EPiC Series in Computing

Volume 83, 2022, Pages 31–42

Proceedings of 14th International Conference
on Bioinformatics and Computational Biology

Signed Rearrangement Distances Considering Repeated Genes and Intergenic Regions

Gabriel Siqueira[1], Alexsandro Oliveira Alexandrino[1], and Zanoni Dias[1]

Institute of Computing, University of Campinas (Unicamp), Campinas, Brazil
{gabriel.siqueira,alexsandro,zanoni}@ic.unicamp.br

Abstract

Genome Rearrangement distance problems are used to infer the evolutionary distance between genomes. These problems look at the number of mutations called rearrangement events necessary to transform one genome into another. Two commonly studied rearrangements are the reversal, which inverts a sequence of genes, and the transposition, which exchanges two consecutive sequences of genes. Seminal works on that topic looked only at the sequence of genes and assumed that no gene has more than one copy. More realistic models have been assuming multiple copies of a gene or have been taking the number of nucleotides between intergenic regions into account. This work combines these two generalizations defining the Signed Intergenic Reversal Distance (SIRD) and the Signed Intergenic Reversal and Transposition Distance (SIRTD) problems. Using a relationship with a problem called Signed Minimum Common Intergenic String Partition, we show $\Theta(k)$-approximation algorithms for the SIRD and the SIRTD problems, where k is the maximum number of copies of a gene in the genomes. Our experimental tests on simulated genomes show that the algorithms tend to find low distances despite the high theorical approximation factor.

1 Introduction

Estimation of the evolutionary distance between genomes is a fundamental task in the field of comparative genomics. Many computational problems help in the estimation of that distance. One important group of such problems is the so-called rearrangement distance problems. A genome rearrangement event is a mutation that affects a large portion of the genome. Given a set of possible events, a rearrangement distance problem aims at finding the minimum number of events necessary to transform one genome into another.

The conservative rearrangement events are a particular class of mutations that do not affect the quantity of genetic material, but instead affect the order in which genes appear. Two important events with that characteristic are the reversal, which inverts a sequence of genes in the genome, and the transposition, which exchanges two consecutive sequences of genes.

When solving rearrangement problems, the genomes may be modeled as strings, with each character corresponding to a gene. In some problems, we take the orientation of the genes into account and a sign ($+$ or $-$) is associated with each character to represent such orientation, in that case we have *signed strings*.

The Signed Reversal Distance problem is the rearrangement distance problem that considers the reversal event and model genomes as signed strings. Such problem has an exact linear time algorithm if no gene has more than one copy [2], but it is in the NP-hard class if more than one copy of the same gene exists [13]. The best known approximation factors for the general case are $16k$ [9] (where k is the maximum number of copies of a character in the strings) and $O(\log n \log^* n)$ [7] (where n is the size of the strings). Those approximation algorithms rely on the relationship between the Reversal Distance in Signed Strings problem and the Signed Minimum Common Partition [6] problem.

If the transposition event is considered alongside the reversal event, we have the Signed Reversal and Transposition Distance problem, which is in the NP-hard class even if no gene has more than one copy [10]. For such restrict case, there is an approximation algorithm of 2 [16]. For the general case, variations of the algorithms for reversal distance ensure approximations of factors $24k$ and $O(\log n \log^* n)$.

These works use only gene order to represent a genome; however, there are more elements that could be considered. For instance, consecutive genes in a genome are separated by nucleotides, which are called intergenic regions. Recent studies argue that including intergenic region information may improve genome comparison [3, 4]. That motivates the study of rearrangement distances considering the distribution of sizes in intergenic regions. These studies considered only the size of intergenic regions because rearrangement events do not break genes, while they break intergenic regions and, therefore, there is no correspondence between intergenic regions content for distinct genomes.

The Signed Intergenic Reversal Distance (SIRD) problem is the rearrangement distance problem that considers the reversal event and models genomes as signed strings and a list of intergenic region sizes. To the best of our knowledge, that problem was only studied when no gene has more than one copy; in that case, the problem is in the NP-hard class, and there is an approximation algorithm of factor 2 [11]. When combining the reversal and the transposition events we have the Signed Intergenic Reversal and Transposition (SIRTD) problem, which was also only studied when no gene has more than one copy; in that case, the problem is in the NP-hard class, and there is an approximation algorithm of factor 3 [12].

This work focuses on the SIRD and SIRTD problems when more than one copy of each gene is allowed. Section 2 formalizes the distance problems and defines other concepts, including a related problem of string partition. Next, Section 3 uses a relationship between distance and partition problems to produce an approximation algorithm for the SIRD and the SIRTD problems. Finally, Section 4 shows experimental results with the proposed algorithms, and Section 5 concludes the paper.

2 Definitions

In the following definitions we use ordered sequences of elements (*lists*). The number of elements in a list X is denoted by $|X|$, and an element at the i-th position of a list X is denoted by X_i. The list $Y = rev(X)$ is equal to the list X in the reverse order (i.e., $|X| = |Y|$ and $Y_i = X_{|X|-i+1}, \forall 1 \leq i \leq |X|$). A list of characters is called a *string*. A string may have a positive or negative sign associated with each element, and in that case we call it a *signed string*; otherwise it is an *unsigned string*. Given a signed string S, we denote by $Y = srev(S)$ the string $rev(S)$ with all its signs swapped (i.e., $|S| = |Y|$ and $Y_i = -S_{|S|-i+1}, \forall 1 \leq i \leq |S|$).

Given a string S, the set Σ_S of distinct elements of S, disregarding signs, is the alphabet of S and each element of Σ_S is called a *label*. Note that characters $+\alpha$ and $-\alpha$ both have label α. The *occurrence* of a label α in a string S is the number of characters of S with label α, and is

denoted by $occ(\alpha, S)$. The *maximum occurrence* of a label in S is $occ(S) = \max_{\alpha \in \Sigma_S}(occ(\alpha, S))$. A character whose label has occurrence one is called a *singleton*, and a character whose label has occurrence at least two is called a *replica*. Two strings S and P are *balanced* if $\Sigma_S = \Sigma_P$ and $occ(\alpha, S) = occ(\alpha, P), \forall \alpha \in \Sigma_S$. In other words, balanced strings are formed by the same characters in possibly different orders and orientations.

Example 1. Three string S, P, and Q with some information describing them. Strings S and P are balanced, while S and Q are not. String S has 2 singletons (A and D) and 2 replicas (B and C).

$$S = [+A\ -B\ +C\ +C\ -D\ -B\ -C],\ S_1 = +A,\ S_7 = -C$$
$$P = [+C\ +C\ +C\ -A\ -B\ -D\ -B],\ P_1 = +C,\ P_7 = -B$$
$$Q = [+A\ +A\ +B\ -D\ -B\ -C\ -C],\ Q_1 = +A,\ Q_7 = -C$$
$$occ(A, S) = occ(A, P) = 1,\ occ(A, Q) = 2$$
$$\Sigma_S = \Sigma_P = \Sigma_Q = \{A, B, C, D\},\ |S| = |P| = |Q| = 7$$

We encode a genome $\mathcal{G} = (S, \breve{S})$ with a sequence of n genes represented by a string S and a sequence of $n-1$ intergenic regions represented by a list $\breve{S}$. The number of genes $n = |S|$ is the genome size. When we take gene orientation into account, the string S is signed (the sign of each character corresponds to the orientation of the gene), and the genome is called a *signed genome*; otherwise, the string S is unsigned, and the genome is called an *unsigned genome*.

Two genomes $\mathcal{G} = (S, \breve{S})$ and $\mathcal{H} = (P, \breve{P})$ are called *co-tailed* if they have the same initial and final gene (i.e., $S_1 = P_1$ and $S_n = P_n$). When modeling a real genome, the initial and final gene are artificially inserted to ensure that any pair of genomes are co-tailed.

The reverse of a genome $\mathcal{G} = (S, \breve{S})$ is the genome $rev(\mathcal{G}) = (srev(S), rev(\breve{S}))$, if $\mathcal{G}$ is signed, or $rev(\mathcal{G}) = (rev(S), rev(\breve{S}))$, if $\mathcal{G}$ is unsigned. We say that two genomes $\mathcal{G}$ and $\mathcal{H}$ are *congruent* ($\mathcal{G} \cong \mathcal{H}$) if $\mathcal{G} = \mathcal{H}$ or $\mathcal{G} = rev(\mathcal{H})$.

Two genomes $\mathcal{G} = (S, \breve{S})$ and $\mathcal{H} = (P, \breve{P})$ of size n are *balanced* if the strings S and P are balanced and the sum of the integers correspondent to intergenic regions are the same (i.e., $\sum_{i=1}^{n} \breve{S}_i = \sum_{i=1}^{n} \breve{P}_i$). Figure 1 shows an example of two balanced co-tailed signed genomes and the reverse of one of them.

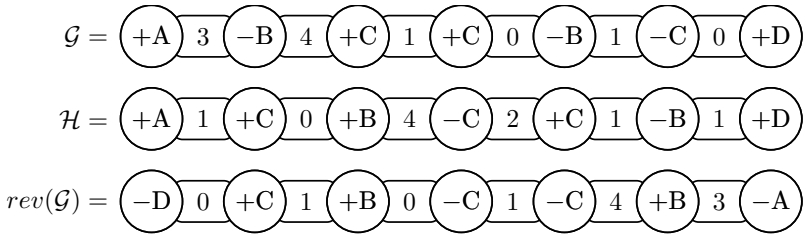

Figure 1: Two balanced co-tailed signed genomes $\mathcal{G}$ and $\mathcal{H}$, and the reverse $rev(\mathcal{G})$ of $\mathcal{G}$. We have $\mathcal{G} = (S, \breve{S})$, such that $S = [+A\ -B\ +C\ +C\ -B\ -C\ +D]$ and $\breve{S} = [3\ 4\ 1\ 0\ 1\ 0]$.

Given two balanced genomes $\mathcal{G} = (S, \breve{S})$ and $\mathcal{H} = (P, \breve{P})$, an *orthologous assignment* ξ between them is a mapping between genes, i.e., for each gene S_i of S there is a correspondent gene $\xi(S_i)$ in P. Each singleton from S is associated with the singleton of same label from P. Each replica from S must be associated with a replica of same label from P. Note that there are multiple ways to perform the association for a replica.

Consider a signed genome $\mathcal{G} = (S, \breve{S})$ of size n and the integers i, j, x, y, with $2 \leq i \leq j \leq n-1$, $0 \leq x \leq \breve{S}_{i-1}$, and $0 \leq y \leq \breve{S}_j$. The *intergenic reversal* $\rho^{(i,j)}_{(x,y)}$ is an operation that transforms $\mathcal{G}$ into a genome $\mathcal{G}.\rho^{(i,j)}_{(x,y)} = (S', \breve{S}')$, where:

$$S' = [S_1 \ldots S_{i-1} \underline{-S_j \ldots -S_i} S_{j+1} \ldots S_n]$$
$$\breve{S}' = [\breve{S}_1 \ldots \breve{S}_{i-2} \underline{x + y} \underline{\breve{S}_{j-1} \ldots \breve{S}_i} \underline{x' + y'} \breve{S}_{j+1} \ldots \breve{S}_{n-1}],$$

with $x' = \breve{S}_{i-1} - x$ and $y' = \breve{S}_j - y$.

Consider a genome $\mathcal{G} = (S, \breve{S})$ of size n and the integers i, j, k, x, y, z, with $2 \leq i < j < k \leq n$, $0 \leq x \leq \breve{S}_{i-1}$, $0 \leq y \leq \breve{S}_{j-1}$, and $0 \leq z \leq \breve{S}_{k-1}$. The *intergenic transposition* $\tau^{(i,j,k)}_{(x,y,z)}$ is an operation that transforms $\mathcal{G}$ into a genome $\mathcal{G}.\tau^{(i,j,k)}_{(x,y,z)} = (S', \breve{S}')$, where:

$$S' = [S_1 \ldots S_{i-1} \underline{S_j \ldots S_{k-1}} \underline{S_i \ldots S_{j-1}} S_k \ldots S_n]$$
$$\breve{S}' = [\breve{S}_1 \ldots \breve{S}_{i-2} \underline{x + y'} \underline{\breve{S}_j \ldots \breve{S}_{k-2}} \underline{z + x'} \underline{\breve{S}_i \ldots S_{j-2}} \underline{y + z'} S_k \ldots S_{n-1}],$$

with $x' = \breve{S}_{i-1} - x$, $y' = \breve{S}_{j-1} - y$, and $z' = \breve{S}_{k-1} - z$.

As shown in the following problem statements, we are interested in finding the minimum number of intergenic operations necessary to transform one signed genome into another. We assume that the genomes are co-tailed.

Signed Intergenic Reversal Distance (SIRD)
Input: Two balanced co-tailed signed genomes $\mathcal{G}$ and $\mathcal{H}$.
Goal: Find a minimum size sequence of intergenic reversals that transforms $\mathcal{G}$ into $\mathcal{H}$.

Signed Intergenic Reversal and Transposition Distance (SIRTD)
Input: Two balanced co-tailed signed genomes $\mathcal{G}$ and $\mathcal{H}$.
Goal: Find a minimum size sequence of intergenic reversals or intergenic transpositions that transforms $\mathcal{G}$ into $\mathcal{H}$.

The minimum number of intergenic reversals necessary to transform one signed genome $\mathcal{G}$ into another signed genome $\mathcal{H}$ is called the *signed intergenic reversals distance* (denoted by $d_{\mathcal{SIR}}(\mathcal{G}, \mathcal{H})$). Similarly, the minimum number of operations among intergenic reversals and intergenic transpositions necessary to transform one signed genome $\mathcal{G}$ into another signed genome $\mathcal{H}$ is called the *signed intergenic reversals and transposition distance* (denoted by $d_{\mathcal{SIRT}}(\mathcal{G}, \mathcal{H})$).

2.1 Intergenic Partitions

In this section, we describe two partition problems that are related to rearrangement distance problems. For that, we need some additional definitions.

A *break* of a genome $\mathcal{G} = (S, \breve{S})$ is an operation that separates $\mathcal{G}$ into two genomes $\mathcal{H} = (P, \breve{P})$ and $\mathcal{K} = (Q, \breve{Q})$ with an intergenic region $\breve{S}_i$ such that: $|P| = i$; $P_j = S_j, \forall 1 \leq j \leq i$; $|Q| = |S| - i$; $Q_{j-i} = S_j, \forall i < j \leq |S|$; $|\breve{P}| = i - 1$; $\breve{P}_j = \breve{S}_j, \forall 1 \leq j < i$; $|\breve{Q}| = |\breve{S}| - i$; and $\breve{Q}_{j-i} = \breve{S}_j, \forall i < j \leq |\breve{S}|$ (i.e., $\mathcal{H}$ and $\mathcal{K}$ consist of all genes and intergenic regions before $\breve{S}_i$ and after $\breve{S}_i$, respectively). We call the intergenic region used in the break operation a *breakpoint*. Note that a genome can be broken into multiple smaller genomes with multiple breakpoints. Figure 2 shows an example of a signed genome $\mathcal{G}$ broken into two signed genomes.

A *signed intergenic partition* between two balanced signed genomes $\mathcal{G} = (S, \breve{S})$ and $\mathcal{H} = (P, \breve{P})$ is a pair of signed genome sequences $(\mathbb{S}, \mathbb{P})$ such that:

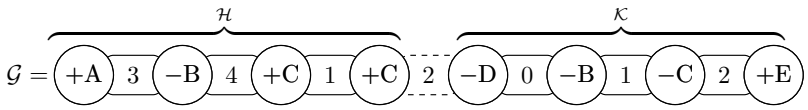

Figure 2: A signed genome $\mathcal{G} = (S, \breve{S})$ broken into two signed genomes $\mathcal{H}$ and $\mathcal{K}$ by the breakpoint $\breve{S}_4 = 2$.

1. The genome $\mathcal{G}$ can be broken into the genomes of $\mathbb{S}$.
2. The genome $\mathcal{H}$ can be broken into the genomes of $\mathbb{P}$.
3. It is possible to change the order and orientation of the genomes of $\mathbb{S}$ to obtain the genomes of $\mathbb{P}$ (i.e., there is at least one permutation ϕ, from the numbers 1 to $|\mathbb{S}|$, such that $\mathbb{P}_i \cong \mathbb{S}_{\phi_i}$, $\forall\ 1 \leq i \leq |\mathbb{S}|$).

Figure 3 shows an example of a signed intergenic partition. A *reverse intergenic partition* between two balanced unsigned genomes $\mathcal{G} = (S, \breve{S})$ and $\mathcal{H} = (P, \breve{P})$ is defined in the same way, but the genomes of $\mathbb{S}$ and $\mathbb{P}$ are also unsigned.

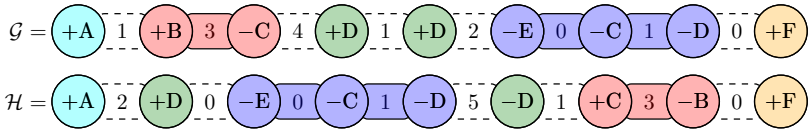

Figure 3: A signed intergenic partition between two genomes $\mathcal{G}$ and $\mathcal{H}$.

In both intergenic partitions, the genomes corresponding to elements of $\mathbb{S}$ and $\mathbb{P}$ are called *blocks*. As the blocks of $\mathbb{S}$ must be combined to form $\mathcal{G}$, the blocks must follow the order in which they appear in $\mathcal{G}$. Additionally, every gene must appear in some block. The same is valid for $\mathbb{P}$ and $\mathcal{H}$. We recall that the breakpoints of $\mathbb{S}$ and $\mathbb{P}$ are the intergenic regions between blocks in $\mathcal{G}$ and $\mathcal{H}$, respectively.

The $cost(\mathbb{S}, \mathbb{P})$ of an intergenic partition $(\mathbb{S}, \mathbb{P})$ is the number of breakpoints of $\mathbb{S}$. The cost can also be calculated by the number of blocks in $\mathbb{S}$ minus one. Note that, as a consequence of the third condition, both sequences $\mathbb{S}$ and $\mathbb{P}$ must have the same number of blocks and, consequently, the cost would be the same if we consider $\mathbb{P}$ instead of $\mathbb{S}$.

An intergenic partition is minimal if no two consecutive blocks can be combined to form an intergenic partition with smaller cost. An orthologous assignment between two genomes $\mathcal{G}$ and $\mathcal{H}$ associates genes of $\mathcal{G}$ with genes of $\mathcal{H}$ and, consequently, induces a unique minimal intergenic partition between $\mathcal{G}$ and $\mathcal{H}$. Note that a partition may be induced by multiple assignments. An assignment ξ is *compatible* with a minimal partition $(\mathbb{S}, \mathbb{P})$ if that partition is induced by ξ.

Let us now show a way to relate reverse intergenic partitions with signed intergenic partitions. For that, we define the unsigned extension of a signed genome $\mathcal{G} = (S, \breve{S})$ as a genome $\mathcal{G}' = (S', \breve{S}')$ created by the following procedure:

1. To create S', replace each gene S_i with two genes $S'_{2i-1} = |S_i|^h$ and $S'_{2i} = |S_i|^t$, if S_i is positive, or with two genes $S'_{2i-1} = |S_i|^t$ and $S'_{2i} = |S_i|^h$, if S_i is negative ($|S_i|$ denotes the character S_i without its sign).
2. To create $\breve{S}'$, we have $\breve{S}'_{2i} = \breve{S}_i$ and $\breve{S}'_{2i-1} = 0$, $\forall 1 \leq i \leq |\breve{S}|$.

Lemma 1. *Let $\mathcal{G}'$ and $\mathcal{H}'$ be the unsigned extension of two signed genomes $\mathcal{G}$ and $\mathcal{H}$, respectively. For every reverse intergenic partition $(\mathbb{S}', \mathbb{P}')$ between $\mathcal{G}'$ and $\mathcal{H}'$ there is a signed intergenic partition $(\mathbb{S}, \mathbb{P})$ between $\mathcal{G}$ and $\mathcal{H}$, such that $cost(\mathbb{S}, \mathbb{P}) \leq cost(\mathbb{S}', \mathbb{P}')$.*

Proof. To transform the reverse intergenic partition $(\mathbb{S}', \mathbb{P}')$ between $\mathcal{G}' = (S', \breve{S}')$ and $\mathcal{H}' = (P', \breve{P}')$ into a signed intergenic partition $(\mathbb{S}, \mathbb{P})$ between $\mathcal{G} = (S, \breve{S})$ and $\mathcal{H} = (P, \breve{P})$, we first ensure that every pair (S'_{2i-1}, S'_{2i}), with $i \in [1, |S|]$, belongs to the same block by constructing a new reverse intergenic partition.

Let (S'_{2i-1}, S'_{2i}) be a pair of genes that are not in the same block in $\mathbb{S}'$. By construction of the unsigned extension, we know that if S'_{2i-1} is in a block $B' = \mathbb{S}'_j$ then it is the last gene of that block and S'_{2i} is the first gene of the block $C' = \mathbb{S}'_{j+1}$. Construct a new sequence $\mathbb{S}''$, where $B'' = \mathbb{S}''_j$ is the block B' with the inclusion of S'_{2i} as the last gene (i.e., $|B''| = |B'| + 1$, $B''_{|B''|} = S'_{2i}$, and $B''_k = B'_k, \forall 1 \leq k \leq |B''| - 1$) and $C'' = \mathbb{S}''_{j+1}$ is the block C' with the exclusion of S'_{2i} (i.e., $|C''| = |C'| - 1$, $C''_k = C'_{k+1}, \forall 1 \leq k \leq |C''|$). Note that if C' has only the element S'_{2i}, then the block B'' is the concatenation of B' and C' into a single block. Apply the same transformation for every pair (S'_{2i-1}, S'_{2i}), with $i \in [1, |S|]$, that are split into different blocks of $\mathbb{S}'$. Let $\mathbb{S}^*$ be the resulting sequence and $\mathbb{P}^*$ a sequence resulting from the same transformations applied to $\mathbb{P}'$. Note that for every block changed in $\mathbb{S}^*$ there is a corresponding block changed in $\mathbb{P}^*$, consequently $(\mathbb{S}^*, \mathbb{P}^*)$ is also a reverse intergenic partition between $\mathcal{G}'$ and $\mathcal{H}'$. Also, we have that $cost(\mathbb{S}^*, \mathbb{P}^*) \leq cost(\mathbb{S}', \mathbb{P}')$ since we do not increase the number of blocks when creating $(\mathbb{S}^*, \mathbb{P}^*)$.

Let $\mathbb{S}$ be a sequence formed by replacing every pair (S'_{2i-1}, S'_{2i}) from $\mathbb{S}^*$, such that $i \in [1, |S|]$, with the original gene from $\mathcal{G}$ and let $\mathbb{P}$ be a sequence formed by replacing every pair (P'_{2i-1}, P'_{2i}) from $\mathbb{P}^*$, such that $i \in [1, |S|]$, with the original gene from $\mathcal{H}$.

Note that, every block or breakpoint in $\mathbb{S}$ corresponds to a block or breakpoint in $\mathbb{S}^*$, every block or breakpoint in $\mathbb{P}$ corresponds to a block or breakpoint in $\mathbb{P}^*$, and reversing a block from $\mathbb{S}$ corresponds to reversing a block from $\mathbb{S}^*$. Since there is a correspondence between blocks of $\mathbb{S}^*$ and blocks of $\mathbb{P}^*$, there is also a correspondence between blocks of $\mathbb{S}$ and blocks of $\mathbb{P}$. Consequently, $(\mathbb{S}, \mathbb{P})$ is a signed partition between $\mathcal{G}$ and $\mathcal{H}$, such that $cost(\mathbb{S}, \mathbb{P}) = cost(\mathbb{S}^*, \mathbb{P}^*) \leq cost(\mathbb{S}', \mathbb{P}')$. □

Lemma 2. *Let $\mathcal{G}'$ and $\mathcal{H}$ be the unsigned extension of two signed genomes $\mathcal{G}$ and $\mathcal{H}$, respectively. For every signed intergenic partition $(\mathbb{S}, \mathbb{P})$ between $\mathcal{G}$ and $\mathcal{H}$ there is a reverse intergenic partition between $\mathcal{G}'$ and $\mathcal{H}'$, such that $cost(\mathbb{S}', \mathbb{P}') = cost(\mathbb{S}, \mathbb{P})$ and no intergenic region with odd index (the ones to which we attributed 0 in the unsigned extension) is a breakpoint.*

Proof. Given a signed intergenic partition $(\mathbb{S}, \mathbb{P})$ between $\mathcal{G}$ and $\mathcal{H}$, we can produce a reverse intergenic partition $(\mathbb{S}', \mathbb{P}')$ between $\mathcal{G}'$ and $\mathcal{H}'$ by applying the unsigned extension on every genome of $\mathbb{S}$ and $\mathbb{P}$ to produce $\mathbb{S}'$ and $\mathbb{P}'$, respectively. Note that, the breakpoints of $(\mathbb{S}', \mathbb{P}')$ are intergenic regions already present in $\mathcal{G}$ or $\mathcal{H}$, hence the cost remains the same and there is no breakpoint with odd index. □

We are interested in minimum cost partitions, as shown in the following problem statements.

Reverse Minimum Common Intergenic String Partition (RMCISP)
Input: Two balanced unsigned genomes $\mathcal{G}$ and $\mathcal{H}$.
Goal: Find a minimum cost reverse intergenic partition between $\mathcal{G}$ and $\mathcal{H}$.

Signed Minimum Common Intergenic String Partition (SMCISP)
Input: Two balanced signed genomes $\mathcal{G}$ and $\mathcal{H}$.
Goal: Find a minimum cost signed intergenic partition between $\mathcal{G}$ and $\mathcal{H}$.

Siqueira *et al.* [14] showed that the RMCISP problem is in the NP-hard class by reducing a version of the problem without intergenic regions (the Reverse Minimum Common String Partition), which is in the NP-hard class [8]. An analogous reduction can be applied between the Signed Minimum Common String Partition (a version of SMCISP without intergenic regions that is also in the NP-hard class [8]) and the SMCISP problem to show its NP-hardness.

3 Using SMCISP to Approximate the Distance Problems

This section presents a correspondence between SMCISP and the distance problems. Such correspondence allows us to adapt an approximation for the SMCISP problem to obtain an approximation for the SIRD and SIRTD problems. The following lemmas establish lower bounds for the distances based on partitions cost.

Lemma 3. *Let $(\mathbb{S}, \mathbb{P})$ be a signed minimal intergenic partition induced by an orthologous assignment between two balanced signed genomes $\mathcal{G} = (S, \breve{S})$ and $\mathcal{H} = (P, \breve{P})$. For any intergenic transposition $\tau^{(i,j,k)}_{(x,y,z)}$, the signed minimal intergenic partition $(\mathbb{R}, \mathbb{Q})$ between the genomes $\mathcal{G}.\tau^{(i,j,k)}_{(x,y,z)}$ and $\mathcal{H}$, induced by the same orthologous assignment, respects the restriction $cost(\mathbb{R}, \mathbb{Q}) \geq cost(\mathbb{S}, \mathbb{P}) - 3$.*

Proof. As the signed intergenic partition $(\mathbb{R}, \mathbb{Q})$ must be induced by the same assignment of $(\mathbb{S}, \mathbb{P})$, we can only reduce the cost of the signed intergenic partition by moving the blocks to allow their combination. The intergenic transposition may be able to combine three pairs of blocks: the block ending in S_{i-1} with the block starting in S_j; the block ending in S_{k-1} with the block starting in S_i; and the block ending in S_{j-1} with the block starting in S_k. In the best case, if all three combinations occur, we have $cost(\mathbb{R}, \mathbb{Q}) = cost(\mathbb{S}, \mathbb{P}) - 3$. □

Lemma 4. *Let $(\mathbb{S}, \mathbb{P})$ be a signed minimal intergenic partition induced by an orthologous assignment between two balanced signed genomes $\mathcal{G} = (S, \breve{S})$ and $\mathcal{H} = (P, \breve{P})$. For any intergenic reversal $\rho^{(i,j)}_{(x,y)}$, the signed minimal intergenic partition $(\mathbb{R}, \mathbb{Q})$ between the genomes $\mathcal{G}.\rho^{(i,j)}_{(x,y)}$ and $\mathcal{H}$, induced by the same orthologous assignment, respects the restriction $cost(\mathbb{R}, \mathbb{Q}) \geq cost(\mathbb{S}, \mathbb{P}) - 2$.*

Proof. Similar to the proof of Lemma 3, considering that the intergenic reversal $\rho^{(i,j)}_{(x,y)}$ can combine at most two pairs of blocks: the block ending in S_{i-1} with the block ending in S_j and the block starting in S_{j+1} with the block starting in S_i. □

Lemma 5. *Let $(\mathbb{S}, \mathbb{P})$ be a signed intergenic partition of minimum cost between two balanced signed genomes $\mathcal{G} = (S, \breve{S})$ and $\mathcal{H} = (P, \breve{P})$. Any sequence of intergenic reversals that transforms S into P must have size at least $\frac{cost(\mathbb{S}, \mathbb{P})}{2}$.*

Proof. Consider a sequence of k intergenic reversals capable of transforming $\mathcal{G}$ into $\mathcal{H}$. Such sequence establishes an orthologous assignment between $\mathcal{G}$ and $\mathcal{H}$. The assignment is recovered by verifying, for each character of S, the new position in P, after the intergenic reversals are applied.

Let $(\mathbb{R}, \mathbb{Q})$ be the signed minimal intergenic partition induced from the orthologous assignment. We know that $\frac{cost(\mathbb{R}, \mathbb{Q})}{2} \leq k$, because each intergenic reversal can remove at most 2 breakpoints

(Lemma 4) and k intergenic reversals are sufficient to turn $\mathbb{R}$ into $\mathbb{Q}$ (i.e., k intergenic reversals can remove all breakpoints). As $(\mathbb{S}, \mathbb{P})$ is a minimum cost signed intergenic partition, we have $\frac{|(\mathbb{S},\mathbb{P})|}{2} \leq \frac{|(\mathbb{R},\mathbb{Q})|}{2} \leq k$. □

Lemma 6. *Let $(\mathbb{S}, \mathbb{P})$ be a signed intergenic partition of minimum cost between two balanced signed genomes $\mathcal{G} = (S, \breve{S})$ and $\mathcal{H} = (P, \breve{P})$. Any sequence composed of intergenic reversals and intergenic transpositions that transforms S into P must have size at least $\frac{cost(\mathbb{S},\mathbb{P})}{3}$.*

Proof. Analogous to the proof of Lemma 5, but using Lemma 3 in combination with Lemma 4. □

With the bounds presented on the previous lemmas, we can establish a relation between partition and distance problems.

Theorem 1. *Let $\mathcal{G} = (S, \breve{S})$ and $\mathcal{H} = (P, \breve{P})$ be two co-tailed genomes and let $(\mathbb{S}, \mathbb{P})$ be the signed intergenic partition between $\mathcal{G}$ and $\mathcal{H}$ returned by an ℓ-approximation for the SMCISP problem. If there is an algorithm that produces a sequence of k reversals, with $k \leq r\ cost(\mathbb{S}, \mathbb{P})$, capable of transform $\mathcal{G}$ into $\mathcal{H}$, then it is a $2r\ell$-approximation for the SIRD problem.*

Proof. Let p be the size of the minimum signed intergenic partition between $\mathcal{G}$ and $\mathcal{H}$. An ℓ-approximation algorithm for the SMCISP problem returns a signed intergenic partition $(\mathbb{S}, \mathbb{P})$, such that $p \leq cost(\mathbb{S}, \mathbb{P}) \leq \ell p$.

By Lemma 5, we know that $d_{\mathcal{SIR}}(\mathcal{G}, \mathcal{H}) \geq \frac{p}{2}$. As we can turn $\mathcal{G}$ into $\mathcal{H}$ with k reversals, we have $d_{\mathcal{SIR}}(\mathcal{G}, \mathcal{H}) \leq k \leq 2r\ell d_{\mathcal{SIR}}(\mathcal{G}, \mathcal{H})$. □

Corollary 1. *An ℓ-approximation for the SMCISP problem ensures a 4ℓ-approximation for the SIRD problem.*

Proof. Let $\mathcal{G} = (S, \breve{S})$ and $\mathcal{H} = (P, \breve{P})$ be two co-tailed genomes and $(\mathbb{S}, \mathbb{P})$ the signed intergenic partition given by the 4ℓ-approximation for the SMCISP problem. First, we apply the unsigned extension to $\mathcal{G}$ and $\mathcal{H}$ turning them into the unsigned strings $\mathcal{G}' = (S', \breve{S}')$ and $\mathcal{H}'$, respectively. By Lemma 2, there is a reverse intergenic partition $(\mathbb{S}', \mathbb{P}')$ of $\mathcal{G}'$ and $\mathcal{H}'$, such that $cost(\mathbb{S}', \mathbb{P}') = cost(\mathbb{S}, \mathbb{P})$ and all breakpoints are on even indexed intergenic regions.

Next, we can use an algorithm from Brito *et al.* [5, Theorem 15] to find a sequence R' of $k = 2cost(\mathbb{S}', \mathbb{P}') = 2cost(\mathbb{S}, \mathbb{P})$ reversals that turn $\mathcal{G}'$ into $\mathcal{H}'$ and only acts on breakpoints of $(\mathbb{S}', \mathbb{P}')$ (for every reversal $\rho^{(i,j)}_{(x,y)}$ of R', $\breve{S}'_{i-1}$ and $\breve{S}'_j$ are breakpoints). As the reversals of R' act only on breakpoints we can construct a correspondent sequence R of k reversals that turns $\mathcal{G}$ into $\mathcal{H}$ by replacing every reversal $\rho^{(i,j)}_{(x,y)}$ of R with a reversal $\rho^{(i/2,j/2)}_{(x,y)}$.

By setting $r = 2$ in Theorem 1, we have a 4ℓ-approximation for the SIRD. □

Theorem 2. *Let $\mathcal{G} = (S, \breve{S})$ and $\mathcal{H} = (P, \breve{P})$ be two co-tailed genomes and let $(\mathbb{S}, \mathbb{P})$ be the sigend intergenic partition between $\mathcal{G}$ and $\mathcal{H}$ returned by an ℓ-approximation for the SMCISP problem. If there is an algorithm that produces a sequence of k reversals or transpositions, with $k \leq r\ cost(\mathbb{S}, \mathbb{P})$, capable of transform $\mathcal{G}$ into $\mathcal{H}$, then it is a $3r\ell$-approximation for the SIRTD problem.*

Proof. Analogous to the proof of Theorem 1, but using Lemma 6 instead of Lemma 5. □

Corollary 2. *An ℓ-approximation for the SMCISP problem ensures a 4.5ℓ-approximation for the SIRTD problem.*

Proof. Analogous to Corollary 1, but using Theorem 2 instead of Theorem 1 and another algorithm from Brito *et al.* [5, Lemma 20] to find a sequence R' of $k = \frac{3}{2}cost(\mathbb{S}', \mathbb{P}') = \frac{3}{2}cost(\mathbb{S}, \mathbb{P})$ reversals and transpositions that turn $\mathcal{G}'$ into $\mathcal{H}'$ and only acts on breakpoints of $(\mathbb{S}', \mathbb{P}')$. □

Now, we are going to reduce the SMCISP to the RMCISP problem, so we can use the 2k-approximation for RMCISP from Siqueira *et al.* [14] (which we call Algorithm-R) to ensure an 2k-approximation to the SMCISP problem, where k is the maximum occurrence of any gene in the input genomes.

Let us define an algorithm to the SMCISP problem (Algorithm-S) with the following steps:

1. Take the unsigned extensions $\mathcal{G}'$ and $\mathcal{H}'$ of $\mathcal{G}$ and $\mathcal{H}$, respectively.
2. Use Algorithm-R to obtain a reverse intergenic partition $(\mathbb{S}', \mathbb{P}')$ between $\mathcal{G}'$ and $\mathcal{H}'$.
3. Take the signed intergenic partition $(\mathbb{S}, \mathbb{P})$ between $\mathcal{G}$ and $\mathcal{H}$ given by Lemma 1.

Theorem 3. *Algorithm-S has approximation factor of* $2k$ *for the SMCISP problem between the genomes* $\mathcal{G} = (S, \breve{S})$ *and* $\mathcal{H} = (P, \breve{P})$*, where* $k = occ(S)$*.*

Proof. Let $(\mathbb{S}', \mathbb{P}')$ be the reverse minimum common intergenic partition between $\mathcal{G}'$ and $\mathcal{H}'$, and let $(\mathbb{R}', \mathbb{P}')$ be the reverse common intergenic partition returned by Algorithm-R. We know that $cost(\mathbb{R}', \mathbb{Q}') \leq 2kcost(\mathbb{S}', \mathbb{P}')$ [14, Theorem 8]. Besides, for a signed minimum common intergenic partition $(\mathbb{S}, \mathbb{P})$ between $\mathcal{G}$ and $\mathcal{H}$, we must have $cost(\mathbb{S}', \mathbb{P}') \leq cost(\mathbb{S}, \mathbb{P})$, because for every signed intergenic partition between $\mathcal{G}$ and $\mathcal{H}$ there is a reverse intergenic partition of same cost between $\mathcal{G}'$ and $\mathcal{H}'$ (Lemma 2). As the partition $(\mathbb{R}, \mathbb{Q})$ given by Lemma 1 applied to $(\mathbb{R}', \mathbb{Q}')$ is such that $cost(\mathbb{R}, \mathbb{Q}) \leq cost(\mathbb{R}', \mathbb{Q}')$, we have $cost(\mathbb{R}, \mathbb{Q}) \leq 2kcost(\mathbb{S}, \mathbb{P})$ and the theorem follows. □

It is worth noting that our algorithm may be applied to a version of the problem without intergenic regions (by setting every intergenic region to 0), keeping the same approximation factor for the problem. That improves the previously known $\Theta(k)$ approximation for that variation of the problem from $8k$ [9] to $2k$.

Corollary 3. *Algorithm-S in combination with the algorithm described in Corollary 1, ensures an approximation factor of* $8k$ *for the SIRD problem between genomes* $\mathcal{G} = (S, \breve{S})$ *and* $\mathcal{H} = (P, \breve{P})$*, where* $k = occ(S)$*.*

Proof. Directly from Corollary 1 and Theorem 3. □

Corollary 4. *Algorithm-S in combination with the algorithm described in Corollary 2, ensures an approximation factor of* $9k$ *for the SIRTD problem between genomes* $\mathcal{G} = (S, \breve{S})$ *and* $\mathcal{H} = (P, \breve{P})$*, where* $k = occ(S)$*.*

Proof. Directly from Corollary 2 and Theorem 3. □

4 Experimental Results

This section presents the experimental results using a database of simulated genomes. Our partition algorithm was implemented in Haskell and the distance algorithms were implemented in C++. These implementations are available in a public repository[1]. The experiments were

[1] https://github.com/compbiogroup/Approximation-Algorithm-for-Rearrangement-Distances-Considering-Repeated-Genes-and-Intergenic-Region

conducted on a PC equipped with a 2.3GHz Intel® Xeon® CPU E5-2470 v2, with 40 cores and 32 GB of RAM, running Ubuntu 18.04.2.

We generated a simulated genome database with 80 sets of genomes. Each set is defined by: (i) the set of operations M used ($M = R$ if reversals were used to create the database and $M = RT$ if reversals and transpositions were used); (ii) the number $o \in \{25, 50, 75, 100\}$ of operations applied to produce the target genome; and (iii) the size $l \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ of the alphabet of possible labels. Each set was created by the following procedure:

1. 1000 source genomes were generated. For each one, we selected 100 characters (genes) by choosing labels from a uniform distribution of ℓ labels (each label could be selected more than once) and assigning $+$ and $-$ signs randomly. Besides, we select 101 integers (intergenic regions) from the interval $[0, 100]$ (each value had the same probability of being chosen).

2. For each source genome $\mathcal{G} = (S, \breve{S})$, we created a target genome $\mathcal{H} = (P, \breve{P})$, by applying o operations in S. If $M = R$, we applied o intergenic reversals $\rho_{(x,y)}^{(i,j)}$, where the values of i, j, x, and y were randomly chosen. If $M = RT$, we applied $\lfloor \frac{o}{2} \rfloor$ intergenic reversals and $\lceil \frac{o}{2} \rceil$ intergenic transpositions. These operations were applied in a random order and the parameters of each one were randomly chosen.

3. For each genome (sources and targets) we include two new genes. One at the beginning of the genome and one at the ending of the genome. The same two genes are used for all genomes, which ensure that all pairs of source and target genomes are co-tailed. Note that, the final genomes have size 102.

In our experimental tests, we generated 200 random orthologous assignments between each pair of genomes, such that 100 of these assignments had no restriction and the other 100 were orthologous assignments compatible with the partitions returned by our algorithm for the SMCISP problem. For each assignment, we computed the intergenic reversal distance when the genome pair was generated using only reversals, and the intergenic reversal and transposition distance when the genome pair was generated using reversals and transpositions. These distances were computed with the algorithms from Brito *et al.* [5].

Table 1 summarizes the results of the distance algorithms applied to the generated instances. Each row shows the average results considering all instances for a fixed M and o.

The third to fifth columns show the results considering instances generated without the use of the partitions, the columns correspond respectively to the average of the following values: the minimum distance considering all orthologous assignments of a genome pair (Distance/Min.), the average distance considering all orthologous assignments of a genome pair (Distance/Avg.), and the time to calculate the distances for the assignments of a genome pair (Time/Distance). The sixth to ninth columns show similar values for the case where the partition is used, in that case an additional column shows the average time to produce the partitions for a genome pair (Time/Partition).

Comparing the values of Table 1, we can see that the use of the partition algorithm not only ensures an approximation, but also improves the practical results. With the only exception of sets generated with 100 operations of reversal and transposition, the minimum distances are lower with partition. Additionally, the average distances using partitions are lower in all cases. The difference between the distances with and without partition is higher when fewer operations were used to create the instances. The average distance was less than 2.2 times the number of operations applied when $o = 25$ and gets closer to the number of operations for higher values of o.

Table 1: Distances for the SIRD (lines with M=R) and SIRTD (lines with M=RT) problems. Each line shows the average distances (minimum and average of all orthologous assignments) and the average execution times (to calculate the distances or the partitions) for the instances generated with o operations of the correspondent problem.

M	o	Without Partition			With Partition			
		Distance		Time	Distance		Time	
		Min.	Avg.	Distance	Min.	Avg.	Partition	Distance
R	**25**	82.90	91.70	0.07	57.68	54.20	0.10	0.03
R	**50**	91.58	97.78	0.11	81.59	78.50	0.14	0.06
R	**75**	95.17	100.18	0.11	91.45	88.44	0.20	0.08
R	**100**	96.88	101.28	0.11	96.05	93.02	0.19	0.09
RT	**25**	80.76	89.09	0.05	54.85	52.10	0.09	0.02
RT	**50**	89.93	95.40	0.06	79.81	76.97	0.15	0.04
RT	**75**	93.96	97.97	0.06	90.79	87.84	0.15	0.05
RT	**100**	95.96	99.13	0.06	96.04	93.02	0.18	0.06

Looking at the running time, we see that considering the time to generate the partitions and to calculate the distances, the case using the partition is slower than the case without partition. However, in most cases, the running time for distances considering the partition is lower than the running time for distances that do not consider the partition, so increasing the number of orthologous assignments used will have a lower impact on time using the partition.

5 Conclusion

We defined the Signed Intergenic Reversal Distance (SIRD) and the Signed Intergenic Reversal and Transposition Distance (SIRTD) problems. Besides, we defined the Signed Minimum Common Intergenic String Partition (SMCISP) problem, and showed that it is related to the distances problems. Using that relation and adapting algorithms for the unsigned versions of the problems, we show a $8k$-approximation for SIRD and a $9k$-approximation for SIRTD, where k is the maximum number of occurrences of a label in the strings.

Our practical tests show that even with the high approximation factor, the distances obtained with the help of our partition algorithm are less than 2.2 times the number of operations applied to produce the instances. Additionally, on average, distances using orthologous assignments that take the partition algorithm into account are lower than distances using any random orthologous assignment.

As future work, one can explore whether other techniques, such as heuristics to produce the orthologous assignment [15] or algorithms using cycle decomposition [11, 12], can decrease the values of the obtained distances. One can also consider the addition of non-conservative events (that change the genetic content of the genomes), such as indels (insertion and deletion of genetic material) [1].

Acknowledgments

This work was supported by the National Council of Technological and Scientific Development, CNPq (grants 161019/2021-8 and 202292/2020-7), the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and the São Paulo Research Foundation, FAPESP (grants 2013/08293-7, 2015/11937-9, and 2017/12646-3).

References

[1] Alexsandro Oliveira Alexandrino, Klairton Lima Brito, Andre Rodrigues Oliveira, Ulisses Dias, and Zanoni Dias. Reversal distance on genomes with different gene content and intergenic regions information. In *Algorithms for Computational Biology*, volume 12715, pages 121–133. Springer International Publishing, 2021.

[2] David A. Bader, Bernard M. E. Moret, and Mi Yan. A Linear-Time Algorithm for Computing Inversion Distance Between Signed Permutations with an Experimental Study. *Journal of Computational Biology*, 8:483–491, 2001.

[3] Priscila Biller, Laurent Guéguen, Carole Knibbe, and Eric Tannier. Breaking Good: Accounting for Fragility of Genomic Regions in Rearrangement Distance Estimation. *Genome Biology and Evolution*, 8(5):1427–1439, 2016.

[4] Priscila Biller, Carole Knibbe, Guillaume Beslon, and Eric Tannier. Comparative Genomics on Artificial Life. In *Pursuit of the Universal*, pages 35–44. Springer International Publishing, 2016.

[5] Klairton Lima Brito, Géraldine Jean, Guillaume Fertin, Andre Rodrigues Oliveira, Ulisses Dias, and Zanoni Dias. Sorting by Genome Rearrangements on both Gene Order and Intergenic Sizes. *Journal of Computational Biology*, 27(2):156–174, 2020.

[6] Xin Chen, Jie Zheng, Zheng Fu, Peng Nan, Yang Zhong, Stefano Lonardi, and Tao Jiang. Assignment of Orthologous Genes via Genome Rearrangement. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(4):302–315, 2005.

[7] Graham Cormode and S. Muthukrishnan. The string edit distance matching problem with moves. *ACM Transactions on Algorithms*, 3(1):1–19, 2007.

[8] Avraham Goldstein, Petr Kolman, and Jie Zheng. Minimum Common String Partition Problem: Hardness and Approximations. In *Proceedings of the 15th International Symposium on Algorithms and Computation (ISAAC'2004)*, pages 484–495. Springer Berlin Heidelberg, 2005.

[9] Petr Kolman and Tomasz Waleń. Reversal Distance for Strings with Duplicates: Linear Time Approximation Using Hitting Set. In *Proceedings of the 4th International Workshop on Approximation and Online Algorithms (WAOA'2006)*, pages 279–289. Springer Berlin Heidelberg, 2007.

[10] Andre Rodrigues Oliveira, Klairton Lima Brito, Ulisses Dias, and Zanoni Dias. On the Complexity of Sorting by Reversals and Transpositions Problems. *Journal of Computational Biology*, 26:1223–1229, 2019.

[11] Andre Rodrigues Oliveira, Geraldine Jean, Guillaume Fertin, Klairton Lima Brito, Laurent Bulteau, Ulisses Dias, and Zanoni Dias. Sorting Signed Permutations by Intergenic Reversals. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2020. In press.

[12] Andre Rodrigues Oliveira, Geraldine Jean, Guillaume Fertin, Klairton Lima Brito, Ulisses Dias, and Zanoni Dias. Sorting permutations by intergenic operations. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021. In press.

[13] Andrew J. Radcliffe, Alex D. Scott, and Elizabeth L. Wilmer. Reversals and Transpositions Over Finite Alphabets. *SIAM Journal on Discrete Mathematics*, 19(1):224–244, 2005.

[14] Gabriel Siqueira, Alexsandro Oliveira Alexandrino, Andre Rodrigues Oliveira, and Zanoni Dias. Approximation algorithm for rearrangement distances considering repeated genes and intergenic regions. *Algorithms for Molecular Biology*, 16:21, 2021.

[15] Gabriel Siqueira, Klairton Lima Brito, Ulisses Dias, and Zanoni Dias. Heuristics for genome rearrangement distance with replicated genes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021. In press.

[16] Maria E. M. T. Walter, Zanoni Dias, and João Meidanis. Reversal and Transposition Distance of Linear Chromosomes. In *Proceedings of the 5th International Symposium on String Processing and Information Retrieval (SPIRE'1998)*, pages 96–102, Los Alamitos, CA, USA, 1998. IEEE Computer Society.