New Results About the Linearization of Scaffolds Sharing Repeated Contigs

Dorine Tabary, Tom Davot, Mathias Weller, Annie Chateau, Rodolphe Giroudeau

New results about the linearization of scaffolds sharing repeated contigs

Dorine Tabary[1], Tom Davot[1], Mathias Weller[2], Annie Chateau[1], and Rodolphe Giroudeau[1]

[1] LIRMM - CNRS UMR 5506 - Montpellier, France
firstname.lastname@lirmm.fr
[2] CNRS, LIGM, Paris, France
firstname.lastname@u-pem.fr

Abstract. Solutions to genome scaffolding problems can be represented as paths and cycles in a "solution graph". However, when working with repetitions, such solution graphs may contain branchings and, thus, they may not be uniquely convertible into sequences. Having introduced various ways of extracting the unique parts of such solutions, we extend previously known NP-hardness results to the case that the solution graph is planar, bipartite, and subcubic, and show that there is no PTAS in this case.

1 Introduction

Extracting information from genomes has become a very largely spread task, at numerous levels, and most of these need to consider their nucleotidic sequence. Large databases contain genomic sequences of a very large range of organisms, or various individuals of a same species. However, difficulties arise when it comes to extract nucleotidic sequences from the DNA molecule. Technical limitations induce a complex inference process, beginning with the *sequencing* step, where a large amount of overlapping, short sequences are produced, going on with the *assembly* step, which takes those short sequences called *reads*, and exploits overlaps to output longer sequences called *contigs*. Those contigs are usually the final product of most of genomes, called *drafted genomes*. NGS data are going to evolve towards longer and longer sequences, but most of the available sequencing data in public databases are huge collections of billions of *short reads* (*i.e.* words of between fifteen and hundreds of characters) [12]. Those genomes are often sufficient to extract useful information, for instance detect and compare genic content. However, the global structure of the genome may be lacking, depending on how these genomes stay fragmented. Intending to cure this fragmentation,

Preprint version.
The final authenticated version is available online at https://doi.org/10.1007/978-3-030-04651-4_7.

and improve the assembly process, it is possible to perform a *scaffolding* of the contigs, that is the inference of relative order and orientation of contigs, using additional information. Most of scaffolding tools are using information from paired-end sequencing, and using various models and methods (see [7, 9] for surveys).

Few of them are considering genomic repeats, which are often disturbing both assembly and scaffolding. In numerous organisms, a significant part of the genome *is* repeated. Such repeats may be of various sizes and present variable copy numbers, according to the species and individuals [3]. Due to the conservatism of some assembly methods, a repeat may cover an entire contig which is separated from the other genomic side fragments [13]. i.e.

In the jungle of problems We focus in this paper on models, graphs and problems aiming to participate to scaffolding with repeated contigs. To this purpose, we essentially manipulate two kinds of graphs, both modeling contigs and their interactions, the *scaffold graph* and the *solution graph*. We denote by $E(G)$ and $V(G)$ the set of edges and vertices, respectively, of a graph G (or E and V if no ambiguity occurs). A solution graph is a special kind of scaffold graph, the latter being defined the following way:

Definition 1 (scaffold graph). *A graph* (G, M^*, ω, m) *is a* scaffold graph *if* V *corresponds to a set of contig extremities, and* E *is composed of two subsets:* M^* *is the set of edges between extremities belonging to a same contig (thus defining a perfect matching in* G*), and* $E \setminus M^*$ *is the set of interactions between contigs. A scaffold graph comes with two functions* $\omega, m : E \to \mathbb{N}$*, defining respectively the confidence level of inter-contigs interactions, and the multiplicity of contigs (their copy number). If* m *is not provided, then all multiplicities are equal to one.*

An example of scaffold graph, and inference of solutions on this graph, can be found in Figure 1.

Given a set of contigs, it is possible to infer their multiplicities using various techniques involving for instance the cover depth in a mapping of reads on contigs (using for instance tools like CRAC [11]), or directly infer multiple contigs from kmer counting [8]. Getting links between contigs necessits additional information, for instance mapping of paired-end reads on contigs [4].

Inferring scaffolds, *i.e.* sequences of contigs at the chromosome scale, is modeled by an optimisation problem in the scaffold graph, similar to Traveling Salesman Problem, but taking into account the chromosomal structure (numbers of linear and circular chromosomes). In the simplified case where every contig is supposed to appear just once, this problem is stated as:

Scaffolding (SCA)
Input: a scaffold graph (G, M^*, ω) and integers $\sigma_{\mathrm{p}}, \sigma_{\mathrm{c}}, k \in \mathbb{N}$
Question: Is there some $S \subseteq E \setminus M^*$ such that $S \cup M^*$ is a collection of $\leq \sigma_{\mathrm{p}}$ alternating paths and $\leq \sigma_{\mathrm{c}}$ alternating cycles and $\sum_{e \in S} \omega(e) \geq k$?

For a vertex v, we define $M^*(v)$ as the unique vertex u with $uv \in M^*$. A path (or a cycle) p is called *alternating* with respect to M^* if, for all vertices u of p, also $M^*(u)$ is a vertex of p.

Scaffolding has been studied in the framework of complexity and approximation [4, 14, 15]. In this case, the produced solution is a collection of disjoint paths alternating between edges from M^* (contigs) and edges from $E \setminus M^*$ (links between contigs), from which it is easy to infer without any ambiguity a set of nucleotidic sequences by reading contig sequences, and for inter-contig links, either detecting possible overlaps missed by the assembly process, or completing with N's.

To improve the realism of the model, it is convenient to take the multiplicities of contigs into account. The main difference induced by allowing a contig to appear several times in the solution, is that the set of edges which are selected in an optimal solution does not necessarily lead to a unique interpretation as a set of scaffolds. The scaffolding problem with multiplicities thus involves a solution which is a graph, corresponding to the fusion of the right number of walks in the original scaffold graph. For each non-contig edge uv, its multiplicity $m(uv)$ equals the smaller of the multiplicities of the contig edges incident to u and v. A *walk* W is a sequence $(u_1, u_2, \ldots, u_\ell)$ of vertices such that, for each two consecutive vertices u_i and u_{i+1}, we have $u_i u_{i+1} \in E$. Then, W is called *closed* if $u_1 = u_\ell$ and W is called *alternating* with respect to M^* if ℓ is even and, for each odd i, we have $u_i u_{i+1} \in M^*$.

Observation 1 *For each vertex u of a solution graph, the sum of multiplicities of its incident non-matching edges is at most the multiplicity of its incident matching edge.*

The scaffolding problem with multiplicities is thus stated as follows:

Scaffolding with Multiplicities (MSCA)
Input: a scaffold graph (G, M^*, ω, m) and $\sigma_{\mathrm{p}}, \sigma_{\mathrm{c}}, k \in \mathbb{N}$
Question: Is there a multiset S of $\leq \sigma_{\mathrm{c}}$ closed and $\leq \sigma_{\mathrm{p}}$ non-closed alternating walks in G such that each $e \in M^*$ occurs at most $m(e)$ times in accross all walks of S and $\sum_{e \in E(S) \setminus M^*} \omega(e) \geq k$?

In this setting, a scaffold graph $(G^*, M^*, \omega^*, m^*)$ is called *solution graph* for (G, M^*, ω, m) if (a) G^* is a subgraph of G, (b) ω^* is the restriction of ω to G^*, (c) $m^*(uv) \leq m(uv)$ for all $uv \in E(G)$, (d) G^* can be decomposed into $\leq \sigma_{\mathrm{c}}$ closed and $\leq \sigma_{\mathrm{p}}$ non-closed walks. Such a decomposition into walks is called a *linearization* of the solution graph and, in general, it is not necessarily unique (see Figure 1). Note that decomposability also implies that no vertex can have more incident non-matching multiplicities than the multiplicity of its incident matching edge.

It turns out that, in presence of repeated contigs, a solution graph implies a unique set of sequences if and only if it does not contain so called *ambiguous paths* [16].

Definition 2 (Ambiguous path). *Let p be an alternating u-v-path in a solution graph. If all edges of p have the same multiplicity μ (that is, $m(e) = \mu$ for all $e \in p$), then p is called* μ-uniform *(or simply* uniform *if μ is unknown). Further, if p is μ-uniform and each of u and v is incident with a non-matching edge of multiplicity strictly less than μ, then p is called "ambiguous".*

Problem	Scaffolding	Scaffolding with Multiplicities	Semi-Brutal Cut
Input	Scaffold Graph	Scaffold Graph	Solution Graph
Output	Scaffolds	Solution Graph	Scaffolds

Table 1: Problems around genome scaffolding.

Thus, the task above can be achieved by destroying all ambiguous paths in the solution graph. A brutal way to do this is to cut the non-contig edges incident to both extremities of each ambiguous path. However, this solution may erase potentially important information. Indeed, to destroy an ambiguous path, it is sufficient to remove the non-contig edges incident to one of its extremities. Further, let v be an extremity of an uniform path, we sometimes say "to cut v", by which we mean removing all non-contig edges incident with v, and in that case v is denoted as a *vertex-cut*. The problem of finding a most parsimonious (with respect to some cost function ω') set X of vertex-cuts which destroys all ambiguous paths is called Semi-Brutal Cut. Several cost-functions ω' make sense in this setting.

Definition 3. *A weight-function $\omega' : 2^V \to \mathbb{N}$ is called*

1. cut-score*, if ω' counts one per vertex-cut (that is, $\omega'(X) = |X|$),*
2. path-score*, if ω' counts one per removed edge (that is, $\omega'(X) := \sum\{m(uv) \mid uv \in E \setminus M^* \wedge uv \cap X \neq \varnothing\}$), and*
3. weight-score*, if ω' counts the total weight of the removed edges (that is, $\omega'(X) := \sum\{m(uv) \cdot \omega(uv) \mid uv \in E \setminus M^* \wedge uv \cap X \neq \varnothing\}$).*

Note that, from the perspective of computational complexity, the path-score is a special case of the weight score, since we can just set $\omega'(e) = 1$ for all edges e. Thus, when saying "both scores" we refer to cut- and weight-score. Formally, the Semi-Brutal Cut problem on which we focus here, is defined the following way:

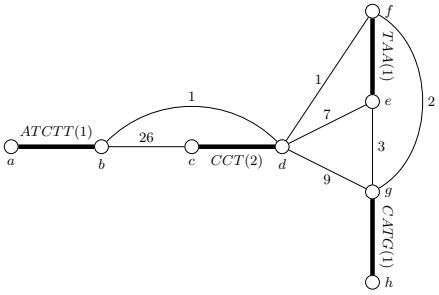
Semi-Brutal Cut (SBC)
Input: a solution graph (G, M^*, ω, m) and some $k \in \mathbb{N}$
Question: Is there a set X of vertex-cuts of G which destroys all ambiguous paths and the score of X is at most k?

We consider the functions defined in Definition 3 as scores for X. In context of approximation, Semi-Brutal Cut refers to its optimization variant, minimizing the score of X.

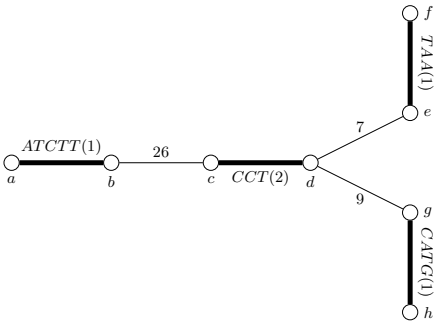
A summary of the different problems involved and their input/output is presented in Table 1.

Related works. In previous work [5, 16], we proposed the first results concerning the complexity and approximation of Semi-Brutal Cut according to the scoring functions mentioned in Definition 3. Some questions remain open concerning the complexity and (Non)-approximation for the cut and weight-score. In this article, we conclude the study of linearization in the framework of complexity and approximation. We prove that Semi-Brutal Cut according to the cut-score is APX-complete.

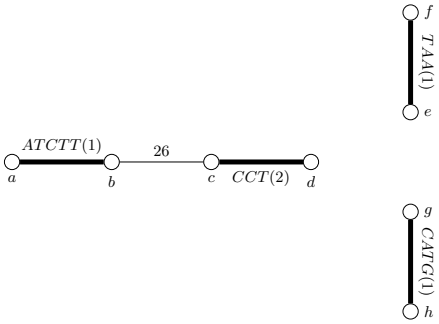
Organization of the article. The Section 2 is devoted to the complexity result, we push this hardness result to bipartite, planar, subcubic graphs whereas Section 3



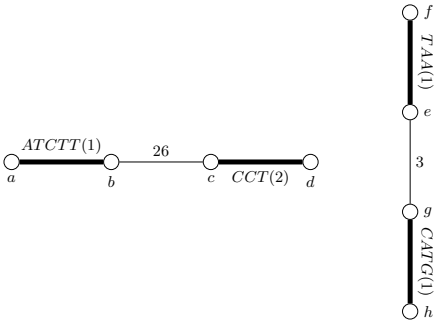
(a) Scaffold graph. This graph illustrates relationship between four contigs, figured by bold edges ab, cd, ef and gh. Labels on these edges show the sequence of the contigs, and their mutliplicity (in parenthesis). Edge cd, whose sequence is `CCT`, has multiplicity two. Other contigs are of multiplicity one. Links between contigs are labeled by their weight. In the input scaffold graph, the real sequences are both paths (a, b, c, d, e, f) and (c, d, g, h)



(b) Solution graph after solving Scaffolding with Multiplicities. The solution graph is obtained as a solution for the MSCA instance asking for two open walks with total weight ≥ 42. In the solution graph, the contig of multiplicity two labeled `CCT` constitutes an ambiguous path, yielding two possible sets of sequences $\{$`ATCCT..CCT..TAAAA`, `CCT..CATG`$\}$ and $\{$`ATCCT..CCT..CATG`, `CCT..TAAAA`$\}$.



(c) Linearisation using Semi-Brutal Cut. Brutal cut would provide a set of four independent sequences of total weight zero (the initial set of contigs), whereas Semi-Brutal Cut with weight-score provides a unique set of four sequences $\{$`ATCCT..CCT`, `CCT`, `TAAAA`, `CATG`$\}$, and weight 26 (minimal weight-score 16). After resolving successively MSCA (with $\sigma_p = 2$ and $\sigma_c = 0$) and SBC (dashed edges are cut), the solution is compatible with the initial hypothesis. The only ambiguous path is the matching edge $\{c, d\}$ and the cut vertex is d.



(d) Direct linearisation from the scaffold graph. Directly searching two maximum weighted alternating paths such that the solution graph does not contain ambiguity yields a chimeric sequence (f, e, g, h).

Fig. 1: Example of scaffold graph (Figure 1a), a solution graph (Figure 1b), scaffolds after solving Semi-Brutal Cut (Figure 1c) and a direct linearization leading to chimeric solution (Figure 1d).

Topologies	Type of cut	Complexity	Lower and upper bound
general	all	NP-hard [16]	
trees	all	linear [16]	
planar, $\Delta \leq 4$	cut-score	NP-hard [16]	approx: 1.37 ($P \neq$ NP) [16], approx: $2-\epsilon$ (UGC) [16], exact: $2^{o(n)}$(ETH) [16]
bip. plan., $\Delta \leq 3$	cut-score	NP-hard ([5])	APX-Hard [5] exact: $2^{o(\sqrt{n+m})}n^{O(1)}$ (ETH) [5] 4-approximation Theorem 3
bip., planar $\Delta \leq 3$	weight-score	NP-hard Theorem 1	2-approximation [5] APX-Complete Corollary 1
$\Delta \leq 3$			1.000056.. Theorem 2

Table 2: Overview of results for Semi-Brutal Cut.

propose some lowers bounds according to complexity hypothesis. In the last section, we develop a polynomial-time approximation algorithm which conclude Semi-Brutal Cut. Table 2 summarizes the overall results.

2 Computational Hardness

We consider in this section a very restricted class of graphs, which are planar, bipartite, subcubic graphs. The choice of this class is simultaneously led by biological and theoretical reasons. Biologically, we noticed that solution graphs are really sparse, and once reduced the non-ambiguous paths, are often equivalent to planar graphs with small degrees. However, this is only empirical observation and to our knowledge there are no general properties on real solution graphs that could be directly exploited. The theoretical reason is a wide literature on those classical classes of graphs, and we know hardness and non-approximation results that could be exploited through classical reductions. We mean then to show that, though not capturing the essential of solution graph properties, the results below give a good indication on how hard the problem is to solve, even under structural constraints.

Although it is know that Semi-Brutal Cut is NP-hard under both cut- and weight-score [16], we extend this hardness for the weight-score to planar, bipartite, subcubic graphs. To this end, we reduce the classic NP-complete problem 3-SAT to Semi-Brutal Cut.

Monotone 3-Satisfiability (3-SAT)
Input: A boolean formula φ in conjunctive normal form where each clause contains exactly three positive literals or three negative literals.
Question: Is there a satisfying assignment β for φ?

Construction 1 *Let φ be an instance of* 3-SAT *with n variables $x_1, x_2, \ldots$ and m clauses $C_1, C_2, \ldots$. For each variable x_i, let ψ_i be the list of indices ℓ such that C_ℓ contains x_i and $|\psi_i|$ is the number of occurrences of x_i in φ. We construct*

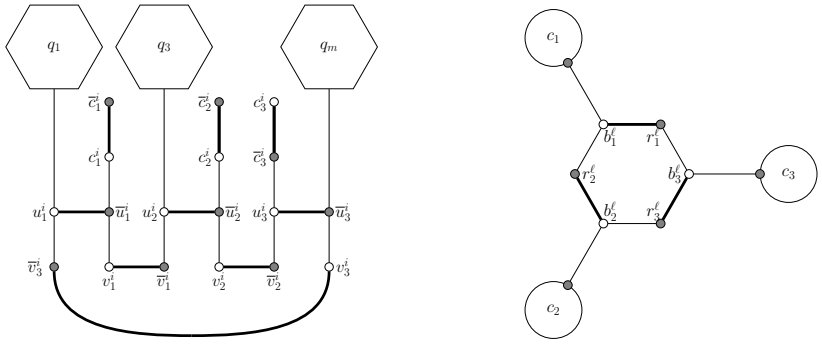


Fig. 2: Matching edges are bold. **Left:** variable gadget c_{x_i} linked to the clause gadgets q_1, q_3 and q_m, where x_i occurs positively in C_1 and C_3 and negatively in C_m. **Right:** clause gadget corresponding to the clause $C_\ell = (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$.

the following solution graph (G^*, M^*, ω, m) *along with a 2-coloring of* G^* *(see Figure 2).*

- *For each* x_i*, we construct a cycle* c_i *on the vertex set* $\bigcup_{j \leq |\psi_i|} \{u^i_j, \overline{u}^i_j, v^i_j, \overline{v}^i_j\}$ *such that, for all* $j \leq |\psi_i|$*,*
  - $\{u^i_j, \overline{u}^i_j\}, \{v^i_j, \overline{v}^i_j\} \in M^*$*, and*
  - *the vertices* u^i_j *and* v^i_j *are blue and the vertices* $\overline{u}^i_j$ *and* $\overline{v}^i_j$ *are red.*
- *For each clause* C_ℓ*, we construct an alternating 6-cycle* q_ℓ *on the vertex set* $\bigcup_{j \leq 3} \{r^\ell_j, b^\ell_j\}$ *such that, for all* $j \leq 3$*,* $\{r^\ell_j, b^\ell_j\} \in M^*$*, and* r^ℓ_j *is red and* b^ℓ_j *is blue.*
- *For each variable* x_i *and each* $j \leq |\psi_i|$*, let* C_ℓ *be the* j^{th} *clause of* ψ_i *and let* t *be such that* lit_i *is the* t^{th} *literal of* C_ℓ*. Then,*
  - *create a single matching edge* $\{c^\ell_j, \overline{c}^\ell_j\}$*, where* c^ℓ_j *is blue and* $\overline{c}^\ell_j$ *is red,*
  - *if* x_i *occurs positively in* C_ℓ*, introduce the edges* $\{r^\ell_t, u^i_j\}$ *and* $\{c^\ell_j, \overline{u}^i_j\}$*, and*
  - *if* x_i *occurs negatively in* C_ℓ*, introduce the edges* $\{b^\ell_t, \overline{u}^i_j\}$ *and* $\{\overline{c^\ell_j}, u^i_j\}$*.*
- *Each non matching edge has multiplicity* 1 *and weight* 1 *and all matching edges have multiplicity* 2 *(thus, each matching edge except the* $\{c^\ell_i, \overline{c}^\ell_i\}$ *is an ambiguous path).*

Clearly, Construction 1 can be carried out in polynomial time. Further, the resulting graph G^* is bipartite and the maximum degree $\Delta(G^*) = 3$. In the following, we call a matching edge *clean* if one of its endpoints has degree one. Note that a scaffold graph whose every matching edge is clean does not contain ambiguous paths.

Theorem 1. Semi-Brutal Cut *is NP-complete for the weight-score, even if the graph is planar, bipartite, subcubic.*

In order to prove Theorem 1, we use the following properties of Construction 1, yielding a "canonical" set of cuts.

Lemma 1. *Let* $S \subseteq V(G^*)$ *be a set of vertex-cuts destroying all ambiguous paths in* (G^*, M^*, ω, m)*, let* c_i *be a variable gadget and let* q_ℓ *be a clause gadget.*

We suppose that we start by cutting the vertices in the variable gadget and then we cut the vertices in the clause gadget. There is a set S' of vertex-cuts with $|S'| \leq |S|$ that also destroys all ambiguous paths and

(a) $\omega'(S \cap V(c_i)) = \omega'(S' \cap V(c_i)) \geq 2 \times |\psi_i|$ *and* $\omega'(S \cap V(q_\ell)) = \omega'(S' \cap V(q_\ell)) \geq 2$ *(S and S' have the same score in variable gadgets and clause gadgets),*

(b) if $\omega'(S' \cap V(c_i)) = 2 \times |\psi_i|$*, then* $S' \cap V(c_i)$ *is either* $\bigcup_{j \leq |\psi_i|} \{u_j^i\}$ *or* $\bigcup_{j \leq |\psi_i|} \{\overline{u}_j^i\}$ *(if S' is optimal on a variable gadget, cuts are only on positive sides or only on negative sides),*

(c) $\omega'(S' \cap V(q_\ell)) = 2$ *if and only if S' contains a vertex adjacent to q_ℓ (only two cuts are needed in a clause gadget iff it has been isolated by a cut in an adjacent variable gadget, meaning that the variable satisfies the clause).*

Proof. (a): For each $j \leq |\psi_i|$, we need to remove two edges to linearize the ambiguous paths $\{u_j^i, \overline{u_j^i}\}$. Then we need to remove at least $2 \times |\psi_i|$ edges in c_i. In the clause q_ℓ, we need to remove at least two edges in the inner cycle.

(b): Note that cutting all vertices in either $\bigcup_{j \leq |\psi_i|} \{u_j^i\}$ or $\bigcup_{j \leq |\psi_i|} \{\overline{u}_j^i\}$ suffices to remove all ambiguous path in x_i and in that case $\omega(S \cap V(c_i)) = 2 \times |\psi_i|$. If S contains some $\overline{u}_j^i$ and does not contain $\overline{u}_{j+1}^i$ for some j, then we need a cut to linearize $\{v_j^i, v_j^i\}$ which will increase by one the score of the solution (and analogously for u_j^i). Hence if $\omega'(S \cap V(c_i)) = 2 \times |\psi_i|$, we can suppose that S contains either $\bigcup_{j \leq |\psi_i|} \{u_j^i\}$ or $\bigcup_{j \leq |\psi_i|} \{\overline{u}_j^i\}$. If S contains a cut in some v_j^i or some $\overline{v}_j^i$ then, since the path $\{v_j^i, \overline{v}_j^i\}$ is already linearized by a cut in $\{\overline{u}_j^i, u_{j+1}^i\}$, we can remove the cut in S'.

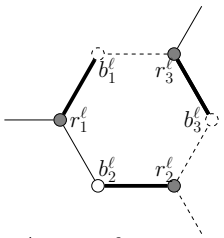


Fig. 3: A cut of size 2 in q_ℓ when one incident edge to q_ℓ is cut. Dashed edges and vertices are part of the cut.

(c): We need to remove at least two edges from the inner cycle of C_ℓ. Suppose that all literals of C_ℓ occur positively. Suppose by symmetry that $\{b_1^\ell, b_2^\ell\} \subseteq S'$. Then if the leaving edge incident to r_3^ℓ is not cut, then we need to remove one more edge from q_ℓ and in that case $\omega'(S' \cap V(q_\ell)) \geq 2$ (see Figure 3). □

Proof (Proof of Theorem 1). Recall that 3-SAT remains NP-complete if the input formula is planar [1] and, in this case, the graph produced by Construction 1 is also planar. Clearly, Semi-Brutal Cut is in NP. We show that Construction 1 is correct, that is, φ is satisfyable if and only if the scaffold graph (G^*, M^*, ω, m) resulting from Construction 1 can be linearized with a score of $8m$.

"⇒": Let β be a satisfying assignment for φ. Then, for each variable x_i and for all $j \leq |\psi_i|$, we cut the vertices u_j^i if $\beta(x_i) = 1$ and the vertices $\overline{u}_j^i$ otherwise. As β is satisfying, this removes at least one edge adjacent to each clause gadget. Thus, according to Lemma 1(c), we can cut two vertices in each clause gadget q_j to turn every matching edge in q_j clean. Since we also cut either the vertices u_j^i or the vertices $\overline{u}_j^i$ for each vertex gadget, we conclude that all matching edges of the result are clean and we remove exactly $2m + \sum_i 2 \times |\psi_i| = 8m$ edges.

"$\Leftarrow$": Let $S \subseteq V$ be the set of vertices such that cutting each vertex of S destroys all ambiguous paths in (G^*, M^*, ω, m) and $\omega'(S) = 8m$. According to Lemma 1(a), each variable gadget remove $|\psi_i|$ edges and each clause gadget remove two edges. Moreover, by Lemma 1(b), for each variable gadget c_i, we can suppose that $S \cap V(c_i)$ equals $\bigcup_{j \leq |\psi_i|} \{u^i_j\}$ or $\bigcup_{j \leq |\psi_i|} \{\overline{u}^i_j\}$. In the former case, we set $\beta(x_i) = 1$ and, in the latter, we set $\beta(x_i) = 0$. To show that β satisfies φ, assume that there is a clause C_ℓ that is not satisfied by β. Then, none of the edges incident to q_ℓ is cut which, by Lemma 1(c), contradicts the fact that there are two removed edges in q_ℓ. □

3 Non-Approximability

In this section, we prove approximation lower bounds for Semi-Brutal Cut. First recall the definition of *L-reduction* between two hard problems Π and Π', described by Papadimitriou [10]. This reduction consists of polynomial-time computable functions f and g such that, for each instance x of Π, $f(x)$ is an instance of Π' and for each feasible solution y' for $f(x)$, $g(y')$ is a feasible solution for x. Moreover there are constants $\alpha, \beta > 0$ such that:

1. $OPT_{\Pi'}(f(x)) \leq \alpha OPT_{\Pi}(x)$ and
2. $|val_{\Pi}(g(y')) - OPT_{\Pi}(x)| \leq \beta |val_{\Pi'}(y') - OPT_{\Pi'}(f(x))|$.

In the following, we present an *L-reduction* from the classical problem Max 3-SAT(4) to Semi-Brutal Cut.

Max 3-SAT(4)
Input: A boolean formula φ in exact 3-CNF where every variable occurs in 4 clauses
Task: Find an assignment that satisfies a maximum number of clauses.

Construction 2 *We reuse Construction 1 and change some variable gadgets and the way we link the variable gadgets to the clause gadgets. First, we change the links between the gadgets: let C_ℓ be a clause and x_i be the j^{th} literal of C_ℓ. Then, attach c_i to r^ℓ_j. The difference with Construction 1 is that we attach the variable gadgets to the red vertices of the clause gadget, no matter if the variable occurs positively or negatively in the clause.*

Now we change some variable gadgets. Let x_i be a variable which occurs positively in the clauses C_p and $C_{p'}$ and negatively in the clauses C_n and $C_{n'}$. We replace the variable gadget associated to x_i by the following gadget r_i:

- *Construct a cycle c_i on the vertex set $\bigcup_{j \leq 2} \{u^i_j, \overline{u}^i_j, v^i_j, \overline{v}^i_j\}$ such that, for all $j \leq 2$, $\{u^i_j, \overline{u}^i_j\}, \{v^i_j, \overline{v}^i_j\} \in M^*$, the vertices u^i_j and v^i_j are blue and $\overline{u}^i_j$ and $\overline{v}^i_j$ are red.*
- *Give multiplicity 1 and weight 1 to all non-matching edges and multiplicity 2 to all matching edges.*
- *Link the clause gadgets $q_p, q_{p'}, q_n$ and $q_{n'}$ to vertices $u^i_1, u^i_2, \overline{u}^i_1$ and $\overline{u}^i_2$ respectively in the same way as previously described.*

Note that all matching edges are ambiguous paths in the variable gadget. The clause gadgets and the other variable gadgets remain unchanged.

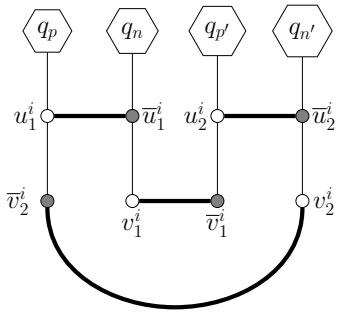


Fig. 4: Matching edges are bold. Example of variable gadget r_{x_i} linked to the clause gadgets q_p, $q_{p'}$, q_n and $q_{n'}$, where x_i occurs positively in C_p and $C_{p'}$ and negatively in C_n and $C_{n'}$.

The resulting graph G^* is bipartite and $\Delta(G^*) = 3$. In the following, when we want to differentiate the variable gadgets, we designate by *rectangle* variable gadget those defined in Construction 2 and by *cycle* variable gadget those defined in Construction 1. An example of a rectangle variable gadget is given in Figure 4. Notice that the properties (a) and (c) of Lemma 1 hold. We can add the following property:

Lemma 2. *Let $S \subseteq V(G^*)$ be an optimal set of vertex-cuts destroying all ambiguous paths in (G^*, M^*, ω, m), let c_i be a cycle variable gadget and $r_{i'}$ be a rectangle variable gadget. There is a set S' of cuts with $\omega(S') = \omega(S)$ that also destroys all ambiguous paths, and*
(a) $S' \cap V(c_i)$ is either $\bigcup_{j \leq |\psi_i|}\{u_j^i\}$ or $\bigcup_{j \leq |\psi_i|}\{\overline{u}_j^i\}$, and
(b) $S' \cap V(r_{i'})$ is either $\{u_1^{i'}, u_2^{i'}\}$ or $\{\overline{u}_1^{i'}, \overline{u}_2^{i'}\}$.

Proof. Recall that, by Lemma 1(a), $\omega(S \cap V(c_i)) \geq |\psi_i|$.

"(a)": By symmetry, suppose that x_i occurs mostly positively in φ. If x_i occurs four times positively, then replacing $S \cap V(c_i)$ by $\bigcup_{j \leq |\psi_i|}\{u_j^i\}$ in S yields a solution S' as sought. Thus, suppose that x_i occurs three times positively. Let C_ℓ be the clause where x_i occurs negatively and let z denote the neighbor of $\overline{u}_j^i$ in c_ℓ. If $|S \cap V(c_i)| > |\psi_i|$, then replacing $S \cap c_i$ by $\bigcup_{j \leq |\psi_i|}\{u_j^i\}$ plus z yields a solution S' as sought. Finally, if $|S \cap V(c_i)| = |\psi_i|$, then S already corresponds to (a) as, otherwise, some ambiguous path $\{v_j^i, \overline{v}_j^i\}$ is not destroyed.

"(b)": Note that one cut in $r_{i'}$ is not enough to destroy all ambiguous paths and cutting either the vertices $\{u_1^{i'}, u_2^{i'}\}$ or the vertices $\{\overline{u}_1^{i'}, \overline{u}_2^{i'}\}$ destroys all ambiguous paths in the rectangle variable gadget. By symmetry, suppose that S contains $v_1^{i'}$, if S contains $\overline{u}_1^{i'}$, then we can remove $v_1^{i'}$ from S. Otherwise, since S is optimal, $S \cap V(r_{i'})\{u_1^{i'}, \overline{u}_2^{i'}, v_1^{i'}\}$. Let $z \notin r_{i'}$ be the vertex adjacent to $u_1^{i'}$. Then, z is clean, since otherwise we can replace $S \cap V(r_{i'})$ by $\{\overline{u}_1^{i'}, \overline{u}_2^{i'}\}$, contradicting the fact that S is optimal. We can then add z in S' and swap $u_1^{i'}$ by $\overline{u}_1^{i'}$. Further if S does not contains any vertices in $\{v_1^{i'}, \overline{v}_1^{i'}, v_2^{i'}, \overline{v}_2^{i'}\}$, then suppose without loss of generality that S contains $\{u_1^{i'}, u_2^{i'}\}$. Let $z_j \notin r_{i'}$ be the vertex incident to $\overline{u}_j^{i'}$. If S contains $\overline{u}_j^{i'}$, then it only serve to remove the leaving edge incident to

$\overline{u}_j^{i'}$ and it also removes the edge $\{\overline{u}_j^{i'}, v_{1+(j+1 \mod 2)}^{i'}\}$, which contradicts the fact that S is optimal. Thus, $S \cap V(r_{i'}) = \{u_1^{i'}, u_2^{i'}\}$.

Theorem 2. *There is a constant $\epsilon_4' > 0$ (the value $\epsilon_4' > 0$ is defined in [2]) for which* Semi-Brutal Cut *cannot be approximated to any factor better than $(1 + {}^{7(\epsilon_4'-1)}/_{65\cdot\epsilon_4'})$, even on graphs of maximum degree three, unless P=NP.*

Proof. Recall that, unless P=NP, Max 3-SAT(4) cannot be approximated to a factor better than $\epsilon_4' = 1,00052$ [2] and that, in an optimal solution of Max 3-SAT(4), at least ${}^7/_8$ of the clauses are satisfied [6], yielding

$$OPT(\varphi) \geq {}^{7m}/_8. \tag{1}$$

To show that Construction 2 constitutes an L-reduction, let f be a function transforming any instance φ of Max 3-SAT(4) into an instance I of Semi-Brutal Cut as above, let S be a feasible solution for I corresponding to the properties of Lemma 1(a), Lemma 1(c) and Lemma 2, and let g be the function that transforms S into an assignment β as constructed in the proof of Theorem 1: each variable x_i is set to true if S cuts u_j^i for all j, and false, otherwise. By Lemma 2, for each clause gadget q_ℓ without an adjacent vertex in S, the "extra" cut occurs in q_ℓ. Hence, for each of the at most ${}^m/_8$ unsatisfied clauses in φ, we have to remove an other edge to linearize I. Thus,

$$OPT(I) \leq 8m + {}^m/_8 \overset{(1)}{\leq} {}^{65}/_7 OPT(\varphi) \tag{2}$$

An important obstacle to overcome (and reason why Construction 1 is not enough for Theorem 2) is that an approximate solution to SBC might spend extra cuts in variable gadgets in order to "change the assignment" of a variable x_i mid-way. However, since each variable occurs at most four times, this only happens for variables that occur two times positively and two times negatively. Now, with our modification to Construction 1, we can observe that each extra cut in any of the variable gadgets allows such a misuse only for a single clause gadget. Thus, the number of satisfied clauses of φ and the clause gadgets in which we have to spend extra cuts adds up to m. Hence,

$$9m = val(g(S)) + val(S) = OPT(I) + OPT(\varphi) \tag{3}$$

Thus, we constructed an L-reduction with $\alpha = {}^{65}/_7$, $\beta = 1$ and, since $\epsilon_4' \cdot val(g(S)) \leq OPT(\varphi)$, we conclude

$$\begin{aligned} val(S) &\overset{(3)}{=} OPT(I) + OPT(\varphi) - val(g(S)) \\ &> OPT(I) + (1 - {}^1/_{\epsilon_4'}) \cdot OPT(\varphi) \\ &\overset{(2)}{\geq} (1 + {}^{7(\epsilon_4'-1)}/_{65.\epsilon_4'}) \cdot OPT(I) \end{aligned}$$

□

This is conclude the proof.

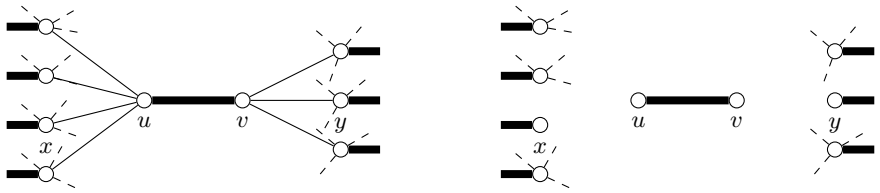


Fig. 5: A forbidden path $xuvy$ (left) and the result of cutting all its vertices (right).

4 Linear-time approximation algorithm

In the following, we present a polynomial-time 4-approximation for Semi-Brutal Cut with cut-score. To this end, we use the following reduction rule introduced by Weller and al. [16].

Rule 1 *Let $\mu \in \mathbb{N}$ and let $uvwx$ be a μ-uniform, alternating path in G. Then, replace $uvwx$ by a matching edge ux with multiplicity μ.*

Rule 1 merges pairs of non-ambiguous contigs into one. Thus, each ambiguous path will consist of a single contig edge. In this sense, we call a contig edge *ambiguous* if it is an ambiguous path and *clean*, otherwise.

Our approximation algorithm works similarly to the well-known classical 2-approximation for Vertex Cover that just returns the extremities of any maximal matching. Contrary to Vertex Cover, our forbidden structures are not edges, but *ambiguous edges*. Thus, we have to consider length-four paths containing an ambiguous edge, and we will cut all four of their vertices. In the following, we call a path $xuvy$ *forbidden* if xu and vy are inter-contig edges and uv is an ambiguous edge such that $m(xu) < m(uv) > m(vy)$ (see Figure 5).

Lemma 3. *Let Q be a maximal packing of vertex-disjoint forbidden paths in (G^*, M^*, ω, m), let X be any solution for SBC with cut-score on (G^*, M^*, ω, m). Then, (a) cutting all vertices of Q destroys all ambiguous edges in G^* and (b) $X \cap p \neq \varnothing$ for all $p \in Q$.*

Proof. (a): Let H be the result of cutting all vertices of Q in G^*. Towards a contradiction, assume that H contains an ambiguous edge uv. By definition, there are inter-contig edges xu and vy in H such that $m(xu) < m(uv) > m(vy)$. But then, the path $xuvy$ is a forbidden path, contradicting the maximality of Q.

(b): Let H be the result of cutting all vertices of X in G^*. Let $xuvy \in Q$ be a forbidden path in (G^*, M^*, ω, m) and assume towards a contradiction that $X \cap xuvy = \varnothing$. Then, none of the edges of $xuvy$ are removed when cutting the vertices of X, that is, $xuvy$ survives in H. Then, however, uv is an ambiguous path in H, contradicting X being a solution for (G^*, M^*, ω, m). □

With Lemma 3, we can show that any maximal packing of forbidden paths constitutes a 4-approximation for Semi-Brutal Cut with cut-score.

Theorem 3. *A 4-approximate solution to* Semi-Brutal Cut *with cut-score can be computed in linear time. This ratio is tight.*

Proof. First, Rule 1 can be exhaustively applied to (G^*, M^*, ω, m) in linear time since the inner vertices of any μ-uniform alternating path have degree two. Second, a packing of forbidden paths in (G^*, M^*, ω, m) can be computed by scanning all contig edges uv and, if uv is ambiguous, then $xuvy$ is a forbidden path for any inter-contig edges xu and vy. By removing x, u, v, and y from G^*, we make sure that the resulting packing is vertex-disjoint. Thus, such a packing can be produced in linear time.

Let Q be any maximal vertex-disjoint packing of forbidden paths in (G^*, M^*, ω, m). By Lemma 3(a), the vertices of Q form a solution for SBC. To show that this solution is 4-approximate, consider any optimal solution X for (G^*, M^*, ω, m). By Lemma 3(b), X intersects each path in Q. Since the paths in Q are mutually vertex disjoint and each of them contains exactly four vertices, we conclude that Q contains at most four times as many vertices as X. Applying this algorithm on a solution graph with a single ambiguous path provides a solution with four vertex-cuts instead of one. Thus, the ratio is tight. □

Corollary 1. Semi-Brutal Cut *with cut-score is APX-complete.*

5 Conclusion

We developed results concerning the complexity, lower bounds and approximability of the linearization problem for genome scaffolds sharing repeated contigs with two possible scoring functions. We managed to strengthen previously known NP-hardness to the very restricted class of planar bipartite subcubic graphs with only two multiplicities for the cut-score. We also provided a simple, linear-time 4-approximation of for cut-scores. Natural perspectives of this work are to extend this result to the weight-score, explore the possibility of FPT algorithms and approximations in the difficult cases, and examine the practical performance of the presented approximation algorithm on larger real-world instances.

Acknowledgments. This work was supported by the Institut de Biologie Computationnelle (ANR Projet Investissements d'Avenir en bioinformatique IBC) and the "Région Occitanie".

References

[1] Berg, M.D., Khosravi, A.: Optimal binary space partitions for segments in the plane. Int. J. Comput. Geometry Appl. **22**(3), 187–206 (2012)

[2] Berman, P., Karpinski, M., Scott, A.D.: Approximation hardness and satisfiability of bounded occurrence instances of SAT. Electronic Colloquium on Computational Complexity (ECCC) **10**(022) (2003)

[3] Biscotti, M.A., Olmo, E., Heslop-Harrison, J.S.: Repetitive DNA in eukaryotic genomes. Chromosome Res. **23**(3), 415–420 (Sep 2015)

[4] Chateau, A., Giroudeau, R.: A complexity and approximation framework for the maximization scaffolding problem. Theoretical Computer Science **595**, 92–106 (2015), doi:10.1016/j.tcs.2015.06.023

http://www.ibc-montpellier.fr/

[5] Davot, T., Chateau, A., Giroudeau, R., Weller, M.: On the hardness of approximating the linearization of scaffolds sharing repeated contigs (accepted to RecombCG 2018)
[6] Håstad, J.: Some optimal inapproximability results. J. ACM **48**(4), 798–859 (2001)
[7] Hunt, M., Newbold, C., Berriman, M., Otto, T.: A comprehensive evaluation of assembly scaffolding tools. Genome Biology **15**(3), R42 (2014)
[8] Koch, P., Platzer, M., Downie, B.R.: RepARK–de novo creation of repeat libraries from whole-genome NGS reads. Nucleic Acids Res. **42**(9), e80 (May 2014)
[9] Mandric, I., Lindsay, J., Măndoiu, I.I., Zelikovsky, A.: Scaffolding algorithms. In: Măndoiu, I., Zelikovsky, A. (eds.) Computational Methods for Next Generation Sequencing Data Analysis, chap. 5, pp. 107–132. Wiley (2016)
[10] Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. Journal of Computer and System Sciences **43**(3), 425–440 (1991)
[11] Philippe, N., Salson, M., Lecroq, T., Léonard, M., Commes, T., Rivals, E.: Querying large read collections in main memory: a versatile data structure. BMC Bioinformatics **12**, 242 (2011). https://doi.org/10.1186/1471-2105-12-242, https://doi.org/10.1186/1471-2105-12-242
[12] Quail, M., Smith, M., Coupland, P., Otto, T., Harris, S., Connor, T., Bertoni, A., Swerdlow, H., Gu, Y.: A tale of three next generation sequencing platforms: comparison of ion torrent, pacific biosciences and illumina miseq sequencers. BMC Genomics **13**(1), 341 (Jul 2012)
[13] Tang, H.: Genome assembly, rearrangement, and repeats. Chemical Reviews **107**(8), 3391–3406 (2007)
[14] Weller, M., Chateau, A., Dallard, C., Giroudeau, R.: Scaffolding problems revisited: Complexity, approximation and fixed parameter tractable algorithms, and some special cases. Algorithmica **80**(6), 1771–1803 (2018)
[15] Weller, M., Chateau, A., Giroudeau, R.: Exact approaches for scaffolding. BMC Bioinformatics **16**(Suppl 14), S2 (2015)
[16] Weller, M., Chateau, A., Giroudeau, R.: On the linearization of scaffolds sharing repeated contigs. In: Proc. 11th COCOA'17. pp. 509–517 (2017)