Consensus Patterns (Probably) Has no EPTAS

Christina Boucher* Christine Lo* Daniel Lokshantov*

June 23, 2015

Abstract

Given n length-L strings $S = \{s_1, \ldots, s_n\}$ over a constant size alphabet Σ together with an integer ℓ, where $\ell \leq L$, the objective of *Consensus Patterns* is to find a length-ℓ string s, a substring t_i of each s_i in S such that $\sum_{\forall i} d(t_i, s)$ is minimized. Here $d(x, y)$ denotes the Hamming distance between the two strings x and y. *Consensus Patterns* admits a PTAS [Li et al., JCSS 2002] is fixed parameter tractable when parameterized by the objective function value [Marx, SICOMP 2008], and although it is a well-studied problem, improvement of the PTAS to an EPTAS seemed elusive. We prove that *Consensus Patterns* does not admit an EPTAS unless FPT=W[1], answering an open problem from [Fellows et al., STACS 2002, Combinatorica 2006]. To the best of our knowledge, *Consensus Patterns* is the first problem that admits a PTAS, and is fixed parameter tractable when parameterized by the value of the objective function but does not admit an EPTAS under plausible complexity assumptions. The proof of our hardness of approximation result combines parameterized reductions and gap preserving reductions in a novel manner.

1 Introduction

Lanctot et al. [15] initiated the study of *distinguishing string selection problems* in bioinformatics, where we seek a representative string satisfying some distance constraints from each of the input strings. The *Consensus Patterns* problem falls within this broad class of stringology problems. Given n length-L strings $S = \{s_1, \ldots, s_n\}$ over a constant size alphabet Σ together with an integer ℓ, where $\ell \leq L$, the objective of *Consensus Patterns* is to find a length-ℓ string s, a length-ℓ substring t_i of each s_i in S such that $\sum_{\forall i} d(t_i, s)$ is minimized. Here $d(x, y)$ denotes the Hamming distance between the two strings x and y. One specific application of *Consensus Patterns* in bioinformatics is the problem of finding transcription factor binding sites [15, 22]. Transcription factors are proteins that bind to promoter regions in the genome and have the effect of regulating the expression of one or more genes. Hence, the region where a transcription factor binds is very well-conserved, and the problem of detecting such regions can be extrapolated to the problem of finding the substrings $\{t_1, \ldots, t_n\}$.

Consensus Patterns is NP-hard even when the alphabet is binary [16], so we do not expect a polynomial-time algorithm for the problem. On the other hand, the problem admits a *polynomial time approximation scheme* (PTAS), which finds a solution that is at most a factor $(1 + \epsilon)$ worse than the optimum [16] in $n^{O(\frac{1}{\epsilon^2} \log \frac{1}{\epsilon})}$-time. While a superpolynomial dependence of the running time on $\frac{1}{\epsilon}$ is implied by the NP-hardness of *Consensus Patterns*, there is still room for faster approximation schemes for the problem and so a significant effort has been invested in attempting on proving tighter bounds on the running time of the PTAS [4, 5]. If the exponent of the polynomial in the running time of a PTAS is independent of ϵ then the PTAS is called

*Department of Computer Science and Engineering, University of California, San Diego

an *efficient PTAS* (EPTAS). An interesting question, posed by Fellows et al. [9] is whether *Consensus Patterns* admits an EPTAS.

The difference in running time of a PTAS and an EPTAS can be quite dramatic. For instance, running a $O(2^{1/\epsilon}n)$-time algorithm is reasonable for $\epsilon = \frac{1}{10}$ and $n = 1000$, whereas running a $O(n^{1/\epsilon})$-time algorithm is infeasible on this same input. Hence, considerable effort has been devoted to improving PTASs to EPTASs, and showing that such an improvement is unlikely for some problems. For example, Arora [2] gave a $n^{O(1/\epsilon)}$-time PTAS for *Euclidean TSP*, which was then improved to a $O(2^{O(1/\epsilon^2)}n^2)$-time algorithm in the journal version of the paper [3]. On the other hand *Independent Set* admits a PTAS on unit disk graphs [14] but Marx [18] showed that it does not admit an EPTAS assuming FPT$\neq$W[1]—a widely believed assumption from parameterized complexity. Many more examples of PTASs that have been improved to EPTASs, and problems for which there exists a PTAS but the existence of an EPTAS has been ruled out under the assumption that FPT$\neq$W[1] can be found in the survey of Marx [19]. In this paper we show that assuming FPT$\neq$W[1], *Consensus Patterns* does not admit an EPTAS, resolving the open problem of Fellows et al. [9]. Since *Consensus Patterns* has a PTAS and is FPT, standard methods for ruling out an EPTAS cannot be applied. We discuss this in more details in Section 1.1. Our proof avoids this obstacle by combining gap preseving reductions and parameterized reductions in a novel manner.

1.1 Methods

Our lower bounds are proved under the assumption FPT$\neq$W[1], a standard assumption in parameterized complexity that we will briefly discuss here. In a parameterized problem every instance $\mathcal{I}$ comes with a *parameter* k. A parameterized problem is said to be *fixed parameter tractable* (FPT) if there is an algorithm solving instances of the problem in time $f(k)|\mathcal{I}|^{O(1)}$ for some function f depending only on k and not on $|\mathcal{I}|$. The class of all fixed parameter tractable problems is denoted by FPT. The class W[1] of parameterized problems is the basic class for fixed parameter intractability, FPT $\subseteq$ W[1] and the containment is believed to be proper. A parameterized problem Π with the property that an FPT algorithm for Π would imply that FPT=W[1] is called W[1]-hard. Thus demonstrating W[1]-hardness of a parameterized problem implies that it is unlikely that the problem is FPT. We refer the reader to the textbooks [6, 8, 11, 21] for a more thorough discussion of parameterized complexity.

W[1]-hardness is frequently used to rule out EPTAS's for optimization problems, since an EPTAS for an optimization problem automatically yields a FPT algorithm for the corresponding decision problem parameterized by the value of the objective function [19]. More specifically, if we set $\epsilon = \frac{1}{2\alpha}$, where α is the value of the objective function, then a $(1+\epsilon)$-approximation algorithm would distinguish between "yes" and "no" instances of the problem. Hence, an EPTAS could be used to solve the problem in $O(f(\epsilon)n^{O(1)}) = O(g(\alpha)n^{O(1)})$-time. Hence, if a problem is W[1]-hard when parameterized by the value of the objective function then the corresponding optimization problem does not admit an EPTAS unless FPT=W[1]. To the best of our knowledge, *all* known results ruling out EPTASs for problems for which a PTAS is known use this approach. However, this approach cannot be used to rule out an EPTAS for *Consensus Patterns* because *Consensus Patterns* parameterized by d has been shown to be FPT by Marx [17]. Thus, different methods are required to rule out an EPTAS for *Consensus Patterns*.

In his survey, Marx [19] introduces a hybrid of FPT reductions and gap preserving reductions and argues that it is conceivable that such a reduction could be used to prove that a problem that has a PTAS and is FPT parameterized by the value of the objective function does not admit an EPTAS unless FPT=W[1]. We show that *Consensus Patterns* does not admit an EPTAS unless FPT=W[1], giving the first example of this phenomenon.

Preliminaries

A PTAS for a minimization problem finds a $(1+\epsilon)$-approximate solution in time $|\mathcal{I}|^{f(1/\epsilon)}$ for some function f. An approximation scheme where the exponent of $|\mathcal{I}|$ in the running time is independent of ϵ is called an *efficient* polynomial time approximation scheme (EPTAS). Formally, an EPTAS is a PTAS whose running time is $f(1/\epsilon)^{O(1)}|\mathcal{I}|^{O(1)}$.

Let $L, L' \subseteq \sum^* \times \mathbb{N}$ be two parameterized problems. We say that L *fpt-reduces* to L' if there are functions $f, g : \mathbb{N} \to \mathbb{N}$, and an algorithm that given an instance $(\mathcal{I}, k)$ runs in time $f(k)|\mathcal{I}|^{f(k)}$ and outputs an instance $(\mathcal{I}', k')$ such that $k' \leq g(k)$ and $(\mathcal{I}, k) \in L \iff (\mathcal{I}', k') \in L'$. These reductions work as expected; if L fpt-reduces to L' and L' is FPT then so is L'. Furthermore, if L fpt-reduces to L' and L is W[1]-hard then so is L'.

Let s be a string over the alphabet Σ. We denote the length of s as $|s|$, and the jth character of s as $s[j]$. Hence, $s = s[1]s[2]\dots s[|s|]$. For a set S of strings of the same length we denote by $S[i]$ as $\{s[i] \ : \ s \in S\}$. Thus, if the same character appears at position i in several strings it is counted several times in $S[i]$. For an interval $P = \{i, i+1, \dots, j-1, j\}$ of integers, define $s[P]$ to be the substring $s[i]s[i+1]\dots s[j]$ of s. For a set S of strings and interval P define $S[P]$ to be the (multi)set $\{s[P] \ : \ s \in S\}$. For a set S of length-ℓ strings we define the *consensus string* of S, denoted as $c(S)$, as the sequence where $c(S)[i]$ is the most-frequent character in $S[i]$ for all $i \leq \ell$. Ties are broken by selecting the lexicographically first such character, however, we note that the tie-breaking will not affect our arguments.

We denote the sum Hamming distance between a string, s, and a set of strings, S, as $d(S, s)$. Observe that the consensus string $c(S)$ minimizes $d(S, c(S))$—implying that no other string x is closer to S than $c(S)$. However, some $x \neq c(S)$ could achieve $d(S, x) = d(S, c(S))$ and we refer to such strings as *majority strings* because they are obtained by picking a most-frequent character at every position with ties broken arbitrarily.

We will use standard concentration bounds for sums of independent random variables. In particular, the following variant of the Hoeffding's bound [13] given by Grimmett and Stirzaker [12, p. 476] will be needed.

Proposition 1. (Hoeffding's bound) *Let $X_1, X_2, ...X_n$ be independent random variables such that $a_i \leq X_i \leq b_i$ for all i. Let $X = \Sigma_i X_i$ and the expected value of X be $E[X]$ then it follows that:*

$$\Pr[X - E[X] \geq t] \leq \exp\left(\frac{-2t^2}{\Sigma_{i=1}^{n}(b_i - a_i)^2}\right).$$

2 Hardness of Approximating Colored Consensus String with Outliers

To show that *Consensus Patterns* does not admit an EPTAS we will first demonstrate hardness the following problem, that we call *Colored Gap-Consensus String with Outliers*. When defining parameterized gap problems, we follow the notation of Marx [19].

Colored Gap-Consensus String with Outliers (CCWSO)	
Input:	A (multi)set of n length-ℓ strings $S = \{s_1, \ldots, s_n\}$ over a finite alphabet Σ, an integer $n^* \leq n$, a partitioning of S into n^* sets $S = S_1 \cup S_2 \ldots S_{n^*}$, a rational ϵ and two integers D_{yes} and D_{no} with $D_{no} \geq D_{yes}(1+\epsilon)$ such that either (a) there exists a set S^* such that $\lvert S^* \cap S_i \rvert = 1$ for every i and $d(S^*, c(S^*)) \leq D_{yes}$ or (b) for every S^* such that $\lvert S^* \cap S_i \rvert = 1$ for every i we have $d(S^*, c(S^*)) \geq D_{no}$.
Parameter:	$\lceil 1/\epsilon \rceil$
Question:	Is there an S^* such that $d(S^*, c(S^*)) \leq D_{yes}$?

The aim of this section is to prove the following lemma.

Lemma 1. Gap-Colored Consensus String with Outliers *is W[1]-hard.*

The proof of Lemma 1 is by reduction from the *MultiColored Clique (MCC)* problem. Here input is a graph G, an integer k and a partition of $V(G)$ into $V_1 \uplus V_2 \ldots V_k$ such that for each i, $G[V_i]$ is an independent set. The task is to determine whether G contains a clique C of size k. Observe that such a clique must contain exactly one vertex from each V_i, since for each i we have $C \cap V_i \leq 1$. It is well-known that MCC is W[1]-hard [10].

Given an instance (G, k) of MCC we produce in $f(k)n^{O(1)}$-time an instance $(S_1, S_2, \ldots S_{n^*})$ of *Colored Gap-Consensus String with Outliers*. We will say that a subset S^* of S such that $\lvert S^* \cap S_i \rvert = 1$ for every $i \leq n^*$ is a *potential solution* to the CCWSO instance. Our constructed instance will have the following property. If G has a k-clique then there exists a potential solution S^* such that $d(S^*, c(S^*)) \leq D_{yes}$. On the other hand, if no k-clique exists in G then for each potential solution S^* we have $d(S^*, c(S^*)) \geq D_{no}$. The values of D_{yes} and D_{no} will be chosen later in the proof, however, we note that the crucial point of the construction is that $D_{no} \geq \left(1 + \frac{1}{h(k)}\right) D_{yes}$. Hence, a $f(\epsilon)(n\ell)^{O(1)}$-time algorithm for *Gap-Consensus String with Outliers* could be used to solve to solve the MCC problem in time $g(k)n^{O(1)}$ by setting $\epsilon = \frac{1}{2h(k)}$. Thus, the reduction is a parameterized, gap-creating reduction where the size of the gap decreases as k increases but the decrease is a function of k only.

Construction. We describe how the instance $(S_1, S_2, \ldots S_{n^*})$ is constructed from (G, k). Our construction is randomized, and will succeed with probability $\frac{2}{3}$. To prove Lemma 1 we have to change the construction to make it deterministic but for now let us not worry about that.

We start by considering the instance (G, k) and let $E(G) = \{e_1, e_2, \ldots e_m\}$. In the reduction we will create one string s_i for every edge $e_i \in E(G)$. We partition the edge set $E(G)$ into sets $\binom{k}{2}$ sets $E_{\{p,q\}}$ where $1 \leq p, q \leq k$ as follows; $e_i \in E_{p,q}$ if e_i has one endpoint in V_p and the other in V_q. The edge $e_i \in E_{p,q}$ has two endpoints, one in V_p and the other in V_q. The string s_i is inserted into the set $S_{\{p,q\}}$ and the set S of strings in the instance of *Gap-Colored Consensus String with Outliers* will be exactly

$$S = \bigcup_{\substack{p,q \\ p \neq q}} S_{\{p,q\}}.$$

We set $n^* = \binom{k}{2}$, and use exactly the partition of S into the sets $S_{\{p,q\}}$ as the partition into n^* sets in the instance. Thus, picking a potential solution S^* corresponds to picking a set of edges with exactly one edge from each of the sets $E_{\{p,q\}}$.

There are $K = k \cdot (k-1) \cdot (k-2)$ ordered triples of integers from 1 to k. Consider the lexicographic ordering of such triples. As an example, if $k = 3$ this ordering is

$$(1,2,3),(1,3,2),(2,1,3),(2,3,1),(3,1,2),(3,2,1).$$

For each i from 1 to K, let $\sigma(i)$ be the i'th triple in this ordering. Thus, for $k = 3$, we have that $\sigma(4) = (2,3,1)$. The functions σ^1, σ^2 and σ^3 return the first, second and third entry of the triple returned by σ. Continuing our example for the case that $k = 3$, we have $\sigma^1(4) = 2$, $\sigma^2(4) = 3$ and $\sigma^3(4) = 1$.

Based on G and k, we select an integer ℓ. The exact value of ℓ will be discussed later in the proof, for now the reader may think of ℓ as some function of k time $\log n$. We construct a set $Z = z_1, z_2, \ldots z_m$ of strings, Z will act as a "pool of random bits" in our construction. For each edge $e_i \in E(G)$ we make a string z_i as follows.

$$z_i = \overline{a}_i^{\sigma(1)} \circ \overline{a}_i^{\sigma(2)} \ldots \circ \overline{a}_i^{\sigma(K)}$$

For every $i \leq m$ and $p \leq K$, the strings $\tilde{a}_i$, $\tilde{a}'_i$ and $\overline{a}_i^{\sigma(p)}$ are random binary strings of length ℓ. For each $p \leq K$ and vertex $u \in V_{\sigma_1(p)}$ we make an identification string $id^p(u)$ of length ℓ. Let i be the smallest integer such that the edge e_i is incident to u. We set $id^p(u) = \overline{a}_i^{\sigma(p)}$. Notice that the other endpoint of e_i a vertex not in V_p. Thus, for any other vertex $v \in V_p$ distinct from u we have that $id^p(v) = \overline{a}_j^{\sigma(p)}$ for some integer $j \neq i$.

We now make the set S of strings in our instance. For each edge $e_i \in E(G)$ we make a string s_i as follows.

$$s_i = a_i^{\sigma(1)} \circ a_i^{\sigma(2)} \ldots \circ a_i^{\sigma(K)}$$

For each $x \leq K$ we define a_i^x using the following rules. Let $e_i = uv$ with $u \in V_p$ and $v \in V_q$. If $\sigma^1(x) = p$ and $\sigma^2(x) = q$ or $\sigma^1(x) = p$ and $\sigma^3(x) = q$, we set $a_i^{\sigma(x)} = id^x(u)$. If $\sigma^1(x) = q$ and $\sigma^2(x) = p$ or $\sigma^1(x) = q$ and $\sigma^3(x) = p$, we set $a_i^{\sigma(x)} = id^x(v)$. Otherwise we set $a_i^{\sigma(x)} = \overline{a}_i^{\sigma(x)}$.

For $1 \leq p \leq K$ we define $B_p = \{(p-1)\ell+1, (p-1)\ell+2, \ldots (p-1)\ell+\ell\}$, and will refer to B_p as the p'th *block* of the instance. Notice that for every $i \leq m$ and $p \leq K$ we have $s_i[B_p] = a_i^{\sigma(p)}$. We set $L = K \cdot \ell$ and $N = |S| = m$, this concludes the construction. Recall that n^* is the size of the solution S^* sought for and observe that L is the length of the constructed strings in S.

Analysis. We consider the constructed strings s_i as random variables, and for every j the character $s_i[j]$ is also a random variable which takes value 1 with probability 1/2 and 0 with probability 1/2. Observe that for any two positions j and j' such that $j \neq j'$ and any i and i' the random variables $s_i[j]$ and $s_{i'}[j']$ are independent. On the other hand $s_i[j]$ and $s_{i'}[j]$ could be dependent. However, if $s_i[j]$ and $s_{i'}[j]$ are dependent then, by construction $s_i[j] = s_{i'}[j]$.

Let $S^* \subseteq S$ be a potential solution. Here we consider S^* as a set of random string variables, rather than a set of strings. We are interested in studying $d(S^*, c(S^*))$ for different choices of the set S^*. We can write out $d(S^*, c(S^*))$ as

$$d(S^*, c(S^*)) = \sum_{p=1}^{K} d(S^*[B_p], c(S^*)[B_p]) \tag{1}$$

and

$$d(S^*[B_p], c(S^*)[B_p]) = \sum_{j \in B_p} d(S^*[j], c(S^*)[j]).$$

Thus, for each $p \leq K$ we have that $d(S^*[B_p], c(S^*)[B_p])$ is a sum of ℓ independent random variables, each taking values from 0 to n^*. Hence, when ℓ is large enough $d(S^*[B_p], c(S^*)[B_p])$

is sharply concentrated around its expected value. Using a union bound (over the choices of p) we can show that $d(S^*, c(S^*))$ is sharply concentrated around its expectation as well.

We turn our attention to $E[d(S^*, c(S^*))]$ for different choices of S^*. The two main cases that we distinguish between is whether S^* corresponds to the set of edges of a clique in G or not. Note that a potential solution S^* corresponds to a set of E^* edges with exactly one edge $e_{\{p,q\}} \in E_{\{p,q\}}$ for every (unordered) pair p,q. *In the remainder of this section S^* is a potential solution and E^* is the edge set corresponding to S^*. For each pair p,q of integers, $e_{\{p,q\}}$ is the unique edge in $E^* \cap \in E_{\{p,q\}}$.* We will determine whether E^* is the set of edges of a clique using the following observation, whose proof is obvious and hence omitted.

Observation 1. *E^* is the edge set of a clique in G if and only if for every ordered triple (a, b, c) of distinct integers between 1 and k the edge $e_{\{a,b\}}$ and the edge $e_{\{a,c\}}$ are incident to the same vertex in V_a.*

In the constructed instance the block B_p such that $\sigma(p) = (a, b, c)$ is responsible for performing the check for the triple (a, b, c). Before proceeding we need some additional definitions regarding random walks on the integers. Let $\vec{v}$ be a vector of positive integers. We define the random variable $X_{\vec{v}} = \vec{W} \cdot \vec{v}$ where $\vec{W}$ is a random vector with same dimension as $\vec{v}$, such that each coordinate of $\vec{W}$ is drawn from $\{-1, 1\}$ uniformly at random. The variable $X_{\vec{v}}$ is interpreted as follows: start a one-dimensional random walk at 0, in each step of the walk we go left or right with probability 1/2. However, the length of the different steps varies, in step i the walk jumps $\vec{v}[i]$ to the left or right. The value of $X_{\vec{v}}$ is the offset from the origin at the end of the walk. The total *length* of the random walk is $\sum_i \vec{v}[i]$ whereas the *number of steps* of the walk is the dimension of $\vec{v}$. We define the random variable $X^i_{r,t} = i + X_{\vec{v}}$ where v is a vector with $r - 2t$ entries that are 1 and t entries that are 2. Intuitively $X^i_{r,t}$ is the offset from 0 of a random walk starting at i of length r, with t steps of length 2 and the remaining steps of length 1. We set $x^i_{r,t} = E[|X^i_{r,t}|]$.

The next lemma characterizes the expectation of $d(S^*[B_p], c(S^*)[B_p])$, subject to the case distinction on whether the solution S^* passes or fails the test of Observation 1 for the triple $\sigma(p)$.

Lemma 2. *Let $p \leq K$ and let $\sigma(p) = (a, b, c)$. If $e_{\{a,b\}}$ and $e_{\{a,c\}}$ are incident to the same vertex in V_a, then*

$$E[d(S^*[B_p], c(S^*)[B_p])] = \ell \cdot (n^*/2 - x^0_{n^*,1}).$$

If $e_{\{a,b\}}$ and $e_{\{a,c\}}$ are not incident to the same vertex in V_a, then

$$E[d(S^*[B_p], c(S^*)[B_p])] = \ell \cdot (n^*/2 - x^0_{n^*,0}).$$

Proof. Every string $s_i \in S^*$ corresponds to an edge $e_i \in E(G)$. If $e_i \notin E_{a,b} \cup E_{a,c}$ then $s_i[B_p] = a_i^{\sigma(p)} = \overline{a}_i^{\sigma(p)}$. On the other hand, if $e_i = E_{a,b} \cup E_{a,c}$ then $s_i[B_p] = a_i^{\sigma(p)} = id^p(u)$, where u is the vertex of V_a incident to e_i.

Let j be a position in B_p. Consider the case that the two edges $e_{\{a,b\}}$ and $e_{\{a,c\}}$ in E^* are *not* incident to the same vertex in V_a. Then $d(S^*[j], c(S^*[j]))$ is distributed as $n^*/2 - |X_{\vec{v}}|$ where $\vec{v}$ is a n^*-dimensional vector of 1s. Specifically for all $s_i \in S^*$ the $s_i[j]$s are independent so $c(S^*[j])$ is the majority character out of n^* characters independently drawn from $\{0, 1\}$, and $d(c(S^*[j], S^*[j]))$ is the number of occurrences of the minority character. This is distributed as $n^*/2 - |X_{\vec{v}}|$.

Consider now the case that the two edges $e_{\{a,b\}}$ and $e_{\{a,c\}}$ in E^* *are* incident to the same vertex in V_a. From the construction of the strings $a_i^{\sigma(p)}$ it follows that all of the characters in $S^*[j]$ are independent with the exception of the two characters in the strings corresponding to the two edges $e_{\{a,b\}}$ and $e_{\{a,c\}}$; these two characters are equal. Therefore $d(S^*[j], c(S^*[j]))$ is

distributed as $n^*/2 - |X_{\vec{v}}|$ where $\vec{v}$ is a $n^* - 1$ dimensional vector with 1 entry of value 2 and $n^* - 1$ entries with value 1. Linearity of expectation implies the lemma. □

We now define E_{yes} as follows.

$$E_{yes} = K \cdot \ell \cdot (n^*/2 - x^0_{n^*,1})$$

Observe that Equation 1, Lemma 2 and linearity of expectation immedeately implies that if E^* is the set of edges of a clique then $E[d(S^*, c(S^*))] = E_{yes}$. Furthermore, By Lemma 2 each triple (a, b, c) of distinct integers from 1 to k such that the edge $e_{\{a,b\}}$ and $e_{\{a,c\}}$ are not incident to the same vertex in V_a will contribute exactly $\ell \cdot (n^*/2 - x^0_{n^*,0})$ instead of $\ell \cdot (n^*/2 - x^0_{n^*,1})$ to the expectation $E[d(S^*, c(S^*))]$. This proves the following lemma.

Lemma 3. *Let t be the number of ordered triples (a, b, c) of distinct integers from 1 to k such that the edge $e_{\{a,b\}}$ and $e_{\{a,c\}}$ are not incident to the same vertex in V_a. Then*

$$E[d(S^*, c(S^*))] = E_{yes} + t \cdot \ell \cdot (x^0_{n^*,1} - x^0_{n^*,0})$$

To conclude the analysis we need to show that as the number of triples t that fail the test of Observation 1 increases, so does the expected value of $d(S^*, c(S^*))$. To that end, all we need to prove is that $x^0_{n^*,1} - x^0_{n^*,0} > 0$. We will prove this by "differentiating" $x^0_{n^*,t}$ with respect to t.

Claim 1. *$x^0_{n^*,0} < x^0_{n^*,1}$. Furthermore we can compute $x^0_{n^*,0}$ and $x^0_{n^*,1}$ in time polynomial in n^*.*

Proof. Recall that $x^i_{r,t} = E[|X^i_{r,t}|]$ where $X^i_{r,t}$ is a random variable denoting the final position of a random walk of length r, with t double steps, starting at i. Here i is an integer and might be negative. Conditional expectation yields the following recurrence for $x^i_{r,t}$, $r \geq 2t \geq 0$.

$$x^i_{r,t} = \begin{cases} |i| & \text{if } r = 0, \\ (x^{i+1}_{r-1,t} + x^{i-1}_{r-1,t})/2 & \text{if } r > 2t, \\ (x^{i+2}_{r-2,t-1} + x^{i-2}_{r-2,t-1})/2 & \text{if } t \geq 1. \end{cases}$$

It is easy to see that one of the three cases must apply when $r \geq 2t \geq 0$ - and $x^i_{r,t}$ is only defined for these values. Observe that if $r > 2t$ and $t \geq 1$ then both the second and the third case apply. The recurrence above also yields a polynomial time algorithm to compute $x^i_{r,t}$. Now, for integers i, r, t such that $r \geq 1$ and $t \geq 2$ define $\delta x^i_{r,t} = x^i_{r,t} - x^i_{r,t-1}$. The recurrence above together with definition of $\delta x^i_{r,t}$ yields the following recurrence for $\delta x^i_{r,t}$, for $r \geq 2t$ and $t \geq 1$.

$$\delta x^i_{r,t} = \begin{cases} 0 & \text{if } r = 2, |i| \geq 2, \\ 1/2 & \text{if } r = 2, |i| = 1, \\ 1 & \text{if } r = 2, |i| = 0, \\ (\delta x^{i+1}_{r-1,t} + \delta x^{i-1}_{r-1,t})/2 & \text{if } r > 2t, \\ (\delta x^{i+2}_{r-2,t-1} + \delta x^{i-2}_{r-2,t-1})/2 & \text{if } t \geq 2. \end{cases}$$

A straightforward induction using this recurrence shows that $\delta x^0_{r,1} > 0$ for all $r \geq 1$, proving that $x^0_{n^*,0} < x^0_{n^*,1}$, as claimed. □

We now define Δ as follows,

$$\Delta = x^0_{n^*,1} - x^0_{n^*,0},$$

and note that Claim 1 implies that $\Delta > 0$. Furthermore, note that Δ depends only on $n^* = \binom{k}{2}$, so Δ is a computable function of k. Define

$$E_{no} = E_{yes} + \Delta \cdot \ell \tag{2}$$

Observe that $E_{no}/E_{yes} \geq 1 + \frac{2\Delta}{K \cdot n^*}$, and that therefore $E_{no}/E_{yes} \geq 1 + 1/h(k)$ for a function h depending only on k. Lemma 3, Claim 1 and the definition of E_{no} implies the following lemma, which summarizes the analysis up until now.

Lemma 4. *If E^* is the edge set of a clique in G, then $E[d(S^*[B_p], c(S^*)[B_p])] = E_{yes}$. Otherwise $E[d(S^*[B_p], c(S^*)[B_p])] \geq E_{no}$.*

From the definitions of E_{yes} and E_{no} it follows that there exist constants κ_{yes} and κ_{no} depending only on k such that $E_{yes} = \kappa_{yes}\ell$ and $E_{no} = \kappa_{no}\ell$. Furthermore, $\kappa_{yes} < \kappa_{no}$ and the value of κ_{yes} and κ_{no} can be computed in time $f(k)$ for some function f. Set $\kappa'_{yes} = (2\kappa_{yes} + \kappa_{no})/3$ and $\kappa'_{no} = (\kappa_{yes} + 2\kappa_{no})/3$. Then $\kappa_{yes} < \kappa'_{yes} < \kappa'_{no} < \kappa_{no}$. We set $D_{yes} = \kappa'_{yes}\ell$ and $D_{no} = \kappa'_{no}\ell$. Notice that

$$\kappa'_{yes} - \kappa_{yes} = \kappa_{no} - \kappa'_{no}.$$

A randomized analogue of Lemma 1. Before proving Lemma 1 we argue that the randomized construction works. Specifically, we show that if *Gap-Consensus String With Outliers* is W[1]-hard under randomized FPT-reductions. The results proved in this section are not used in the proof of Lemma 1, but they provide useful insights on how the deterministic construction works.

Lemma 5. *For any potential solution S^*, any $p \leq K$ and any real $x > 0$ we have the following inequality.*

$$P\Big[|d(S^*[B_p], c(S^*)[B_p]) - E[d(S^*[B_p], c(S^*)[B_p])]| > x \cdot \ell\Big] \leq 2\exp\left(-2\left(\frac{x}{n^*}\right)^2 \ell\right).$$

Proof. We have that $d(S^*[B_p], c(S^*)[B_p]) = \sum_{j \in B_p} d(S^*[j], c(S^*)[j])$. The $d(S^*[j], c(S^*)[j])$'s are independent of each other, and therefore $d(S^*[B_p], c(S^*)[B_p])$ is the sum of ℓ independent random variables taking values from 0 to n^*. The statement of the lemma follows from Hoeffding's inequality (Proposition 1). □

We now define ℓ. This value for ℓ is only valid for the randomized construction, and a different value for ℓ is used in the proof of Lemma 1.

$$\ell = \left(\frac{Kn^*}{\kappa'_{yes} - \kappa_{yes}}\right)^2 \ln\left(20Km^{n^*}\right). \tag{3}$$

Recall that m is the number of edges in the graph G, so $m \leq n^2$ and hence $\ell_1 \leq f \cdot \log n$ for some f depending only on k.

Lemma 6. *If G has a k-clique C, let S^* be the set of strings corresponding to edges of C. Then $P[d(S^*, c(S^*)) > D_{yes}] \leq \frac{1}{10(m)^{n^*}}$. If G does not contain a k-clique, then the probability that S contains a potential solution S^* such that $d(S^*, c(S^*)) < D_{no}$ is at most $1/10$.*

Proof. If G has a k-clique C, let S^* be the set of strings corresponding to edge endpoints of edges in C. By Lemma 5 it follows that for any $p \leq K$ we have

$$\begin{aligned} P\Big[|d(S^*[B_p], c(S^*)[B_p]) - E[d(S^*[B_p], c(S^*)[B_p])]| > \frac{\kappa_{yes'} - \kappa_{yes}}{K} \cdot \ell\Big] \\ \leq 2\exp\left(-2\left(\frac{\kappa'_{yes} - \kappa_{yes}}{Kn^*}\right)^2 \ell\right) \leq \frac{1}{10Km^{n^*}}. \end{aligned}$$

The union bound over all choices of $p \leq K$, together with Equation 1 yields

$$P\Big[|d(S^*, c(S^*)) - E[d(S^*, c(S^*))]| > (\kappa_{yes'} - \kappa_{yes}) \cdot \ell\Big] \leq \frac{1}{10m^{n^*}}.$$

Since $D_{yes} - E_{yes} = (\kappa_{yes'} - \kappa_{yes})\ell$, it follows that $P[d(S^*, c(S^*)) > D_{yes}] \leq \frac{1}{10m^{n^*}}$.

On the other hand, consider a set S^* of size n^* that does not correspond to the edge endpoints of a clique. An argument identical to the one above (but using that $\kappa'_{yes} - \kappa_{yes} = \kappa_{no} - \kappa'_{no}$) shows that $P[d(S^*, c(S^*)) < D_{no}] \leq \frac{1}{10m^{n^*}}$. Since there are at most m^{n^*} choices for potential solutions S^*, the union bound implies the second statement of the lemma. □

We now prove a randomized analogue of Lemma 1.

Lemma 7. *If* Colored Gap-Consensus String With Outliers *is FPT then W[1] ⊆ randomized FPT.*

Proof. Assiming that *Colored Gap-Consensus String With Outliers* has an algorithm with running time $f(\epsilon)(n\ell)^{O(1)}$ we give a randomized fixed parameter tractable algorithm for *MCC* with two sided error. We construct the instance to *Colored Gap-Consensus String With Outliers* as described, and set $\epsilon = \frac{D_{no}}{D_{yes}} - 1 = \frac{\kappa'_{no}}{\kappa'_{yes}} - 1$. If the algorithm for *Colored Gap-Consensus String With Outliers* concludes that there potential solution S^* such that $d(S^*, c(S^*)) \leq D_{yes}$ the algorithm returns that the input graph G contains a k-clique, otherwise we return that G has no k-clique. The construction takes time $O(f(k)n^{O(1)})$ for some function f, and ϵ depends only on k. Hence the total running time is $g(k)n^c$ for some function g. Thus the algorithm terminates in FPT time.

If G contains a k-clique, then by Lemma 6, with probability at least $1 - \frac{1}{10(m)^{n^*}} \geq 1 - \frac{1}{n^k}$ there is a set S^* of size n^* such that $d(S^*, c(S^*)) \leq D_{yes}$. If this event occurs, the algorithm for *Colored Gap-Consensus String With Outliers* will correctly find such a set and correctly return "yes". Hence the probability of false negatives is at most $\frac{1}{n^k}$.

If G does not contain a k-clique, then by Lemma 6, with probability at least 9/10 for every set S^* of size n^* we have $d(S^*, c(S^*)) > D_{no}$. If this event occurs the algorithm correctly returns "no" and hence the probability if false positives is at most 1/10. This implies that there is a randomized fixed parameter tractable algorithm for *MCC*, which in turn shows that W[1] ⊆ randomized FPT. □

A Deterministic Construction and Proof of Lemma 1. In order to prove Lemma 1 we need to make the construction deterministic. We only used randomness to construct the set Z, all other steps are deterministic. We now show how Z can be computed deterministically instead of being selected at random, preserving the properties of the reduction. For this, we need the concept of near p-wise independence defined by Naor and Naor [20]. The original definition of near p-wise independence is in terms of sample spaces, we define near p-wise independence in terms of collections of binary strings. This is only a notational difference, and one may freely translate between the two variants.

Definition 1 ([20])**.** *A set $C = \{c_1, c_2, \ldots c_t\}$ of length ℓ binary strings is (ϵ, p)-independent if for any subset C' of C of size p, if a position $i \leq t$ is selected uniformly at random, then*

$$\sum_{\alpha \in \{0,1\}^p} |P[C'[i] = \alpha] - 2^{-p}| \leq \epsilon.$$

Naor and Naor [20] and Alon et al. [1] give determinsitic constructions of small nearly k-wise independent sample spaces. Reformulated in our terminology, Alon et al. [1] prove a slightly stronger version of the following theorem.

Theorem 1 ([1]). *For every t, p, and ϵ there is a (ϵ, p)-independent set $C = \{c_1, c_2, \ldots c_t\}$ of binary strings of length ℓ, where $\ell = O(\frac{2^k \cdot k \log t}{\epsilon})$. Furthermore, C can be computed in time $O(|C|^{O(1)})$.*

We use Theorem 1 to construct the set Z. We set

$$\epsilon = \frac{\kappa'_{yes} - \kappa_{yes}}{K \cdot n^*}$$

and construct an (ϵ, n^*)-independent set C of $2m$ strings. These strings have length $\ell = f \cdot \log(n)$ for some f depending only on k, and C can be constructed in time $O(gn^{O(1)})$ for some g depending only on k. i we set

$$z_i = c_i \circ c_i \circ \ldots \circ c_i,$$

where we used K copies of c_i such that z_i is a string of length L. That is, in the construction of z_i we set $\overline{a}_i^{\sigma(p)} = c_i$ for all $p \leq K$. The remaining part of the construction, i.e the construction of S from Z remains unchanged. To distinguish between the deterministically constructed S and the randomized construction, we refer to the deterministically constructed S as S_{det}. We now prove that for every potential solution $S^*_{det} \subseteq S_{det}$, if S^* is the set of strings in the randomized construction that corresponds to the same edges as S^*_{det}, then $d(S^*_{det}, c(S^*_{det}))$ is almost equal to $E[d(S^*, c(S^*))]$. When considering $E[d(S^*, c(S^*))]$ we consider the randomized construction, but with the same choice of ℓ as in the construction of S_{det}, so that the strings in S and S_{det} have the same length.

For a subset I of $\{1, 2, \ldots, m\}$ define $S^*(I) = \{s_i \in S \ : \ i \in I\}$ and $S^*_{det}(I) = \{s_i \in S_{det} \ : \ i \in I\}$. The construction of S_{det} (and S) from Z implies that for every $x \leq K$, there exists a function $f_x : \mathbb{N} \to \mathbb{N}$ such that for any $i \leq m$, $s_i[B_x] = z_{f(i)}[B_x]$. For any $I \subseteq \{1, 2, \ldots, m\}$ and $x \leq K$ we define $Z^*(I, x)$ to be an arbitrarily chosen subset of Z of size n^* such that $\{z_{f_x(i)} \ : \ i \in I\} \subseteq Z^*(I, x)$. The reason we did not define $Z^*(I, x)$ as exactly $\{z_{f_x(i)} \ : \ i \in I\}$ is that the function f_x is not injective, and we want to ensure $|Z^*(I, x)| = n^*$. The definition of $Z^*(I, x)$ ensures that for every $I \subseteq \{1, 2, \ldots, m\}$ of size n^*, the string sets $S^*(I)[B_x]$ and $S^*_{det}(I)[B_x]$ are functions of $Z^*(I, x)[B_x]$. Even stronger, for every $j \in B_x$ we have that the *strings* $S^*(I)[j]$ and $S^*_{det}(I)[j]$ are functions of $Z^*(I, x)[j]$. Strictly speaking $S^*(I)[j]$, $S^*_{det}(I)[j]$ and $Z^*(I, x)[j]$ are multi-sets of characters, but we can think of them as strings by, for example, reading the characters in $S^*(I)[j]$ as $s_i[j]$ for all $i \in I$ in increasing order. Since the deterministic and randomized constructions are identical (except for the construction of Z) the strings $S^*(I)[j]$ and $S^*_{det}(I)[j]$ depend on $Z^*(I, x)[j]$ in exactly the same way.

An immediate implication of the fact that $S^*(I)[B_x]$ and $S^*_{det}(I)[B_x]$ are functions of $Z^*(I, x)[B_x]$, is that the distances $d(S^*(I)[j], c(S^*(I)[j]))$ and $d(S^*_{det}(I)[j], c(S^*_{det}(I)[j]))$ are also functions of $Z^*(I, x)[j]$. We now give these functions a name. For every set $I \subseteq \{1, 2, \ldots, m\}$ of size n^* and integer $x < K$ define $d^I_x : \{0, 1\}^{n^*} \to \{0, 1, \ldots, n^*\}$ to be a function such that for any $j \in B_x$, if $Z^*(I)[j] = \alpha$ then $d(S^*(I)[j], c(S^*(I)[j])) = d^I_x(\alpha)$ and $d(S^*_{det}(I)[j], c(S^*_{det}(I)[j])) = d^I_x(\alpha)$.

For every set $I \subseteq \{1, 2, \ldots, m\}$ of size n^* and integer $x \leq K$ we have the following expression for $d(S^*(I)_{det}[B_x], c(S^*(I)_{det}[B_x]))$.

$$d\left(S^*_{det}(I)[B_x], c(S^*_{det}(I)[B_x])\right) = \ell \cdot \sum_{\alpha \in \{0,1\}^{n^*}} P[Z^*(I)[j] = \alpha] \cdot d^I_j(\alpha) \tag{4}$$

Here the probability $P[Z^*(I)[j] = \alpha]$ is taken over random selections of j from B_x. For the randomized construction we have that $P[Z^*(I)[j] = \alpha] = \frac{1}{2^{n^*}}$, which yields the following expression.

$$E\left[d\left(S^*(I)[B_x], c(S^*(I)[B_x])\right)\right] = \ell \cdot \sum_{\alpha \in \{0,1\}^{n^*}} \frac{1}{2^{n^*}} \cdot d_j^I(\alpha) \tag{5}$$

Combining Equations 4 and 5 yields the following bound.

$$\begin{aligned} &\left| d\left(S^*_{det}(I)[B_x], c(S^*_{det}(I)[B_x])\right) - E[d(S^*(I)[B_x], c(S^*(I)[B_x]))] \right| \\ &\qquad = \ell \cdot \left| \sum_{\alpha \in \{0,1\}^{n^*}} \left(P[Z^*(I)[p] = \alpha] - \frac{1}{2^{n^*}} \right) \cdot d_j^I(\alpha) \right| \qquad (6) \\ &\qquad \leq \ell \cdot \epsilon \cdot n^* \end{aligned}$$

Summing Equation 6 over $1 \leq x \leq K$ yields the desired bound for every $I \subseteq \{1, 2, \ldots, 2m\}$ of size n^*.

$$\left| d\left(S^*_{det}(I), c(S^*_{det}(I))\right) - E[d(S^*(I), c(S^*(I)))] \right| \leq \ell \cdot K \cdot \epsilon \cdot n^* \leq \ell \cdot (\kappa_{yes'} - \kappa_{yes}) \tag{7}$$

Equation 7 allows us to finish the proof of Lemma 1. For any potential solution S^* that corresponds to a clique in G, we have that $E[d(S^*(I), c(S^*(I)))] = E_{yes} = \ell\kappa_{yes}$, and so by Equation 7,

$$d\left(S^*_{det}(I), c(S^*_{det}(I))\right) \leq \ell\kappa'_{yes} = D_{yes}.$$

For any potential solution S^* of size n^* that does not correspond to a clique in G, we have that $E[d(S^*(I), c(S^*(I)))] \geq E_{no} = \ell\kappa_{no}$, and so by Equation 7,

$$d\left(S^*_{det}(I), c(S^*_{det}(I))\right) \geq \ell\kappa'_{no} = D_{no}.$$

Since $\frac{D_{no}}{D_{yes}} \geq 1 + \delta$ for some δ depending only on k, the construction is an fpt-reduction from MCC to *Gap-Consensus String With Outliers*, completing the proof of Lemma 1. □

3 Hardness of Approximating Consensus Patterns

To show that *Consensus Patterns* does not have an EPTAS unless $FPT = W[1]$ we introduce the following gap variant of the problem.

Gap-Consensus Patterns

Input:	A set $S = \{s_1, \ldots, s_n\}$ of length-L strings over a constant size alphabet Σ together with an integer ℓ, where $\ell \leq L$, a rational ϵ and intgers D_{yes} and D_{no} with $D_{no} \geq D_{yes}(1+\epsilon)$ such that the following holds. Either there is a length-ℓ substring t_i of each s_i in S such that $\sum_{\forall i} d(t_i, s) \leq D_{yes}$ or for every collection $t_1, \ldots t_n$ such that t_i is a length-ℓ substring s_i we have $\sum_{\forall i} d(t_i, s) \geq D_{no}$.
Parameter:	$\lceil 1/\epsilon \rceil$
Question:	Is there a length-ℓ substring t_i of each s_i in S such that $\sum_{\forall i} d(t_i, s) \leq D_{yes}$?

We will now give a fpt-reduction from *Gap-Colored Consensus String with Outliers* to *gap-Consensus Patterns*. The main ingredient in our reduction is a gadget string w. The string w has length L_1 (to be determined later), and for every $i \geq 1$, $w[i] = 1$ if $i = j^2$ for an integer j and $w[i] = 0$ otherwise. We will say that an integer i is a *square* if $i = j^2$ for some integer j. Thus $w[i]$ is 1 if and only if i is a square.

Lemma 8. *For positive integers x, y and z such that $z \geq \frac{L_1}{4}$, $x < y$ and $y + z \leq L_1$ we have $d(w[\{x, x+1, \ldots, x+z\}], w[\{y, y+1, \ldots, y+z\}]) \geq \lfloor \frac{\sqrt{L_1}}{16} \rfloor$*

Proof. To lower bound $d(w[\{x, x+1, \ldots, x+z\}], w[\{y, y+1, \ldots, y+z\}])$ it is sufficient to find the number of values for i between 0 and z such that $w[x+i] = 1$ but $w[y+i] = 0$, that is $x+i$ is a square but $y+i$ is not. Let $i_1, i_2, \ldots i_t$ be all the values for i such that $x+i$ is square, in increasing order. We prove that if $y + i_j$ is square then $y + i_{j+1}$ is not. In particular, suppose $y + i_j$ is square. Let r_x and r_y be the integers such that $x + i_j = r_x^2$ and $y + i_j = r_y^2$. Since $x < y$ we have $r_x < r_y$. Furthermore, $x + i_{j+1} = (r_x + 1)^2$. Hence

$$y + i_{j+1} = r_y^2 + i_{j+1} - i_j = r_y^2 + ((r_x+1)^2 - r_x^2) < r_y^2 + ((r_y+1)^2 - r_y^2) = (r_y+1)^2.$$

But then $y + i_{j+1}$ can't be square. It follows that there are at least $\lfloor \frac{t}{2} \rfloor$ values for i such that $x + i$ is a square but $y + i$ is not. It remains to lower bound t.

The gap between a square number i and the next square number i' is less than $2\sqrt{i'} \leq 2\sqrt{L_1}$. Thus the number of square numbers in $\{x, x+1, \ldots, x+z\}$ is at least $\frac{L_1}{4} \cdot \frac{1}{2\sqrt{L_1}} \geq \lfloor \frac{\sqrt{L_1}}{8} \rfloor$. Hence $d(w[\{x, x+1, \ldots, x+z\}], w[\{y, y+1, \ldots, y+z\}]) \geq \lfloor \frac{\sqrt{L_1}}{16} \rfloor$. □

Given an instance n^*, $S = S_1 \uplus S_2 \uplus \ldots \uplus S_{n^*}$ of *Gap-Colored Consensus String with Outliers* we construct an instance of *Gap-Cosensus Patterns* as follows. First we ensure that all of the (multi) sets S_i contain the same number of strings; if $|S_i| < |S_j|$ for some i, j we can make duplicates of strings in S_i until equality is obtained. This does not affect any other aspects of the instance, since a solution S^* has to pick one string from each S_i.

Let ℓ be the length of all the strings in S. We choose L_1 such that $\lfloor \frac{\sqrt{L_1}}{16} \rfloor > n^* \cdot \ell$ and construct a gadget string w of length L_1. For every $i \leq n^*$ we make a string $\hat{s}_i$ from the set S_i. Let $S_i = s_i^1, s_i^2, \ldots, s_i^t$. We define

$$\hat{s}_i = w \circ s_i^1 \circ w \circ s_i^2 \circ w \ldots \circ w \circ s_i^t.$$

and set $L = L_1 + \ell$. We keep the values of D_{yes} and D_{no}. This concludes the construction.

Lemma 9. *For every $S^* = \{s_1^*, \ldots, s_{n^*}^*\} \subset S$ such that $s_i^* \in S_i$ for all i there is a collection $T^* = t_1^*, \ldots t_{n^*}^*$ such that t_i^* is a length L substring of $\hat{s}_i$ and $d(c(T^*), T^*) \leq d(C(S^*), S^*)$.*

Proof. For every i, set $t_1^* = w \circ s_i^*$. Since $s_i^* \in S_i$ we have that t_1^* is a length L substring of $\hat{s}_i$. Set $c = w \circ c(S^*)$, we have that $d(c(T^*), T^*) \leq d(c, T^*) \leq d(C(S^*), S^*)$. □

Lemma 10. *For every collection $T^* = t_1^*, \ldots t_{n^*}^*$ such that t_i^* is a length L substring of $\hat{s}_i$ and $d(c(T^*), T^*) \leq n^* \cdot \ell$ there is a subset $S^* = \{s_1^*, \ldots, s_{n^*}^*\} \subset S$ such that $s_i^* \in S_i$ for all i and $d(C(S^*), S^*) \leq d(c(T^*), T^*)$.*

Proof. For every i we can decompose t_i^* into $t_i^* = w[\{a_i + 1, \ldots, L\}] \circ s_i^* \circ w[\{1, \ldots, a_i\}]$ for a non-negative integer $a_i \leq L$, where $s_i^* \in S_i$. If $a_i = 0$ then $t_i^* = w \circ s_i^*$ while $a_i = L$ gives $t_i^* = s_i^* \circ w$. Set $S^* = \{s_1^*, \ldots, s_{n^*}^*\}$. We need to show that $d(C(S^*), S^*) \leq d(c(T^*), T^*)$. It is sufficient to show that for every i, j we have $a_i = a_j$ because then all the s_i^*'s align in the decomposition of the t_i^*'s and so $d(C(S^*), S^*) \leq d(c(T^*), T^*)$ holds.

We prove that if $a_i \neq a_j$ for some i, j then $d(c(T^*), T^*) \geq d(t_i^*, t_j^*) > d(t_i^*, t_j^*) > n^* \cdot \ell$, contradicting the assumption of the lemma. If $a_i \neq a_j$, without loss of generality $a_i < a_j$. Then we can decompose $t_i^* = w_i^1 \circ z_i \circ w_i^2 \circ s_i^* \circ w_i^3$ and $t_j^* = w_j^1 \circ s_j \circ w_j^2 \circ s_j^* \circ w_j^3$ such that the following properties hold. The lengths of w_i^1, w_i^2 and w_i^3 equals the lengths of w_j^1, w_j^2 and w_j^3 respectively, z_i and z_j both have length ℓ, and w_i^1, w_i^2, w_i^3, w_j^1, w_j^2, w_j^3 are all substrings of w. Since $\ell \leq \frac{L_1}{4}$ we have that one of w_i^1, w_i^2, w_i^3 have length at least $\frac{L_1}{4}$. Without loss of

generality this is w_i^1. We have that $d(t_i^*, t_j^*) \geq d(w_i^1, w_j^1)$. Furthermore, since $a_i \neq a_j$ we have $w_i^1 \neq w_j^1$ and hence by Lemma 8 it follows that $d(w_i^1, w_j^1) \geq \lfloor \frac{\sqrt{L_1}}{16} \rfloor > n^* \cdot \ell$. But this implies that $d(t_i^*, t_j^*) > n^* \cdot \ell$. By the triangle inequality we have $d(c(T^*), T^*) \geq d(t_i^*, t_j^*) > n^* \cdot \ell$ yielding the desired contradiction. □

The construction, together with Lemmata 1, 9 and 10 yield the following result.

Lemma 11. Gap-Consensus Patterns *is W[1]-hard.*

Since an EPTAS for *Consensus Patterns* could be used to solve *Gap-Consensus Patterns* in time $f(\epsilon)(nL)^{O(1)}$, Lemma 11 implies our main result.

Theorem 2. Consensus Patterns *does not have an EPTAS unless FPT=W[1].*

4 Conclusions and Future Work

We have shown that *Consensus Patterns* does not admit an EPTAS unless FPT=W[1]. Our result rules out the possibility of a $(1+\epsilon)$ approximation algorithms with running time $f(1/\epsilon)n^{O(1)}$, while the best PTAS for *Consensus Patterns* has running time $n^{O(1/\epsilon^4)}$. Hence there is still a significant gap between the known upper and lower bounds, and obtaining tighter bounds warrants further investigation.

References

[1] N. Alon, O. Goldreich, J. Håstad and R. Peralta, Simple Construction of Almost k-wise Independent Random Variables. *Random Struct. Algor.*, 3(3): 289–304, 1992.

[2] S. Arora. Polynomial Time Approximation Schemes for Euclidean TSP and Other Geometric Problems. *Proc of 37th FOCS*, pages 2-11, 1996.

[3] S. Arora, Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and other Geometric Problems. *J. ACM*, 45, 5:753–782, 1998.

[4] B. Brejová, D.G. Brown, I.M. Harrower, and T. Vinar. New Bounds for Motif Finding in Strong Instances. *Proc. of 17th CPM*, pages 94–105, 2006.

[5] B. Brejová, D.G. Brown, I.M. Harrower, A. López-Ortiz and T. Vinar. Sharper Upper and Lower Bounds for an Approximation Scheme for Consensus-Pattern. *Proc. of 16th CPM*, pages 1–10, 2005.

[6] M. Cygan, F. V. Fomin, D. Lokshtanov, L. Kowalik, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms.* Springer, 2015. In press.

[7] C. Lo, B. Kakaradov, D. Lokshtanov, and C. Boucher. SeeSite: Efficiently Finding Co-occurring Splice Sites and Exon Splicing Enhancers. arXiv:1206.5846v1.

[8] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013.

[9] M.R. Fellows, J. Gramm, and R. Niedermeier. On the parameterized intractability of motif search problems. *Combinatorica*, 26:141–167, 2006.

[10] M.R. Fellows, D. Hermelin, F.A. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009

[11] J. Flum, and M. Grohe. Parameterized Complexity Theory. *Springer-Verlag*, 2006.

[12] F. Grimmett, and D. Stirzaker. Probability and random processes. *Oxford University Press*, 3 edition, 2001.

[13] W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *J. Amer. Statistical Assoc.*, 58(301): 13–30, 1963.

[14] H.B. Hunt III, M.V. Marathe, V. Radhakrishnan, S.S. Ravi, D.J. Rosenkrantz, and R.E. Stearns, NC-Approximation Schemes for NP- and PSPACE-Hard Problems for Geometric Graphs. *J. Algorithms*, 26(2):238–274, 1998

[15] J.K. Lanctot, M. Li, B. Ma, S. Wang, and L. Zhang. Distinguishing string selection problems. *Inform. Comput.*, 185(1):41–55, 2003. Preliminary version appeared in Proc. 10th SODA, pages 41-55, 1999.

[16] M. Li, B. Ma, and L. Wang. Finding similar regions in many sequences. *J. Comput. System Sci.*, 65(1):73–96, 2002.

[17] D. Marx. Closest Substring Problems with Small Distances. *SIAM J. Comput.*, 38(4):1283–1410, 2008.

[18] D. Marx. Efficient Approximation Schemes for Geometric Problems? *Proc. of 13th ESA*, 51(1): 448–459, 2005.

[19] D. Marx. Parameterized complexity and approximation algorithms. *Comput. J.*, 51(1): 60–78, 2008.

[20] J. Naor and M. Naor. Small-Bias Probability Spaces: Efficient Constructions and Applications. *SIAM J. Comput.*, 22(4): 838–856, 1993.

[21] R. Niedermeier. Invitation to Fixed-Parameter Algorithms. *Oxford University Press*, 2006.

[22] P. Pevzner and S. Sze. Combinatorial approaches to finding subtle signals in DNA strings. In *Proc. of the 8th ISMB*, pages 269–278, 2000.