Non-breaking Similarity of Genomes with Gene Repetitions*

Zhixiang Chen[1], Bin Fu[1], Jinhui Xu[2], Boting Yang[3], Zhiyu Zhao[4], and Binhai Zhu[5]

[1] Department of Computer Science, University of Texas-American, Edinburg, TX 78739-2999, USA
chen, binfu@cs.panam.edu

[2] Department of Computer Science, SUNY-Buffalo, Buffalo, NY 14260, USA
jinhui@cse.buffalo.edu

[3] Department of Computer Science, University of Regina, Regina, Saskatchewan, S4S 0A2, Canada
boting@cs.uregina.ca

[4] Department of Computer Science, University of New Orleans, New Orleans, LA 70148, USA
zzha2@cs.uno.edu.

[5] Department of Computer Science, Montana State University, Bozeman, MT 59717-3880, USA
bhz@cs.montana.edu.

Abstract. In this paper we define a new similarity measure, the *non-breaking similarity*, which is the complement of the famous breakpoint distance between genomes (in general, between any two sequences drawn from the same alphabet). When the two input genomes $\mathcal{G}$ and $\mathcal{H}$, drawn from the same set of n gene families, contain gene repetitions, we consider the corresponding Exemplar Non-breaking Similarity problem (ENbS) in which we need to delete repeated genes in $\mathcal{G}$ and $\mathcal{H}$ such that the resulting genomes G and H have the maximum non-breaking similarity. We have the following results.

- For the Exemplar Non-breaking Similarity problem, we prove that the Independent Set problem can be linearly reduced to this problem. Hence, ENbS does not admit any factor-$n^{1-\epsilon}$ polynomial-time approximation unless P=NP. (Also, ENbS is W[1]-complete.)
- We show that for several practically interesting cases of the Exemplar Non-breaking Similarity problem, there are polynomial time algorithms.

1 Introduction

In the genome comparison and rearrangement area, the breakpoint distance is one of the most famous distance measures [15]. The implicit idea of breakpoints was initiated as early as in 1936 by Sturtevant and Dobzhansky [14].

* This research is supported by Louisiana Board of Regents under contract number LEQSF(2004-07)-RD-A-35, NSF grant CCF-0546509, NSERC grant 261290-03 and Montana EPSCOR's Visiting Scholar's Program.

B. Ma and K. Zhang (Eds.): CPM 2007, LNCS 4580, pp. 119–130, 2007.

Until a few years ago, in genome rearrangement research, it is always assumed that every gene appears in a genome exactly once. Under this assumption, the genome rearrangement problem is in essence the problem of comparing and sorting signed/unsigned permutations [10,11]. In the case of breakpoint distance, given two perfect genomes (in which every gene appears exactly once, i.e., there is no gene repetition) it is easy to compute their breakpoint distance in linear time.

However, perfect genomes are hard to obtain and so far they can only be obtained in several small virus genomes. For example, perfect genomes do not occur on eukaryotic genomes where paralogous genes are common [12,13]. On the one hand, it is important in practice to compute genomic distances, e.g., using Hannenhalli and Pevzner's method [10], when no gene duplication arises; on the other hand, one might have to handle this gene duplication problem as well. In 1999, Sankoff proposed a way to select, from the duplicated copies of genes, the common ancestor gene such that the distance between the reduced genomes (*exemplar genomes*) is minimized [13]. A general branch-and-bound algorithm was also implemented in [13]. Recently, Nguyen, Tay and Zhang proposed using a divide-and-conquer method to compute the exemplar breakpoint distance empirically [12].

For the theoretical part of research, it was shown that computing the exemplar signed reversal and breakpoint distances between (imperfect) genomes are both NP-complete [1]. Two years ago, Blin and Rizzi further proved that computing the exemplar conserved interval distance between genomes is NP-complete [2]; moreover, it is NP-complete to compute the minimum conserved interval matching (i.e., without deleting the duplicated copies of genes). In [6,3] it was shown that the exemplar genomic distance problem does not admit any approximation (regardless of the approximation factor) unless P=NP, as long as $G = H$ implies that $d(G, H) = 0$, for any genomic distance measure $d(\ )$. This implies that for the exemplar breakpoint distance and exemplar conserved interval distance problems, there are no polynomial-time approximations. In [6] it was also shown that even under a weaker definition of (polynomial-time) approximation, the exemplar breakpoint distance problem does not admit any weak approximation of factor $n^{1-\epsilon}$ for any $0 < \epsilon < 1$, where n is the maximum length of the input genomes. In [3,4] it was shown that under the same definition of weak approximation, the exemplar conserved interval distance problem does not admit any weak approximation of a factor which is superlinear (roughly $n^{1.5}$).

In [5] three new kinds of genomic similarities were considered. These similarity measures, which are not distance measures, do not satisfy the condition that $G = H$ implies that $d(G, H) = 0$. Among them, the exemplar common interval measure problem seems to be the most interesting one. When gene duplications are allowed, Chauve, *et al.* proved that the problem is NP-complete and left open a question whether there is any inapproximability result for it.

In this paper, we define a new similarity measure called *non-breaking similarity*. Intuitively, this is the complement of the traditional breakpoint distance measure. Compared with the problem of computing exemplar breakpoint distance, which is a minimization problem, for the exemplar non-breaking similarity problem we need to maximize the number of non-breaking points. Unfortunately we show in this paper that Independent Set can be reduced to ENbS; moreover, this reduction implies that ENbS is W[1]-complete (and ENbS does not have a factor-n^ϵ polynomial-time approximation). This reduction works even when one of the two genomes is given exemplar.

The W[1]-completeness (see [8] for details) and the recent lower bound results [7] imply that if k is the optimal solution value, unless an unlikely collapse occurs in the parameterized complexity theory, ENbS is not solvable in time $f(k)n^{o(k)}$, for any function f. However, we show that for several practically interesting cases of the problem, there are polynomial time algorithms. This is done by parameterizing some quantities in the input genomes, followed with some traditional algorithmic techniques.

This effort is not artificial: in real-life datasets, usually there are some special properties in the data. For example, as reported in [12], the repeated genes in some bacteria genome pairs are often pegged, i.e., the repeated genes are usually separated by a peg gene which occurs exactly once. Our solution can help solving cases like these, especially when the number of such repeated genes is limited.

This paper is organized as follows. In Section 2, we go over the necessary definitions. In Section 3, we reduce Independent Set to ENbS, hence showing the inapproximability result. In Section 4, we present polynomial time algorithms for several practically interesting cases. In Section 5, we conclude the paper with some discussions.

2 Preliminaries

In the genome comparison and rearrangement problem, we are given a set of genomes, each of which is a signed/unsigned sequence of genes[1]. The order of the genes corresponds to the position of them on the linear chromosome and the signs correspond to which of the two DNA strands the genes are located. While most of the past research are under the assumption that each gene occurs in a genome once, this assumption is problematic in reality for eukaryotic genomes or the likes where duplications of genes exist [13]. Sankoff proposed a method to select an *exemplar genome*, by deleting redundant copies of a gene, such that in an exemplar genome any gene appears exactly once; moreover, the resulting exemplar genomes should have a property that certain genomic distance between them is minimized [13].

The following definitions are very much following those in [1,6]. Given n *gene families* (alphabet) $\mathcal{F}$, a genome $\mathcal{G}$ is a sequence of elements of $\mathcal{F}$. (Throughout this paper, we will consider unsigned genomes, though our results can be applied to signed genomes as well.) In general, we allow the repetition of a gene family in any genome. Each occurrence of a gene family is called a *gene*, though we will not try to distinguish a gene and a gene family if the context is clear.

[1] In general a genome could contain a set of such sequences. The genomes we focus on in this paper are typically called *singletons*.

The number of a gene g appearing in a genome $\mathcal{G}$ is called the occurrence of g in $\mathcal{G}$, written as $occ(g, \mathcal{G})$. A genome $\mathcal{G}$ is called *r-repetitive*, if all the genes from the same gene family occur at most r times in $\mathcal{G}$. For example, if $\mathcal{G} = abcbaa$, $\text{occ}(b, \mathcal{G}) = 2$ and $\mathcal{G}$ is a 3-repetitive genome.

For a genome $\mathcal{G}$, alphabet($\mathcal{G}$) is the set of all the characters (genes) that appear at least once in $\mathcal{G}$. A genome G is an exemplar genome of $\mathcal{G}$ if $\text{alphabet}(G) = \text{alphabet}(\mathcal{G})$ and each gene in alphabet($\mathcal{G}$) appears exactly once in G; i.e., G is derived from $\mathcal{G}$ by deleting all the redundant genes (characters) in $\mathcal{G}$. For example, let $\mathcal{G} = bcaadage$ there are two exemplar genomes: $bcadge$ and $bcdage$.

For two exemplar genomes G and H such that $\text{alphabet}(G) = \text{alphabet}(H)$ and $|\text{alphabet}(G)| = |\text{alphabet}(H)| = n$, a breakpoint in G is a two-gene substring $g_i g_{i+1}$ such that $g_i g_{i+1}$ is not a substring in H. The number of breakpoints in G (symmetrically in H) is called the *breakpoint distance*, denoted as $\text{bd}(G, H)$. For two genomes $\mathcal{G}$ and $\mathcal{H}$, their *exemplar breakpoint distance* $\text{ebd}(\mathcal{G}, \mathcal{H})$ is the minimum $\text{bd}(G, H)$, where G and H are exemplar genomes derived from $\mathcal{G}$ and $\mathcal{H}$.

For two exemplar genomes G and H such that $\text{alphabet}(G) = \text{alphabet}(H)$ $|\text{alphabet}(G)| = |\text{alphabet}(H)| = n$, a *non-breaking point* is a common two-gene substring $g_i g_{i+1}$ that appears in both G and H. The number of non-breaking points between G and H is also called the *non-breaking similarity* between G and H, denoted as $\text{nbs}(G, H)$. Clearly, we have $\text{nbs}(G, H) = n - 1 - \text{bd}(G, H)$. For two genomes $\mathcal{G}$ and $\mathcal{H}$, their *exemplar non-breaking similarity* $\text{enbs}(\mathcal{G}, \mathcal{H})$ is the maximum $\text{nbs}(G, H)$, where G and H are exemplar genomes derived from $\mathcal{G}$ and $\mathcal{H}$. Again we have $\text{enbs}(\mathcal{G}, \mathcal{H}) = n - 1 - \text{ebd}(\mathcal{G}, \mathcal{H})$.

The Exemplar Non-breaking Similarity (ENbS) Problem is formally defined as follows:

Instance: Genomes $\mathcal{G}$ and $\mathcal{H}$, each is of length $O(m)$ and each covers n identical gene families (i.e., at least one gene from each of the n gene families appears in both $\mathcal{G}$ and $\mathcal{H}$); integer K.
Question: Are there two respective exemplar genomes of $\mathcal{G}$ and $\mathcal{H}$, G and H, such that the non-breaking similarity between them is at least K?

In the next two sections, we present several results for the optimization versions of these problems, namely, to compute or approximate the maximum value K in the above formulation. Given a maximization problem Π, let the optimal solution of Π be OPT. We say that an approximation algorithm $\mathcal{A}$ provides a *performance guarantee* of α for Π if for every instance I of Π, the solution value returned by $\mathcal{A}$ is at least OPT/α. (Usually we say that $\mathcal{A}$ is a factor-α approximation for Π.) Typically we are interested in polynomial time approximation algorithms.

3 Inapproximability Results

For the ENbS problem, let O_{ENbS} be the corresponding optimal solution value. First we have the following lemma.

Lemma 1. $0 \leq O_{ENbS} \leq n-1$.

Proof. Let the n gene families be denoted by $1, 2, ..., n$. We only consider the corresponding exemplar genomes G, H. The lower bound of O_{ENbS} is achieved by setting $G = 123 \cdots (n-1)n$ and H can be set as follows: when n is even, $H = (n-1)(n-3) \cdots 531n(n-2) \cdots 642$; when n is odd, $H = (n-1)(n-3) \cdots 642n135 \cdots (n-4)(n-2)$. It can be easily proved that between G, H there is no non-breaking point. The upper bound of O_{ENbS} is obtained by setting $G = H$ in which case any two adjacent genes form a non-breaking point. □

The above lemma also implies that different from the Exemplar Breakpoint Distance (EBD) problem, which does not admit any polynomial-time approximation at all (as deciding whether the optimal solution value is zero is NP-complete), the same cannot be said on ENbS. Given $\mathcal{G}$ and $\mathcal{H}$, it can be easily shown that deciding whether $O_{ENbS} = 0$ can be done in polynomial time (hence it is easy to decide whether there exists some approximation for ENbS—for instance, as $O_{ENbS} \leq n-1$, if we can decide that $O_{ENbS} \neq 0$ then it is easy to obtain a factor-$O(n)$ approximation for ENbS). However, the next theorem shows that even when one of $\mathcal{G}$ and $\mathcal{H}$ is given exemplar, ENbS still does not admit a factor-$n^{1-\epsilon}$ approximation.

Theorem 1. *If one of $\mathcal{G}$ and $\mathcal{H}$ is exemplar and the other is 2-repetitive, the Exemplar Non-breaking Similarity Problem does not admit any factor $n^{1-\epsilon}$ polynomial time approximation unless P=NP.*

Proof. We use a reduction from Independent Set to the Exemplar Non-breaking Similarity Problem in which each of the n genes appears in $\mathcal{G}$ exactly once and in $\mathcal{H}$ at most twice. Independent Set is a well known NP-complete problem which cannot be approximated within a factor of $n^{1-\epsilon}$ [9].

Given a graph $T = (V, E), V = \{v_1, v_2, \cdots, v_N\}, E = \{e_1, e_2, \cdots, e_M\}$, we construct $\mathcal{G}$ and $\mathcal{H}$ as follows. (We assume that the vertices and edges are sorted by their corresponding indices.) Let A_i be the sorted sequence of edges incident to v_i. For each v_i we add v'_i as an additional gene and for each e_i we add x_i, x'_i as additional genes. We have two cases: $N+M$ is even and $N+M$ is odd. We mainly focus on the case when $N+M$ is even. In this case, the reduction is as follows.

Define $Y_i = v_i A_i v'_i$, if $i \leq N$ and $Y_{N+i} = x_i x'_i$, if $i \leq M$.

$\mathcal{G} : v_1 v'_1 v_2 v'_2 \cdots v_N v'_N x_1 e_1 x'_1 x_2 e_2 x'_2 \cdots x_M e_M x'_M$.

$\mathcal{H} : Y_{N+M-1} Y_{N+M-3} \cdots Y_1 Y_{N+M} Y_{N+M-2} \cdots Y_2$.

(Construct $\mathcal{H}$ as $Y_{N+M-1} Y_{N+M-3} \cdots Y_2 Y_{N+M} Y_1 Y_3 \cdots Y_{N+M-2}$ when $N+M$ is odd. The remaining arguments will be identical.)

We claim that T has an independent set of size k iff the exemplar non-breaking similarity between $\mathcal{G}$ and $\mathcal{H}$ is k. Notice that $\mathcal{G}$ is already an exemplar genome, so $G = \mathcal{G}$.

If T has an independent set of size k, then the claim is trivial. Firstly, construct the exemplar genome H as follows. For all i, if v_i is in the independent set, then delete A_i in $Y_i = v_i A_i v'_i$ (also arbitrarily delete all redundant edges in A_s in $\mathcal{H}$

for which v_s is not in the independent set of T). There are k non-breaking points between G, H—notice that any vertex v_i which is in the independent set gives us a non-breaking point $v_i v_i'$. The final exemplar genomes obtained, G and H, obviously have k exemplar non-breaking points.

If the number of the exemplar non-breaking points between $\mathcal{G}$ and $\mathcal{H}$ is k, the first thing to notice is that $Y_i = x_i x_i'$ $(N < i \leq N + M)$ cannot give us any non-breaking point. So the non-breaking points must come from $Y_i = v_i A_i v_i'$ $(i \leq N)$, with some A_i properly deleted (i.e., such a Y_i becomes $v_i v_i'$ in H). Moreover, there are exactly k such A_i's deleted. We show below that any two such completely deleted A_i, A_j correspond to two independent vertices v_i, v_j in T. Assume that there is an edge e_{ij} between v_i and v_j, then as both A_i, A_j are deleted, both of the two occurrences of the gene e_{ij} will be deleted from $\mathcal{H}$. A contradiction. Therefore, if the number of the exemplar non-breaking points between $\mathcal{G}$ and $\mathcal{H}$ is k, there is an independent set of size k in T.

To conclude the proof of this theorem, notice that the reduction take polynomial time (proportional to the size of T). □

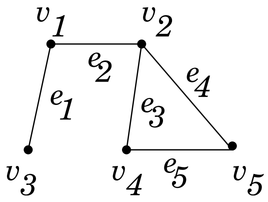

Fig. 1. Illustration of a simple graph for the reduction

In the example shown in Figure 1, we have
$\mathcal{G} : v_1v_1'v_2v_2'v_3v_3'v_4v_4'v_5v_5'x_1e_1x_1'x_2e_2x_2'x_3e_3x_3'x_4e_4x_4'x_5e_5x_5'$ and
$\mathcal{H} : x_4x_4'x_2x_2'v_5e_4e_5v_5'v_3e_1v_3'v_1e_1e_2v_1'x_5x_5'x_3x_3'x_1x_1'v_4e_3e_5v_4'v_2e_2e_3e_4v_2'$.
Corresponding to the optimal independent set $\{v_3, v_4\}$, we have
$H : x_4x_4'x_2x_2'v_5e_5v_5'v_3v_3'v_1e_1e_2v_1'x_5x_5'x_3x_3'x_1x_1'v_4v_4'v_2e_3e_4v_2'$. The two non-breaking points are $[v_3v_3']$, $[v_4v_4']$.

We comment that EBD and ENbS, even though complement to each other, are still different problems. With respect to the above theorem, when $\mathcal{G}$ is exemplar and $\mathcal{H}$ is not, there is a factor-$O(\log n)$ approximation for the EBD problem [6]. This is significantly different from ENbS, as shown in the above theorem.

4 Polynomial Time Algorithms for Some Special Cases

The proof of Theorem 1 also implies that ENbS is W[1]-complete, as Independent Set is W[1]-complete [8]. Following the recent lower bound results of Chen, *et al.*, if k is the optimal solution value for ENbS then unless an unlikely collapse occurs in the parameterized complexity theory, ENbS is not solvable in time $f(k)n^{o(k)}$, for any function f [7]. Nevertheless, we show below that for several practically interesting cases of the problem, there are polynomial time algorithms. The idea is to set a parameter in the input genomes (or sequences, as we will use interchangeably from now on) and design a polynomial time algorithm when such a parameter is $O(\log n)$.

In practical datasets, usually there are some special properties in the data. For instance, the repeated genes in the five bacteria genome pairs (Baphi-Wigg, Pmult-Hinft, Ecoli-Styphi, Xaxo-Xcamp and Ypes) are usually pegged, i.e., the repeated genes are usually separated by a peg gene which occurs exactly once [12]. When the total number of such repeated genes is a constant, our algorithm can solve this problem in polynomial time.

We first present a few extra definitions. For a genome $\mathcal{G}$ and a character g, $\text{span}(g, \mathcal{G})$ is the maximal distance between the two positions that are occupied by g in the genome $\mathcal{G}$. For example, if $\mathcal{G} = abcbaa$, $\text{span}(a, \mathcal{G}) = 5$ and $\text{span}(b, \mathcal{G}) = 2$. For a genome $\mathcal{G}$ and $c \geq 0$, we define $\text{totalocc}(c, \mathcal{G}) = \sum_{g \text{ is a character in } \mathcal{G} \text{ and } \text{span}(g,\mathcal{G}) \geq c} \text{occ}(g, \mathcal{G})$.

Assume that c and d are positive integers. A (c, d)*-even partition* for a genome $\mathcal{G}$ is $\mathcal{G} = \mathcal{G}_1\mathcal{G}_2\mathcal{G}_3$ with $|\mathcal{G}_2| = c$ and $|\mathcal{G}_1| + \lfloor |\mathcal{G}_2|/2 \rfloor = d$.

For a genome $\mathcal{G}$ and integers $c, d > 0$, a (c, d)-split G_1, G_2, G_3 for $\mathcal{G}$ is derived from a (c', d)-even partition $\mathcal{G} = \mathcal{G}_1\mathcal{G}_2\mathcal{G}_3$ for $\mathcal{G}$ for some $c \leq c' \leq 2c$ and satisfies the following conditions 1)-6):

(1) $\text{alphabet}(\mathcal{G}) = \text{alphabet}(G_1G_2G_3)$.

(2) We can further partition $\mathcal{G}_2$ into $\mathcal{G}_2 = \mathcal{G}_2^1\mathcal{G}_2^2\mathcal{G}_2^3$ such that $|\mathcal{G}_2^2| \leq c+1$, and there is at least one gene g with all its occurrences in $\mathcal{G}$ being in $\mathcal{G}_2^2$. We call such a gene g as a whole gene in $\mathcal{G}_2^2$.

(3) G_2 is obtained from $\mathcal{G}_2^2$ by deleting some genes and every gene appears at most once in G_2. And, G_2 contains one occurrence of every whole gene in $\mathcal{G}_2^2$.

(4) G_1 is obtained from $\mathcal{G}_1\mathcal{G}_2^1$ by deleting all genes in $\mathcal{G}_1\mathcal{G}_2^1$ which also appear in G_2.

(5) G_3 is obtained from $\mathcal{G}_2^3\mathcal{G}_3$ by deleting all genes in $\mathcal{G}_2^3\mathcal{G}_3$ which also appear in G_2.

(6) G_2 has no gene common with either G_1 or G_3.

Finally, for a genome $\mathcal{G}$ and integers $c, d \geq 0$, a (c, d)-decomposition is G_1x, G_2G_3, where G_1, G_2, G_3 is a (c, d)-split for $\mathcal{G}$ and x is the first character of G_2. We have the following lemma. From now on, whenever a different pair of genomes are given we assume that they are drawn from the same n gene families.

Lemma 2. *Assume that c, d are integers satisfying $c \geq 0$ and $|\mathcal{G}| - 2c \geq d \geq 2c$. and $\mathcal{G}$ is a genome with* $\text{span}(g, \mathcal{G}) \leq c$ *for every gene g in $\mathcal{G}$. Then, (1) the number of (c, d)-decompositions is at most 2^{c+1}; (2) every exemplar genome of $\mathcal{G}$ is also an exemplar genome of $G_1G_2G_3$ for some (c, d)-split G_1, G_2, G_3 of $\mathcal{G}$.*

Proof. (1). Since $\text{span}(g, \mathcal{G}) \leq c$ for every gene g in $\mathcal{G}$, it is easy to see that there is a c', $c \leq c' \leq 2c$, such that we can find (c, d)-splits G_1, G_2 and G_3 from a (c', d)-even partition $\mathcal{G} = \mathcal{G}_1\mathcal{G}_2\mathcal{G}_3$ with $\mathcal{G}_2 = \mathcal{G}_2^1\mathcal{G}_2^2\mathcal{G}_2^3$. Since $|\mathcal{G}_2^2| \leq c + 1$, there are at most 2^{c+1} possible ways to obtain G_2. Therefore, the total number of decompositions is at most 2^{c+1}. (2) is easy to see. □

Lemma 3. *Let c be a positive constant and ϵ be an arbitrary small positive constant. There exists an $O(n^{c+2+\epsilon})$-time algorithm such that given an exemplar genome G, in which each genes appears exactly once, and $\mathcal{H}$, in which* $\mathrm{span}(g, \mathcal{H}) \leq c$ *for every g in $\mathcal{H}$, it returns* $\mathrm{enbs}(G, \mathcal{H})$*.*

Proof. We use the divide-and-conquer method to compute $\mathrm{enbs}(G, \mathcal{H})$. The separator is put at the middle of $\mathcal{H}$ with width c. The genes within the region of separator are handled by a brute-force method.

```
Algorithm
$A(G, \mathcal{H})$
          Input: $G$ is a genome with no gene repetition,
                    and $\mathcal{H}$ is a genome such that $\mathrm{span}(g, \mathcal{H}) \leq c$ for each gene in $\mathcal{H}$.
          let $s = 0$ and $d = |\mathcal{H}|/2$.
          for every $(c, d)$-decomposition $H_1x, H_2H_3$ of $\mathcal{H}$)
                    begin
                              if the length of $H_1x$ and $H_2H_3$ is $\leq \log n$
                                        then compute $A(G, H_1x)$ and $A(G, H_2H_3)$ by brute-force;
                                        else let $s' = A(G, H_1x) + A(G, H_2H_3)$;
                              if $(s < s')$ then $s = s'$
                    end
          return $s$;
```

The correctness of the algorithm is easy to verify. By Lemma 2 and the description of the algorithm, the computational time is based on the following recursive equation: $T(n) \leq (2^{c+1}(2T(n/2 + c)) + c_0 n$, where c_0 is a constant. We show by induction that $T(n) \leq c_1 n^{c+2+\epsilon}$, where c_1 is a positive constant. The basis is trivial when n is small since we can select constant c_1 large enough. Assume that $T(n) \leq c_1 n^{c+2+\epsilon}$ is true all $n < m$.

$T(m) \leq 2^{c+1}(2T(m/2+c)+c_0 m \leq 2(2^{c+1}c_1(m/2+c)^{c+2+\epsilon})+c_0 m < c_1 m^{c+2+\epsilon}$ for all large m. □

We now have the following theorem.

Theorem 2. *Let $\mathcal{G}$ and $\mathcal{H}$ be two genomes with $t = \mathrm{totalocc}(1, \mathcal{G}) + \mathrm{totalocc}(c, \mathcal{H})$, for some arbitrary constant c. Then $\mathrm{enbs}(\mathcal{G}, \mathcal{H})$ can be computed in $O(3^{\lfloor t/3 \rfloor} n^{c+2+\epsilon})$ time.*

Proof. Algorithm:

```
          $d = 0$;
          for each gene $g_1$ in $\mathcal{G}$ with $\mathrm{span}(g_1, \mathcal{G}) \geq 1$
          begin
                    for each position $p_1$ of $g_1$ in $\mathcal{G}$
                    begin
                              remove all $g_1$'s at all positions other than $p_1$;
                    end
                    assume that $\mathcal{G}$ has been changed to $G$;
                    for each gene $g_2$ in $\mathcal{H}$ with $\mathrm{span}(g_2, \mathcal{H}) > c$
                    begin
                    for each position $p_2$ of $g_2$ in $\mathcal{H}$
                    begin
                         remove all $g_2$'s at all positions other than $p_2$;
                    end
                    assume that $\mathcal{H}$ has been changed to $\mathcal{H}'$;
                    compute $d_0 = \text{enbs}(G, \mathcal{H}')$ following Lemma 3;
                    if $(d < d_0)$ then $d = d_0$;
          end
     end
     return $d$;
```

Let g_i, $1 \leq i \leq m$, be the genes in $\mathcal{G}$ and $\mathcal{H}$ with $\text{span}(g_1, \mathcal{G}) \geq 1$ in $\mathcal{G}$ or $\text{span}(g_2, \mathcal{H}) > c$ in $\mathcal{H}$. We have $t = k_1 + \cdots + k_m$. Let k_i be the number of occurrences of g_i. Notice that $k_i \geq 2$. The number of cases to select the positions of those genes in $\mathcal{G}$ and the positions of those genes in $\mathcal{H}$ is at most $k_1 \cdots k_m$, which is at most $4 \cdot 3^{\lfloor t/3 \rfloor}$ following Lemma 6. In G, every gene appears exactly once. In $\mathcal{H}'$, every gene has span bounded by c. Therefore, their distance can be computed in $O(n^{c+2+\epsilon})$ steps by Lemma 3. □

Next, we define a new parameter measure similar to the Maximum Adjacency Disruption (MAD) number in [5].

Assume that $\mathcal{G}$ and $\mathcal{H}$ are two genomes/sequences. For a gene g, define $\text{shift}(g, \mathcal{G}, \mathcal{H}) = \max_{\mathcal{G}[i]=g, \mathcal{H}[j]=g} |i - j|$, where $\mathcal{G}[i]$ is the gene/character of $\mathcal{G}$ at position i. A *space-permitted* genome $\mathcal{G}$ may have space symbols in it. For two space-permitted genomes $\mathcal{G}_1$ and $\mathcal{G}_2$, a non-breaking point $g_1 g_2$ satisfies that g_1 and g_2 appear at two positions of $\mathcal{G}$ without any other genes/characters except some spaces between them, and also at two positions of $\mathcal{H}$ without any other genes except spaces between them.

For a genome $\mathcal{G}$ and integers $c, d > 0$, an exact (c, d)-split G_1, G_2, G_3 for $\mathcal{G}$ is obtained from a (c, d)-even partition $\mathcal{G} = \mathcal{G}_1 \mathcal{G}_2 \mathcal{G}_3$ for $\mathcal{G}$ and satisfies the following conditions (1)-(5):

(1) $\text{alphabet}(\mathcal{G}) = \text{alphabet}(G_1 G_2 G_3)$.

(2) G_2 is obtained from $\mathcal{G}_2$ by replacing some characters with spaces and every non-space character appears at most once in G_2.

(3) G_1 is obtained from $\mathcal{G}_1$ by changing all $\mathcal{G}$ characters that also appear in G_2 into spaces.

(4) G_3 is obtained from $\mathcal{G}_3$ by changing all $\mathcal{G}_3$ characters that also appear in G_2 into spaces.

(5) G_2 has no common non-space character with either G_1 or G_3.

We now show the following lemmas.

Lemma 4. *Let c, k, d be positive integers. Assume that $\mathcal{G}$ is a space-permitted genome with* $\text{span}(g, \mathcal{G}) \leq c$ *for every character g in $\mathcal{G}$, and $\mathcal{G}$ only has spaces at the first kc positions and spaces at the last kc positions. If $|\mathcal{G}| > 2(k+4)c$ and $(k+2)c < d < |\mathcal{G}| - (k+2)c$, then $\mathcal{G}$ has at least one exact $(2c, d)$-split and for every exact $(2c, d)$-split G_1, G_2, G_3 for $\mathcal{G}$, G_2 has at least one non-space character.*

Proof. For $(k+2)c < d < |\mathcal{G}|-(k+2)c$, it is easy to see that $\mathcal{G}$ has a subsequence S of length $2c$ that starts from the d-th position in $\mathcal{G}$ and has no space character. For every subsequence S of length $2c$ of $\mathcal{G}$, if S has no space character, it has at least one character in $\mathcal{G}$ that only appears in the region of S since $\text{span}(g, \mathcal{G}) \leq c$ for every character g in $\mathcal{G}$. □

Lemma 5. *Let c be a positive constant. There exists an $O(n^{2c+1+\epsilon})$ time algorithm such that, given two space-permitted genomes/sequences $\mathcal{G}$ and $\mathcal{H}$, it returns* $\text{enbs}(\mathcal{G}, \mathcal{H})$*, if* $\text{shift}(g, \mathcal{G}, \mathcal{H}) \leq c$ *for each non-space character g, $\mathcal{G}$ and $\mathcal{H}$ only have spaces at the first and last* $4c$ *positions, and* $|\mathcal{G}| \geq 16c$ *and* $|\mathcal{H}| \geq 16c$.

Proof. Since $\text{shift}(g, \mathcal{G}, \mathcal{H}) \leq c$ for every gene/character g in $\mathcal{G}$ or $\mathcal{H}$, we have $\text{span}(g, \mathcal{G}) \leq 2c$ and $\text{span}(g, \mathcal{H}) \leq 2c$ for every character g in $\mathcal{G}$ or $\mathcal{H}$.

```
Algorithm
    $B(\mathcal{G}, \mathcal{H})$
    Input: $\mathcal{G}, \mathcal{H}$ are two space-permitted genomes.
    assume that $|\mathcal{G}| \leq |\mathcal{H}|$;
    set $s = 0$ and $d = \lfloor |\mathcal{G}|/2 \rfloor$;
    for every exact $(2c, d)$-split $G_1, G_2, G_3$ of $\mathcal{G}$
    begin
        for every exact $(2c, d)$-split $H_1, H_2, H_3$ of $\mathcal{H}$
        begin
            if the length of $\mathcal{G}$ and $\mathcal{H}$ is $\leq \log n$
                then compute $\text{enbs}(\mathcal{G}, \mathcal{H})$ by brute-force;
                else $s = B(G_1G_2, H_1H_2) + B(G_2G_3, H_2H_3) - B(G_2, H_2)$;
            if $(s < s')$ then $s = s'$;
        end
    end
    return $s$;
```

Following the divide-and-conquer method, it is easy to see that G_1G_2, H_1H_2, G_2G_3 and H_2H_3 have spaces in the first and last $2c$ positions. This is because $\text{span}(g, \mathcal{G}) \leq 2c, \text{span}(g, \mathcal{H}) \leq 2c$ for every character g. $B(G_2, H_2)$ can be determined by a linear scan, since both of them are exemplar. The computational time is determined by the recurrence relation: $T(n) = (2^{2c} + 2c)(2T(\frac{n}{2} + 2c) + O(n))$, which has solution $T(n) = O(n^{2c+1+\epsilon})$ as we show in the Lemma 3. □

Lemma 6. *Let $k \geq 3$ be a fixed integer. Assume that $k_1, k_2, \cdots, k_m$ are m integers that satisfies $k_i \geq 2$ for $i = 1, 2, \cdots, m$ and $k_1 + k_2 + \cdots + k_m = k$. Then $k_1k_2 \cdots k_m \leq 4 \cdot 3^{\lfloor \frac{k}{3} \rfloor}$.*

Proof. We assume that for fixed k, m is the largest integer that makes the product $k_1k_2 \cdots k_m$ maximal and $k_1 + k_2 + \cdots + k_m = k$. We claim that $k_i \leq 3$ for all $i = 1, 2, \cdots, m$. Otherwise, without loss of generality, we assume that $k_m > 3$. Clearly, $2 \cdot (k_m - 2) \geq k_m$. Replace k_m by $k'_m = 2$ and $k_{m+1'} = k_m - 2$. We still have that $k_1 + k_2 + \cdots + k_{m-1} + k'_m + k'_{m+1} = k$ and $k_1k_2 \cdot k_{m-1}k'_mk'_{m+1} \geq k_1k_2 \cdots k_m$. This contradicts that m is maximal. Therefore, each $k_i (i = 1, 2, \cdots, m)$ is either 2 or 3 while $k_1 + k_2 + \cdots + k_{m-1} + k_m = k$

and $k_1k_2\cdots k_m$ is still maximal. It is impossible that there are at least three 2s among $k_1, k_2, \cdots, k_m$. This is because that $2+2+2 = 3+3$ and $2\cdot 2\cdot 2 < 3\cdot 3$. On the other hand, the number of 3s among $k_1, k_2, \cdots, k_m$ is at most $\lfloor \frac{k}{3} \rfloor$ since $k_1 + k_2 + \cdots + k_{m-1} + k_m = k$. □

Finally, we have the following theorem.

Theorem 3. *Let $\mathcal{G}$ and $\mathcal{H}$ be two genomes with a total of t genes g satisfying* $\text{shift}(g, \mathcal{G}, \mathcal{H}) > c$*, for some arbitrary positive constant c. Then* $\text{enbs}(\mathcal{G}, \mathcal{H})$ *can be computed in $O(3^{\lfloor t/3 \rfloor} n^{2c+1+\epsilon})$ time.*

The idea to prove this theorem is as follows. We consider all possible ways to replace every gene g, $\text{shift}(g, \mathcal{G}, \mathcal{H}) > c$, with space in $\mathcal{G}$ and $\mathcal{H}$, while keeping one occurrence of g in $\mathcal{G}$ and $\mathcal{H}$. For each pair of such resulting $\mathcal{G}'$ and $\mathcal{H}'$, we consider to use the algorithm in Lemma 5 to compute $\text{enbs}(\mathcal{G}', \mathcal{H}')$. Notice that we may have spaces not only in the two ends but also in the middle of $\mathcal{G}'$ or $\mathcal{H}'$. However, we can modify the method of selecting exact (c, d)-splits for the two genome. The new method is to start at the middle position of $\mathcal{G}'$ (or $\mathcal{H}'$) to find the nearest non-space gene either in the right part or the left of the middle position. Say, such a gene is u in the right part of the middle position of $\mathcal{H}'$. Then, we determine $\mathcal{H}_2$ by including c positions to the right of u and also including c or more positions to the left to make sure that the middle position is also included. The rest part in the left of $\mathcal{H}_2$ is $\mathcal{H}_1$, and the rest in the right of $\mathcal{H}_2$ is $\mathcal{H}_3$. It is easy to see that the number of genes (not spaces) in $\mathcal{H}_2$ is no more than $2c$. Similarly, we can determine an even partition for $\mathcal{G}_1$. Notice also that spaces do not contribute to constructing exact (c, d)-splits. Therefore, $\text{enbs}(\mathcal{G}', \mathcal{H}')$ can be computed, following the spirit of the algorithm in Lemma 5.

5 Concluding Remarks

We define a new measure—non-breaking similarity of genomes and prove that the exemplar version of the problem does not admit an approximation of factor $n^{1-\epsilon}$ even when one of the input genomes is given exemplar; and moreover, the problem is W[1]-complete. This differs from the corresponding result for the dual exemplar breakpoint distance problem, for which a factor-$O(\log n)$ approximation exists when one of the input genomes is exemplar (and for the general input there is no polynomial time approximation) [6]. On the other hand, we present polynomial time algorithms for several practically interesting cases under this new similarity measure. In practice, the practical datasets usually have some special properties [12], so our negative results might not hold and our positive results might be practically useful. We are currently working along this line.

References

1. Bryant, D.: The complexity of calculating exemplar distances. In: Sankoff, D., Nadeau, J. (eds.) Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment, and the Evolution of Gene Families, pp. 207–212. Kluwer Acad. Pub. Boston, MA (2000)

2. Blin, G., Rizzi, R.: Conserved interval distance computation between non-trivial genomes. In: Wang, L. (ed.) COCOON 2005. LNCS, vol. 3595, pp. 22–31. Springer, Heidelberg (2005)
3. Chen, Z., Fu, B., Fowler, R., Zhu, B.: Lower bounds on the application of the exemplar conserved interval distance problem of genomes. In: Chen, D.Z., Lee, D.T. (eds.) COCOON 2006. LNCS, vol. 4112, pp. 245–254. Springer, Heidelberg (2006)
4. Chen, Z., Fu, B., Fowler, R., Zhu, B.: On the inapproximability of the exemplar conserved interval distance problem of genomes. J. Combinatorial Optimization (to appear)
5. Chauve, C., Fertin, G., Rizzi, R., Vialette, S.: Genomes containing duplicates are hard to compare. In: Proc. 2nd Intl. Workshop on Bioinformatics Research and Applications (IWBRA'06), LNCS 3992, pp. 783–790 (2006)
6. Chen, Z., Fu, B., Zhu, B.: The approximability of the exemplar breakpoint distance problem. In: Cheng, S.-W., Poon, C.K. (eds.) AAIM 2006. LNCS, vol. 4041, pp. 291–302. Springer, Heidelberg (2006)
7. Chen, J., Huang, X., Kanj, I., Xia, G.: Linear FPT reductions and computational lower bounds. In: Proc. 36th ACM Symp. on Theory Comput. (STOC'04), pp. 212–221 (2004)
8. Downey, R., Fellows, M.: Parameterized Complexity. Springer, Heidelberg (1999)
9. Hästad, J.: Clique is hard to approximate within $n^{1-\epsilon}$. Acta. Mathematica, 182, 105–142 (1999)
10. Hannenhalli, S., Pevzner, P.: Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. J. ACM, 46(1), 1–27 (1999)
11. Gascuel, O. (ed.): Mathematics of Evolution and Phylogeny. Oxford University Press, Oxford, UK (2004)
12. Nguyen, C.T., Tay, Y.C., Zhang, L.: Divide-and-conquer approach for the exemplar breakpoint distance. Bioinformatics 21(10), 2171–2176 (2005)
13. Sankoff, D.: Genome rearrangement with gene families. Bioinformatics 16(11), 909–917 (1999)
14. Sturtevant, A., Dobzhansky, T.: Inversions in the third chromosome of wild races of drosophila pseudoobscura, and their use in the study of the history of the species. In: Proc. Nat. Acad. Sci. USA, vol. 22 pp. 448–450 (1936)
15. Watterson, G., Ewens, W., Hall, T., Morgan, A.: The chromosome inversion problem. J. Theoretical Biology 99, 1–7 (1982)