9-18-2014

On the Approximability of the Exemplar Adjacency Number Problem for Genomes with Gene Repetitions

Zhinxiag Chen
The University of Texas Rio Grande Valley

Bin Fu
The University of Texas Rio Grande Valley

Randy Goebel
University of Alberta

Guohui Lin
University of Alberta

Weitian Tong
University of Alberta

See next page for additional authors

On the Approximability of the Exemplar Adjacency Number Problem for Genomes with Gene Repetitions

Zhixiang Chen[a], Bin Fu[a], Randy Goebel[b], Guohui Lin[b], Weitian Tong[b], Jinhui Xu[c], Boting Yang[d], Zhiyu Zhao[e], Binhai Zhu[f,*]

[a]*Department of Computer Science, University of Texas-American, Edinburg, TX 78739-2999, USA.*
[b]*Department of Computing Science, University of Alberta, Edmonton, Alberta T6G 2E8, Canada.*
[c]*Department of Computer Science, SUNY-Buffalo, Buffalo, NY 14260, USA.*
[d]*Department of Computer Science, University of Regina, Regina, Saskatchewan S4S 0A2, Canada.*
[e]*Department of Computer Science, University of New Orleans, New Orleans, LA 70148, USA.*
[f]*Department of Computer Science, Montana State University, Bozeman, MT 59717-3880, USA.*

Abstract

In this paper, we apply a measure, *exemplar adjacency number*, which complements and extends the well-studied breakpoint distance between two permutations, to measure the similarity between two genomes (or in general, between any two sequences drawn from the same alphabet). For two genomes $\mathcal{G}$ and $\mathcal{H}$ drawn from the same set of n gene families and containing gene repetitions, we consider the corresponding Exemplar Adjacency Number problem (EAN), in which we delete duplicated genes from $\mathcal{G}$ and $\mathcal{H}$ such that the resultant exemplar genomes (permutations) G and H have the maximum adjacency number. We obtain the following results. First, we prove that the one-sided 2-repetitive EAN problem, i.e., when one of $\mathcal{G}$ and $\mathcal{H}$ is given exemplar and each gene occurs in the other genome at most twice, can be

*Corresponding Author.

Email addresses: chen@cs.panam.edu (Zhixiang Chen), binfu@cs.panam.edu (Bin Fu), rgoebel@ualberta.ca (Randy Goebel), guohui@ualberta.ca (Guohui Lin), weitian@ualberta.ca (Weitian Tong), jinhui@cse.buffalo.edu (Jinhui Xu), boting@cs.uregina.ca (Boting Yang), sylvia@cs.uno.edu (Zhiyu Zhao), bhz@cs.montana.edu (Binhai Zhu)

linearly reduced from the Maximum Independent Set problem. This implies that EAN does not admit any $O(n^{0.5-\epsilon})$-approximation algorithm, for any $\epsilon > 0$, unless P = NP. This hardness result also implies that EAN, parameterized by the optimal solution value, is W[1]-hard. Secondly, we show that the two-sided 2-repetitive EAN problem has an $O(n^{0.5})$-approximation algorithm, which is tight up to a constant factor.

Keywords: Genome comparison, adjacency, breakpoint, NP-hard, approximation algorithm

---

1. Introduction

In genome comparison and rearrangement studies, the breakpoint distance is one of the most well-known distance measures [19]. The implicit idea of breakpoints was initiated as early as in 1936 by Sturtevant and Dobzhansky [18]. While gene duplication/loss, etc, is an inseparable part of evolution, due to the difficulty of handling duplicated genes, until only a few years ago, in computational genomics it had been largely assumed that every gene appears in a genome exactly once. Under this assumption, the genome rearrangement problem is essentially the problem of comparing and sorting unsigned (or signed) permutations [12, 10]. Computing the breakpoint distance between two permutations over the same alphabet can be done in linear time.

Genomes in the form of permutations are hard to obtain and so far, can only be obtained in several small virus genomes. In fact, these kinds of 'perfect' genomes do not occur on eukaryotic genomes where paralogous genes are common [16, 17]. In practice, it is important to compute genomic distances between genomes in the form of permutations, such as is done by using the Hannenhalli-Pevzner method [12]. However, more often than never, one might have to handle the gene duplication problem. (Interested readers are referred to a recent survey on this topic [20].) In 1999, Sankoff proposed a way to select, from the duplicated copies of a gene, the common ancestral gene such that the distance between the reduced genomes (called *exemplar genomes*) is minimized. For this case, Sankoff produced a branch-and-bound algorithm [17]. In a subsequent work, Nguyen, Tay and Zhang proposed a divide-and-conquer method to compute the exemplar breakpoint distance empirically [16].

From the algorithmic complexity research point of view, it has been shown

that computing the *exemplar signed reversal distance* and computing the *exemplar breakpoint distance* between two general genomes (i.e., with gene duplications) are both NP-hard [3]. A few years ago, Blin and Rizzi further proved that computing the *exemplar conserved interval distance* between two such genomes is NP-hard [2]; furthermore, it is NP-hard to compute the *minimum conserved interval matching*, that is, without deleting the duplicated copies of genes. On the approximability, for any exemplar genomic distance measure $d(\cdot, \cdot)$ satisfying coincidence axiom (i.e., $d(G, H) = 0$ if and only if $G = H$ or the reversal of H), it was shown that the problem does not admit any approximation algorithms, even when each gene appears at most three times in each input genome unless P = NP [8, 6]. A few years later, this bound was tightened, as deciding when $d(\mathcal{G}, \mathcal{H}) = 0$ is NP-complete even if each gene appears in the input genomes $\mathcal{G}$ and $\mathcal{H}$ at most twice [1, 13]. It follows that for the exemplar breakpoint distance and the exemplar conserved interval distance problems, there are no polynomial time approximation algorithms. Furthermore, even under a weaker definition of polynomial time approximation algorithms, the exemplar breakpoint distance problem is shown not to admit any weak $O(n^{1-\epsilon})$-approximation algorithm, for any $0 < \epsilon < 1$, where n is the maximum length of the two input genomes [8]. The exemplar conserved interval distance problem is also shown not to admit any weak $O(n^{1.5})$-approximation algorithm [6, 7].

Complementary to the genomic distances, computing certain genomic similarities between two genomes has also been studied in [4]. In general, genomic similarity measures do not satisfy coincidence axiom. Among others, Chauve *et al.* proved that computing the maximum *exemplar common interval similarity* between two general genomes is NP-hard, while leaving open the problem approximability [4].

Here we study the *exemplar adjacency number* between two (general) genomes, which complements the breakpoint distance measure. Formally, given an alphabet Σ of n genes and two genomes $\mathcal{G}$ and $\mathcal{H}$ drawn from Σ, the *Exemplar Adjacency Number* problem (*EAN* for short) is to delete duplicated genes from $\mathcal{G}$ and $\mathcal{H}$ such that the number of adjacencies between the two resultant exemplar genomes (i.e., permutations), G and H, is maximized. The EAN problem is NP-hard, and here we study the approximability. When one of the input genomes is already exemplar, the problem is called one-sided EAN; the general case is also called two-sided EAN. We first present a linear reduction from the *Maximum Independent Set* (MIS) problem to the one-sided 2-repetitive EAN problem. This reduction implies that the one-

sided EAN problem is W[1]-hard, and that it does not admit an $O(n^{0.5-\epsilon})$-approximation algorithm, for any $\epsilon > 0$, unless P = NP. The W[1]-hardness (see [9] for details) and the recent lower bound results [5] imply that, if k is the optimal solution value to the one-sided EAN problem, then barring an unlikely collapse in the parameterized complexity hierarchy, the problem is not solvable in time $f(k)n^{o(k)}$, for any function f. Our second positive result is an $O(n^{0.5})$-approximation for the two-sided 2-repetitive EAN problem. Ignoring constants, the negative hardness result and the positive algorithmic result match perfectly for this case.

The rest of the paper is organized as follows. In Section 2, we summarize some of the necessary background definitions. Section 3 presents the linear reduction from the MIS problem to the one-sided EAN problem, and we draw the conclusion on inapproximability. The positive algorithmic result is presented in Section 4, with both the design and the analysis of the $O(n^{0.5})$-approximation algorithm. Section 5 concludes the paper with some discussions.

2. Preliminaries

In the (pairwise) genome comparison and rearrangement problems, we are given two genomes, each of which is a sequence of signed (or unsigned) genes. Note that in general a genome can be a set of such sequences (i.e., with multiple chromosomes); yet in this paper we focus on such one-sequence genomes, often called *singletons*. The order of the genes in one genome corresponds to their physical positions on the genome, and the sign of a gene indicates which one of the two DNA strands the gene is located. In the literature, most of the research assumes that each gene occurs exactly once in a genome; such an assumption is problematic in reality for eukaryotic genomes and the like where duplications of genes exist [17]. For such a general genome, Sankoff proposed to select an *exemplar genome*, by deleting duplicated copies of each gene, in which every gene appears exactly once. The deletion is to minimize certain genomic distance between the resultant exemplar genomes [17].

The following definitions are very much the same as those in [3, 8]. In this paper, we consider only unsigned genomes, though our results can be applied to signed genomes. We assume a gene alphabet Σ that consists of n distinct genes. A genome $\mathcal{G}$ is a sequence of elements of Σ, under the constraint that each element occurs at least once in $\mathcal{G}$. We allow repetitions of every gene

in any genome. Specifically, if each gene occurs exactly once in a genome, then the genome is called *exemplar*; otherwise *non-exemplar*. A genome $\mathcal{G}$ is called *r-repetitive* if each gene occurs at most r times in $\mathcal{G}$. For example, if $\Sigma = \{a, b, c\}$, then genome $\mathcal{G} = abcbaa$ is 3-repetitive.

Given a non-exemplar genome $\mathcal{G}$, for each gene family one can delete all but one duplicated genes to obtain an exemplar sub-genome G. In G, each gene from Σ occurs exactly once. Hence G is a permutation of Σ. For example, if $\Sigma = \{a, b, c\}$ and genome $\mathcal{G} = abcbaa$, then there are four distinct exemplar genomes for $\mathcal{G}$ by deleting two copies of a and one copy of b: $G_1 = abc$, $G_2 = acb$, $G_3 = bca$, and $G_4 = cba$.

For two exemplar genomes G and H drawn from a common n-gene alphabet Σ, a *breakpoint* in G is a two-gene substring $g_i g_{i+1}$ such that $g_i g_{i+1}$, and its reversal $g_{i+1} g_i$, do not occur as a substring in H. The number of breakpoints in G (symmetrically the number of breakpoints in H) is called the *breakpoint distance* between G and H, and denoted as $\mathrm{bd}(G, H)$. For two non-exemplar genomes $\mathcal{G}$ and $\mathcal{H}$, their *exemplar breakpoint distance* $\mathrm{ebd}(\mathcal{G}, \mathcal{H})$ is the minimum $\mathrm{bd}(G, H)$, where G and H are exemplar genomes of $\mathcal{G}$ and $\mathcal{H}$, respectively.

For two exemplar genomes G and H drawn from a common n-gene alphabet Σ, an *adjacency* in G is a two-gene substring $g_i g_{i+1}$ such that $g_i g_{i+1}$, or its reversal $g_{i+1} g_i$, also occurs as a substring in H. Likewise, the number of adjacencies in G (symmetrically the number of adjacencies in H) is called the *adjacency number* between G and H, and denoted as $\mathrm{an}(G, H)$. Similarly, for two non-exemplar genomes $\mathcal{G}$ and $\mathcal{H}$, their *exemplar adjacency number* $\mathrm{ean}(\mathcal{G}, \mathcal{H})$ is the maximum $\mathrm{an}(G, H)$, where G and H are exemplar genomes of $\mathcal{G}$ and $\mathcal{H}$, respectively. Clearly, for permutations, (exemplar) breakpoint distance and (exemplar) adjacency number are complement to each other, and they sum to exactly $n - 1$.

We comment that the adjacency definition we used is really on permutations, while in [1] the adjacency definition is directly on strings without any element deletion. In the latter case, even if two strings P, Q have no breakpoint between them, it does not mean $P = Q$ or Q's reversal. The recent work on filling scaffolds (to construct a complete genome) also uses the latter (string) adjacencies [14, 15].

Formally, in the *Exemplar Adjacency Number* (EAN) problem, we are given two genomes $\mathcal{G}$ and $\mathcal{H}$ drawn from a common n-gene alphabet Σ, and the goal is to compute $\mathrm{ean}(\mathcal{G}, \mathcal{H})$ and the associated exemplar genomes G and H of $\mathcal{G}$ and $\mathcal{H}$ respectively.

The EAN problem is a maximization problem. For any instance I, we use $OPT(I)$ to denote the optimal solution value of I. For an approximation algorithm A designed for the EAN problem, we use $A(I)$ to denote the solution value returned by A on I. Algorithm A has a *performance guarantee* of α, if $A(I) \geq OPT(I)/\alpha$ for all I. In this case, A is also called an α-approximation algorithm. We note that an approximation algorithm, by default, runs in polynomial time.

3. Inapproximability Result

For (any instance of) the EAN problem, OPT denotes the optimal solution value. We first have the following lemma.

Lemma 1. $0 \leq OPT \leq n-1$, *where* n (≥ 4) *is the size of the gene alphabet.*

Proof. Let the n (≥ 4) distinct genes be denoted as $1, 2, 3, \ldots, n$. We only consider the exemplar genomes. The upper bound of OPT is achieved by setting $G = H$; the lower bound of OPT is achieved by setting $G = 123\ldots(n-1)n$ (the identity permutation) and H as follows:

$$H = \begin{cases} (n-1)(n-3)\ldots531n(n-2)\ldots642, & \text{if } n \text{ is even,} \\ (n-1)(n-3)\ldots642n135\ldots(n-4)(n-2), & \text{otherwise.} \end{cases}$$

It can be easily confirmed that between this pair of G and H there is no adjacency. □

It is interesting to note that, given $\mathcal{G}$ and $\mathcal{H}$, whether or not $OPT = 0$ can be easily decided in polynomial time. For instance, one can use a brute-force method on each pair of distinct genes to check whether it is possible to make them into an adjacency. Such an observation implies that there is a trivial $O(n)$-approximation algorithm for the EAN problem. Note that the complement Exemplar Breakpoint Distance problem is different, which does not admit any polynomial time approximation at all since deciding whether its optimal solution value is zero is NP-complete [8, 1, 13]. The next theorem shows that the one-sided EAN problem does not admit any $O(n^{0.5-\epsilon})$-approximation algorithm, for any $0 < \epsilon < 0.5$.

Theorem 1. *Even if one of* $\mathcal{G}$ *and* $\mathcal{H}$ *is exemplar and the other is* 2*-repetitive, the* EAN *problem does not admit any* $O(n^{0.5-\epsilon})$*-approximation algorithm, for any* $0 < \epsilon < 0.5$*, unless* P = NP*, where* n *is the size of the gene alphabet.*

Proof. It is easy to see that the decision version of the EAN problem is in NP. We next present a reduction from the Maximum Independent Set (MIS) problem to the EAN problem in which the optimal solution value is preserved. The MIS problem is a well known NP-hard problem that cannot be approximated within a factor of $|V|^{1-\epsilon}$, for any $0 < \epsilon < 1$, unless P = NP, where V is the vertex set of the input graph [21].

Let (V, E) be an instance of the MIS problem, where V is the vertex set and E is the edge set. Let $N = |V|$ and $M = |E|$, and the vertices of V are $v_1, v_2, v_3, \ldots, v_N$, the edges of E are $e_1, e_2, e_3, \ldots, e_M$. We construct a gene alphabet Σ and two genomes $\mathcal{G}$ and $\mathcal{H}$ as follows. For each vertex v_i, two distinct genes v_i and v_i' are created; for each edge e_j, three distinct genes e_j, x_j and x_j' are created. The alphabet Σ contains in total $2N + 3M$ distinct genes. Let A_i denote the sequence of all edges incident at vertex v_i, sorted by their indices. Let $Y_i = v_i A_i v_i'$, for $i = 1, 2, \ldots, N$, and $Y_{N+j} = x_j x_j'$, for $j = 1, 2, \ldots, M$.

Let

$$\mathcal{G} = v_1 v_1' v_2 v_2' \ldots v_N v_N' x_1 e_1 x_1' x_2 e_2 x_2' \ldots x_M e_M x_M'.$$

Clearly, $\mathcal{G}$ is exemplar. We distinguish two cases to construct $\mathcal{H}$ (as in the proof of Lemma 1):

$$\mathcal{H} = \begin{cases} Y_{N+M-1} Y_{N+M-3} \ldots Y_1 Y_{N+M} Y_{N+M-2} \ldots Y_2, & \text{if } N+M \text{ is even,} \\ Y_{N+M-1} Y_{N+M-3} \ldots Y_2 Y_{N+M} Y_1 Y_3 \ldots Y_{N+M-2}, & \text{otherwise.} \end{cases}$$

Clearly, in either case, $\mathcal{H}$ is 2-repetitive. The remaining argument is identical for both cases.

We claim that the graph (V, E) has a maximum independent set of size k iff $\text{ean}(\mathcal{G}, \mathcal{H}) = k$. First of all, since $\mathcal{G}$ is exemplar, $G = \mathcal{G}$. If the graph (V, E) has an independent set of size k, then the claim is trivial. To see this, we construct the exemplar genome H as follows. For all i, if v_i is in the independent set, then we delete A_i from $Y_i = v_i A_i v_i'$. Next, all other redundant edges can be arbitrarily deleted to form H. This way, $v_i v_i'$ is an adjacency between G and H, and thus $\text{ean}(\mathcal{G}, \mathcal{H}) = k$. On the other hand, if $\text{ean}(\mathcal{G}, \mathcal{H}) = k$, the first thing to notice is that $Y_j = x_j x_j'$ $(N+1 \leq j \leq N+M)$ cannot give us any adjacency; so the adjacency between G and H must all come from $Y_i = v_i A_i v_i'$ $(1 \leq i \leq N)$, with A_i being deleted to create an adjacency $v_i v_i'$. It follows that there are exactly k such A_i's being deleted. For any two such deleted A_i and A_j, there is no edge between v_i and v_j, for otherwise both copies of the edge would be deleted and consequently H

would not be exemplar. Therefore, these vertices form into an independent set in graph (V, E).

The above reduction takes polynomial time. Since $n = |\Sigma| = 2N + 3M \in O(N^2)$ and the MIS problem does not admit any $O(N^{1-\epsilon})$-approximation algorithm, for any $0 < \epsilon < 1$, unless P = NP, our EAN problem does not admit any $O(n^{0.5-\epsilon})$-approximation algorithm, with ϵ subsequently scaled to $0 < \epsilon < 0.5$. □

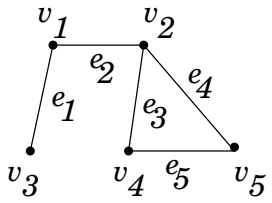

Figure 1. Illustration of a simple graph for the reduction.

In the example shown in Figure 1, we have
$\mathcal{G} : v_1 v_1' v_2 v_2' v_3 v_3' v_4 v_4' v_5 v_5' x_1 e_1 x_1' x_2 e_2 x_2' x_3 e_3 x_3' x_4 e_4 x_4' x_5 e_5 x_5'$ and
$\mathcal{H} : x_4 x_4' x_2 x_2' v_5 e_4 e_5 v_5' v_3 e_1 v_3' v_1 e_1 e_2 v_1' x_5 x_5' x_3 x_3' x_1 x_1' v_4 e_3 e_5 v_4' v_2 e_2 e_3 e_4 v_2'$.
Corresponding to the optimal independent set $\{v_3, v_4\}$, we have
$H : x_4 x_4' x_2 x_2' v_5 e_5 v_5' v_3 v_3' v_1 e_1 e_2 v_1' x_5 x_5' x_3 x_3' x_1 x_1' v_4 v_4' v_2 e_3 e_4 v_2'$. The two adjacencies are $[v_3 v_3']$ and $[v_4 v_4']$.

It is natural to consider using fixed-parameter tractable (FPT) algorithms for this kind of problems. However, regarding the FPT-tractability of the EAN problem, parameterized by the optimal solution value k, we note that the above reduction is an fpt-reduction. This is due to that, in the constructed EAN instance, the optimal solution value only depends on the size of the independent set (and not on the rest of the input at all). Hence, EAN is as hard as the parameterized version of Independent Set, which is W[1]-hard. (See [9] for the details regarding the hierarchies on the parameterized complexity, e.g., W[1].) This implies that, unless an unlikely collapse of the parameterized complexity hierarchies occurs, EAN cannot be solved in time $f(k)n^{o(k)}$, where f is any computable function [5]. We hence have the following corollary.

Corollary 1. *Even if one of $\mathcal{G}$ and $\mathcal{H}$ is exemplar and the other is 2-repetitive, the* EAN *problem parameterized by the optimal solution value k is W[1]-hard.*

4. An $O(n^{0.5})$-Approximation Algorithm

Here we consider the two-sided 2-repetitive EAN problem in this section. Let $\Sigma = \{1, 2, \ldots, n\}$ be the gene alphabet, and $\mathcal{G} = (g_1 g_2 \ldots g_p)$ and $\mathcal{H} = (h_1 h_2 \ldots h_q)$ be the two 2-repetitive genomes. For the ease of presentation, for each gene $i \in \Sigma$, an occurrence in $\mathcal{G}$ or its exemplar sub-genomes is denoted by i^+, while an occurrence in $\mathcal{H}$ or its exemplar sub-genomes is denoted by i^-. To implement our algorithm, we construct an interval element y_i^+ between g_i and g_{i+1} for $i = 1, ..., p-1$; likewise, we construct an interval element y_j^- between h_j and h_{j+1} for $j = 1, ..., q-1$. Moreover, for each gene i we construct a gene element x_i.

4.1. The Algorithm

Between any two exemplar genomes G and H derived from $\mathcal{G}$ and $\mathcal{H}$ respectively, an adjacency ij un-ambiguously points to two positions i_1 and j_1 in $\mathcal{G}$ and two positions i_2 and j_2 in $\mathcal{H}$ such that $\{g_{i_1}, g_{j_1}\} = \{i^+, j^+\}$ and $\{h_{i_2}, h_{j_2}\} = \{i^-, j^-\}$; furthermore, to obtain G from $\mathcal{G}$, the substring $\mathcal{G}[i_1 + 1..j_1 - 1]$ is deleted (similarly, to obtain H from $\mathcal{H}$, the substring $\mathcal{H}[i_2 + 1..j_2 - 1]$ is deleted). Motivated by this observation, we create a set $S(i_1, j_1; i_2, j_2)$ when $\{g_{i_1}, g_{j_1}\} = \{i^+, j^+\}$ and $\{h_{i_2}, h_{j_2}\} = \{i^-, j^-\}$ for some pair of distinct genes i and j $(i < j)$, for all possible quadruples $1 \leq i_1 < j_1 \leq p$ and $1 \leq i_2 < j_2 \leq q$. Set $S(i_1, j_1; i_2, j_2)$ contains those genes in $\mathcal{G}[i_1 + 1..j_1 - 1]$ and those genes in $\mathcal{H}[i_2 + 1..j_2 - 1]$, and additionally $j_1 - i_1$ interval elements $y_{i_1}^+, y_{i_1+1}^+, \ldots, y_{j_1-1}^+$, $j_2 - i_2$ interval elements $y_{i_2}^-, y_{i_2+1}^-, \ldots, y_{j_2-1}^-$, and two gene elements x_i and x_j. Clearly, the total number of such constructed sets is $O(n^2)$.

We next remove some of these constructed sets for further consideration, since they do not correspond to feasible adjacencies. There are two cases: In one case, $\mathcal{G}[i_1 + 1..j_1 - 1]$ contains a gene which occurs only once in $\mathcal{G}$; in the other case, $\mathcal{G}[i_1 + 1..j_1 - 1]$ contains both copies of a gene. Since deleting the whole substring $\mathcal{G}[i_1 + 1..j_1 - 1]$ of genes leads to no exemplar sub-genomes of $\mathcal{G}$, $g_{i_1} g_{j_1}$ is not a feasible adjacency. The same procedure applies to $\mathcal{H}$, that if $\mathcal{H}[i_2 + 1..j_2 - 1]$ contains a gene which occurs only once in $\mathcal{H}$ or contains both copies of a gene, then $h_{i_2} h_{j_2}$ is not a feasible adjacency either. Let $\mathcal{S}$ denote the collection of the constructed sets after the above removing procedure, where each set corresponds to a feasible adjacency.

Let $\Sigma^+ = \{1^+, 2^+, \ldots, n^+\}$, $\Sigma^- = \{1^-, 2^-, \ldots, n^-\}$, $X = \{x_1, x_2, \ldots, x_n\}$, $Y^+ = \{y_1^+, y_2^+, \ldots, y_{p-1}^+\}$, and $Y^- = \{y_1^-, y_2^-, \ldots, y_{q-1}^-\}$. We construct an

instance I of *Set Packing* using the ground set $U = \Sigma^+ \cup \Sigma^- \cup X \cup Y^+ \cup Y^-$ and the collection $\mathcal{S}$ of subsets of U. Then, the linear time (in $|U|$) approximation algorithm in [11] for the Set Packing problem can be applied on I to produce an approximate solution, which is a sub-collection $Approx(I)$ of $\mathcal{S}$ containing mutually disjoint sets. By the following Lemma 3, the set of adjacencies extracted from $Approx(I)$ can be extended into an exemplar genome G of $\mathcal{G}$ and an exemplar genome H of $\mathcal{H}$, such that $\text{an}(G, H) \geq |Approx(I)|$. We return the pair G and H as the final solution to the EAN problem. A high-level description of the approximation algorithm A is in Figure 1.

Input: $\Sigma = \{1, 2, \ldots, n\}$ and two 2-repetitive genomes $\mathcal{G}$ and $\mathcal{H}$
Output: Two exemplar genomes G and H of $\mathcal{G}$ and $\mathcal{H}$ respectively
1. Construct set $S(i_1, j_1; i_2, j_2)$ for all possible quadruples;
2. Remove infeasible sets and form set collection $\mathcal{S}$;
3. Construct an instance I of Set Packing:
ground set $U = \Sigma^+ \cup \Sigma^- \cup X \cup Y^+ \cup Y^-$ and collection $\mathcal{S}$;
4. Run the linear time Set Packing approximation algorithm on I:
obtain a solution $Approx(I)$;
5. Extend $Approx(I)$ into exemplar genomes G and H.

Figure 1: A high-level description of the approximation algorithm A.

4.2. Performance Analysis

We first have the following result for the Set Packing problem:

Lemma 2. [11] *The* Set Packing *problem admits an $O(|U| + |\mathcal{S}|)$-time $|U|^{0.5}$-approximation algorithm, where U is the ground set and $\mathcal{S}$ is the collection of subsets.*

Next, we exploit relationships between the feasible solutions of the EAN and the Set Packing problems.

Lemma 3. *If there is a set packing $\mathcal{S}' \subseteq \mathcal{S}$ of size k, then there is a pair of exemplar genomes G and H, derived from $\mathcal{G}$ and $\mathcal{H}$ respectively, such that* $\text{an}(G, H) \geq k$.

Proof. Let $S_1, S_2, \ldots, S_k$ be the sets in the set packing $\mathcal{S}'$. Note that these k sets are mutually disjoint, i.e., no two of them contain a common element from $U = \Sigma^+ \cup \Sigma^- \cup X \cup Y^+ \cup Y^-$.

From the construction process of the sets of $\mathcal{S}$, we know each S_i is associated with an interval of $\mathcal{G}$ and an interval of $\mathcal{H}$, and S_i contains all the associated interval elements. Two disjoint sets S_i and S_j are thus associated with two non-overlapping intervals of $\mathcal{G}$ (and of $\mathcal{H}$, respectively). Therefore, all the adjacencies corresponding to sets $S_1, S_2, \ldots, S_k$ can be formed by deleting all genes from the intervals associated with sets $S_1, S_2, \ldots, S_k$. Moreover, if a gene i occurs only once in $\mathcal{G}$ (in $\mathcal{H}$, respectively), then i^+ (i^-, respectively) does not belong to any of $S_1, S_2, \ldots, S_k$. Likewise, if a gene i occurs twice in $\mathcal{G}$ (in $\mathcal{H}$, respectively), then i^+ (i^-, respectively) belongs to at most one of $S_1, S_2, \ldots, S_k$. If it belongs to exactly one S_i, gene i will form an adjacency together with some other gene. Otherwise, there will still be two copies of the gene remaining in each of the two genomes after deleting the intervals associated with $S_1, S_2, \ldots, S_k$.

In the former case, element x_i is covered by exactly one of $S_1, S_2, \ldots, S_k$ and thus gene i is in a unique adjacency. In the latter case, we may keep an arbitrary copy of i^+ in $\mathcal{G}$ and an arbitrary copy of i^- in $\mathcal{H}$, while deleting the others if any. This way, we obtain an exemplar genome G from $\mathcal{G}$ and an exemplar genome H from $\mathcal{H}$, for which all the adjacencies corresponding to sets $S_1, S_2, \ldots, S_k$ are kept. That is, $\mathrm{an}(G, H) \geq k$. This proves the lemma. In addition, we see that such a pair of exemplar genomes can be obtained from $\mathcal{S}'$ in a linear scan through the genomes $\mathcal{G}$ and $\mathcal{H}$. □

Lemma 4. *If* $\mathrm{ean}(\mathcal{G}, \mathcal{H}) = k$*, then the optimal set packing has size at least* $\frac{k}{2}$.

Proof. Let G^* and H^* denote the exemplar genomes of $\mathcal{G}$ and $\mathcal{H}$ respectively such that $\mathrm{an}(G^*, H^*) = \mathrm{ean}(\mathcal{G}, \mathcal{H})$. Clearly, adjacencies between G^* and H^*, when regarded as edges connecting the two involved genes, form disjoint paths. For each such path containing ℓ adjacencies, a maximum of $\lceil \frac{\ell}{2} \rceil$ disjoint adjacencies can be obtained; here two adjacencies are disjoint if they do not share any common gene. It follows from the proof of Lemma 3 that the optimal set packing has size at least $\frac{k}{2}$. □

Theorem 2. *The two-sided* 2*-repetitive* EAN *problem admits an* $O(n^3)$*-time* $O(n^{0.5})$*-approximation algorithm, where* n *is size of the gene alphabet.*

Proof. Let the two 2-repetitive genomes be $\mathcal{G}$ and $\mathcal{H}$. Their lengths are thus bounded above by $2n$. For each position pair (i_1, j_1) in $\mathcal{G}$, we only need to look up at most 4 possibilities to construct sets, each of which contains $O(n)$

elements. Therefore, the instance I of Set Packing can be constructed in $O(n^3)$ time, with $|U| \leq 7n$ and $|\mathcal{S}| \in O(n^2)$. Running the approximation algorithm for Set Packing on I takes $O(n^2)$ time, with the returned solution $|Approx(I)| \leq n$. Finally, a pair of exemplar genomes G and H can be extended from $Approx(I)$ in $O(n)$ time. Therefore, the overall running time is $O(n^3)$.

From Lemmas 2–4,

$$\text{an}(G, H) \geq |Approx(I)| \geq \frac{\text{ean}(\mathcal{G}, \mathcal{H})}{2}/|U|^{0.5} = \frac{\text{ean}(\mathcal{G}, \mathcal{H})}{2\sqrt{7}n^{0.5}}.$$

Therefore, our approximation algorithm has a performance ratio in $O(n^{0.5})$. □

5. Concluding Remarks

In this paper we studied the exemplar adjacency number problem, which complements and extends the exemplar breakpoint distance problem, between two general genomes (with gene repetitions). (An earlier version of this paper appeared at CPM'07, where some different terminologies were used.) We proved that the EAN problem cannot be approximated within $O(n^{0.5-\epsilon})$ for $0 < \epsilon < 0.5$, even in the one-sided 2-repetitive case, where n is the size of the gene alphabet. On the positive side, we presented a cubic time $O(n^{0.5})$-approximation algorithm for the two-sided 2-repetitive EAN problem. Therefore, within the context of 2-repetitiveness, our negative inapproximability and positive algorithmic results merge perfectly. We believe that our techniques could extend the approximation algorithm for the r-repetitive case, for any fixed r; but we are not sure whether the general EAN problem admits an $O(n^{0.5})$-approximation algorithm, even in the one-sided case.

On the other hand, the approximability for the (complement) one-sided Exemplar Breakpoint Distance problem, even when each gene appears in the non-exemplar genome at most twice, is still open. The only known negative result is APX-hardness [1], and the only positive result is the trivial $O(n)$-factor approximation.

Finally, while the EAN problem we discussed in this paper is hard in terms of designing efficient FPT algorithms, using FPT algorithms to handle problems in computational genomics, in many situations, is feasible. We refer

interested readers to the recent survey for the current status of this research direction [20].

Acknowledgments

BF was supported by NSF under projects CAREER-0845376 and 1137764. The research of RG, GL and WT was supported in part by NSERC, Alberta Innovates Technology Futures, and a PhD Early Achievement Award to WT. BY was supported by NSERC. BZ was supported by NSF of China.

References

[1] S. Angibaud, G. Fertin, I. Rusu, A. Thevenin, and S. Vialette. On the approximability of comparing genomes with duplicates. *Journal of Graph Algorithms and Applications*, 13:19–53, 2009.

[2] G. Blin and R. Rizzi. Conserved interval distance computation between non-trivial genomes. In *Proceedings of the 11th International Annual Computing and Combinatorics Conference (COCOON'05)*, LNCS 3595, pages 22–31, 2005.

[3] D. Bryant. The complexity of calculating exemplar distances. In D. Sankoff and J. Nadeau, eds., *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment, and the Evolution of Gene Families*, pages 207–212. Kluwer Academic Publisher, 2000.

[4] G. Blin, C. Chauve, G. Fertin, R. Rizzi, and S. Vialette. Comparing genomes with duplications: a computational complexity point of view. *IEEE/ACM Trans. Comput. Biology and Bioinformatics*, 4:523–534, 2007.

[5] J. Chen, X. Huang, I. Kanj, and G. Xia. Linear FPT reductions and computational lower bounds. In *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC'04)*, pages 212–221, 2004.

[6] Z. Chen, B. Fu, R. Fowler, and B. Zhu. Lower bounds on the approximation of the exemplar conserved interval distance problem of genomes. In *Proceedings of the 12th International Annual Computing and Combinatorics Conference (COCOON'06)*, LNCS 4112, pages 245–254, 2006.

[7] Z. Chen, B. Fu, R. Fowler, and B. Zhu. On the inapproximability of the exemplar conserved interval distance problem of genomes. *Journal of Combinatorial Optimization*, 15:201–221, 2008.

[8] Z. Chen, B. Fu, and B. Zhu. The approximability of the exemplar breakpoint distance problem. In *Proceedings of the 2nd International Conference on Algorithmic Aspects in Information and Management (AAIM'06)*, LNCS 4041, pages 291–302, 2006. (Correction: LNCS 7285, pp. 368, 2012).

[9] R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

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

[11] M. M. Halldórsson, J. Kratochvil, and J. Telle. Independent sets with domination constraints. *Discrete Applied Mathematics*, 99:39–54, 2000.

[12] S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM*, 46:1–27, 1999.

[13] M. Jiang. The zero exemplar distance problem. *Journal of Computational Biology*, 18:1077–1086, 2011.

[14] H. Jiang, C. Zheng, D. Sankoff and B. Zhu. Scaffold filling under the breakpoint and related distances. *IEEE/ACM Trans. Comput. Biology and Bioinformatics*, 9(4):1220-1229, 2012.

[15] N. Liu, H. Jiang, D. Zhu and B. Zhu. An improved approximation algorithm for scaffold filling to maximize the common adjacencies. *IEEE/ACM Trans. Comput. Biology and Bioinformatics*, 10(4):905-913, 2013.

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

[17] D. Sankoff. Genome rearrangement with gene families. *Bioinformatics*, 16:909–917, 1999.

[18] 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. *Proceedings of the National Academy of Sciences of USA*, 22:448–450, 1936.

[19] G. Watterson, W. Ewens, T. Hall, and A. Morgan. The chromosome inversion problem. *Journal of Theoretical Biology*, 99:1–7, 1982.

[20] B. Zhu. A retrospective on genomic preprocessing for comparative genomics. In Chauve et al., eds., *Models and Algorithms for Genome Evolution*, pages 183-206. Springer, 2013.

[21] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3:103-128, 2007.