On the hardness of approximating Linearization of Scaffolds sharing Repeated Contigs

Tom Davot, Annie Chateau, Rodolphe Giroudeau, Mathias Weller

On the hardness of approximating Linearization of Scaffolds sharing Repeated Contigs

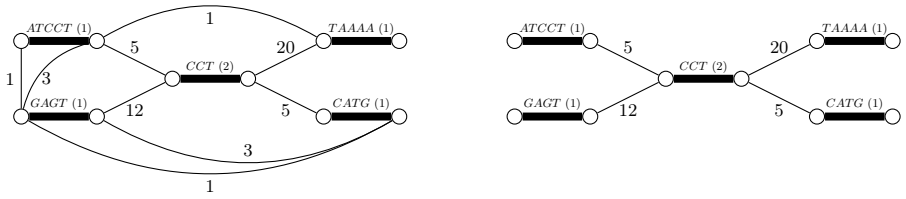
Tom Davot[1], Annie Chateau[1,2], Rodolphe Giroudeau[1], Mathias Weller[3]

[1] LIRMM - CNRS UMR 5506 - Montpellier, France
[2] IBC - Montpellier, France
[3] CNRS, LIGM, Université Paris Est, Marne-la-Vallée, France
{davot,chateau,rgirou}@lirmm.fr, mathias.weller@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 graph may contain branchings and they may not be uniquely convertible into sequences. Having introduced, in a previous work, 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 the APX-completeness in this case. We also provide some practical tests.

1 Introduction

Motivation. The process of generating proper biological genomes, from Next-Generation Sequencing (NGS) data to a full sequence of nucleotides, is a path strewn with pitfalls [6]. 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) [17] which have to be assembled into longer sequences called *contigs*. Those contigs represent fragments of the final genome but they usually do not reach the size of chromosomes and the thusly obtained *draft genomes* may therefore be highly fragmented, especially due to repeats in the genomes [18]. Though some emerging methods aim to use partially assembled genomes to infer global information on genomes [3], reducing this fragmentation is of great interest when it comes to consider whole-genome rearrangements. This fragmentation can be reduced by an additional operation, the *scaffolding*, that aims at providing an order and relative orientation of contigs that is consistent with most of the original NGS data [14]. Especially when reads are paired, it is possible to construct a *scaffold graph* summarizing the putative hypotheses concerning ordering and orientation of contigs [4]. Herein, a scaffold graph is a weighted, undirected graph G consisting of 1. a perfect matching M^* that corresponds to the contigs and 2. non-contig edges uv whose weights indicate the confidence that the contig-extremity u is adjacent to the contig-extremity v in the target genome. This paper focuses on the following problem: suppose that (a) we know for each contig how often it occurs in the genome (its *multiplicity* – which can be inferred using various possibilities), and (b) an optimal subgraph (the *solution graph*) has been extracted



(a) Scaffold graph (b) Solution graph

Fig. 1: A scaffold graph, its solution graph, and sequences that can be inferred. Contigs are represented by bold edges, labeled by the corresponding sequence and their multiplicity (in parentheses). Inter-contig edges are labeled by their weight. The solution graph is obtained as a solution for the MSCA instance asking for two 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 $\{\texttt{ATCCT..CCT..TAAAA}, \texttt{GAGT..CCT..CATG}\}$ and $\{\texttt{ATCCT..CCT..CATG}, \texttt{GAGT..CCT..TAAAA}\}$. Brutal cut would provide a set of six 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 $\{\texttt{ATCCT}, \texttt{GAGT}, \texttt{CCT..TAAAA}, \texttt{CCT..CATG}\}$, and weight 25 (minimal weight-score 17).

from the input scaffold graph (see [4, 20, 22] for methods to infer such solution graphs), which we consider as a given input. Then, the task is to infer sequences from the solution graph (see Figure 1). We consider several score functions, examine several special cases and performed tests on a dataset of various species.

Repeats. If each contig occurs exactly once in the target genome, then all vertices of the solution graph will have degree at most two and the problem becomes easy. However, 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 [2]. Due to the conservatism of some assembly methods, a repeat may cover an entire contig which is separated from the other genomic side fragments [18]. 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* [21]. 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. The problem of finding a most parsimonious (with respect to some cost function) set X of edges such that removing X from the given solution graph destroys all ambiguous paths is called Semi-Brutal Cut.

Definitions and problems. 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 scaffold graph (G, M^*, ω) consists of a simple loopless multigraph G associated with a perfect matching M^*, a weight function $\omega : E \setminus M^* \to \mathbb{N}$. The match-

ing M^* represents the contigs and ω represents the confidence that two contigs occur consecutively (respecting relative orientation implied by the edge) in the target sequence. The maximum degree of a graph G is denoted by $\Delta(G)$. For a vertex v, we define $M^*(v)$ as the unique vertex u with $uv \in M^*$. A *path* (resp. a *cycle*) is a sequence $(u_1, u_2, \ldots, u_\ell)$ of distinct vertices (resp. distinct vertices except the first and the last) such that, for each two consecutive vertices u_i and u_{i+1}, we have $u_iu_{i+1} \in E$. 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. The Scaffolding problem is defined as follows:

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 the weight-score $\sum_{e \in S} \omega(e) \geq k$?

Scaffolding has been studied in the framework of complexity and approximation [4, 20, 22]. If contigs may appear repeatedly in the genome, we add a multiplicity function $m : E \to \mathbb{N}$ to the scaffold graph. For contig edges, the multiplicity equals the number of times the contig occurs in the genome and this can be estimated from the data [9]. 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 $u_iu_{i+1} \in E$. 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 $u_iu_{i+1} \in M^*$. The difference between path and walk (resp. cycle and closed walk) is that the vertices do not need to be distinct. The Scaffolding with Multiplicities problem is the following:

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 across 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 $uv \in E$, (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).

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.*

In earlier work [21], we showed that a largest uniquely linearizable subgraph can be obtained by destroying all *ambiguous paths*, that is, all alternating paths p such that all edges of the path have the same multiplicity m_p and both extremi-

Topologies	Score	Complexity	Lower bound
general	all	NP-hard [21]	
trees	all	linear [21]	
planar, $\Delta \leq 4$	cut-score	NP-hard [21]	approx: 1.37 ($P \neq$ NP) [21], $2-\epsilon$ (UGC) [21], exact: $2^{o(n)}$(ETH) [21]
general, $\Delta \leq 2$	all	linear (Prop. 1)	
complete bipartite	cut-score	linear (Prop. 2)	
bip. plan., $\Delta \leq 3$	cut-score	NP-hard (Th. 1)	APX-Hard (Th. 2)
			exact: $2^{o(\sqrt{n+m})}n^{O(1)}$ (ETH) (Cor. 1)

Table 1: Overview of results for Semi-Brutal Cut.

ties of p are incident to a non-contig edge with multiplicity strictly less than m_p. Thus, the main problem considered in this work is the following.

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 non-contig edges of G such that $G - X$ does not contain ambiguous paths and the score of X is at most k?

We choose to separate MSCA and Semi-Brutal Cut, which is justified by the danger of producing chimeric sequencing when combining optimisation of weight on the scaffold graph and the linearisability constraint (see Figure 2). Moreover, the solution graph is by itself an interesting object to study. It embeds all *possibilities*, and may be a reasonable representation of a genome under our current knowledge. We expect that additional information may disambiguate a solution graph, such as finer study of the nature of involved repeats, dynamic of transposed elements, etc. Thus, Semi-Brutal Cut was raised as a problem aiming to propose a standard output (*e.g.* fasta files) from the solution graph. Several cost-functions ω' make sense in this setting.

Definition 1. *A weight function $\omega' : 2^E \to \mathbb{N}$ is called*

1. cut-score*, if ω' counts one per cut vertex (that is, $\omega'(X)$ is the size of a smallest vertex cover of X),*
2. path-score*, if ω' counts one per removed edge (that is, $\omega'(X) = |X|$), and*
3. weight-score*, if ω' counts the total weight of the removed edges (that is, $\omega'(X) = \sum_{e \in X} \omega(e)$).*

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. Further, when talking about cut-score, we sometimes say "to cut a vertex v", by which we mean cutting all non-contig edges incident with v. In context of approximation, Semi-Brutal Cut refers to its optimization variant, minimizing the score of X.

Related works. In previous work [21], we proposed the first results concerning the complexity of Semi-Brutal Cut according to the scoring functions mentioned in Definition 1. In that article, two main results are proved: the NP-completeness for general graphs and a polynomial-time algorithm for trees based on dynamic programming. Here, we push this hardness result to bipartite, planar, subcubic graphs and give polynomial-time algorithms for more special cases

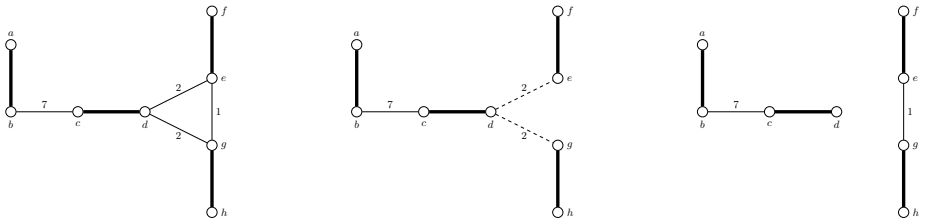


Fig. 2: **(Left)** Edge $\{c, d\}$ has multiplicity two. Other multiplicities are equal to one. The labels on the edges correspond to their weight. In the input scaffold graph, the real sequences are both paths (a, b, c, d, e, f) and (c, d, g, h). **(Middle)** After resolving successively MSCA (with $\sigma_p = 2$ 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. **(Right)** Directly searching two maximum weighted alternating paths such that the solution graph does not contain ambiguity yields a chimeric sequence (f, e, g, h).

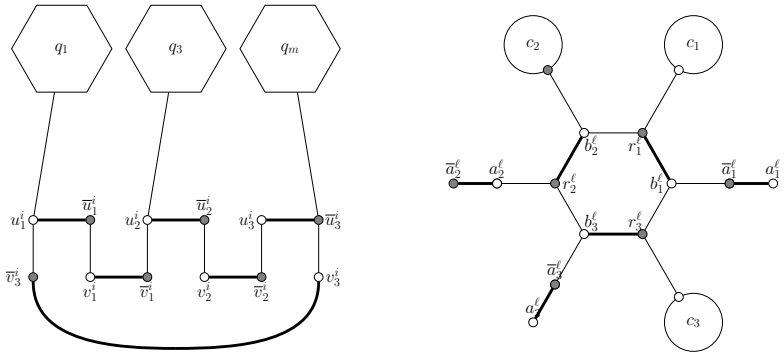


Fig. 3: Matching edges are bold. **Left:** variable gadget c_{x_i} linked to the clause gadgets q_1, q_3 and q_m, where x_i occurs positively in C_1 and C_3 and negatively in C_m. **Right:** clause gadget corresponding to the clause $C_\ell = (x_1 \vee \overline{x_2} \vee x_3)$.

(especially for $\Delta \leq 2$) marking the boundary between the NP-completeness and the polynomiality. Table 1 summarizes the complexity results.

2 Computational Hardness

While Semi-Brutal Cut is known to be NP-hard for both cut- and weight-score [21], we extend the cut-score hardness to planar, bipartite, subcubic graphs. To this end, we reduce the classic NP-complete 3-SAT [7] problem to SBC.

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

Construction 1 *Let φ be an instance of* 3-SAT *with n variables $x_1, \ldots, x_n$ and m clauses $C_1, \ldots, C_m$. 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 the following solution graph (G^*, M^*, ω, m) with a 2-coloring of G^* (see Figure 3).*

- *For each x_i, we construct a cycle c_i on the vertex set $\bigcup_{j \leq |\psi_i|} \{u_j^i, \overline{u}_j^i, v_j^i, \overline{v}_j^i\}$ such that, for all $j \leq |\psi_i|$,*
  - *$\{u_j^i, \overline{u}_j^i\}, \{v_j^i, \overline{v}_j^i\} \in M^*$, and*
  - *the vertices u_j^i and v_j^i are blue and the vertices $\overline{u}_j^i$ and $\overline{v}_j^i$ are red.*
- *For each C_ℓ, we construct an alternating 6-cycle q_ℓ on the vertex set $\bigcup_{j \leq 3} \{r_j^\ell, b_j^\ell\}$ such that, for all $j \leq 3$, $\{r_j^\ell, b_j^\ell\} \in M^*$, and r_j^ℓ is red and b_j^ℓ is blue.*
- *For each clause C_ℓ and each $j \leq 3$, let x_i be the j^{th} literal of C_ℓ and let t be such that C_ℓ is the t^{th} clause in which x_i occurs. Then,*
  - *create a single matching edge $\{a_j^\ell, \overline{a}_j^\ell\}$, where a_j^ℓ is blue and $\overline{a}_j^\ell$ is red,*
  - *if x_i is a positive literal, introduce the edges $\{r_j^\ell, u_t^i\}$ and $\{b_j^\ell, \overline{a}_j^\ell\}$, and*
  - *if x_i is a negative literal, introduce the edges $\{b_j^\ell, \overline{u}_t^i\}$ and $\{r_j^\ell, a_j^\ell\}$.*
- *Each non matching edge has multiplicity 1 and all matching edges have multiplicity 2 (thus, each matching edge except the $\{a_i^\ell, \overline{a}_i^\ell\}$ is an ambiguous path).*

Clearly, Construction 1 can be carried out in polynomial time. Further, the resulting graph G^* is bipartite and $\Delta(G^*) = 3$. 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 cut-score, even if the graph is planar, bipartite, subcubic and all multiplicities are one or two.*

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. There is a set S' of cuts with $|S'| \leq |S|$ that also destroys all ambiguous paths and*

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

(b) if $|S' \cap V(c_i)| = |\psi_i|$, 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) $|S' \cap V(q_\ell)| = 2$ 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): Since, in a cycle with y ambiguous paths, each cut can destroy at most two of them, we need at least $\lceil y/2 \rceil$ cuts to linearize this cycle. A clause gadget and a variable gadget contain a cycle of three ambiguous paths and a cycle of $2|\psi_i|$ ambiguous paths, respectively. Thus, we need at least two cuts in a clause gadget and $|\psi_i|$ cuts in a variable gadget to linearize G^*.

(b): If $v^i_j \in S$ for some j, then we can swap it for $\overline{u}^i_j$ in S (and analogously for $\overline{v}^i_j$). This operation does not increases the cardinality of S. Thus, we suppose that S contains neither v^i_j nor $\overline{v}^i_j$ for any j, implying that S contains either u^i_j or $\overline{u}^i_j$ for each j. Since $|S \cap V(c_i)| = |\psi_i|$ we know that exactly one of u^i_j and $\overline{u}^i_j$ is in S for each j. Now, if S contains some u^i_j and some $\overline{u}^i_{j'}$, then it also contains such vertices for consecutive j and j'. Hence, we can suppose that $j' = (j+1) \mod |\psi_i|$, implying that $\{v^i_j, \overline{v}^i_j\}$ is an ambiguous path.

(c): Note that both r^ℓ_1 and b^ℓ_1 are incident with a non-matching edge leaving q_ℓ. To destroy the ambiguous path $\{r^\ell_1, b^\ell_1\}$, one of these edges has to be removed. By symmetry the same holds for $\{r^\ell_2, b^\ell_2\}$ and $\{r^\ell_3, b^\ell_3\}$. Since at least three edges leaving q_ℓ have to be removed, we need a total of three cuts inside of q_ℓ unless a vertex adjacent to q_ℓ is cut. Conversely, suppose by symmetry that the edge incident to the vertex r^ℓ_2 is cut. Then, q_ℓ can be linearized by cutting b^ℓ_1 and b^ℓ_3 (see Figure 4). □

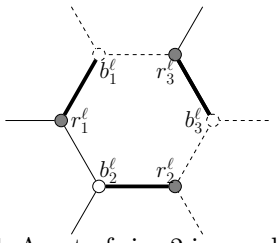


Fig. 4: A cut of size 2 in q_ℓ when 1 incident edge to q_ℓ is cut. Dashed edges and vertices are part of the cut.

Proof. of Theorem 1

Recall that 3-SAT remains NP-complete if the input formula is planar [12] and, in this case, since each gadget is planar and the edges between the clause gadget and the variable gadget can be placed in any order on the gadgets, the graph produced by Construction 1 can also be assumed to be planar. Since, clearly, Semi-Brutal Cut ∈ NP, it remains to show that Construction 1 is correct, that is φ is satisfiable if and only if the scaffold graph (G^*, M^*, ω, m) resulting from Construction 1 can be linearized with $5m$ cuts.

"⇒": 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^i_j if $\beta(x_i) = 1$ and the vertices $\overline{u}^i_j$ 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^i_j or the vertices $\overline{u}^i_j$ for each vertex gadget, we conclude that all matching edges of the result are clean and we cut exactly $2m + \sum_i |\psi_i| = 5m$ vertices.

"⇐": 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 $|S| = 5m$. According to Lemma 1(a), each variable gadget contains $|\psi_i|$ cuts and each clause gadget contains two cuts. 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 cuts in q_ℓ. □

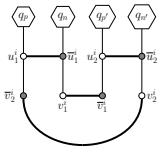


Fig. 5: Matching edges are bold. Example of variable gadget r_{x_i} linked to the clause gadgets q_p, $q_{p'}$, q_n and $q_{n'}$, where x_i occurs positively in C_p and $C_{p'}$ and negatively in C_n and $C_{n'}$.

Since Construction 1 is linear in the number of vertices and planar 3-SAT does not admit a $2^{o(\sqrt{n+m})}n^{O(1)}$-time algorithm [13], there is also no $2^{o(\sqrt{n+m})}n^{O(1)}$-time algorithm for Semi-Brutal Cut.

Corollary 1. *Assuming ETH, there is no $2^{o(\sqrt{n+m})}n^{O(1)}$-time algorithm for* Semi-Brutal Cut *in bipartite planar subcubic graphs for the cut-score.*

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 and Yannakakis [16]. 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, let $\Pi'' \in \{\Pi, \Pi'\}$, we denote by $OPT_{\Pi''}$ the value of an optimal solution of Π'' and by $val_{\Pi''}(y'')$ the value of a solution y'' of an instance of Π''. 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. 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 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 in Construction 1.*

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

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 5. 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 $|S'| = |S|$ that also destroys all ambiguous paths, and*
(a) $S' \cap V(c_i)$ is either $\bigcup_{j \leq |\psi_i|}\{u^i_j\}$ or $\bigcup_{j \leq |\psi_i|}\{\overline{u}^i_j\}$, and
(b) $S' \cap V(r_{i'})$ is either $\{u^{i'}_1, u^{i'}_2\}$ or $\{\overline{u}^{i'}_1, \overline{u}^{i'}_2\}$.

Proof. Recall that S covers the edges of M^* and, by Lemma 1(a), $|S \cap V(c_i)| \geq |\psi_i|$.

"(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 $\bigcup_{j \leq |\psi_i|}\{u^i_j\}$ 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 $\overline{u}^i_j$ in c_ℓ. If $|S \cap V(c_i)| > |\psi_i|$, then replacing $S \cap c_i$ by $\bigcup_{j \leq |\psi_i|}\{u^i_j\}$ 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 $\{v^i_j, \overline{v}^i_j\}$ is not destroyed.

"(b)": Note that one cut in $r_{i'}$ is not enough to destroy all ambiguous paths and cutting either the vertices $\{u^{i'}_1, u^{i'}_2\}$ or the vertices $\{\overline{u}^{i'}_1, \overline{u}^{i'}_2\}$ destroys all ambiguous paths in the rectangle variable gadget. Further if S cuts $\{v^{i'}_1, v^{i'}_2\}$ or $\{\overline{v}^{i'}_1, \overline{v}^{i'}_2\}$, then we can instead cut $\{u^{i'}_1, u^{i'}_2\}$ or $\{\overline{u}^{i'}_1, \overline{u}^{i'}_2\}$, respectively, without creating ambiguous paths. Suppose without loss of generality that $\{u^{i'}_1, u^{i'}_2\} \subseteq S$. Suppose further that there is some $\overline{u} \in S \cap V(r_{i'}) \setminus \{u^{i'}_1, u^{i'}_2\}$. Then, there is some clause gadget q_n linked to $\overline{u}$ since, otherwise, $S - \overline{u}$ is also a solution, contradicting optimality of S. Since all matching edges of $r_{i'}$ are already clean, the cut can only remove the edge between $\overline{u}$ and q_n. Thus we can replace $\overline{u}$ by its neighbor in q_n without changing the cardinality of S. By swapping the one or two cuts in $S \cap V(r_{i'}) \setminus \{u^{i'}_1, u^{i'}_2\}$, we obtain $S' \cap V(r_j) = \{u^{i'}_1, u^{i'}_2\}$. □

Theorem 2. *There is a constant $\epsilon'_4 > 0$ (the value $\epsilon'_4 > 0$ is defined in [1]) for which* Semi-Brutal Cut *cannot be approximated to any factor better than $(1 + {}^{7\epsilon'_4}/_{41})$, even on bipartite 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 $\epsilon'_4 = 1,00052$ [1] and that, in an optimal solution of Max 3-SAT(4), at least ${}^7/_8$ of the clauses are satisfied [8], 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^i_j 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 spend another cut to linearize I. Thus,

$$OPT(I) \leq 5m + m/8 \overset{(1)}{\leq} 41/7 OPT(\varphi) \tag{2}$$

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,

$$6m = val(g(S)) + val(S) = OPT(I) + OPT(\varphi) \tag{3}$$

Thus, we constructed an L-reduction with $\alpha = 41/7$, $\beta = 1$ and, since $val(g(S)) < (1 - \epsilon'_4) \cdot OPT(\varphi)$, we conclude

$$\begin{aligned} val(S) &\overset{(3)}{=} OPT(I) + OPT(\varphi) - val(g(S)) \\ &> OPT(I) + \epsilon'_4 OPT(\varphi) \\ &\overset{(2)}{\geq} (1 + 7\epsilon'_4/41) \cdot OPT(I) \end{aligned}$$ □

Relaxing on planarity et or maximum degree, we obtain better lower bounds:

Theorem 3. *There is a constant $\epsilon'_4 > 0$ for which* Semi-Brutal Cut *cannot be approximated to any factor better than $1 + \epsilon'_4/10$, even on bipartite, subcubic graphs with multiplicities in the set $\{1, 2\}$ unless P=NP.*

The value $\epsilon'_4 > 0$ is defined in [1].

Proof. We use a gap preserving reduction from Max 3-SAT(4) (assume w.l.o.g. that each clause of ϕ has exactly three literals (this can be easily done by repeating the literals within a clause, if necessary)) to Semi-Brutal Cut even on bipartite, subcubic graphs with multiplicities in the set $\{1, 2\}$ that transforms a Boolean formula ϕ to a graph using Construction 1 such that:

1. if $OPT(\varphi) = m$ then $OPT(G) = 5m$ (see Theorem 1), and
2. if $OPT(\varphi) < (1 - \epsilon'_4)m$ then $OPT(G) \geq 5m + \epsilon'_4 m/2$.

First item is obtained by Theorem 1. Second item comes from the observation that if the optimal solution of Max 3-SAT(4) does not satisfy m clauses, but $k \leq m - 1$, it means that an extra cut was necessary in the transformed instance to linearize the graph G. If this extra cut is placed in a variable gadget, it can linearize between one and two clauses (the extra cut can not linearize more

than two clauses since otherwise changing the value of the variable increases the number of satisfied clauses which is a contradiction). Thus, for k between $m-1$ and $m-2$, we know that at least one extra cut is needed. We generalize this argument to any number of satisfied clauses in the optimal solution, to finally get:

$$OPT(G) \geq 5m + \left\lceil \frac{m - OPT(\varphi)}{2} \right\rceil.$$

By hypothesis we have $OPT(\varphi) < (1-\epsilon'_4)m$, thus $OPT(G) \geq 5m + \lceil \epsilon'_4 m/2 \rceil \geq 5m + \epsilon'_4 m/2$. The Theorem follows. □

Hereafter, we consider Max 3-SAT(B) which is the restricted special case of Max 3-SAT where every variable occurs in at most B clauses. Recall that for Max 3-SAT(B) the best possible approximation is at least $7/8 + \Omega(1/B)$ (as a function of B) and at most $7/8 + O(1/\sqrt{B})$ unless NP=RP[19]. Based on the same arguments as given in Theorem 2, we have $val(S) \geq (335/328 - O(1/\sqrt{B}))OPT(\varphi)$, this leads us to following result:

Theorem 4. Semi-Brutal Cut *cannot be approximated to any factor better than* $335/328 - O(1/\sqrt{B})$*, even on bipartite graphs of maximum degree three, unless RP=NP.*

4 Polynomial Cases

In this section, we consider the Semi-Brutal Cut problem in graphs with maximum degree two and in complete bipartite graphs. We show a linear-time algorithm for both cut-score and weight-score. Recall that Semi-Brutal Cut can be solved in linear time in trees [21]. In the following, we suppose that the input solution graph contains at least one ambiguous path and that G is connected as otherwise, we can treat each connected component individually.

Proposition 1. Semi-Brutal Cut *can be solved in linear time on a collection of paths and cycles* ($\Delta(G) = 2$) *for the weight function given by Definition 1 .*

Proof. First, since $\Delta(G) = 2$, the cut-score is a special case of the weight-score with $\omega(e) = 1$ for all non-matching edges e. Thus, suppose that the weight-score is used. Second, since SBC can be solved in linear time on paths [21], suppose that G is a cycle. Let p be any ambiguous path in G, let e_1 and e_2 be the two unique non-matching edges incident to the extremities of p. Since $G_1 = G - e_1$ and $G_2 = G - e_2$ are trees, we can find and compare optimal solutions X_1 and X_2 for G_1 and G_2, respectively, in linear time. Further, as p is ambiguous, all optimal solutions delete e_1 or e_2 (or both) and, thus, one of $X_1 \cup \{e_1\}$ and $X_2 \cup \{e_2\}$ is optimal for G. □

Proposition 2. Semi-Brutal Cut *can be solved in linear time for cut-score on complete bipartite graphs.*

Proof. Let $K_{n,n}$ be a complete bipartite graph and note that $n > 2$ as, otherwise, there is no ambiguous path in $K_{n,n}$. Also note that, by Observation 1, all matching edges are ambiguous paths. Then, it is sufficient to cut all but one

Algorithm 1: Greedy Algorithm

Data: A solution graph (G^*, M^*, ω, m).
Result: A set $X \subseteq E \setminus M^*$ whose removal makes G^* uniquely linearizable.

$X \leftarrow \varnothing$;
$A \leftarrow$ list of extremities of ambiguous paths;
while $A \neq \varnothing$ **do**
  $u \leftarrow \text{argmin}_{x \in A}\, \omega_X(x)$;
  remove the two extremities of the ambiguous path containing u from A;
  add all non-matching edges incident with u to X;
end
return X;

Fig. 6: Tightness of the approximation ratio. Edges are bold ($\in M^*$), solid ($\in X_{opt}$) or dashed ($\in X$) and all edges have weight one. The multiplicities of the matching edges are equal to two and the multiplicities of the non-matching edges are equal to one. Thus, $\omega(X) = 2$ and $\omega(X_{opt}) = 1$.

vertex of any of the two cells of the bipartition to turn all matching edges clean. To show that $n-1$ cuts are also necessary, assume that there is a solution X with cut-score $n-2$ and let u and v be the vertices that are not cut. Then, u and v are in the same cell of the bipartition since, otherwise, there is a matching edge xy with $ux, vy \notin X$ and, thus, xy is an ambiguous path in $G - X$. But then, $\{u, M^*(v)\}$ and $\{M^*(u), v\}$ are not in X, implying that $\{u, M^*(u)\}$ is an ambiguous path in $G - X$. □

5 Approximable Cases

We propose a greedy strategy (see Algorithm 1) for the Semi-Brutal Cut problem under the weight-score function. Let (G^*, M^*, ω, m) be a solution graph and let $X \subseteq E \setminus M^*$ be a set of non-matching edges. For a vertex x, we let $\omega_X(x)$ denote the sum of the weights of all non-matching edges incident with x that are not in X. More formally, we define $\omega_X(x) := \sum_{e \in E \setminus (M^* \cup X)} \omega(e) \cdot \chi_e(x)$, where $\chi_e(x) := |e \cap \{x\}|$ is the characteristic function of e. The principle of our algorithm is to successively visit each ambiguous path and cut the edges incident to the extremity with the lowest value of w_S, where S contains all previously cut edges.

Proposition 3. *In $\mathcal{O}((|V|+|E|)\log|V|)$ time, Algorithm 1 computes a solution for* Semi-Brutal Cut *under the weight-score with an approximation ratio of 2 and this ratio is tight.*

Proof. Since each time some extremities are removed from A, the ambiguous path they belonged to has been destroyed, there are no more ambiguous paths remaining when $A = \varnothing$. Thus, the set X that is returned is indeed a solution. Let X_{opt} be an optimal solution. Let p_i denote the ambiguous path of G^* considered in step i of Algorithm 1, let u and v be its extremities, and let X_i be the set of

edges added to X in step i. If X_{opt} contains all non-matching edges incident to u, then let Q_i contain them. Otherwise, X_{opt} contains all non-matching edges incident to v, and we let Q_i contain those. Then, $\omega'(X_i) \leq \omega'(Q_i)$ for all i and, thus, $\omega'(X) \leq \sum_i \omega'(Q_i)$. Further, $\bigcup_i Q_i = X_{opt}$ and, since each edge of G^* occurs in at most two sets Q_i, we conclude $\sum_i \omega'(Q_i) \leq 2\omega'(X_{opt})$. The claimed approximation factor of two follows and, by Figure 6, it is tight.

Concerning the running time, the list of ambiguous paths is build in $\mathcal{O}(|E| + |V|)$ with a depth-first search algorithm. The sorting of this list can be done in $\mathcal{O}(|V| \log |V|)$. The maintain of the sorting of the list at each cut yields a $\mathcal{O}((|V| + |E|) \log |V|)$. □

6 Experiments

In order to observe the behavior of our algorithm on real instances, we tested it on datasets described below and we compare the obtained solutions of three different algorithms.

Description of the datasets They were generated in the following way:

1. A set of reference genomes in the Nucleotide NCBI database (see Table 2) have been chosen for their diversity in genome sizes, and types of organisms.
2. Paired-end reads, have been simulated using `wgsim` [11].then assembled using the De Bruijn Graph based *de novo* assembly tool `minia` [5].
3. Reads have been mapped to the contigs, using `bwa` [10] and contigs on the reference genome, using `megablast` [15], in order to find their multiplicities and generate scaffold graphs. Table 3 presents some statistics about produced scaffolding graphs. Notice that those graphs may be large, however their sparsity and mean degree explain why we consider very constrained classes of graphs (degree bounded by three, planar, bipartite, etc.). They do not fit these constraints, but they are quite close to.
4. We generated the solution graphs from the scaffold graphs using our ILP formulation [23] for Scaffolding with Multiplicities and using the *cplex* solver. Statistics on solution graphs are available on Table 4.

Results We ran Algorithm 1 on the datasets and we compared it with two other algorithms:

1. an exact algorithm obtained by an ILP formulation of Semi-Brutal Cut
2. a naive algorithm cutting arbitrary extremities of ambiguous path as long as such paths exist.

The idea of implementing the naive algorithm is that it provides an upper bound of Semi-Brutal Cut. We wanted answer to the following question: is the ratio of Algorithm 1 closer than 1 to those provided by the naive algorithm on real-world instances? Statistics on produced solutions are presented in Table 5. We notice that Algorithm 1 finds an optimal solution in most of the cases, even if the number of cuts is of the order of several dozens (i.e. gloeobacter, pseudomonas). The algorithm does not find an optimal solution for two instances:

Table 2: Sequences selected for experiments.

species	Tax.	alias	size (bp)	type	acc. number
Bacillus anthracis str. Sterne	Bacteria	anthrax	5228663	Chrom.[1]	NC_005945.1
Gloeobacter violaceus PCC 7421	Bacteria	gloeobacter	4659019	Chrom.[1]	NC_005125.1
Lactobacillus acidophilus NCFM	Bacteria	lactobacillus	1993560	Chrom.[1]	NC_006814.3
Pandoravirus salinus	Virus	pandora	2473870	Comp.[2]	NC_022098.1
Pseudomonas aeruginosa PAO1	Bacteria	pseudomonas	6264404	Chrom.[1]	NC_002516.2
Oryza sativa Japonica	Plant	rice	134525	Chlor.[3]	X15901.1
Saccharomyces cerevisiae	Yeast	sacchr3	316613	Chrom.[1] 3	X59720.2
Saccharomyces cerevisiae	Yeast	sacchr12	1078177	Chrom.[1] 12	NC_001144.5

1. chromosome 2. complete genome 3. chloroplast

Table 3: Scaffold graphs.

Data	$\|V\|$	$\|E\|$	Min/Max/Avg degree
anthrax	8110	11013	1 / 7 / 2.72
gloeobacter	9034	12402	1 / 12 / 2.75
lactobacillus	3796	5233	1 / 12 / 2.76
pandora	4902	6722	1 / 7 / 2.74
pseudomonas	10496	14334	1 / 9 / 2.73
rice	168	223	1 / 6 / 2.65
sacchr3	592	823	1 / 7 / 2.78
sacchr12	1778	2411	1 / 10 / 2.12

Table 4: Sequences selected for experiments.

data	#AP[1]	#NAP[2]	total weight	avg. deg.[3]	max/min deg.
anthrax	13	260	329	5.31	4 / 2
gloeobacter	44	432	694	5.68	6 / 2
lactobacillus	15	135	225	5.27	5 / 2
pandora	5	183	210	5.00	4 / 2
pseudomonas	47	413	650	5.20	5 / 2
rice	6	9	29	4.17	3 / 2
sacchr3	5	25	54	5.40	4 / 2
sacchr12	23	74	190	4.87	4 / 2

1. ambiguous paths 2. non-ambiguous paths 3. average degree of extremities of amb. paths

rice and sacchr12 and the ratio of the computed solutions are 1.33 and 1.11, respectively. The high ratio of the rice can be explained by the low score in the optimal solution. Thus, the answer to our question seems to be that the ratio of Algorithm 1 is close to 1. However, the tested instances are relatively small and it is interesting to run tests on bigger instances.

Table 5: Results statistics.

	exact		naive algorithm			Algorithm 1		
data	score	#cuts	score	ratio	#cuts	score	ratio	#cuts
anthrax	17	13	20	1.17	13	17	1.00	12
gloeobacter	68	45	80	1.17	43	68	1.00	41
lactobacillus	19	15	21	1.10	14	19	1.00	14
pandora	6	5	7	1.16	5	6	1.00	5
pseudomonas	51	50	65	1.27	41	51	1.00	40
rice	3	6	5	1.66	4	4	1.33	4
sacchr3	6	5	6	1.00	4	6	1.00	5
sacchr12	18	23	24	1.33	16	20	1.11	17

7 Conclusion

In this article, we develop 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. 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 greedy algorithm on larger real-world instances.

Acknowledgments This work was supported by the Institut de Biologie Computationnelle[4] (ANR Projet Investissements d'Avenir en bioinformatique IBC).

References

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

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

[4] http://www.ibc-montpellier.fr/

[3] D. L. Cameron, J. Schroder, J. S. Penington, H. Do, R. Molania, A. Dobrovic, T. P. Speed, and A. T. Papenfuss. GRIDSS: sensitive and specific genomic rearrangement detection using positional de Bruijn graph assembly. *Genome Res.*, 27 (12):2050–2060, Dec 2017.

[4] A. Chateau and R. Giroudeau. 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.

[5] R. Chikhi and G. Rizk. Space-efficient and exact de Bruijn graph representation based on a Bloom filter. In *WABI*, pages 236–248, 2012.

[6] R. Ekblom and J. B. Wolf. A field guide to whole-genome sequencing, assembly and annotation. *Evol Appl*, 7(9):1026–1042, Nov 2014.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[8] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[9] P. Koch, M. Platzer, and B. R. Downie. RepARK–de novo creation of repeat libraries from whole-genome NGS reads. *Nucleic Acids Res.*, 42(9):e80, May 2014.

[10] H. Li and R. Durbin. Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics*, 26(5):589–595, 2010.

[11] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. T. Marth, G. R. Abecasis, and R. Durbin. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[12] D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982.

[13] D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the Exponential Time Hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011.

[14] I. Mandric, J. Lindsay, I. I. Măndoiu, and A. Zelikovsky. Scaffolding algorithms. In I. Măndoiu and A. Zelikovsky, editors, *Computational Methods for Next Generation Sequencing Data Analysis*, chapter 5, pages 107–132. Wiley, 2016.

[15] A. Morgulis, G. Coulouris, Y. Raytselis, T. L. Madden, R. Agarwala, and A. A. Schäffer. Database indexing for production megablast searches. *Bioinformatics*, 24(16):1757–1764, 2008. doi: 10.1093/bioinformatics/btn322.

[16] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[17] M.A. Quail, M. Smith, P. Coupland, T.D. Otto, S.R. Harris, T.R. Connor, A. Bertoni, H.P. Swerdlow, and Y. Gu. 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.

[18] H. Tang. Genome assembly, rearrangement, and repeats. *Chemical Reviews*, 107 (8):3391–3406, 2007.

[19] L. Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 453–461, 2001.

[20] M. Weller, A. Chateau, and R. Giroudeau. Exact approaches for scaffolding. *BMC Bioinformatics*, 16(Suppl 14):S2, 2015.

[21] M. Weller, A. Chateau, and R. Giroudeau. On the linearization of scaffolds sharing repeated contigs. In *Proc. 11th COCOA'17*, pages 509–517, 2017.

[22] M. Weller, A. Chateau, C. Dallard, and R. Giroudeau. Scaffolding problems revisited: Complexity, approximation and fixed parameter tractable algorithms, and some special cases. *Algorithmica*, 80(6):1771–1803, 2018.

[23] M. Weller, A. Chateau, R. Giroudeau, and M. Poss. Scaffolding with repeated contigs using flow formulations. 2018.