these data, in order to obtain relevant biological conclusions, is one of the challenges in Biology. One way to structure these data is using comparison of genomes, i.e., the search for similarities and differences between two or more organisms. The central question of this paper proposes

*2000 *AMS Subject Classification.* 68R10, 05C75, 05C85.
Key Words and Phrases: power of a path, unit interval graph, genome, gene clusters.
*This research was supported by CNPq and FAPERJ.

a problem in this area by asking: given a set of genes in a genome, called *cluster* is it always possible to find two genomes such that their intersection is exactly this cluster? First, we show the modeling presented by Adam et al. [1] and Sankoff and Xu [7] which will be used in this paper.

A *marker* is a gene with a known location on a chromosome. Let V_X be the set of n markers in the genome X. These markers are partitioned among a number of total orders called *chromosomes.* For markers g and h in V_X on the same chromosome in X, let $gh \in E_X$ if the number of genes intervening between g and h in X is less than θ, where $\theta \geqslant 1$ is a fixed *neighbourhood parameter.* We call $G_X = (V_X, E_X)$ a *θ-adjacency graph* if its edges are determined by a neighbourhood parameter θ.

Consider the θ-adjacency graphs $G_S = (V_S, E_S)$ and $G_T = (V_T, E_T)$ with a non-null set of vertices in common $V_{ST} = V_S \cap V_T$. We say that a subset of $V \subseteq V_{ST}$ is a *generalized adjacency cluster* if it consists of the vertices of a maximal connected subgraph of $G_{ST} = (V_{ST}, E_S \cap E_T)$. We call $G = G_{ST}[V]$ the subgraph induced by set V.

Let $G = (V(G), E(G))$ be a graph with vertex set $V(G)$ and edge set $E(G)$, such that $|V(G)| = n$. Let v, $\bar{v} \in V(G)$. The *distance* between vertices v and $\bar{v}$, denoted by $d_G(v, \bar{v})$, is the number of edges in a shortest path between v and $\bar{v}$ in G. A *path* between two vertices v_0 and v_t of the graph G is a sequence of vertices $v_1, v_2, \ldots, v_t$ such that $v_i v_{i+1}$ is an edge of G, $1 \leqslant i \leqslant t{-}1$. Let P_n be a graph which is a path with n vertices. A *θ-power of a path P_{n_θ}*, denoted by $P^{\theta}_{n_\theta}$, $\theta > 0$, is graph such that: $V(P^{\theta}_{n_\theta}) = V(P_{n_\theta})$ and $E(P^{\theta}_{n_\theta}) = \{v\bar{v} : d_{P_{n_\theta}}(v, \bar{v}) \leqslant \theta \text{ with } v, \bar{v} \in V(P^{\theta}_{n_\theta})\}$. For the benefit of the reader, we denote the power of a path $P^{\theta}_{n_\theta}$ by P^{θ}. The definition of a chromosome with n_θ markers in a θ-adjacency graph is similar to a power of a path $P^{\theta}_{n_\theta}$. Now, the central question of this work can be reformulated as follows:

Question 1. *[2, 5] Given a connected graph G, does there exist two θ-powers of paths G_S and G_T, whose intersection contains G as an induced subgraph?*

If the answer is yes, we are also interested in finding the minimum value of power θ and number vertices n_θ for these two θ-powers of a path.

In this work, we consider the case when G is an unit interval graph. We say that G is an *unit interval graph* if there exists a family I of intervals (a, b) on the real line such that: each $v \in V(G)$ can be put in a one-to-one correspondence with $(a_v, b_v) \in I$; the intervals in I are of same length; and $v\bar{v}$ is a edge of $E(G)$ if, and only if, $(a_v, b_v) \cap (a_{\bar{v}}, b_{\bar{v}}) \neq \emptyset$. There exist linear-time recognition algorithms for unit interval graphs, for example Figueiredo et al. [4] and Corneil et al. [3].

Brandstädt et al. [2] and Lin et al. [5] proved independently the following structural property:

Theorem 1. *[2, 5] A graph G is an induced subgraph of a power of a path if, and only if, G is an unit interval graph.* ■

Thus, given an unit interval graph G, there exists a power of a path $P^\theta_{n_\theta}$ that contains G as an induced subgraph. But the proofs of the structural characterization given by Theorem 1 [2, 5] does not lead to an algorithm that constructs G_S and G_T for Question 1.

In this paper, we present an $\mathcal{O}(n^2)$ time algorithm that generates, from a connected unit interval graph G, the smallest θ-powers of paths G_S and G_T (with respect to θ and to number of the vertices n of G) whose intersection contains G as an induced subgraph. The proofs will be omitted in this extended abstract due to space.

2 The algorithm

Our result is based on the ordering of the vertex set of G given by Algorithm *Recognize* [3], which satisfies the property proved by Roberts in [6]: *"A graph G is an unit interval graph if and only if there is an order $<$ on vertices such that: for all vertices v, the closed neighborhood of v is a set of consecutive vertices with respect to the order $<$."* Since all powers of paths are unit interval graphs, we can insert the vertices of $V(G)$, in the vertex set of a power of a path $P^\theta_{n_\theta}$ until this power of a path contains G as an induced subgraph.

This construction is done as follows. First, we take $v_1 < v_2 < \ldots < v_n$ an ordering of $V(G)$ given by Algorithm *Recognize* [3]. We consider θ_0 as the number of vertices of the maximal clique, that contains v_1, minus one; and we insert the vertices of this clique in P^{θ_0}. The algorithm constructs a sequence of power of a paths $P^{\theta_0} \subset P^{\theta_1} \subset \ldots \subset P^{\theta_{l-1}} \subset P^{\theta_l}$ such that $\theta_i = \theta_{i-1} + 1$.

Let v be the first vertex non-adjacent to v_1 in the order on $V(G)$. If v is adjacent to v_2, the algorithm must insert v in the vertex of P^{θ_0} that is at distance $\theta_0 + 1$ from vertex v_1 in P^{θ_0}. Similarly, if v is not adjacent to v_t, but is adjacent to v_{t+1}, the algorithm must insert v in the vertex of P^{θ_0} that is at a distance $\theta_0 + 1$ from vertex v_t in P^{θ_0}. This is done by inserting $t - 1$ vertices between the vertex of largest index adjacent to v_1 and v in P^{θ_0}. Now, suppose that there exists at least two vertices v, $\bar{v}$ that are not adjacent to v_1 and adjacent to v_2. Let $\bar{v}$ be the second vertex of this set. In order to minimize the number of vertices of P^{θ_0}, vertex $\bar{v}$ must be a vertex of P^{θ_0} at distance $\theta_0 + 2$ of vertex v_1 in P^{θ_0}. Then, the algorithm must call Procedure *SHIFT* to increase θ_0 to $\theta_1 := \theta_0 + 1$ because of the edge $\bar{v}v_2$. On the other hand, this procedure adds several edges in P^{θ_0} which are not in $E(G)$. Thus, Procedure *SHIFT* adjusts the power of a path P^{θ_0} for the new θ_1, by inserting vertices in P^{θ_0} in order to preserve the adjacencies and non-adjacencies of the vertices of G and generates a new P^{θ_1}. Algorithm proceeds until all vertices of $V(G)$ are included in $P^{\theta}_{n_\theta}$, a smallest power of a path with respect to θ and n_θ.

Before describing the algorithm, we shall give some notations. Given an ordering of $V(G)$ returned by Algorithm *Recognize* [3], then $\text{order}_G(v)$ is the position of vertex v in the ordering of the vertex set of G; $\xi_G(v) = \max\{\text{order}_G(\overline{v}) : \overline{v} \in N_G[v]\}$ and $\eta_G(v) = \min\{\text{order}_G(\overline{v}) : \overline{v} \in N_G[v]\}$. Let $v \in V(G)$ and $v \in V(P^\theta)$. We refer to $\text{order}_{P^\theta}(v)$ as the position of vertex v in the ordering of the vertex set of P^θ, i.e., $\text{order}_{P^\theta}(v) = i$, if $u_i = v$ in P^θ. Given $u \in V(P^\theta)$, we denote $\xi_{P^\theta}(u) = \max\{\text{order}_{P^\theta}(\bar{u}) : \bar{u} \in N_{P^\theta}[u]\}$ and $\eta_{P^\theta}(u) = \min\{\text{order}_{P^\theta}(\bar{u}) : \bar{u} \in N_{P^\theta}[u]\}$.

Next, we present Algorithm *CPP* and Procedure *SHIFT*.

Algorithm. *CONSTRUCTING_POWER_OF_PATH (CPP)*

- *Input: a connected unit interval graph G and an ordering of $V(G)$, $v_1 < \ldots < v_n$, given by Algorithm* Recognize *[3].*
- *Output: a smallest power of a path $P^{\theta}_{n_\theta}$, with respect to θ and to number of the vertices n_θ, which contains G as an induced subgraph.*

```
1. $\theta := \xi_G(v_1) - 1$.
2. $P^\theta := (u_1, u_2, \ldots, u_{\theta(n-1)}, u_{\theta(n-1)+1})$ null-vector.
3. For $j := 1$ to $\xi_G(v_1)$ do
       $u_j := v_j$.
4. For $i := 1$ to $\eta_G(v_n) - 1$ do
       For $j := 1$ to $\xi_G(v_{i+1}) - \xi_G(v_i)$ do
           $u_{order_{P^\theta}(v_i)+\theta+j} := v_{\xi_G(v_i)+j}$.
           If $|order_{P^\theta}(v_{\xi_G(v_i)+j}) - order_{P^\theta}(v_{i+1})| > \theta$ then
               SHIFT($P^\theta[u_1, u_2, \ldots, u_{order_{P^{\theta-1}}(v_i)+\theta+j}]$).
5. Return $P^\theta := (u_1, u_2, \ldots, u_{order_{P^\theta}(v_n)})$.
```

□

The following procedure is called in Step 4 of algorithm *CPP*, and it receives as input a smallest power of a path P^θ that contains $G[v_1, \ldots, v_{l-1}]$, $\xi_G(v_1) + 1 \leqslant l \leqslant n$ as an induced subgraph in P^θ. This power of a path also contains the last vertex v_l inserted by Algorithm *CPP*. Vertex v_l raises the Procedure *SHIFT* because it is not adjacent to some vertex v_{l-t} in P^θ, but $v_{l-t}v_l \in E(G)$.

Procedure. *SHIFT*

- *Input: a smallest power of a path P^{θ} that contains $G[v_1, \ldots, v_{l-1}]$ as an induced subgraph.*

- *Output: a smallest power of a path $P^{\theta+1}$ that contains $G[v_1, \ldots, v_l]$ as an induced subgraph.*

```
1. $\theta := \theta + 1$.
2. $P^{\theta} := (w_1, w_2, \ldots, w_{\theta(l-1)+1})$ null-vector.
3. $k := \max\{order_{P^{\theta-1}}(v) :$
           $order_{P^{\theta-1}}(v) < \eta_{P^{\theta-1}}(u_{n_{\theta-1}}) - 1,\ v \in V(G)\}$
   $s := min\{t \geqslant 1 : t \equiv k \bmod \theta\}$.
4. For $j := 1$ to $s$ do
      $w_j := u_j$.
5. For $j := s+1$ to $k+1$ do
      If $j \equiv (s+1) \bmod \theta$
         then $w_{order_{P^{\theta}}(u_{j-1})+2} := u_j$;
         else $w_{order_{P^{\theta}}(u_{j-1})+1} := u_j$.
6. For $j := k+2$ to $n_{\theta-1}$ do
      $w_{order_{P^{\theta}}(u_{j-1})+1} := u_j$.
7. Return $P^{\theta}$.
```

□

Algorithm *CPP* returns $P^{\theta}_{n_\theta}$, the smallest power of a path (with respect to θ and n_θ) that contains G as an unit interval graph. We construct the two powers of paths, $G_T = (V_T, E_T)$ and $G_S = (V_S, E_S)$, form $P^{\theta}_{n_\theta}$, as follows. First, $V_T = V_S = V(P^{\theta}_{n_\theta})$. Then, vertices V_T, which are not in V, receive different labels from vertices in $V(P^{\theta}_{n_\theta})$.

Next, we present a sketch of the proofs of correctness of the Procedure *SHIFT* and Algorithm *CPP*, respectively.

Lemma 1. *Let P^θ be a smallest power of a path that contains $G_{l-1} = G[v_1, \ldots, v_{l-1}]$ as an induced subgraph, with respect to order $v_1 < \ldots < v_{l-1}$. Let $v_l \in V(G)$ be the next vertex inserted in P^θ and $v_{l-t-1}v_l \notin E(G)$, $v_{l-t}v_l \in E(G)$ and $d_{P_{n_\theta}}(v_{l-t}, v_l) = \theta + 1$. Then, the output of the Procedure* SHIFT *$P^{\theta+1}$ is a smallest power of a path that contains $G_l = G[v_1, \ldots, v_{l-1}, v_l]$ as an induced subgraph, with respect to order $v_1 < \ldots < v_{l-1} < v_l$.*

Sketch of the proof. We observe that since the clique formed by the vertices $\{v_{l-t}, \ldots, v_l\}$ must be preserved in P^θ, the value of θ must be increased by one unit. Then, to preserve the non-adjacencies of vertices of G_l in the new P^θ, the procedure must insert a vertex between the largest vertex of $V(G_l)$ that is non-adjacent to v_l in P^θ and its consecutive vertex in P_{n_θ}, i.e., between $u_{\text{order}_{P^\theta}(v_{l-t-1})}$ and $u_{\text{order}_{P^\theta}(v_{l-t-1})+1}$; and the procedure must insert a vertex each $\theta + 1$ vertices numbered in descending order from the $u_{\text{order}_{P^\theta}(v_{l-t-1})}$ in P^θ. This assures $P^\theta[u_1, \ldots, u_{\text{order}_{P^\theta}(v_{l-t-1})}]$ is induced subgraph of $P^{\theta+1}$. Since $G_l[v_1, \ldots, v_{\xi_{G_l}(v_{l-t-1})}]$ is induced subgraph of $P^\theta[u_1, \ldots, u_{\text{order}_{P^\theta}(v_{l-t-1})}]$, by transitivity, $G_l[v_1, \ldots, v_{\xi_{G_l}(v_{l-t-1})}]$ is induced subgraph of $P^{\theta+1}$. As the clique formed by the vertices $\{v_{l-t}, \ldots, v_l\}$ was preserved in $P^{\theta+1}$ and the vertex set $\{u_{\text{order}_{P^\theta}(v_{l-t-1})+1}, \ldots, u_{\text{order}_{P^\theta}(v_{l-t})-1}\}$ contains no vertices of $V(G_l)$, we have $G_l[v_{l-t-1}, \ldots, v_l]$ is induced subgraph of $P^{\theta+1}$. Thus, G_l is induced subgraph of $P^{\theta+1}$.

This insertion of vertices in P^θ is minimal. In fact, given $v \in V(G_l)$ the first vertex non-adjacent to v in P^θ, with respect ordering of $V(P^\theta)$, is the first vertex non-adjacent to v in $P^{\theta+1}$, then these vertices cannot be omitted. ■

Finally, the correctness of the Algorithm *CPP* is given by Theorem 2.

Theorem 2. *Let G be an unit interval graph. Algorithm* CPP *returns a smallest power of a path $P^\theta_{n_\theta}$ with respect to n_θ and θ, which contains G as an induced subgraph.*

Sketch of the proof. Using similar techniques of Lemma 1, we prove that the output of Algorithm *CPP* is a smallest power of a path which contains G as an induced subgraph, with respect the ordering of $V(G)$ given by [3]. On the other hand, if there exists a power of a path P^σ such that P^σ is a smallest power of a path which contains G as an induced subgraph, then $\sigma \leqslant \theta$ and $n_\sigma \leqslant n_\theta$. We prove that the ordering of $V(G)$ induced by the ordering of P^σ is equal to the ordering of $V(G)$ returned by Algorithm *Recognize*, up to of indistinguishable vertices; in this case, we can change the positions of the indistinguishable vertices belonging to $V(G)$ in P^σ. Changing these positions, the ordering of $V(G)$ induced by the ordering of P^σ is equal to the ordering of $V(G)$ returned by Algorithm *Recognize*. Since P^θ is a smallest power of a path that contains G as an induced subgraph with respect this ordering, we have $\theta \leqslant \sigma$ and $n_\theta \leqslant n_\sigma$. Then, $\sigma = \theta$ and $n_\sigma = n_\theta$ ■

The Algorithm *CPP* analyzes each vertex of G in the ordering returned by Algorithm *Recognize* [3] a single time. In the worst case, the algorithm calls Procedure *SHIFT* for each vertex $v_l \in V(G)$ only once. Since, for each vertex v_l, the Procedure *SHIFT* passes by the set of vertices of G_l at most once, the complexity of Algorithm *CPP* is $\mathcal{O}(n^2)$.

3 Conclusion

In this work, we developed an $\mathcal{O}(n^2)$ time algorithm that generates, from a connected unit interval graph G, the smallest θ-powers of paths G_S and G_T (with respect to θ and to number of the vertices n of G) whose intersection contains G as an induced subgraph.

We remark that θ can be greater than or equal to the size of a maximum clique of the graph G, denoted by $\omega(G)$. Figure 1 shows an example where G has $\omega(G) = 4$ and the Algorithm *CPP* returns $\theta = 5$, but the difference between θ and $\omega(G)$ can be greater than 1.

As future work, we intend to investigate other classes of graphs. An example is graph $C_4 = (V, E)$, with $V = \{v_1, v_2, v_3, v_4\}$ and $E = \{v_1v_2, v_1v_3, v_2v_4, v_3v_4\}$, which is not an unit interval graph. In this case, we have G_S and G_T, with $V_S = V_T = \{v_1, v_2, v_3, v_4\}$, $E_S = \{v_1v_2, v_1v_3, v_2v_3, v_2v_4, v_3v_4\}$ and $E_T = \{v_1v_2, v_1v_3, v_1v_4, v_2v_4, v_3v_4\}$.

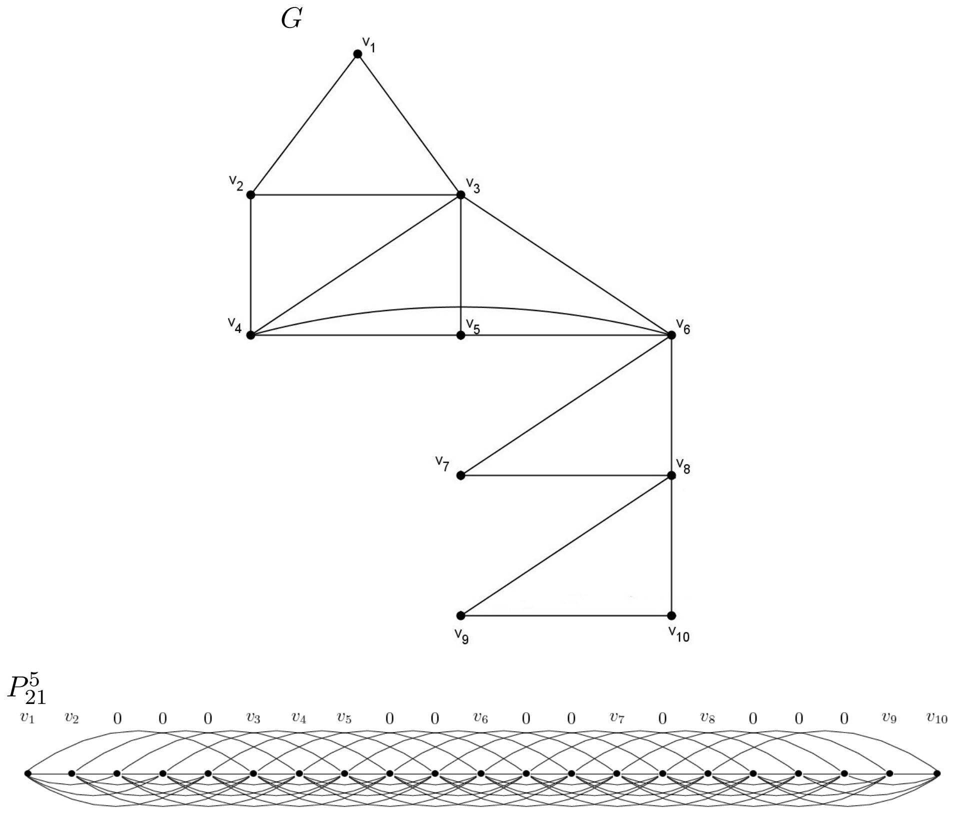
G
v1
v2
v3
v4
v5
v6
v7
v8
v9
v10
P^5_{21}
v_1 v_2 0 0 0 v_3 v_4 v_5 0 0 v_6 0 0 v_7 0 v_8 0 0 0 v_9 v_{10}

Figure 1: Graph G with $n = 10$ and $\omega(G) = 4$ and its corresponding output returned by Algorithm *CPP*: $P^{\theta}_{n_\theta}$ with $n_\theta = 21$ and $\theta = 5$.

References

[1] Adam, Z., Choi, V., Sankoff, D. and Zhu, Q., Generalized gene adjacencies, graph bandwidth and clusters in yeast evolution. In: *Mandoiu, I., Sunderraman, R., Zelikovsky, A.* (eds.) ISBRA 2008. (LNBI), v. 4983, pp. 134-145, Springer, 2008.

[2] Brandstädt, A., Hundt, C., Mancini, F. and Wagner, P., Rooted directed path graphs are leaf powers. *Discrete Mathematics*, v. 310, pp. 897-910, 2010.

[3] Corneil, D. G. and Kim, H., Natarajan, S., Olariu, S. and Sprague, A., Simple linear time recognition of unit interval graphs. *Information Processing Letters*, v. 55, pp. 99-104, 1995.

[4] Figueiredo, C. M. H, Meidanis, J. and Mello, C. P., A linear-time algorithm for proper interval graph recognition. *Information Processing Letters*, v. 56, pp. 179–184, 1995.

[5] Lin, M. C., Rautenbach, D., Soulignac, F. J. and Szwarcfiter, J. L., Powers of cycles, powers of paths, and distance graph. *Discrete Applied Mathematics*, 2010, DOI: 10.1016/j.dam.2010.03.012

[6] Roberts, F. S., Representations of indifference relations. Stanford University, Ph. D. Thesis, 1968.

[7] Sankoff, D. and Xu, X., Tests for gene clusters satisfying the generalized criterion. In: *Lecture Notes in Computer Science*, v. 5167, pp. 152-160, Springer, 2008.

Vítor Costa and Simone Dantas
Instituto de Matemática
Univ. Federal Fluminense
24.020-140, Niterói, Brazil
Email: vitorsilcost@mat.uff.br
Email: sdantas@im.uff.br

David Sankoff
Department of Math. and Statistics
University of Ottawa
Ottawa, Canada
Email: sankoff@uottawa.ca

Ximing Xu
Department of Statistics
University of Toronto
Toronto, Canada
Email: ximing@utstat.utoronto.ca