The Approximability of the Exemplar Breakpoint Distance Problem *

Zhixiang Chen † Bin Fu ‡ Binhai Zhu §

Abstract

In this paper we present the first set of approximation/inapproximability results for the Exemplar Breakpoint Distance Problem. Our inapproximability results hold for the simplest case between only two genomes $\mathcal{G}$ and $\mathcal{H}$, each containing only one sequence of genes (possibly with repetitions).

- For the general Exemplar Breakpoint Distance Problem, we prove that the problem does not admit any approximation unless P=NP; in fact, this result holds even when a gene appears in $\mathcal{G}$ ($\mathcal{H}$) at most three times.
- Even on a weaker definition of approximation (which we call weak approximation), we show that the problem does not admit a weak approximation with a factor $m^{1-\epsilon}$, where m is the maximum length of $\mathcal{G}$ and $\mathcal{H}$.
- We present a factor-$2(1+\log n)$ approximation for an interesting special case, namely, one of the two genomes is a *k-span* genome (i.e., all genes in the same gene family are within a distance $k = O(\log n)$), where n is the number of gene families in $\mathcal{G}$ and $\mathcal{H}$.

Keywords: Approximation Algorithms, Exemplar Genomic Distance, NP-hardness, Breakpoint, Reversal

*This research is supported by Louisiana Board of Regents under contract number LEQSF(2004-07)-RD-A-35 and MSU-Bozeman's Short-term Professional Development Leave Program.

†Department of Computer Science, University of Texas-American, Edinburg, TX 78739-2999, USA. Email: chen@cs.panam.edu.

‡Department of Computer Science, University of New Orleans, New Orleans, LA 70148 and Research Institute for Children, 200 Henry Clay Avenue, New Orleans, LA 70118, USA. Email: fu@cs.uno.edu.

§Department of Computer Science, Montana State University, Bozeman, MT 59717-3880, USA. Email: bhz@cs.montana.edu.

1 Introduction

In the genome comparison and rearrangement area, a standard problem is to compute the number (i.e., genetic distances) and the actual sequence of genetic operations needed to convert a source genome to a target genome. This problem is important in evolutionary molecular biology. Typical genetic distances include edit [15], signed reversal [18, 16, 1] and breakpoint [23], etc. (The idea of signed reversal and, implicitly, breakpoint, was initiated as early as in 1936 by Sturtevant and Dobzhansky [21].) Recently, conserved interval distance was also proposed to measure the similarity of multiple sequences of genes [4]. (Interested readers are referred to [11, 12] for a summary of the research performed in this area.)

Until very recently, in genome rearrangement research, it is always assumed that each gene appears in a genome exactly once. Under this assumption, the genome rearrangement problem is in essence the problem of comparing and sorting signed permutations [11, 12]. However, this assumption is very restrictive and is only justified in several small virus genomes. For example, this assumption does not hold on eukaryotic genomes where paralogous genes exist [17, 20]. On the one hand, it is important in practice to compute genomic distances, e.g., Hannenhalli and Pevzner's method [11], when no gene duplications arise; 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 [20]. A general branch-and-bound algorithm was also implemented in [20]. Recently, Nguyen, Tay and Zhang proposed to use a divide-and-conquer method to compute the exemplar breakpoint distance empirically [17].

For the theoretical part of research, it was shown that computing the signed reversals and breakpoint distances between exemplar genomes are both NP-complete [2]. Recently, Blin and Rizzi further proved that computing the conserved interval distance between exemplar genomes is NP-complete [3]; moreover, it is NP-complete to compute the minimum conserved interval matching (i.e., without deleting the duplicated copies of genes). Before this work, there has been no formal theoretical results on the approximability of the exemplar genomic distance problems except the NP-completeness proofs [2, 3].

In this paper, we present the first set of inapproximability/approximation results for the Exemplar Breakpoint Distance problem, given two genomes each containing only one sequence of genes drawn from n identical gene families. (Some of the results hold subsequently for the Exemplar Reversal Distance problem.) For the One-sided Exemplar Breakpoint Distance Problem, which is also known to be NP-complete, we obtain a factor-$2(1 + \log n)$, polynomial-time approximation. The approximation algorithm follows the greedy strategy for Set-Cover, but constructing the family of sets is non-trivial and is related to a new problem of *longest constrained common subsequences* which is related to but different from the recently studied *constrained longest common subsequences* [5].

The paper is organized as follows. In Section 2, we present formal definitions of all the necessary concepts. In Section 3, we present several inapproximability results. In Section 4, we extend some inapproximability result to a weak approximation model. In Section 5, we present the details of the $2(1 + \log n)$-approximation for the One-sided Exemplar Breakpoint Distance Problem. In Section 6, we conclude the paper with several interesting open problems.

2 Preliminaries

In the genome comparison and rearrangement problem, we are given a set of genomes, each of which is a signed 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 [20]. 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 genetic distance between them is minimized [20].

The following definitions are very much following those in [3]. Given n *gene families* (alphabet) $\mathcal{F}$, a genome $\mathcal{G}$ is a sequence of elements of $\mathcal{F}$ such that each element is with a sign (+ or −). 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. Given a genome $G = g_1 g_2 ... g_m$ with no repetition of any gene, we say that gene g_i *immediately precedes* g_j if $j = i + 1$. Given genomes G, H, if gene a immediately precedes b in G and neither a immediately precedes b nor $-b$ immediately precedes $-a$ in H, then they constitute a *breakpoint* in G. The *breakpoint distance* is the number of breakpoints in G (symmetrically, it is the number of breakpoints in H).

The number of a gene g appearing in a genome $\mathcal{G}$ is called the cardinality of g in $\mathcal{G}$, written as $card(g, \mathcal{G})$. A gene in $\mathcal{G}$ is called *trivial* if g has cardinality exactly 1; otherwise, it is called *non-trivial*. In this paper, we assume that all the genomes we discuss could contain both trivial and non-trivial genes. A genome $\mathcal{G}$ is called *r-repetitive*, if all the genes from the same gene family appear at most r times in $\mathcal{G}$. A genome $\mathcal{G}$ is called a *k-span* genome, if all the genes from the same gene family are within distance at most k in $\mathcal{G}$. For example, $\mathcal{G} = -adc - bdaeb$ is 2-repetitive and it is a 5-span genome.

Given a genome $\mathcal{G} = g_1 g_2 \cdots g_m$, an interval $[g_i, g_j]$ is simply the substring $g_i g_{i+1} \cdots g_j$ (which will also be denoted as $\mathcal{G}[i, j]$). Example: given $\mathcal{G}' = bdc - ag - e - fh, \mathcal{G}'' = bdce - gafh$, between the interval $I_1 = dc - ag - e - f, I_2 = dce - gaf$, there are 2 breakpoints. A *signed reversal* on a genome $\mathcal{G}$ simply reverses the order and signs of all the elements in an interval of $\mathcal{G}$. In the previous example, if a signed reversal operation is conducted on I_1 then we obtain a new genome $\mathcal{G}^* = bfe - ga - c - dh$. (All the reversals concerned in this paper are signed reversals. Henceforth, we simply use *reversal* to make the presentation simpler.) The *reversal distance* between genomes G and H is the minimum number of reversals to transfer G into H.

Given a genome $\mathcal{G}$ over $\mathcal{F}$, an *exemplar genome* of $\mathcal{G}$ is a genome G' obtained from $\mathcal{G}$ by deleting duplicating genes such that each gene family in $\mathcal{G}$ appears exactly once in G'. For example, let $\mathcal{G} = bcaadagef$ there are two exemplar genomes: $bcadgef$ and $bcdagef$.

The Exemplar Breakpoint (Reversal) Distance Problem is defined as follows:

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

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 breakpoint (reversal) distance between them is at most K?

In the next three sections, we present inapproximability/approximation results for the optimization versions of these problems, namely, to compute/approximate the minimum value K in the above formulation. Given a minimization 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 most $\alpha \times OPT$. (Usually we say that $\mathcal{A}$ is a factor-α approximation for Π.) Typically we are interested in polynomial time approximation algorithms.

In many biological problems, the optimal solution value OPT could be zero. (For example, in some minimum recombination haplotype reconstruction problems the optimal solution could be zero.) In that case, if computing such a zero optimal solution value is NP-complete then the problem does not admit *any* approximation (unless P=NP). However, in reality one would be happy to obtain a solution with value one or two. Due to this reason, we relax the above (traditional) definition of approximation to a *weak approximation*. Given a minimization problem Π, let the optimal solution of Π be OPT. We say that a weak approximation algorithm $\mathcal{B}$ provides a *performance guarantee* of α for Π if for every instance I of Π, the solution value returned by $\mathcal{B}$ is at most $\alpha \times (OPT + 1)$.

3 Inapproximability Bounds

In this section, we present a series of inapproximability bounds on the Exemplar Breakpoint Distance Problem.

Theorem 3.1 If both $\mathcal{G}$ and $\mathcal{H}$ are 2-repetitive genomes, then the Exemplar Breakpoint Distance Problem cannot be approximated within a factor 1.36.

Proof. We use a reduction from Vertex Cover to the Exemplar Breakpoint Distance Problem in which each gene appears in $\mathcal{G}$ ($\mathcal{H}$) at most twice. Dur and Safra proved that Vertex Cover cannot be approximated within a factor 1.36 [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 and $-A_i$ be the signed reversal of A_i. (# is not a gene and is used only for the readability purpose.)

$\mathcal{G} : A_1 \# A_2 \# \cdots \# A_{n-1} \# A_n$

$\mathcal{H} : -A_1 \# -A_2 \# \cdots \# -A_{n-1} \# -A_n$

We claim that T has a vertex cover of size K iff the exemplar breakpoint distance between $\mathcal{G}$ and $\mathcal{H}$ is $K - 1$.

If T has a vertex cover of size K, then the claim is trivial. Firstly, construct the exemplar genomes G, H as follows. For all i, if v_i is in the vertex cover, then leave A_i in $\mathcal{G}$ and $-A_i$ in $\mathcal{H}$ and delete all $A_j, -A_j$ in $\mathcal{G}, \mathcal{H}$ for which v_j is not in the vertex cover of T. Finally, if e_i appears twice in the current genomes $\mathcal{G}$ and $\mathcal{H}$, say in A_s, A_t, then delete one copy of e_i in either A_s or A_t arbitrarily (say in A_s), and delete the corresponding copy of $-e_i$ in $-A_s$. The final exemplar

genomes obtained, G and H, obviously have a breakpoint distance of $K-1$. In fact, a breakpoint in G, H can only occur at the # positions—between some A_i and A_j in G ($-A_i$ and $-A_j$ in H).

If the exemplar breakpoint distance between $\mathcal{G}$ and $\mathcal{H}$ is $K-1$, the first thing to notice is that there is no breakpoint in A_i and $-A_i$; in other words, deleting e_j in A_i inconsistently (say, by deleting e_j in A_i and deleting $-e_j$ in $-A_s$ instead of in $-A_i$) would increase the number of breakpoints in the exemplar genomes G, H. Therefore, we can obtain a pair of exemplar genomes G, H by enforcing the breakpoints to be in between A_i and A_j in G (and symmetrically, $-A_i$ and $-A_j$ in H), with all redundant edges between them deleted. Clearly, the remaining A_i's in G (and $-A_i$'s in H) correspond to a vertex cover of size K in T.

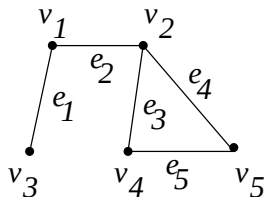

Figure 1. Illustration of a simple graph for the reduction.

In the example shown in Figure 1, we have
$\mathcal{G}: e_1e_2\#e_2e_3e_4\#e_1\#e_3e_5\#e_4e_5$ and
$\mathcal{H}: -e_2-e_1\#-e_4-e_3-e_2\#-e_1\#-e_5-e_3\#-e_5-e_4$.
Corresponding to the optimal vertex cover $\{v_1, v_2, v_5\}$, we have $G: e_1e_2\#e_3e_4\#e_5$ and $H: -e_2-e_1\#-e_4-v_3\#-e_5$. □

Corollary 3.2 If both $\mathcal{G}$ and $\mathcal{H}$ are 2-repetitive, then the Exemplar Reversal Distance Problem cannot be approximated within a factor 1.36.

Proof. Construction is the same as above. The claim that T has a vertex cover of size K iff the exemplar reversal distance between $\mathcal{G}$ and $\mathcal{H}$ is K can be proved similarly. □

In [17] it was claimed that the Exemplar Breakpoint Distance Problem cannot be approximated within a constant factor. But the proof, which was included in Nguyen's thesis, in fact implies a stronger $c \log n$ inapproximability bound as the reduction was from Set Cover. We extend Theorem 3.1 below to obtain a much simpler and clean proof of the $c \log n$ inapproximability bound, even though this is not the strongest inapproximability bound in this section.

Corollary 3.3 The Exemplar Breakpoint Distance Problem cannot be approximated within a factor $c \log n$, for some constant $c > 0$.

Proof. Similar to the proof of Theorem 3.1, we use a reduction from Dominating Set to the Exemplar Breakpoint Distance Problem in which each gene appears in $\mathcal{G}$ ($\mathcal{H}$) as many as $n-1$ times. Raz and Safra proved that Dominating Set cannot be approximated within a factor $c \log n$, for some $c > 0$ [19].

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 B_i be the sorted sequence of vertices incident to v_i and $-B_i$ be the signed reversal of B_i. (# is not a gene and is again used only for the readability purpose.)

$\mathcal{G}: v_1B_1\#v_2B_2\#\cdots\#v_{n-1}B_{n-1}\#v_nB_n$

$\mathcal{H}: -B_1 - v_1\# - B_2 - v_2\# \cdots \# - B_{n-1} - v_{n-1}\# - B_n - v_n$

We claim that T has a dominating set of size K iff the exemplar breakpoint distance between $\mathcal{G}$ and $\mathcal{H}$ is $K-1$.

If T has a dominating set of size K, then the claim is again trivial. Firstly, construct the exemplar genomes G, H as follows. For all i, if v_i is in the dominating set, then leave v_iB_i in $\mathcal{G}$ and $-B_i - v_i$ in $\mathcal{H}$ and delete all other $v_jA_j, -A_j - v_j$ in $\mathcal{G}, \mathcal{H}$ for which v_j is not in the dominating set of T. Finally, if v_i appears x times in the current genomes $\mathcal{G}$ and $\mathcal{H}$, then arbitrarily delete $x-1$ copies of v_i in all v_sB_s which contains v_i, and delete the corresponding copy of $-v_i$ in $-B_s - v_s$. The final exemplar genomes obtained, G and H, obviously have a breakpoint distance of $K-1$. In fact, a breakpoint in G, H can only occur at the # positions—between some v_iB_i and v_jB_j in G ($-B_i - v_i$ and $-B_j - v_j$ in H).

If the exemplar breakpoint distance between $\mathcal{G}$ and $\mathcal{H}$ is $K-1$, the first thing to notice is that there is no breakpoint in v_iB_i and $-B_i - v_i$; in other words, deleting v_j in v_iB_i inconsistently (say, by deleting v_j in v_iB_i and deleting $-v_j$ in $-B_s - v_s$ instead of in $-B_i - v_i$) would increase the number of breakpoints in the exemplar genomes G and H. Therefore, we can obtain a pair of exemplar genomes G and H by enforcing the breakpoints to be in between v_iB_i, v_jB_j in G (and symmetrically, $-B_i - v_i, -B_j - v_j$ in H), with all redundant v_l's deleted. Clearly, the remaining v_iB_i's in G (and $-B_i - v_i$'s in H) correspond to a dominating set of size K in T.

In the example shown in Figure 1, we have

$\mathcal{G}: v_1v_2v_3\#v_2v_1v_4v_5\#v_3v_1\#v_4v_2v_5\#v_5v_2v_4$ and

$\mathcal{H}: -v_3 - v_2 - v_1\# - v_5 - v_4 - v_1 - v_2\# - v_1 - v_3\# - v_5 - v_2 - v_4\# - v_4 - v_2 - v_5$.

Corresponding to the optimal dominating set $\{v_1, v_4\}$, we have $G: v_1v_2v_3\#v_4v_5$ and $H: -v_3 - v_2 - v_1\# - v_5 - v_4$. □

Corollary 3.4 The Exemplar Reversal Distance Problem cannot be approximated within a factor $c \log n$, for some constant $c > 0$.

Proof. Construction is the same as above. The claim that T has a dominating set of size K iff the exemplar reversal distance between $\mathcal{G}$ and $\mathcal{H}$ is K can be proved similarly. □

Next, we show an even stronger negative result for the Exemplar Breakpoint Distance Problem; namely, deciding whether the exemplar distance between $\mathcal{G}$ and $\mathcal{H}$ is zero is NP-complete. This implies that for the Exemplar Breakpoint Distance Problem there is no approximation unless P=NP. From now on we simply call this problem the *zero breakpoint distance (ZBD)* problem.

Theorem 3.5 Deciding if two genomes $\mathcal{G}$ and $\mathcal{H}$ have zero breakpoint distance is NP-complete.

Proof. We construct a reduction from the SAT problem [10] to the ZBD problem.

Let $F = f_1 \bigwedge f_2 \bigwedge \cdots \bigwedge f_q$ be a conjunctive normal form, where each sub-formula f_i is a disjunctive clause like $(x_2 \bigvee x_5 \bigvee \neg x_7)$. We construct a pair of sequences $\mathcal{G}$ and $\mathcal{H}$ such that F is satisfiable iff $\mathcal{G}$ and $\mathcal{H}$ have breakpoint distance zero.

Assume that $x_1, x_2, \cdots, x_n$ are the boolean variables in the formula F. For each variable x_i, we construct two sequences S_i and S_i^*. Let $f_{i_1}, \cdots, f_{i_u}$ be the sub-formulas in F that contains x_i, and let $f_{j_1}, \cdots, f_{j_v}$ be the sub-formulas of F that contains $\neg x_i$. Let $S_i = f_{i_1} \cdots f_{i_u} f_{j_1} \cdots f_{j_v}$ and $S_i^* = f_{j_1} \cdots f_{j_v} f_{i_1} \cdots f_{i_u}$, where $f_1, \cdots, f_q$ are considered as the names of q genes in $\mathcal{G}$ and $\mathcal{H}$.

Let $\mathcal{G} = S_1 g_1 S_2 g_2 \cdots g_{n-1} S_n$ and $\mathcal{H} = S_1^* g_1 S_2^* g_2 \cdots g_{n-1} S_n^*$, where $g_1, \cdots, g_n$ are (peg) genes that occur only once in $\mathcal{G}$ or $\mathcal{H}$.

Assume that $x_1 = b_1, \cdots, x_n = b_n$ are assignments that make F true. If $b_i = 1$, adjust both S_i and S_i^* to $S_i' = f_{i_1} \cdots f_{i_u}$ and $S_i^{*'} = f_{i_1} \cdots f_{i_u}$, respectively. If $b_i = 0$, adjust both S_i and S_i^* to

$S_i' = f_{j_1} \cdots f_{j_v}$ and $S_i^{*'} = f_{j_1} \cdots f_{j_v}$, respectively. It is easy to see that $G' = S_1' g_1 S_2' \cdots S_{n-1}' g_{n-1} S_n'$ is the same as $H' = S_1^{*'} g_1 S_2^{*'} \cdots S_{n-1}^{*'} g_{n-1} S_n^{*'}$. Since the assignments make F true, each sub-formula $f_t \in \{f_1, \cdots, f_q\}$ is true due to $x_i = b_i$ for some i. That is, f_t must occur in S_i and S_i^*. If f_t occurs more than once in G' and H' then we can delete their corresponding occurrences in G' and H'. Finally, notice that both G' and H' contain all $q + n - 1$ genes in $\{f_1, \cdots, f_q, g_1, \cdots, g_{n-1}\}$.

Assume that $\mathcal{G}$ is converted into G'' and $\mathcal{H}$ is converted into H'' via removing some genes such that $G'' = H''$ and they contain all genes in $\{f_1, \cdots, f_q, g_1, \cdots, g_{n-1}\}$. Let $S_i^{"}$ and $S_i^{*"}$ be the substrings in G'' and H'' with respect to S_i and S_i^* in $\mathcal{G}$ and $\mathcal{H}$ respectively. This implies that $S_i^{"}$ and $S_i^{*"}$ are the common subsequence of either $f_{i_1} \cdots f_{i_u}$ or $f_{j_1} \cdots f_{j_v}$, because $S_i = f_{i_1} \cdots f_{i_u} f_{j_1} \cdots f_{j_v}$ and $S_i^* = f_{j_1} \cdots f_{j_v} f_{i_1} \cdots f_{i_u}$. If $S_i^{"}$ is empty then we can assign a value to x_i arbitrarily. If $S_i^{"}$ is not empty and it is a subsequence of $f_{i_1} \cdots f_{i_u}$ then we assign $x_i = 1$. If $S_i^{"}$ is not empty and it is a subsequence of $f_{j_1} \cdots f_{j_v}$ then we assign $x_i = 0$. As each $f_t \in \{f_1, \cdots, f_q\}$ occurs in G'', H'' once, it must occur in a non-empty $S_i^{"}$. It is easy to see that F is true by the assignments to those variables $x_1, \cdots, x_n$.

The reduction takes linear (in the length of F, $|F|$) time. A sub-formula f_j with y literals appears in $\mathcal{G}$ ($\mathcal{H}$) exactly y times and there are $n - 1$ additional peg genes in $\mathcal{G}$ ($\mathcal{H}$). Therefore, the length of $\mathcal{G}$ and $\mathcal{H}$ are both bounded by $c|F|$ for some constant $c > 1$. □

The above theorem implies that the Exemplar Breakpoint Distance problem does not admit any approximation unless P=NP—if such a polynomial-time approximation existed then it would be able to decide whether $\mathcal{G}$ and $\mathcal{H}$ have zero breakpoint distance in polynomial time hence contradicting Theorem 3.5. If we parameterize the ZBD problem to kZBD, which is to decide if two k-repetitive sequences have zero break point distance, then the above theorem can be further strengthened as follows.

Theorem 3.6 Deciding if two 3-repetitive genomes have zero breakpoint distance is NP-complete.

Proof. Using the same reduction, a 3SAT sub-formula f_j with three literals appears in $\mathcal{G}$ ($\mathcal{H}$) exactly three times. Therefore, we can reduce 3SAT to 3ZBD in linear time. □

We now have the following corollary.

Corollary 3.7 Unless P=NP, the Exemplar Reversal Distance Problem cannot be approximated even if both $\mathcal{G}$ and $\mathcal{H}$ are 3-repetitive.

4 Weak Inapproximability Bounds

In this section, we try to generalize Theorem 3.5 to obtain some inapproximability bound under a weak approximation model. Let $opt(\mathcal{G}, \mathcal{H})$ be the optimal exemplar breakpoint distance between $\mathcal{G}$ and $\mathcal{H}$. (We also use $d(X, Y)$ to denote the minimum breakpoint distance between two genomes X and Y, where X and Y do not have to be exemplar.) We obtain the following inapproximability bounds under a much weaker model of approximation.

Theorem 4.1 Let $\epsilon > 0$ and $g(x) : N \to N$ be a function computable in polynomial time. If there is a polynomial time algorithm such that given $\mathcal{G}$ and $\mathcal{H}$ of length at most m it can return exemplar genomes G and H satisfying $d(G, H) \leq g(m) opt(\mathcal{G}, \mathcal{H}) + m^{1-\epsilon}$, then P=NP.

Proof. Let f be a SAT formula. Let $G(f), H(f)$ be the sequences as constructed in Theorem 3.5 such that f is satisfiable if and only if $d(G(f), H(f)) = 0$.

Let u be the length of f. Then $|G(f)| = |H(f)| \leq cu$ for some positive constant $c > 1$. Let x be a number such that $u^x > u^{(1+x)(1-\frac{\epsilon}{2})}$. Let $M = u^x$.

Let $\Sigma(S)$ be the alphabet of a sequence S. If Σ_i is a different set of letters with $|\Sigma_i| = |\Sigma(S)|$, we define $S(\Sigma_i)$ to be a new sequence obtained by replacing all letters in S, in one to one fashion, by those in Σ_i.

Let $\Sigma_1, \Sigma_2, \cdots, \Sigma_M$ be M disjoint sets of letters of size $|\Sigma(G(f))|$. Let $G_1 = G(f)(\Sigma_1), G_2 = G(f)(\Sigma_2), \cdots, G_M = G(f)(\Sigma_M)$ be the sequences derived from $G(f)$. Let $H_1 = H(f)(\Sigma_1), H_2 = G(f)(\Sigma_2), \cdots, H_M = G(f)(\Sigma_M)$ be the sequences derived from $H(f)$.

Define $\mathcal{G} = G_1 s_1 G_2 s_2 \cdots G_M s_M$ and $\mathcal{H} = H_1 s_1 H_2 s_2 \cdots H_M s_M$, where s_i is a peg gene appearing only once in $\mathcal{G}$ and $\mathcal{H}$. The total length of $\mathcal{G}, \mathcal{H}$ is bounded by $c(u+1)M \leq 2cu^{x+1}$. Let m be the maximum length of $\mathcal{G}$ and $\mathcal{H}$, then $m \leq c'u^{x+1}$ for some $c' > 2$.

Assume that some polynomial time algorithm $\mathcal{A}$ outputs G, H such that G is an exemplar genome of $\mathcal{G}$ and H is an exemplar genome of $\mathcal{H}$, and $d(G,H) \leq g(m)d(\mathcal{G},\mathcal{H}) + m^{1-\epsilon}$, we can then decide if f is satisfiable by checking whether $d(G,H) \leq m^{1-\epsilon}$. If f is satisfiable, it is easy to see that $d(\mathcal{G},\mathcal{H}) = 0$ then $d(G,H) \leq m^{1-\epsilon}$. If f is not satisfiable, then from Theorem 3.5 $d(G_i, H_i) \geq 1$. As no letter is shared by G_i, G_j, we have $d(\mathcal{G},\mathcal{H}) \geq M = u^x > u^{(1+x)(1-\frac{\epsilon}{2})} \geq (\frac{m}{c'})^{1-\frac{\epsilon}{2}} > m^{1-\frac{3}{4}\epsilon}$ when m is sufficiently large. Since G, H are exemplar genomes of $\mathcal{G}, \mathcal{H}$, $d(G,H) > m^{1-\frac{3}{4}\epsilon}$. □

Corollary 4.2 Let $\epsilon > 0$. If there is a polynomial time algorithm such that given $\mathcal{G}$ and $\mathcal{H}$ of length at most m it can return exemplar genomes G and H satisfying $d(G,H) \leq m^{1-\epsilon}[opt(\mathcal{G},\mathcal{H}) + 1]$, then P=NP.

This negative result shows that even under a much weaker model, it is not possible to obtain a good approximation unless P=NP. In next section, we will present a factor-$2(1+\log n)$ approximation for the One-Sided Exemplar Reversal Distance Problem in which one of the two genomes is a k-span genome. It is not surprising that this problem is also known to be NP-complete, in fact, it is NP-complete even when $k = 1$ [2].

5 A $2(1+\log n)-$approximation for the One-Sided Case

Given a *k-span* genome $\mathcal{G}_k$ and a general genome $\mathcal{H}$, each is a sequence containing $O(m)$ signed or unsigned genes (drawn from the n gene families and genes from the same family in $\mathcal{G}_k$ are at most k positions away and there are possibly any kind of repetitions in $\mathcal{H}$), the problem is to compute the minimum exemplar breakpoint distance between two exemplar genomes G, H (obtained by deleting redundant genes in $\mathcal{G}_k$ and $\mathcal{H}$). Let $\mathcal{G}_k = a_1 a_2 \cdots a_{n_1}, \mathcal{H} = b_1 b_2 \cdots b_{m_1}$. Throughout this section we assume that $k = O(\log n)$.

Let $opt(\mathcal{G}_k, \mathcal{H})$ be the size of the optimal solution of the above One-sided Exemplar Breakpoint Distance Problem.

Let $A = [a_i, a_{i+s_{p-1}}] \in \mathcal{G}_k$ and $B = [b_j, b_{j+t_{p-1}}] \in \mathcal{H}$. If a gene family, which is a multi-set of genes in $\mathcal{G}_k(\mathcal{H})$, all appear in A (B) then it is called a multi-set of *whole-family* genes in A (B). Example: Let $\mathcal{G}_3 = ga - fgedbedc - e$ and $\mathcal{H} = acefgac - fbebdach - g$. Consider the interval $I_G = a - fgedbed$ in $\mathcal{G}_3$ and the interval $I_H = gac - fbebdc$ in $\mathcal{H}$. The multi-set of whole-family genes in I_H is $\{\{b,b\},\{d\}\}$.

Given $A = [a_i, a_{i+s_{p-1}}] \in \mathcal{G}_k$ and $B = [b_j, b_{j+t_{p-1}}] \in \mathcal{H}$, an interval $I = c_1 c_2 ... c_p$ or its signed reversal $-I$ is called a *Non-Breaking Interval* (NB-interval for short) if I contains no repetition of

any gene, for each multi-set of whole-family genes in A and B one of them must appear in I, and I appears in $\mathcal{G}_k$ with $c_1 = a_i, c_2 = a_{i+s_1}, \cdots, c_p = a_{i+s_{p-1}}$ (or $c_1 = -a_{i+s_{p-1}}, c_2 = -a_{i+s_{p-2}}, \cdots, c_p = -a_i$) and in $\mathcal{H}$ with $c_1 = b_j, c_2 = b_{j+t_1}, \cdots, c_p = b_{j+t_{p-1}}$ (or $c_1 = -b_{j+t_{p-1}}, c_2 = -b_{j+t_{p-2}}, \cdots, c_p = -b_j$) for some $s_{p-1} > s_{p-2} > \cdots > s_1 > 0$ and some $t_{p-1} > t_{p-2} > \cdots > t_1 > 0$. The length p is called the *size* of I. Given $A = [a_i, a_{i+s_{p-1}}] \in \mathcal{G}_k$ and $B = [b_j, b_{j+t_{p-1}}] \in \mathcal{H}$, we are interested in computing a NB-interval of maximum size (length). Notice that a maximum NB-interval is very much a *longest constrained common subsequence* of A and B, it is related to but different from the recently studied *constrained longest common subsequence* [5]. From now on, we will only talk about maximum NB-intervals, which we will simply use NB-intervals if the context is clear.

Now let $A = g_1 g_2 \cdots g_N, B = h_1 h_2 \cdots h_M$ be strings on z identical gene families, and $g_1 = h_1, g_M = h_N$. We assume that both A, B are long enough, say, at least of length $20k$ (otherwise we can simply use a brute-force method). Let $W(A[i,j]), W(B[s,t])$ be the whole-family gene sets in $A[i,j]$ and $B[s,t]$ respectively. We show below a polynomial time dynamic programming algorithm to compute the NB-interval between strings A, B. Let $A[i,j] = P_a H P_b$, where $|P_a| = |P_b| = k$. Since A is a k-span genome, P_a, P_b have no common genes when $|H| \geq k$. Let H_a, H_b be exemplar genomes selected from P_a, P_b respectively. In the dynamic programming table, $table(i, j, H_a, H_b, s, t)$ stores a longest constrained common subsequence $H_a V H_b$ of $A[i,j]$ and $B[s,t]$ such that $W(A[i,j]), W(B[s,t])$ all appear in $H_a V H_b$ and there is no repetition of any gene in $H_a V H_b$.

Let $A[i,j] = P_a H P_b$, with $H = H_1 P_c P_d H_2$ and $|P_c| = |P_d| = k$. Assume that $A[i, j_1] = P_a H_1 P_c$ and $A[j_1, j] = P_d H_2 P_b$, we can merge $table(i, j_1, U_a, U_b, s, t_1)$ and $table(j_1 + 1, j, T_a, T_b, t_1 + 1, t)$ into $table(i, j, H_a, H_b, s, t)$—if $U_b T_a$ is exemplar and selected from $P_c P_d$ then all whole family genes in $P_c P_d$ must be in $U_b T_a$ and no gene is repeated in $U_b T_a$; moreover, among all such candidates we select the longest one as $U_b T_a$. So when j_1, t_1 is fixed this merge takes $O(k^2 + n) = O(n)$ time. As we need to try different combinations j_1 and t_1, the final content in $table(i, j, H_a, H_b, s, t)$, which should be the longest, can be computed in $O(n^3)$ time, provided that $table(i, j_1, U_a, U_b, s, t_1)$ and $table(j_1 + 1, j, T_a, T_b, t_1 + 1, t)$ are already available.

There are at most 2^k ways to select H_a from P_a (H_b from P_b). Therefore, this dynamic programming algorithm uses $O(2^{2k} n^5)$ space (there are $O(2^{2k} n^4)$ cells in the table, each could store a sequence of length $O(n)$) and it takes $O(2^{2k} n^7)$ time to compute the (maximum) NB-interval between A and B, which is stored in $table(1, N, -, -, 1, M)$. Finally, notice that each signed/unsigned gene in $\mathcal{G}_k$ or $\mathcal{H}$ is a degenerate (maximum) NB-interval of length one.

This dynamic programming algorithm will be used as a subroutine in our final approximation for the One-sided Exemplar Breakpoint Distance Problem. Now consider the problem of covering all genes in $\mathcal{G}_k$ and $\mathcal{H}$ using the minimum number of (disjoint) NB-intervals. Let $C^*(\mathcal{G}_k, \mathcal{H})$ be the size of the optimal solution for this covering problem.

Lemma 5.1 $C^*(\mathcal{G}_k, \mathcal{H}) \leq opt(\mathcal{G}_k, \mathcal{H}) + 1$.

Proof. Trivial, as each breakpoint in the exemplar genomes G, H can only occur between two NB-intervals. □

We now show how to obtain a factor $2(1 + \log n)$ approximation for $C^*(\mathcal{G}_k, \mathcal{H})$ by converting it to a set-cover problem $(X, \mathcal{F})$. In this case, each (degenerate and non-degenerate) NB-interval

is a set $S \in \mathcal{F}$. X contains all of the n genes. The problem is to compute the minimum number of (disjoint) NB-intervals which cover all the genes. The algorithm follows the greedy method [7, 13, 14].

(1) Start with $\mathcal{G}_k, \mathcal{H}$. Enumerate all pairs of intervals $A = [a_i, a_{i+s}]$ and $B = [b_j, b_{j+t}]$ with $a_i = b_j, a_{i+s} = b_{j+t}$. For each such pair (A, B), use the above dynamic programming algorithm to compute a maximum length NB-interval.

(2) Among all the maximum NB-intervals computed at Step (1), select one with the maximum size, I, and put it in the approximation solution.

(3) Delete all the (signed/unsigned) genes in I to have two updated versions of $\mathcal{G}_k, \mathcal{H}$. Repeat Step (1)-(2) until all the genes are covered.

Let $App(\mathcal{G}_k, \mathcal{H})$ be the number of the NB-intervals obtained in the above approximation solution. Following [7, 13, 14], we have the following lemma.

Lemma 5.2 $App(\mathcal{G}_k, \mathcal{H}) \leq (1 + \log n) \cdot C^*(\mathcal{G}_k, \mathcal{H})$.

We have the following theorem.

Theorem 5.3 $App(\mathcal{G}_k, \mathcal{H}) \leq 2(1 + \log n) \cdot opt(\mathcal{G}_k, \mathcal{H})$.

Proof. By Lemmas 5.1 and 5.2, $App(\mathcal{G}_k, \mathcal{H}) \leq (1 + \log n) \cdot opt(\mathcal{G}_k, \mathcal{H}) + \log n + 1$. When $opt(\mathcal{G}_k, \mathcal{H}) > 0$, $App(\mathcal{G}_k, \mathcal{H}) \leq (1 + \log n) \cdot opt(\mathcal{G}_k, \mathcal{H}) + \log n + 1 \leq (1 + \log n) \cdot opt(\mathcal{G}_k, \mathcal{H}) + (1 + \log n) \cdot opt(\mathcal{G}_k, \mathcal{H}) = 2(1 + \log n) \cdot opt(\mathcal{G}_k, \mathcal{H})$. When $opt(\mathcal{G}_k, \mathcal{H}) = 0$, which can be identified by the above dynamic programming algorithm, we can ignore using this approximation algorithm. $\Box$.

The running time of the above approximation algorithm is as follows: There could be $O(n)$ rounds in the greedy selection process. At each round we could have enumerated $O(n^2)$ intervals and each call to the dynamic programming procedure takes $O(2^{2k}n^7)$ time. Therefore, the overall running time of the approximation algorithm is $O(2^{2k}n^{10})$. The approximation algorithm uses $O(2^{2k}n^5)$ space.

We comment that for this problem, when $k = 1$, the above factor-$2(1+\log n)$ approximation can be greatly simplified. The complex dynamic programming method can be replaced by a Longest Common Subsequence computation [6] and the algorithm runs in $O(n^5)$ time and $O(n^2)$ space, which is clearly much more efficient.

6 Concluding Remarks

We present the first set of inapproximability/approximation results for the Exemplar Breakpoint Distance Problem. Although it seems that the general problem does not admit any approximation, for a special one-sided case, decent approximation does exist. This also partially conforms with the real-life dataset that repetitions of genes are typically pegged and not very far away [17]. It would be interesting to study some meaningful special cases. For example, can be obtain a good approximation when $\mathcal{G}$ is 2-repetitive and $\mathcal{H}$ is a 3-span genome?

References

[1] V. Bafna and P. Pevzner, Sorting by reversals: Genome rearrangements in plant organelles and evolutionary history of X chromosome, *Mol. Bio. Evol.*, 12:239-246, 1995.

[2] D. Bryant. The complexity of calculating exemplar distances. *In D. Sankoff and J. Nadeau, editors, Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment, and the Evolution of Gene Families*, pp. 207-212. Kluwer Acad. Pub., 2000.

[3] G. Blin and R. Rizzi. Conserved interval distance computation between non-trivial genomes. *Proc. 11th Intl. Ann. Comput. and Combinatorics (COCOON'05),* LNCS 3595, pp. 22-31, 2005.

[4] A. Bergeron and J. Stoye. On the similarity of sets of permutations and its applications to genome comparison. *Proc. 9th Intl. Ann. Comput. and Combinatorics (COCOON'03),* LNCS 2697, pp. 68-79, 2003.

[5] S. Bereg and B. Zhu. RNA multiple structural alignment with longest common subsequences. *Proc. 11th Intl. Ann. Comput. and Combinatorics (COCOON'05),* LNCS 3595, pp. 32-41, 2005.

[6] T. Cormen, C. Leiserson, R. Rivest, C. Stein. *Introduction to Algorithms*, second edition, MIT Press, 2001.

[7] V. Chvátal, "A greedy heuristic for the set-covering problem," *Math. Oper. Res.*, vol. 4, 1979, pp. 233-235.

[8] T. Cormen, C. Leiserson and R. Rivest, *Introduction to Algorithms*, The MIT Press, 1990.

[9] I. Dur and S. Safra. The importance of being biased. In *Proc. 34th ACM Symp. on Theory Comput. (STOC'02)*, pages 33-42, 2002.

[10] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* Freeman, San Francisco, CA, 1979.

[11] S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *J. ACM,* **46**(1):1-27, 1999.

[12] O. Gascuel, editor. *Mathematics of Evolution and Phylogeny.*Oxford University Press, 2004.

[13] D. Johnson, "Approximation algorithms for combinatorial problems," *J. Comput. System Sci.*, vol. 9, 1974, pp. 256-278.

[14] L. Lovász, "On the ratio of optimal integral and fractional covers," *Discrete Math.*, vol. 13, 1975, pp. 383-390.

[15] M. Marron, K. Swenson and B. Moret. Genomic distances under deletions and insertions. *Theoretical Computer Science,* **325**(3):347-360, 2004.

[16] C. Makaroff and J. Palmer. Mitochondrial DNA rearrangements and transcriptional alternatives in the male sterile cytoplasm of Ogura radish. *Mol. Cell. Biol.*, **8**:1474-1480, 1988.

[17] C.T. Nguyen, Y.C. Tay and L. Zhang. Divide-and-conquer approach for the exemplar breakpoint distance. *Bioinformatics*, **21**(10):2171-2176, 2005.

[18] J. Palmer and L. Herbon. Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence. *J. Mol. Evolut.*, **27**:87-97, 1988.

[19] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP. In *Proc. 29th ACM Symp. on Theory Comput. (STOC'97)*, pages 475-484, 1997.

[20] D. Sankoff. Genome rearrangement with gene families. *Bioinformatics*, **16**(11):909-917, 1999.

[21] A. Sturtevant and T. Dobzhansky. Inversions in the third chromosome of wild races of *drosophila pseudoobscura*, and their use in the study of the history of the species. *Proc. Nat. Acad. Sci. USA,* 22:448-450, 1936.

[22] E. Tannier and M-F. Sagot. Sorting by reversals in subquadratic time. *Proc. 15th Symp. Combinatorial Pattern Matching (CPM'04), Istanbul, Turkey, Pages 1-13, July, 2004 (LNCS series, 3109).*

[23] G. Watterson, W. Ewens, T. Hall and A. Morgan. The chromosome inversion problem. *J. Theoretical Biology*, **99**:1-7, 1982.