Super short operations on both gene order and intergenic sizes

Andre Oliveira, Géraldine Jean, Guillaume Fertin, Ulisses Dias, Zanoni Dias

Oliveira *et al. Algorithms Mol Biol* (2019) 14:21
https://doi.org/10.1186/s13015-019-0156-5

Algorithms for Molecular Biology

RESEARCH Open Access

Super short operations on both gene order and intergenic sizes

Andre R. Oliveira[1*], Géraldine Jean[2], Guillaume Fertin[2], Ulisses Dias[3] and Zanoni Dias[1]

Abstract

Background: The evolutionary distance between two genomes can be estimated by computing a minimum length sequence of operations, called *genome rearrangements*, that transform one genome into another. Usually, a genome is modeled as an ordered sequence of genes, and most of the studies in the genome rearrangement literature consist in shaping biological scenarios into mathematical models. For instance, allowing different genome rearrangements operations at the same time, adding constraints to these rearrangements (e.g., each rearrangement can affect at most a given number of genes), considering that a rearrangement implies a cost depending on its length rather than a unit cost, etc. Most of the works, however, have overlooked some important features inside genomes, such as the presence of sequences of nucleotides between genes, called *intergenic regions*.

Results and conclusions: In this work, we investigate the problem of computing the distance between two genomes, taking into account both gene order and intergenic sizes. The genome rearrangement operations we consider here are constrained types of reversals and transpositions, called *super short reversals* (SSRs) and *super short transpositions* (SSTs), which affect up to two (consecutive) genes. We denote by *super short operations* (SSOs) any SSR or SST. We show 3-approximation algorithms when the orientation of the genes is not considered when we allow SSRs, SSTs, or SSOs, and 5-approximation algorithms when considering the orientation for either SSRs or SSOs. We also show that these algorithms improve their approximation factors when the input permutation has a higher number of inversions, where the approximation factor decreases from 3 to either 2 or 1.5, and from 5 to either 3 or 2.

Keywords: Genome rearrangements, Intergenic regions, Super short operations, Approximation algorithms

Background

Given two genomes $\mathcal{G}_1$ and $\mathcal{G}_2$, one way to estimate their evolutionary distance is to compute the minimum number of large scale events, called *genome rearrangements*, that are needed to transform $\mathcal{G}_1$ into $\mathcal{G}_2$. The minimality is required due to the commonly accepted parsimony principle, while the allowed genome rearrangements depend on the model, i.e. on the classes of events that supposedly happen during evolution.

Prior to counting rearrangement events, one needs to model the input genomes. Previous works [1–3] have defined genomes as ordered sequences of elements (*genes*). Variants within this setting can occur. For instance, each gene may appear either once or several times in a genome. In the latter case, genomes are modeled as strings, while in the former case they are modeled as *permutations*. Besides, genomes modeled as permutations may be signed or unsigned (the sign of an element represents the orientation of that gene in the DNA strand it lies on).

Concerning genome rearrangements, the most commonly studied events are *reversal*, which consists in taking a continuous sequence in the genome, reversing it, and putting it back at the same location [4], and *transposition*, which consists in taking a continuous sequence in the genome and putting it back in a different location [5]. A more recent and general type of genome rearrangement is the DCJ (Double-Cut and Join) [3], that cuts a genome between adjacent genes a and b, and adjacent genes c and d, and joins either a to c and b to d, or a to d and b to c.

*Correspondence: andrero@ic.unicamp.br
[1] Institute of Computing, University of Campinas, Campinas, Brazil
Full list of author information is available at the end of the article

Since the mid-nineties, a very large amount of work has been done for computing distances between pairs of genomes, depending on the genome model and the allowed set of rearrangements. We refer the reader to Fertin et al. book [6] for a survey of algorithmic aspects.

In populations where the number of rearrangement events that affect a very large portion of the genes are rare, we can restrict events to span at most k genes, for some value of k [7, 8]. During an analysis with closely-related pairs of bacterial genomes, the number of short inversions (inversions affecting up to three genes) was discovered to be very high, especially inversions of a single gene (which we call 1-reversal) [9]. There are also other works showing the prevalence of short inversions in bacterial genomes [10] and eukaryotes genomes [11, 12].

As previously mentioned, most of the works have assumed that a genome is an ordered sequence of genes. It has been argued that this model could underestimate the "true" evolutionary distance, and that other genome features should be taken into account to circumvent this problem [13, 14]. Indeed, genomes carry more information than just their ordered sequences of genes. In particular, consecutive genes are separated by DNA sequences called *intergenic regions*, each having different lengths in terms of number of nucleotides. These lengths may be used along with gene order to generate a more realistic model for genomes.

This recently led some authors to model a genome as an ordered sequence of genes, together with an ordered list of its intergenic sizes, and to consider the problem of computing the DCJ distance, either in the case where insertions and deletions of nucleotides are forbidden [15], or allowed [16].

Biller and coauthors [13] used the intergenic regions to define what they called *fragile regions*, regions where rearrangements are more likely to act. After identifying these fragile regions, practical tests showed that considering rearrangements on non-fragile regions can yield incoherent distance estimations. When using the equiprobable model (i.e., rearrangements can occur in any position with the same probability), practical tests [16] showed that statistical properties of the inferred scenarios for DCJs using intergenic regions are closer to the true ones than scenarios which do not use them.

In this work, we also consider genomes as ordered sequences of genes together with their intergenic sizes, in cases where the gene sequence is an unsigned or signed permutation and the considered rearrangement operations are *super short reversal* (or SSR, i.e. a reversal of (gene) length at most two), *super short transposition* (or SST, i.e. a transposition affecting only two genes), or both (*super short operation* or SSO). In this context, our goal is to determine the minimum number of SSRs/SSTs/SSOs that transform one genome into another.

This paper is organized as follows. In Section 2 we provide the notations that we will use throughout the paper, and we introduce novel ideas that will prove useful for studying the problem. In sections 3-7 we derive lower and upper bounds on the sought distance for five different variants, which help us design an approximation algorithm of constant factor for each of these five problems. Section 8 presents a practical analysis of the five algorithms on simulated instances. Section 9 concludes the paper.

Definitions

We represent a genome $\mathcal{G}$ with n genes as an instance with (i) an n-tuple and (ii) $n+1$ intergenic regions. If there is no duplicated genes, the n-tuple is a (possibly signed) permutation $\pi = (\pi_1 \pi_2 \cdots \pi_{n-1} \pi_n)$, with $|\pi_i| \in \{1, 2, ..., (n-1), n\}$, for $1 \leq i \leq n$, and $|\pi_i| = |\pi_j|$ if, and only if, $i = j$. If gene orientation is known, each element from π has a $+$ or $-$ sign that indicates the gene orientation it represents, and we say that π is a *signed permutation*; π is an *unsigned permutation* otherwise.

We denote by ι the *identity permutation*, in which all elements are in ascending order and with positive signs. The *extended permutation* is obtained from π by adding two new elements: $\pi_0 = 0$ and $\pi_{n+1} = (n+1)$.

The intergenic region r_i^π is located before element π_i from the extended permutation π, for $1 \leq i \leq n+1$. We denote by $\ell(r_i^\pi)$ the *length* of intergenic region r_i^π, i.e., the number of nucleotides in r_i^π, with $\ell(r_i^\pi) \in \mathbb{N}$ for $1 \leq i \leq n+1$. Let $r^\pi = (\ell(r_1^\pi), ..., \ell(r_{n+1}^\pi))$. An *instance* here is then formed by (π, r^π).

A *reversal* $\rho(i, j, x, y)$ applied over an instance (π, r^π), with $1 \leq i \leq j \leq n$, $0 \leq x \leq \ell(r_i^\pi)$, $0 \leq y \leq \ell(r_{j+1}^\pi)$, and $\{x, y\} \in \mathbb{N}$, is an operation that generates $(\pi', r^{\pi'})$ by (i) reversing the order and the orientation of the elements in the subset of adjacent elements $\{\pi_i, ..., \pi_j\}$; (ii) reversing the order of intergenic regions in the subset of adjacent intergenic regions $\{r_{i+1}^\pi, ..., r_j^\pi\}$ when $j > i+1$; (iii) *cutting* two intergenic regions: r_i^π after x nucleotides and r_{j+1}^π after y nucleotides such that $\ell(r_i^{\pi'}) = x + y$ and $\ell(r_{j+1}^{\pi'}) = (\ell(r_i^\pi) - x) + (\ell(r_{j+1}^\pi) - y)$.

A reversal $\rho(i, j, x, y)$ is also called a g-reversal, where $g = (j - i) + 1$. A *super short reversal* is a 1-reversal or a 2-reversal, i.e. a reversal that affects only one or two elements from π.

A *transposition* $\tau(i, j, k, x, y, z)$ applied over an instance (π, r^π), with $1 \leq i < j < k \leq n+1$, $0 \leq x \leq \ell(r_i^\pi)$, $0 \leq y \leq \ell(r_j^\pi)$, $0 \leq z \leq \ell(r_k^\pi)$, and $\{x, y, z\} \in \mathbb{N}$, is an operation that generates $(\pi', r^{\pi'})$ by (i) exchanging subsets of adjacent elements $\{\pi_i, ..., \pi_{j-1}\}$ and $\{\pi_j, ..., \pi_{k-1}\}$; (ii) moving subsets of adjacent intergenic regions $\{r_{j+1}^\pi, ..., r_{k-1}^\pi\}$

and $\{r^{\pi}_{i+1}, ..., r^{\pi}_{j-1}\}$ to start at positions $(i+1)$ and $(i+k-j+1)$, respectively; (iii) *cutting* three intergenic regions: r^{π}_{i}, r^{π}_{j}, and r^{π}_{k} such that $\ell(r^{\pi'}_{i}) = x + \ell(r^{\pi}_{j}) - y$, $\ell(r^{\pi'}_{i+k-j}) = \ell(r^{\pi}_{i}) - x + z$, and $\ell(r^{\pi'}_{k}) = \ell(r^{\pi}_{k}) - z + y$.

A transposition $\tau(i, j, k, x, y, z)$ is called a g-transposition, where $g = k - i$, and we say that a g-transposition is *super short* if $g = 2$.

Figure 1 shows a sequence of two super short reversals and one super short transposition that transforms the permutation $\pi = (1\ 3\ 4\ 2\ 5)$ with $r^{\pi} = (3, 5, 2, 1, 2, 8)$ into $\iota = (1\ 2\ 3\ 4\ 5)$ with $r^{\iota} = (3, 2, 6, 4, 5, 1)$.

Given an instance (π, r^{π}), a pair of elements (π_i, π_j) from π is called an *inversion* if $\pi_i > \pi_j$ and $i < j$, with $\{i, j\} \in [1..n]$. We denote the number of inversions in a permutation π by $inv(\pi)$. For the example in Fig. 1a, pairs (3, 2) and (4, 2) are the only inversions, thus $inv(\pi) = 2$.

Given two instances (π, r^{π}) and (α, r^{α}) representing genomes $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively such that π and α have the same number of elements, $(\ell(r^{\pi}_{i}) - \ell(r^{\alpha}_{i}))$ is the *imbalance* between intergenic regions r^{π}_{i} and r^{α}_{i}, with $1 \leq i \leq m$.

Given two instances (π, r^{π}) and (α, r^{α}) such that (i) π and α have the same number of elements and (ii) $\sum_{i=1}^{m} \ell(r^{\pi}_{i}) = \sum_{i=1}^{m} \ell(r^{\alpha}_{i})$, let $\Delta_j(r^{\pi}, r^{\alpha}) = \sum_{i=1}^{j} (\ell(r^{\pi}_{i}) - \ell(r^{\alpha}_{i}))$ denote the *cumulative sum of imbalances* between intergenic regions of π and α from positions 1 to j, with $1 \leq j \leq m$. Since $\sum_{i=1}^{m} \ell(r^{\pi}_{i}) = \sum_{i=1}^{m} \ell(r^{\alpha}_{i})$, we have that $\Delta_m(r^{\pi}, r^{\alpha}) = 0$.

From now on, we will consider that (i) the target permutation α is such that $\alpha = \iota$; (ii) π and ι have the same number of elements; and (iii) $\sum_{i=1}^{m} \ell(r^{\pi}_{i}) = \sum_{i=1}^{m} \ell(r^{\alpha}_{i})$. By doing this, we can compute the *distance* of π, denoted by $d(\pi)$, that consists in finding the minimum number of super short operations that sorts π and transforms r^{π} into r^{ι}.

Let (π, r^{π}) and (ι, r^{ι}) be two instances such that π and ι have the same number of elements and $\sum_{i=1}^{m} \ell(r^{\pi}_{i}) = \sum_{i=1}^{m} \ell(r^{\iota}_{i})$. The *intergenic graph*, denoted by $I(\pi, r^{\pi}, r^{\iota}) = (V, E)$, is such that V is composed by two sets of vertices: intergenic vertices (one for each $r^{\pi}_{i} \in r^{\pi}$), and permutation vertices (one for each π_i of the extended permutation π). The set E is composed by inversion edges: an edge $e = (r^{\pi}_{i}, r^{\pi}_{i+2}) \in E$ if there is a $j \neq i$ such that (π_i, π_j) or (π_j, π_{i+1}) is an inversion, with $1 \leq i \leq n-1$ and $1 \leq j \leq n$.

We divide vertices of an intergenic graph $I(\pi, r^{\pi}, r^{\iota})$ into *components*. A component starts and ends with permutation vertices. Besides, the first component starts with the permutation vertex π_0, and the last component ends with the permutation vertex π_{n+1}. Consecutive components share exactly one permutation vertex, i.e., the last permutation vertex π_i of a component is the first permutation vertex of its adjacent component to the right.

If a component c starts with vertex π_i and ends with vertex π_j, with $i < j$, then $r^{\pi}_{k} \in c$ for $i < k \leq j$ and $\pi_k \in c$

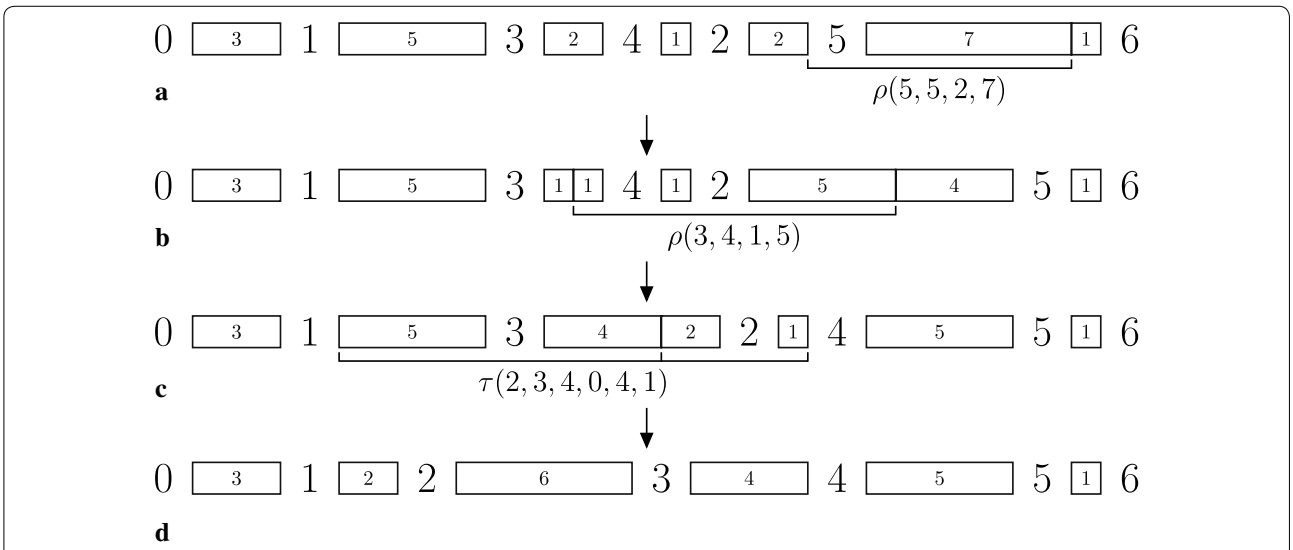

Fig. 1 A sequence of two super short reversals and one super short transposition that transforms $\pi = (1\ 3\ 4\ 2\ 5)$, with $r^{\pi} = (3, 5, 2, 1, 2, 8)$ into $\iota = (1, 2, 3, 4, 5)$, with $r^{\iota} = (3, 2, 6, 4, 5, 1)$. Intergenic regions are represented by rectangles, whose dimensions vary according to their sizes. The 1-reversal $\rho(5, 5, 2, 7)$ applied in **a** transforms π into $\pi' = \pi$, and it cuts π after position 2 at r^{π}_{5} and after position 7 at r^{π}_{6}, resulting in $\ell(r^{\pi'}_{5}) = 9$, $\ell(r^{\pi'}_{6}) = 1$, and $r^{\pi'} = (3, 5, 2, 1, 9, 1)$. The 2-reversal $\rho(3, 4, 1, 5)$ applied in **b** transforms π' into $\pi'' = (1\ 3\ 2\ 4\ 5)$, and it cuts π' after position 1 at $r^{\pi'}_{3}$ and after position 5 at $r^{\pi'}_{5}$, resulting in $\ell(r^{\pi''}_{3}) = 6$, $\ell(r^{\pi''}_{5}) = 5$, and $r^{\pi''} = (3, 5, 6, 1, 5, 1)$. Finally, the 2-transposition $\tau(2, 3, 4, 0, 4, 1)$ applied in **c** transforms π'' into ι, and it cuts π'' in position 0 at $r^{\pi''}_{2}$, after position 4 at $r^{\pi''}_{3}$, and after position 1 at $r^{\pi''}_{4}$, resulting in $\ell(r^{\pi''}_{3}) = 6$, $\ell(r^{\pi''}_{5}) = 5$, and $r^{\pi''} = (3, 5, 6, 1, 5, 1)$. as shown in **d**

for $i < k < j$. Besides, any two intergenic vertices that are connected to each other by an inversion edge must belong to the same component. Thus, if c ends with π_j, then $e = (r^\pi_j, r^\pi_{j+2}) \notin E$.

The idea is that components break (π, r^π) according to $I(\pi, r^\pi, r^\iota)$ into smaller pieces, where it is possible to make a local redistribution of intergenic regions and elements from π (with no need to exchange them between components) transforming (π, r^π) into (ι, r^ι). This requires that any component c starting with π_i and ending with π_j must have $\sum_{k=i+1}^{j} \ell(r^\pi_k) - \ell(r^\iota_k) = 0$.

Formally, given an intergenic graph $I(\pi, r^\pi, r^\iota)$, a *component* c is a minimal set of vertices from V in which: (i) any two intergenic vertices that are connected to each other by an inversion edge must belong to the same component, (ii) if $(r^\pi_i, r^\pi_j) \in g$, with $i < j$, then $\{\pi_{i-1}, \pi_j\} \in c$ and for any $i < k < j$ $\{r^\pi_k, \pi_k\} \in c$, and (iii) the sum of imbalances of its intergenic regions from r^π with respect to r^ι is equal to zero, i.e. $\sum_{\forall\, k\, s.t.\, r^\pi_k \in c} \ell(r^\pi_k) - \ell(r^\iota_k) = 0$.

A component with one intergenic vertex is called *trivial*, and is called *non-trivial* otherwise. The number of intergenic vertices in a component c is denoted by c_r. A component c is *odd* if c_r is odd, and it is *even* otherwise. The number of components in an intergenic graph $I(\pi, r^\pi, r^\iota)$ is denoted by $C(I(\pi, r^\pi, r^\iota))$, the number of odd components is denoted by $C_{odd}(I(\pi, r^\pi, r^\iota))$, and the number of even components is denoted by $C_{even}(I(\pi, r^\pi, r^\iota))$. Figure 2 shows three examples of intergenic graphs.

In the next two lemmas, we analyze the impact of applying super short operations on the number of components.

Lemma 1 *Given an instance* (π, r^π) *and a target instance* (ι, r^ι), *let* $(\pi', r^{\pi'})$ *be the resulting instance after applying a* 1*-reversal. It follows that* $C(I(\pi', r^{\pi'}, r^\iota)) \leq C(I(\pi, r^\pi, r^\iota)) + 1$.

Proof Recall that a 1-reversal $\rho(i, i, x, y)$ is applied over intergenic regions r^π_i and r^π_{i+1}, with $1 \leq i \leq n$. Besides, since 1-reversals do not create nor remove inversions from π, intergenic graphs $I(\pi', r^{\pi'}, r^\iota) = (V', E')$ and $I(\pi, r^\pi, r^\iota) = (V, E)$ satisfy $E = E'$.

If $r^\pi_i \in c$ and $r^\pi_{i+1} \notin c$, this 1-reversal is applied over two different components, which means that r^π_i is the last intergenic region of c, so $\Delta_i(r^\pi, r^\iota) = 0$. If $x + y \neq \ell(r^\pi_i)$, we have that $C(I(\pi', r^{\pi'}, r^\iota)) = C(I(\pi, r^\pi, r^\iota)) - 1$, as shown in Fig. 3a.

Consider now that $\{r^\pi_i, r^\pi_{i+1}\} \in c$. If $i < n$ and $(r^{\pi'}_i, r^{\pi'}_{i+2}) \in E'$, or $i > 1$ and $(r^{\pi'}_{i-1}, r^{\pi'}_{i+1}) \in E'$, then $C(\pi', r^{\pi'}, r^\iota) = C(\pi, r^\pi, r^\iota)$. Otherwise, we have two cases to consider: $C(\pi', r^{\pi'}, r^\iota) = C(\pi, r^\pi, r^\iota)$, if $\Delta_i(r^{\pi'}, r^\iota) \neq 0$ (as shown in Fig. 3b); and $C(\pi', r^{\pi'}, r^\iota) = C(\pi, r^\pi, r^\iota) + 1$ if $\Delta_i(r^{\pi'}, r^\iota) = 0$ (as shown in Fig. 3c). □

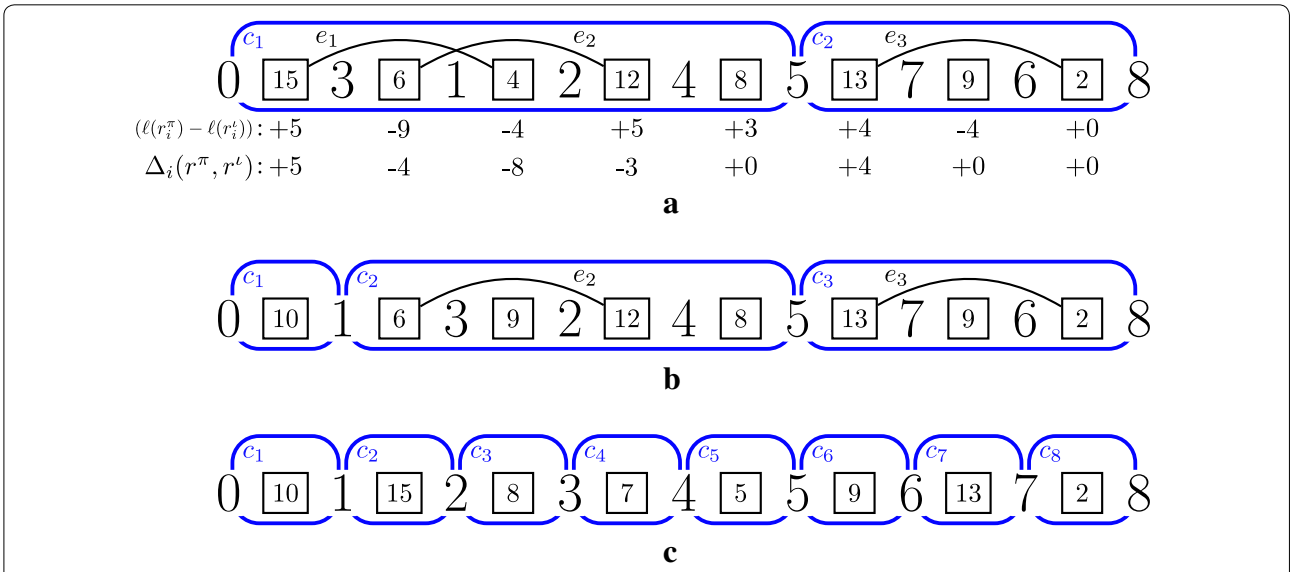

Fig. 2 Intergenic graphs $I(\pi, r^\pi, r^\iota)$, $I(\pi', r^{\pi'}, r^\iota)$, and $I(\iota, r^\iota, r^\iota)$, with π = (3 1 2 4 5 7 6), r^π = (15, 6, 4, 12, 8, 13, 9, 2), π' = (1 3 2 4 5 7 6), $r^{\pi'}$ = (10, 6, 9, 12, 8, 13, 9, 2), ι = (1 2 3 4 5 6 7), and r^ι = (10, 15, 8, 7, 5, 9, 13, 2). Black squares represent intergenic vertices, and the number inside it indicate their sizes. Rounded rectangles in blue represent components. Note that in (**a**) there are three edges in $I(\pi, r^\pi, r^\iota)$, and $C(I(\pi, r^\pi, r^\iota)) = 2$, and $C_{odd}(I(\pi, r^\pi, r^\iota)) = 2$ since there are five intergenic vertices in c_1 and three intergenic vertices in c_2. We also have in (a) all values for $(\ell(r^\pi_i) - \ell(r^\iota_i))$ and $\Delta_i(r^\pi, r^\iota)$, with $1 \leq i \leq 8$. The instance $(\pi', r^{\pi'})$ is the result of applying $\rho(1, 2, 8, 2)$ to (π, r^π). In (**b**) we can see that compared to $I(\pi, r^\pi, r^\iota)$, $I(\pi', r^{\pi'}, r^\iota)$ has one more component, and e_1 was removed. In (**c**) we can see that when we reach the target instance (ι, r^ι) the number of components is equal to the number of intergenic regions in ι (i.e., $C(I(\iota, r^\iota, r^\iota)) = n + 1 = 8$)

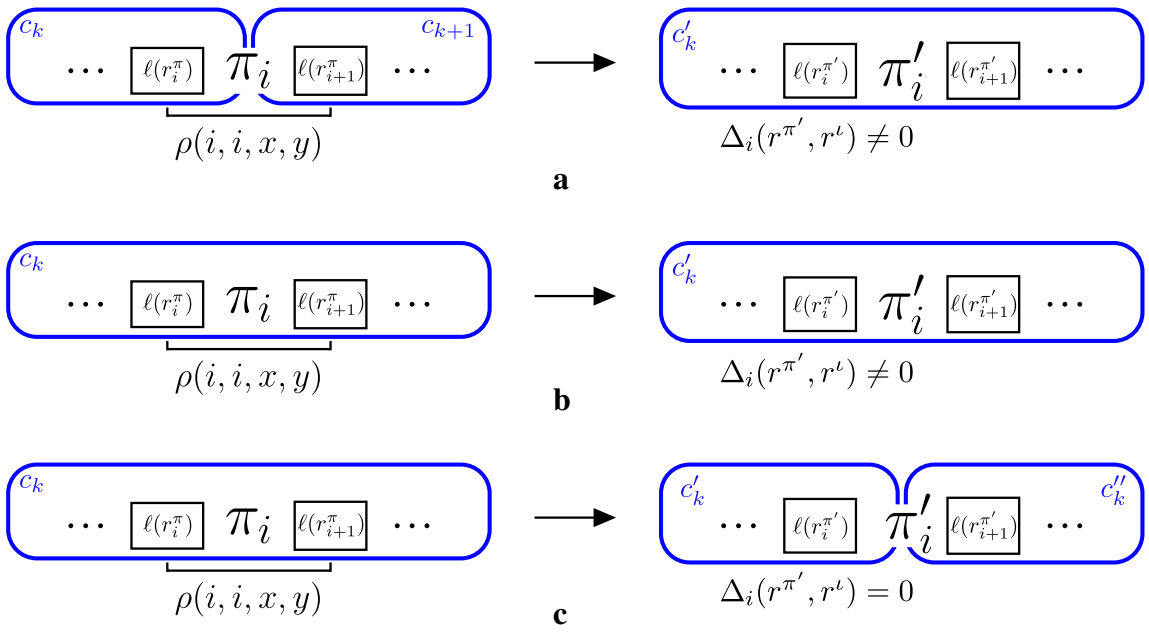

Fig. 3 Example of intergenic graphs for all possible values of $C(I(\pi', r^{\pi'}, r^\iota))$ with respect to $C(I(\pi, r^\pi, r^\iota))$ where $(\pi', r^{\pi'})$ is the resulting instance after applying a 1-reversal to (π, r^π). When the 1-reversal is applied over two components at the same time and $r_i^{\pi'} \neq r_i^\pi$, $C(I(\pi', r^{\pi'}, r^\iota)) = C(I(\pi, r^\pi, r^\iota)) - 1$, as shown in **a**. Otherwise, we have that $C(I(\pi', r^{\pi'}, r^\iota)) = C(I(\pi, r^\pi, r^\iota)) + k, k \in \{0, 1\}$, as shown in **b** and **c**

Lemma 2 *Given an instance* (π, r^π) *and a target instance* (ι, r^ι)*, let* $(\pi', r^{\pi'})$ *be the resulting instance after applying either a* 2*-reversal or a* 2*-transposition. It follows that* $C(I(\pi', r^{\pi'}, r^\iota)) \leq C(I(\pi, r^\pi, r^\iota)) + 2$.

Proof If a 2-reversal or 2-transposition is applied to intergenic regions of two different components in $I(\pi, r^\pi, r^\iota)$, then we are necessarily creating a new inversion, and the graph $I(\pi', r^{\pi'}, r^\iota)$ has either $C(I(\pi', r^{\pi'}, r^\iota)) = C(I(\pi, r^\pi, r^\iota)) - 2$ (as shown in Fig. 4a) or $C(I(\pi', r^{\pi'}, r^\iota)) = C(I(\pi, r^\pi, r^\iota)) - 1$ (as shown in Fig. 4b).

Consider now that this operation is applied to intergenic regions of a same component in $I(\pi, r^\pi, r^\iota)$, and exchanges elements π_i and π_{i+1}, with $1 \leq i < n - 1$. If the intergenic graph $I(\pi', r^{\pi'}, r^\iota) = (V', E')$ has $(r_i^{\pi'}, r_{i+2}^{\pi'}) \in E'$, then $C(I(\pi', r^{\pi'}, r^\iota)) = C(I(\pi, r^\pi, r^\iota))$. Otherwise, we have three cases to consider:

1. $C(I(\pi', r^{\pi'}, r^\iota)) = C(I(\pi, r^\pi, r^\iota))$, if $\Delta_i(r^{\pi'}, r^\iota) \neq 0$ and $\Delta_{i+1}(r^{\pi'}, r^\iota) \neq 0$ (as shown in Fig. 4c);
2. $C(I(\pi', r^{\pi'}, r^\iota)) = C(I(\pi, r^\pi, r^\iota)) + 1$ if either $\Delta_i(r^{\pi'}, r^\iota) = 0$ or $\Delta_{i+1}(r^{\pi'}, r^\iota) = 0$ (as shown in Fig. 4d);
3. $C(I(\pi', r^{\pi'}, r^\iota)) = C(I(\pi, r^\pi, r^\iota)) + 2$ otherwise (as shown in Fig. 4e). □

In the following sections, we will explore five different problems concerning super short operations but also considering intergenic regions, namely Sorting by Super Short Reversals (SbSSR), Sorting by Super Short Transpositions (SbSST), Sorting by Super Short Reversals and Super Short Transpositions (SbSSO), Sorting by Signed Super Short Reversals (SbSigSSR), and Sorting by Signed Super Short Reversals and Super Short Transpositions (SbSigSSO). Table 1 summarizes our results concerning general permutations (GP), permutations with $n\ell$ inversions for some $\ell \geq 1$ (ℓIP), permutations with at least n inversions (1IP), and permutations with at least $2n$ inversions (2IP).

Sorting by Super Short Reversals

In this section, we analyze the version of the problem when only super short reversals (i.e., 1-reversals and 2-reversals) are allowed to transform (π, r^π) into (ι, r^ι). First, we state that if a non-trivial component c of an intergenic graph $I(\pi, r^\pi, r^\iota)$ has no edge (i.e., there is no inversion inside c), then it is always possible to split c into two components with a 1-reversal.

Lemma 3 *If a component* c *of an intergenic graph* $I(\pi, r^\pi, r^\iota)$ *with* $c_r \geq 2$ *contains no edge, then there is always a pair of consecutive intergenic regions to which we can apply a* 1*-reversal that splits* c *into two components* c' *and* c'' *such that* $c'_r + c''_r = c_r$.

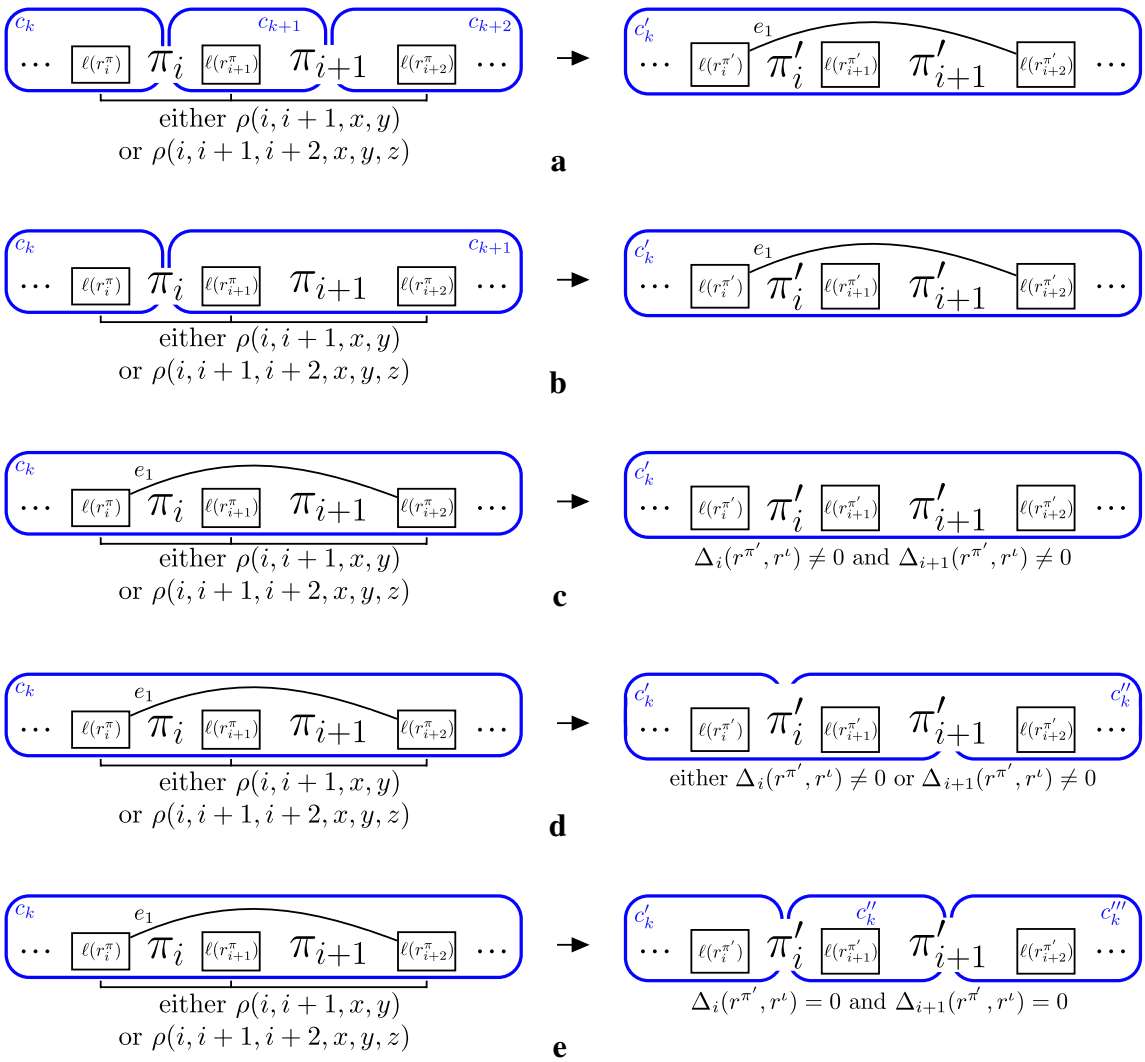

Fig. 4 Example of intergenic graphs for all possible values of $C(I(\pi', r^{\pi'}, r^\iota))$ with respect to $C(I(\pi, r^\pi, r^\iota))$ where $(\pi', r^{\pi'})$ is the resulting instance after applying a 2-reversal or a 2-transposition to (π, r^π). When the 2-reversal or 2-transposition is applied over two components at the same time $C(I(\pi', r^{\pi'}, r^\iota)) < C(I(\pi, r^\pi, r^\iota))$, as shown in **a** and **b**. Otherwise, we have that $C(I(\pi, r^\pi, r^\iota)) \leq C(I(\pi', r^{\pi'}, r^\iota)) \leq C(I(\pi, r^\pi, r^\iota)) + 2$, as shown in **c**–**e**

Proof Let p_i be the index in r^π of the i-th intergenic region inside component c. The last intergenic region of c is at position p_{c_r}. By definition of a component, and since c contains no edge, for any $p_1 \leq j < p_{c_r}$ we have that $\Delta_j(r^\pi, r^\iota) \neq 0$. Note that since $c_r > 1$ we have that $\Delta_{p_1}(r^\pi, r^\iota) = (\ell(r^\pi_{p_1}) - \ell(r^\iota_{p_1})) \neq 0$.

If $\Delta_{p_1}(r^\pi, r^\iota) > 0$, let $p_i = p_1$ and let k be the index of element from π located right after $r^\pi_{p_1}$. Apply the reversal $\rho(k, k, \ell(r^\iota_{p_1}), 0)$.

Otherwise, we have that $\Delta_{p_1}(r^\pi, r^\iota) < 0$, and we need to find two intergenic regions $r^\pi_{p_i}$ and $r^\pi_{p_{i+1}}$ for $1 \leq i < c_r$ such that $\Delta_{p_i}(r^\pi, r^\iota) < 0$ and $\Delta_{p_{i+1}}(r^\pi, r^\iota) \geq 0$. Since, by definition of a component, $\Delta_{p_{c_r}}(r^\pi, r^\iota) = 0$, such pair always exists. Let k be the index of element from π located right after r_{p_i}. Apply the reversal $\rho(k, k, \ell(r^\pi_{p_i}), -\Delta_{p_i}(r^\pi, r^\iota))$.

In both cases, the resulting permutation π' has $\Delta_{p_i}(r^{\pi'}, r^\iota) = 0$, $\Delta_{p_{i+1}}(r^{\pi'}, r^\iota) = \Delta_{p_{i+1}}(r^\pi, r^\iota) + \Delta_{p_i}(r^\pi, r^\iota)$, and for any $i+2 \leq j \leq c_r$ we have that $\Delta_{p_j}(r^{\pi'}, r^\iota) = \Delta_{p_j}(r^\pi, r^\iota)$; thus, as before, all intergenic regions from $r^{\pi'}_{p_{i+1}}$ to $r^{\pi'}_{p_{c_r}}$ must belong to the same component.

This 1-reversal splits c into two components: c' with all intergenic regions in positions p_1 to p_i, and c'' with all intergenic regions in positions p_{i+1} to p_{c_r}. □

Let $\Delta_{odd}(r^\pi, r^\iota) = \sum_{i=1, i \pmod 2=1}^{n+1}(\ell(r_i^\pi) - \ell(r_i^\iota))$ denote the cumulative sum of imbalances of intergenic regions from π and ι in odd positions only. Using Lemmas 1, 2 and 3, we show in the following two lemmas the minimum and maximum number of super short reversals needed to transform π into ι and r^π into r^ι.

Lemma 4 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, m the number of intergenic regions in r^π and r^ι, and let $\varphi^r = 0$ if $\Delta_{odd}(r^\pi, r^\iota) = 0$ and $\varphi^r = 1$ otherwise. It follows that $d(\pi) \geq \max\{\frac{m - C(I(\pi, r^\pi, r^\iota))}{2}, inv(\pi) + \varphi^r\}$.*

Proof In order to sort π, we need to remove all inversions, and since a 2-reversal can remove only one inversion, we have that $d(\pi) \geq inv(\pi)$. Besides, since 2-reversals exchange material between intergenic regions of same parity only, then $d(\pi) \geq inv(\pi) + \varphi^r$, with $\varphi^r = 1$ if $\Delta_{odd}(r^\pi, r^\iota) \neq 0$ (in this case we will need at least one 1-reversal to exchange material between an intergenic region located at an odd position and an intergenic region located at an even position), and $\varphi^r = 0$ otherwise.

On the other hand, by Lemmas 1 and 2, we can increase the number of components by at most two with a super short reversal, so to reach m trivial components we need at least $\frac{m - C(I(\pi, r^\pi, r^\iota))}{2}$ super short reversals. □

Lemma 5 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let m be the number of intergenic regions in r^π and r^ι. We have that $d(\pi) \leq inv(\pi) + m - C(I(\pi, r^\pi, r^\iota))$.*

Proof While $\pi \neq \iota$, π has at least one pair of consecutive elements (π_i, π_{i+1}) that is an inversion. Suppose that we first remove all inversions from π using $inv(\pi)$ 2-reversals of type $\rho(i, i+1, \ell(r_i^\pi), 0)$ i.e., without modifying its intergenic regions lengths. Let π' be the resulting permutation, that has $r^{\pi'} = r^\pi$. The number of components in $I(\pi', r^{\pi'}, r^\iota)$ cannot be smaller than $C(I(\pi, r^\pi, r^\iota))$, since any 2-reversal removing an inversion is applied inside a same component. By Lemma 3, we can go from $C(I(\pi', r^{\pi'}, r^\iota))$ to m components using $m - C(I(\pi', r^{\pi'}, r^\iota))$ 1-reversals, which results in no more than $m - C(I(\pi, r^\pi, r^\iota))$ 1-reversals. □

Finally, using Lemmas 4 and 5, we prove that it is possible to obtain 3-approximation for this problem.

Theorem 1 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. The value of $d(\pi)$ is 3-approximable.*

Proof Let $k = C(I(\pi, r^\pi, r^\iota))$, and let $\varphi^r = 0$, if $\Delta_{odd}(r^\pi, r^\iota) = 0$, or $\varphi^r = 1$ otherwise. If $\frac{m-k}{2} \geq inv(\pi) + \varphi^r$ then, by Lemma 4, $d(\pi) \geq \frac{m-k}{2}$, and, by Lemma 5, $d(\pi) \leq m - k + inv(\pi) \leq m - k + \frac{m-k}{2} \leq 3\frac{m-k}{2}$. Otherwise, $\frac{m-k}{2} < inv(\pi) + \varphi^r$, so $m - k < 2inv(\pi) + 2\varphi^r$. By Lemma 4, $d(\pi) \geq inv(\pi) + \varphi^r$, and, by Lemma 5, $d(\pi) \leq m - k + inv(\pi) \leq 2inv(\pi) + 2\varphi^r + inv(\pi) \leq 3inv(\pi) + 2\varphi^r$. □

Algorithm 1 describes a 3-approximation algorithm that transforms an instance (π, r^π) into (ι, r^ι) using super short reversals. Computing $inv(\pi)$ takes $O(n \log n)$ and it takes up to $O(n^2)$ to build $I(\pi, r^\pi, r^\iota)$. Computing $\Delta_i(r^\pi, r^\iota)$ and $C(I(\pi, r^\pi, r^\iota))$ take $O(n)$, and it takes constant time to update them. The while loop in line 5 (resp. line 14) iterates up to $O(n^2)$ times, so the overall complexity of Algorithm 1 is $O(n^2)$.

Algorithm 1: A 3-approximation algorithm for Sorting by Super Short Reversals. The inputs are two instances (π, r^π) and (ι, r^ι), and the output is a sequence of super short reversals that transforms (π, r^π) into (ι, r^ι).

```
   Data: (π, r^π, r^ι)
1  seq ← {}
2  ℓ ← inv(π)
3  Compute Δ_i(r^π, r^ι) and build I(π, r^π, r^ι)
4  k ← C(I(π, r^π, r^ι))
     ▷ Remove all inversions using ℓ 2-reversals (that
       never merge two components). As a heuristic, we
       always try to set Δ_i(r^π, r^ι) = 0, which may split
       this component.
5  while There is a pair (π_i, π_{i+1}) ∈ π that is an inversion
     do
6      if Δ_i(r^π, r^ι) ≥ 0 then
7          aux ← max{0, ℓ(r_i^π) − Δ_i(r^π, r^ι)}
8          op ← ρ(i, i + 1, aux, 0)
9      else
10         aux ← min{ℓ(r_{i+2}^π), −Δ_i(r^π, r^ι)}
11         op ← ρ(i, i + 1, ℓ(r_i^π), aux)
12     seq.append(op)
13     Apply op to (π, r^π)
     ▷ At this point inv(π) = 0, and we can split every
       non-trivial component into two trivial
       components using up to m − k 1-reversals
14 while r^π ≠ r^ι do
15     op ← a 1-reversal as explained in Lemma 3
16     seq.append(op)
17     Apply op to (π, r^π)
18 return seq
```

Let δ_n denote the set of all permutations π with n elements, and let $\delta_{n,k}$ denote the number of all permutations $\pi \in \delta_n$ such that $inv(\pi) \leq k$. For $n = 12$ there are 762,007 permutations in $\delta_{12,12}$, which corresponds to 0.16% of the 12! permutations from δ_{12}, and for $n > 12$ the number of permutations in $\delta_{n,n}$ never corresponds to more than 0.05% of the $n!$ permutations from δ_n [17]. Besides, for $n > 18$ the number of permutations in $\delta_{n,2n}$ never exceeds 0.03% of the $n!$ permutations from δ_n [17].

Algorithm 1 has a better approximation factor when the number of inversions is at least n, as explained in the following theorem.

Theorem 2 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. If $inv(\pi) \geq n$, Algorithm 1 has an approximation factor of $(1 + \frac{1}{\ell})$, where $\ell = \frac{inv(\pi)}{n} \geq 1$.*

Proof Let $k = C(I(\pi, r^\pi, r^\iota))$, and let $\varphi^r = 0$ if $\Delta_{odd}(r^\pi, r^\iota) = 0$ and $\varphi^r = 1$ otherwise. Suppose now that $inv(\pi) = n\ell$ for some $\ell \geq 1$. Since $\frac{m-k}{2} < n$, by Lemma 4 we have that $d(\pi) \geq n\ell$. Algorithm 1 applies $n\ell$ 2-reversals and up to $m - k < n$ 1-reversals, which results in no more than $n\ell + n - 1 < n(\ell + 1)$ super short reversals. □

Corollary 2.1 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. If $inv(\pi) \geq n$ (resp. $inv(\pi) \geq 2n$) Algorithm 1 has an approximation factor of at most 2 (resp. 1.5).*

Sorting by Super Short Transpositions

In this section, we analyze the version of the problem when only Super Short Transpositions are allowed. First, we investigate how 2-transpositions split non-trivial components from an intergenic graph $I(\pi, r^\pi, r^\iota)$.

Lemma 6 *If a component c of an intergenic graph $I(\pi, r^\pi, r^\iota)$ with $c_r > 2$ (resp. $c_r = 2$) has no edge, then we can apply two 2-transpositions that split c into three components c', c'', and c''' such that $c'_r + c''_r + c'''_r = c_r$ (resp. two components c' and c'' such that $c'_r = c''_r = 1$).*

Proof Note that any 2-transposition will increase or decrease the number of inversions by one. By Lemma 2, a 2-transposition that removes an inversion can increase the number of components by at most two units, and a 2-transposition creating an inversion cannot increase the number of components. Since there is no inversion in c, for each 2-transposition removing an inversion from c we have a 2-transposition creating that inversion before.

Now we explain how to increase the number of components by two units when $c_r \geq 3$. Let p_i be the index in r^π of the i-th intergenic region inside component c. If there is no intergenic region r_j inside c in which the cumulative sum is negative, apply $\tau(p_1, p_2, p_3, x, y, 0)$ in such a way that $x = \min\{\ell(r^\iota_{p_1}) + \ell(r^\iota_{p_2}), \ell(r^\pi_{p_1})\}$ and $y = \ell(r^\pi_{p_2}) + \ell(r^\iota_{p_1}) + \ell(r^\iota_{p_2}) - x$. Now apply $\tau(p_1, p_2, p_3, \ell(r^\iota_{p_1}), 0, 0)$. These two 2-transpositions split c into three components: c' with r_{p_1}, c'' with r_{p_2} and c''' with the remaining intergenic regions from c. Note that c' and c'' are odd, and c''' has the same parity as c.

Otherwise, we can find a pair of consecutive intergenic regions r_{p_i} and $r_{p_{i+1}}$ inside c such that $\Delta_{p_i}(r^\pi, r^\iota) < 0$ and $\Delta_{p_{i+1}}(r^\pi, r^\iota) \geq 0$, and since $\Delta_{p_{c_r}}(r^\pi, r^\iota) = 0$, such pair always exists. If c_r is even or if c_r is odd but p_i is even, apply $\tau(p_{i-1}, p_i, p_{i+1}, x, y, 0)$ such that $x = \ell(r^\pi_{p_{i-1}})$ and $y = \ell(r^\pi_{p_i}) + \Delta_{pi-1}(r^\pi, r^\iota)$, followed by $\tau(p_{i-1}, p_i, p_{i+1}, x', 0, y')$ such that $x' = \ell(r^\iota_{p_{i-1}})$ and $y' = \ell(r^\iota_{p_i})$.

If c_r and p_i are odd, apply $\tau(p_i, p_{i+1}, p_{i+2}, x, y, 0)$ such that $x = \ell(r^\pi_{p_i})$ and $y = \ell(r^\pi_{p_{i+1}}) + \Delta_{pi}(r^\pi, r^\iota)$, followed by $\tau(p_{i-1}, p_i, p_{i+1}, x', 0, y')$ such that $x' = \ell(r^\iota_{p_i})$ and $y' = \ell(r^\iota_{p_{i+1}})$. These two 2-transpositions split c into three components so if c_r is even then we will end up with two odd components and one even component, and if c is odd we will end up with three odd components due to the choice of the position defined above.

If $c_r = 2$ and $p_1 > 1$ we apply $\tau(p_1 - 1, p_1, p_2, x, y, 0)$ such that $x = \ell(r^\pi_{p_1 - 1})$ and $y = \ell(r^\pi_{p_1})$, followed by $\tau(p_1 - 1, p_1, p_2, x', 0, y')$ such that $x' = x$ and $y' = \ell(r^\iota_{p_1})$. If $c_r = 2$ and $p_1 = 1$ we apply $\tau(p_1, p_2, p_2 + 1, \ell(r^\pi_{p_1}), 0, 0)$ followed by $\tau(p_1, p_2, p_2 + 1, \ell(r^\iota_{p_1}), 0, 0)$. These two transpositions transform c into two trivial components. □

The following lemma gives the number of transpositions needed to transform a permutation π and its intergenic regions r^π into ι with its intergenic regions r^ι when $inv(\pi) = 0$.

Lemma 7 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. If $inv(\pi) = 0$, then $d(\pi) = m - C(I(\pi, r^\pi, r^\iota)) + C_{even}(I(\pi, r^\pi, r^\iota))$.*

Proof If a 2-transposition applied on a component c of $I(\pi, r^\pi, r^\iota)$ increases the number of components by two units, we can assume by the proof of Lemma 6 that it transforms c into three components c', c'', and c''' such that two of them are odd components and the other has the same parity as c_r.

If c is odd, and if we can always increase the number of components by two units, we end up with a component with only one intergenic region, but if c is even, at some point we will have to increase the number of components by one unit, creating two odd components. This means that for each even component we need to apply two 2-transpositions that increase the number of components by one unit only. Since we can always apply pairs of transpositions that do not increase the number of even components, it follows that $d(\pi) = m - C(I(\pi, r^\pi, r^\iota)) + C_{even}(I(\pi, r^\pi, r^\iota))$. □

Lemmas 8 and 9 respectively show the lower and upper bounds for finding $d(\pi)$ using super short transpositions.

Lemma 8 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. It follows that $d(\pi) \geq \max\{\frac{m - C(I(\pi, r^\pi, r^\iota)) + C_{even}(I(\pi, r^\pi, r^\iota))}{2}, inv(\pi)\}$.*

Proof In order to sort π we need to remove all inversions, and since a 2-transposition can remove only one inversion, we necessarily have that $d(\pi) \geq inv(\pi)$. Besides, by Lemma 2, we can increase the number of components by at most two with a super short transposition. Let $k = C(I(\pi, r^\pi, r^\iota)) - C_{even}(I(\pi, r^\pi, r^\iota))$. To reach m trivial components, and considering also Lemma 7, we need at least $\frac{m+k}{2}$ super short transpositions. Thus, $d(\pi) \geq \max\{\frac{m+k}{2}, inv(\pi)\}$. □

Lemma 9 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. We have that $d(\pi) \leq inv(\pi) + m - C(I(\pi, r^\pi, r^\iota)) + C_{even}(I(\pi, r^\pi, r^\iota))$.*

Proof Suppose that we first remove all inversions of π using $inv(\pi)$ 2-transpositions of type $\tau(i, i+1, i+2, \ell(r_i^\pi), 0, 0)$, and let π' be the resulting permutation. The value of $C(I(\pi', r^{\pi'}, r^\iota))$ cannot be smaller than $C(I(\pi, r^\pi, r^\iota))$ since any 2-transposition removing an inversion is applied inside a same component. Let $k = C(I(\pi, r^\pi, r^\iota)) - C_{even}(I(\pi, r^\pi, r^\iota))$ and let $k' = C(I(\pi', r^{\pi'}, r^\iota)) - C_{even}(I(\pi', r^{\pi'}, r^\iota))$

Let us analyze the parity of any component that a 2-transposition breaks: (i) if it transforms an odd component into two, then one component must be odd; (ii) if it transforms an even component into two, then both components are odd or even; (iii) if it transforms an even component into three, then two components must be odd; (iv) if it transforms an odd component into three, then either two components are even or the three components are odd. This means that $k' \geq k$.

By Lemma 7, we can go from $C(I(\pi', r^{\pi'}, r^\iota))$ to m components using $m - k'$ 2-transpositions, which results, by the analysis above, in no more than $m - k$ 2-transpositions. □

Finally, using Lemmas 8 and 9, we prove that it is possible to obtain a 3-approximable solution for this problem.

Theorem 3 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. The value of $d(\pi)$ is 3-approximable.*

Proof Let $k = C(I(\pi, r^\pi, r^\iota)) - C_{even}(I(\pi, r^\pi, r^\iota))$. If $\frac{m-k}{2} \geq inv(\pi)$ then, by Lemma 8, $d(\pi) \geq \frac{m-k}{2}$, and, by Lemma 5, $d(\pi) \leq m - k + inv(\pi) \leq m - k + \frac{m-k}{2} \leq 3\frac{m-k}{2}$. Otherwise, $\frac{m-k}{2} < inv(\pi)$, so $m - k < 2\, inv(\pi)$. By Lemma 8, $d(\pi) \geq inv(\pi)$, and, by Lemma 9, $d(\pi) \leq m - k + inv(\pi) \leq 2\, inv(\pi) + inv(\pi) \leq 3\, inv(\pi)$. □

Algorithm 2 describes a 3-approximation algorithm that transforms an instance (π, r^π) into (ι, r^ι) using Super Short Transpositions. Similarly to Algorithm 1, Algorithm 2 has a time complexity of $O(n^2)$.

Algorithm 2: A 3-approximation algorithm for Sorting by Super Short Transpositions. The inputs are two instances (π, r^π) and (ι, r^ι), and the output is a sequence of super short transpositions that transforms (π, r^π) into (ι, r^ι).

```
Data: (π, r^π, r^ι)
seq ← {}
ℓ ← inv(π)
Compute Δ_i(r^π, r^ι) and build I(π, r^π, r^ι)
k ← C(I(π, r^π, r^ι)) − C_even(I(π, r^π, r^ι))
  ▷ Remove all inversions using ℓ 2-transpositions
    (that never merge two components). As a
    heuristic, we always try to set Δ_i(r^π, r^ι) = 0,
    which may split this component.
while There is a pair (π_i, π_{i+1}) ∈ π that is an inversion
  do
    if Δ_i(r^π, r^ι) ≥ 0 then
        aux ← max{0, ℓ(r_i^π) − Δ_i(r^π, r^ι)}
        op ← τ(i, i+1, i+2, aux, ℓ(r_{i+1}^π), 0)
    else
        aux ← min{ℓ(r_{i+1}^π), −Δ_i(r^π, r^ι)}
        op ← τ(i, i+1, i+1, ℓ(r_i^π), aux, 0)
    seq.append(op)
    Apply op to (π, r^π)
  ▷ At this point inv(π) = 0, and we can split every
    non-trivial component into two trivial
    components using up to m − k 2-transpositions
while r^π ≠ r^ι do
    ops ← a pair of 2-transpositions as explained in
      Lemma 7
    seq.append(ops)
    Apply transpositions from ops to (π, r^π)
return seq
```

Algorithm 2 has a better approximation factor when the number of inversions is strictly greater than n, as stated in the following theorem.

Theorem 4 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. If $inv(\pi) > n$, Algorithm 2 has an approximation factor of $(1 + \frac{1}{\ell})$, where $\ell = \frac{inv(\pi)}{n} \geq 1$.*

Proof Similar to proof of Theorem 2, given that $m - C(I(\pi, r^\pi, r^\iota)) + C_{even}(I(\pi, r^\pi, r^\iota)) \leq n + 1$. □

Corollary 4.1 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. If $inv(\pi) \geq n$ (resp. $inv(\pi) \geq 2n$) Algorithm 2 has an approximation factor of at most 2 (resp. 1.5).*

Sorting by Super Short Reversals and Super Short Transpositions

In this section we analyze the version of the problem when both super short reversals and Super Short Transpositions are allowed to transform any (π, r^π) into (ι, r^ι).

Lemma 10 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. It follows that $d(\pi) \geq \max\{\frac{m - C(I(\pi, r^\pi, r^\iota))}{2}, inv(\pi)\}$.*

Proof Directly from Lemmas 2, 4, and 8. □

Lemma 11 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. We have that $d(\pi) \leq inv(\pi) + m - C(I(\pi, r^\pi, r^\iota))$.*

Proof Suppose that first we remove all inversions of π using $inv(\pi)$ 2-reversals of type $\rho(i, i+1, \ell(r_i^\pi), 0)$, and let π' (resp. $r^{\pi'}$)) be the resulting permutation (resp. intergenic regions). Let $k = C(I(\pi, r^\pi, r^\iota))$ and let $k' = C(I(\pi', r^{\pi'}, r^\iota))$. We have that $k' \geq k$, since 2-reversals removing inversions are always applied inside a same component.

Analogous to Lemma 9, and assuming that $k' = k + \ell$ for some $\ell \geq 0$, then $C_{even}(I(\pi', r^{\pi'}, r^\iota)) \leq C_{even}(I(\pi, r^\pi, r^\iota)) + \ell$. We use the procedure described in Lemma 9 on components c with $c_r \geq 3$, applying two 2-transpositions that increase the number of components by two units. For components c with $c_r = 2$, we apply a 1-reversal as described in Lemma 1, breaking them into two odd components.

The above procedure applies $inv(\pi)$ 2-reversals, $n - k' - C_{even}(I(\pi', r^{\pi'}, r^\iota))$ 2-transpositions, and $C_{even}(I(\pi', r^{\pi'}, r^\iota))$ 1-reversals, which results in no more than $inv(\pi) + m - C(I(\pi), r^\iota))$. □

Now we prove that it is possible to obtain a 3-approximable solution for this problem.

Theorem 5 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. The value of $d(\pi)$ is 3-approximable.*

Proof Similar to proof of Theorem 1, using Lemmas 10 and 11. □

Algorithm 3 describes a 3-approximation algorithm that transforms an instance (π, r^π, r^ι) into $(\iota, r^\iota, r^\iota)$ using both super short reversals and super short transpositions. As algorithms 1 and 2, it has a time complexity of $O(n^2)$.

Algorithm 3: A 3-approximation algorithm for Sorting by Super Short Reversals and Super Short Transpositions. The inputs are two instances (π, r^π) and (ι, r^ι), and the output is a sequence of super short reversals and super short transpositions that transforms (π, r^π) into (ι, r^ι).

```
Data: (π, r^π, r^ι)
seq ← {}
ℓ ← inv(π)
Compute Δ_i(r^π, r^ι) and build I(π, r^π, r^ι)
  ▷ Remove all inversions using ℓ 2-reversals (that
    never merge two components). As a heuristic, we
    always try to set Δ_i(r^π, r^ι) = 0, which may split
    this component.
while There is a pair (π_i, π_{i+1}) ∈ π that is an inversion
  do
    if Δ_i(r^π, r^ι) ≥ 0 then
        aux ← max{0, ℓ(r_i^π) − Δ_i(r^π, r^ι)}
        op ← ρ(i, i + 1, aux, 0)
    else
        aux ← min{ℓ(r_{i+2}^π), −Δ_i(r^π, r^ι)}
        op ← ρ(i, i + 1, ℓ(r_i^π), aux)
    seq.append(op)
    Apply op to (π, r^π)
  ▷ At this point inv(π) = 0
k ← C(I(π, r^π, r^ι))
k' ← C_even(I(π, r^π, r^ι))
  ▷ Split every non-trivial component c with c_r ≥ 3
    into three using m − k − k' 2-transpositions
    (grouped in pairs)
while r^π ≠ r^ι do
    op ← a pair of 2-transpositions as explained in Lemma 7
    seq.append(op)
    Apply transpositions from op to (π, r^π)
  ▷ Now we can split every non-trivial component
    into two trivial components using k' 1-reversals
while r^π ≠ r^ι do
    op ← a 1-reversal as explained in Lemma 3
    seq.append(op)
    Apply op to (π, r^π)
return seq
```

As in the previous algorithms, Algorithm 3 has a better approximation factor when the number of inversions is at least n, as explained in the following theorem.

Theorem 6 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. If $inv(\pi) \geq n$ Algorithm 3 has an approximation factor of $(1 + \frac{1}{\ell})$, where $\ell = \frac{inv(\pi)}{n} \geq 1$.*

Proof Analogous to proof of Theorem 2. □

Corollary 6.1 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. If $inv(\pi) \geq n$ (resp. $inv(\pi) \geq 2n$) Algorithm 3 has an approximation factor of at most 2 (resp. 1.5).*

Sorting by Signed Super Short Reversals

In this section, we analyze the version of the problem when super short reversals are allowed to transform (π, r^π) into (ι, r^ι), where π and ι are signed permutations.

Given a signed permutation π, let $S_\pi^{even^-}$ be the set of elements from π such that $||\pi_i| - i|$ is even and $\pi_i < 0$, and let $S_\pi^{odd^+}$ be the set of elements from π such that $||\pi_i| - i|$ is odd and $\pi_i > 0$. Sets $S_\pi^{even^-}$ and $S_\pi^{odd^+}$ capture the negative and positive elements from π that end with negative signs after any sequence of 2-reversals that puts all elements in their correct positions (i.e., remove all inversions). Let φ^{neg} be the number of elements in $S_\pi^{even^-} \cup S_\pi^{odd^+}$.

The following lemma, proved by Galvão et al. [8], gives the exact number of super short reversals needed to transform π into ι.

Lemma 12 *Given a signed permutation* π, $d(\pi) = inv(\pi) + \varphi^{neg}$.

This lemma helps us to state the following lower bound for our problem.

Lemma 13 *Let* (π, r^π) *be an instance,* (ι, r^ι) *be the target instance, and let* $m = n + 1$ *be the number of intergenic regions in* r^π *and* r^ι. *We have that* $d(\pi) \geq inv(\pi) + \max\{\varphi^r, \varphi^{neg}\}$.

Proof Directly from Lemmas 4 and 12. □

The following lemma states an upper bound for this problem.

Lemma 14 *Let* (π, r^π) *be an instance,* (ι, r^ι) *be the target instance, and let* $m = n + 1$ *be the number of intergenic regions in* r^π *and* r^ι. *We have that* $d(\pi) \leq inv(\pi) + \max\{\varphi^r, \varphi^{neg}\} + 2(m - C(I(\pi, r^\pi, r^\iota))$.

Proof Let $k = C(I(\pi, r^\pi, r^\iota))$ and let $\ell = \max\{\varphi^r, \varphi^{neg}\}$. Suppose that we first remove all inversions of π using $inv(\pi)$ 2-reversals of type $\rho(i, i+1, \ell(r_i^\pi), 0)$, and let π' (resp. $r^{\pi'}$) be the resulting permutation (resp. intergenic regions).

Let $k' = C(I(\pi', r^{\pi'}, r^\iota))$. We have that $k' \geq k$. We apply $m - k' \leq m - k$ 1-reversals that split every non-trivial component from $I(\pi', r^{\pi'}, r^\iota)$ into two components according to Lemma 3, and let π'' (resp. $r^{\pi''}$) be the resulting permutation (resp. intergenic regions).

At this point, we have a permutation π'' such that $r^{\pi''} = r^\iota$, and π'' has no more than $\ell + (m - k') \leq \ell + (m - k)$ negative elements. We just need to apply up to $\ell + (m - k')$ 1-reversals of type $\rho(i, i, \ell(r_i^{\pi''}), 0)$ (i.e., without modifying the length of its intergenic regions) to each negative element from π', and the lemma follows. □

Using Lemmas 13 and 14, we prove that the value of $d(\pi)$ is 5-approximable.

Theorem 7 *Let* (π, r^π) *be an instance,* (ι, r^ι) *be the target instance, and let* $m = n + 1$ *be the number of intergenic regions in* r^π *and* r^ι. *The value of* $d(\pi)$ *is 5-approximable.*

Proof Let $k = C(I(\pi, r^\pi, r^\iota))$, and let $\ell = \max\{\varphi^r, \varphi^{neg}\}$. If $\frac{m-k}{2} \geq inv(\pi) + \ell$ then, by Lemma 12, $d(\pi) \geq \frac{m-k}{2}$, and, by Lemma 14, $d(\pi) \leq 2(m-k) + inv(\pi) + \ell \leq 2(m-k) + \frac{m-k}{2} \leq 5\frac{m-k}{2}$.

Otherwise, $\frac{m-k}{2} < inv(\pi) + \ell$, so $2(m-k) < 4(inv(\pi) + \ell)$. By Lemma 12, $d(\pi) \geq inv(\pi) + \ell$, and, by Lemma 14, $d(\pi) \leq 2(m-k) + inv(\pi) + \ell \leq 4(inv(\pi) + \ell) + inv(\pi) + \ell \leq 5(inv(\pi) + \ell)$. □

Algorithm 4 describes a 5-approximation algorithm that transforms a signed instance (π, r^π, r^ι) into $(\iota, r^\iota, r^\iota)$ using Signed Super Short Reversals. As in previous algorithms, the time complexity of Algorithm 4 is $O(n^2)$.

Algorithm 4: A 5-approximation algorithm for Sorting by Signed Super Short Reversals. The inputs are two instances (π, r^π) and (ι, r^ι), and the output is a sequence of signed super short reversals that transforms (π, r^π) into (ι, r^ι).

```
Data: (π, r^π, r^ι)
seq ← {}
ℓ ← inv(π)
Compute Δ_i(r^π, r^ι) and build I(π, r^π, r^ι)
  ▷ Remove all inversions using ℓ 2-reversals. As a
    heuristic, we always try to set Δ_i(r^π, r^ι) = 0,
    which may split this component.
while There is a pair (π_i, π_{i+1}) ∈ π that is an inversion
  do
    if Δ_i(r^π, r^ι) ≥ 0 then
        aux ← max{0, ℓ(r_i^π) − Δ_i(r^π, r^ι)}
        op ← ρ(i, i + 1, aux, 0)
    else
        aux ← min{ℓ(r_{i+2}^π), −Δ_i(r^π, r^ι)}
        op ← ρ(i, i + 1, ℓ(r_i^π), aux)
    seq.append(op)
    Apply op to (π, r^π)
  ▷ At this point inv(π) = 0
k ← C(I(π, r^π, r^ι))
  ▷ Split every non-trivial component into two using
    m − k 1-reversals
while r^π ≠ r^ι do
    op ← a 1-reversal as explained in Lemma 3
    seq.append(op)
    Apply op to (π, r^π)
  ▷ Now we apply up to φ^neg + m − k 1-reversals to
    every negative element of π
for ( π_i ∈ π ):
    if π_i < 0 then
        op ← ρ(i, i, ℓ(r_i^π), 0)
        seq.append(op)
        Apply op to (π, r^π)
return seq
```

As in the previous algorithms, Algorithm 4 has a better approximation factor when the number of inversions is at least n, as explained in the following theorem.

Theorem 8 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n+1$ be the number of intergenic regions in r^π and r^ι. If $inv(\pi) \geq n$, Algorithm 3 has an approximation factor of $(1+\frac{2}{\ell})$, where $\ell = \frac{inv(\pi)}{n} \geq 1$.*

Proof Let $k = C(I(\pi, r^\pi, r^\iota))$. Suppose now that $inv(\pi) = n\ell$ for some $\ell \geq 1$. Since $\frac{m-k}{2} < n$, by Lemma 4 we have that $d(\pi) \geq n\ell$. Algorithm 4 applies $n\ell$ 2-reversals, up to $m-k < n$ 1-reversals, and up to n 1-reversals to flip the sign of each negative element, which results in no more than $n\ell + n - 1 + n < n(\ell+2)$ super short reversals. □

Corollary 8.1 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n+1$ be the number of intergenic regions in r^π and r^ι. If $inv(\pi) \geq n$ (resp. $inv(\pi) \geq 2n$) Algorithm 4 has an approximation factor of at most 3 (resp. 2).*

Sorting by Signed Super Short Reversals and Super Short Transpositions

In this section, we analyze the version of the problem when both super short reversals and Super Short Transpositions are allowed to sort signed permutations.

Let $H(\pi)$ be the *inversion graph* [18] of the signed permutation π, such that $V(H(\pi)) = \{\pi_1, \pi_2, ..., \pi_n\}$ and $E(H(\pi))$ is formed by pairs of elements from π that are inversions. In $H(\pi)$, a component is defined as a maximal subgraph in which any two vertices are connected to each other by paths. A component from $H(\pi)$ is *negative* if it contains an odd number of negative elements (vertices), and it is *positive* otherwise.

Let φ^{odd} be the number of negative components of $H(\pi)$. The following lemma, proved by Galvão et al. [8], gives the exact number of super short reversals and Super Short Transpositions needed to transform π into ι, which is a lower bound for our problem.

Lemma 15 *Given a signed permutation π, $inv(\pi) + \varphi^{odd}$ super short operations are required to transform π into ι.*

Now we state in the following lemma an upper bound for this problem.

Lemma 16 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n+1$ be the number of intergenic regions in r^π and r^ι. We have that $d(\pi) \leq inv(\pi) + \varphi^{odd} + 2(m - C(I(\pi, r^\pi, r^\iota)))$.*

Proof Suppose that we first remove all inversions of π using the polynomial algorithm presented in [8], that uses $inv(\pi) + \varphi^{odd}$ super short operations such that all 2-reversals are of type $\rho(i, i+1, \ell(r_i^\pi), 0)$, all 2-transpositions are of type $\tau(i, i+1, i+2, \ell(r_i^\pi), 0, 0)$, and all the φ^{odd} 1-reversals are ignored (i.e., not applied), and let π' be the resulting permutation.

The number of components $I(\pi', r^{\pi'}, r^\iota)$ in π' cannot be smaller than $C(I(\pi, r^\pi, r^\iota))$, since the 2-reversals and 2-transpositions are applied inside a same component only. Let $k' = C(I(\pi', r^{\pi'}, r^\iota)) \geq C(I(\pi, r^\pi, r^\iota))$.
By Lemma 3, we can go from k' to m components using $m - k'$ 1-reversals, which results in no more than $m - C(I(\pi, r^\pi, r^\iota))$ 1-reversals. After that, we will have a permutation π'' with up to $\min\{n, m - k' + \varphi^{odd}\}$ negative elements, so we can apply up to $\min\{n, m - k' + \varphi^{odd}\}$ 1-reversals of type $\rho(i, i, \ell(r_i^\iota), 0)$ to each negative element of π''. □

Using Lemmas 15 and 16, we prove that it is possible to obtain a 5-approximable solution for this problem.

Theorem 9 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n+1$ be the number of intergenic regions in r^π and r^ι. The value of $d(\pi)$ is 5-approximable.*

Proof Let $k = C(I(\pi, r^\pi, r^\iota))$, and let $\ell = \varphi^{odd}$. If $\frac{m-k}{2} \geq inv(\pi) + \ell$ then, by Lemma 15, $d(\pi) \geq \frac{m-k}{2}$, and, by Lemma 16, $d(\pi) \leq 2(m-k) + inv(\pi) + \ell \leq 2(m-k) + \frac{m-k}{2} \leq 5\frac{m-k}{2}$.

Otherwise, $\frac{m-k}{2} < inv(\pi) + \ell$, so $2(m-k) < 4(inv(\pi) + \ell)$. By Lemma 15, $d(\pi) \geq inv(\pi) + \ell$, and, by Lemma 16, $d(\pi) \leq 2(m-k) + inv(\pi) + \ell \leq 4(inv(\pi) + \ell) + inv(\pi) + \ell \leq 5(inv(\pi) + \ell)$. □

Algorithm 5 describes a 5-approximation algorithm that transforms a signed instance (π, r^π, r^ι) into $(\iota, r^\iota, r^\iota)$ using both signed super short reversals and Super Short Transpositions. Regarding the complexity, by previous algorithms we know that lines 1-3 and 6-17 take up to $O(n^2)$, and according to [8] the while loop in line 4 takes $O(n^3)$, which is then the time complexity of Algorithm 5.

Algorithm 5: A 5-approximation algorithm for Sorting by Signed Super Short Reversals and Super Short Transpositions. The inputs are two instances (π, r^π) and (ι, r^ι), and the output is a sequence of signed super short operations that transforms (π, r^π) into (ι, r^ι).

Data: (π, r^π, r^ι)
$seq \leftarrow \{\}$
$\ell \leftarrow inv(\pi)$
Compute $\Delta_i(r^\pi, r^\iota)$ and build $I(\pi, r^\pi, r^\iota)$
▷ Remove all inversions using k super short operations, according to the algorithm presented in [8], but do not apply 1-reversals. As a heuristic, we always try to set $\Delta_i(r^\pi, r^\iota) = 0$, which may split this component.
while π_i, π_{i+1} **do**
  $op \leftarrow$ a 2-reversal or a 2-transposition according to [8]
  $seq.append(op)$
  Apply op to (π, r^π)
▷ Let π' be the resulting permutation. At this point $inv(\pi') = 0$, and there are φ^{odd} negative elements in π'
$k \leftarrow C(I(\pi', r^{\pi'}, r^\iota))$
▷ Split every non-trivial component into two using $m - k < n$ 1-reversals
while $r^{\pi'} \neq r^\iota$ **do**
  $op \leftarrow$ a 1-reversal as explained in Lemma 3
  $seq.append(op)$
  Apply op to $(\pi', r^{\pi'})$
▷ Now we apply up to n 1-reversals to every negative element of π'
for ($\pi'_i \in \pi'$):
  if $\pi'_i < 0$ **then**
    $op \leftarrow \rho(i, i, \ell(r^{\pi'}_i), 0)$
    $seq.append(op)$
    Apply op to $(\pi', r^{\pi'})$
return seq

As for previous algorithms, Algorithm 5 also has a better approximation factor when the number of inversions is at least n. This is the purpose of the following theorem.

Theorem 10 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. If $inv(\pi) \geq n$, Algorithm 5 has an approximation factor of $(1 + \frac{2}{\ell})$, where $\ell = \frac{inv(\pi)}{n} \geq 1$.*

Proof Let $k = C(I(\pi, r^\pi, r^\iota))$. Suppose now that $inv(\pi) = n\ell$. Since $\frac{m-k}{2} < n$, by Lemma 4 we have that $d(\pi) \geq \ell$. Algorithm 4 applies $n\ell$ operations between 2-reversals and 2-transpositions, up to $m - k < n$ 1-reversals, and up to n 1-reversals to flip the sign of each negative element, which results in no more than $n\ell + n - 1 + n < n(\ell + 2)$ super short operations. □

Corollary 10.1 *Let (π, r^π) be an instance, (ι, r^ι) be the target instance, and let $m = n + 1$ be the number of intergenic regions in r^π and r^ι. If $inv(\pi) \geq n$ (resp. $inv(\pi) \geq 2n$) Algorithm 5 has an approximation factor of at most 3 (resp. 2).*

Experimental tests

We implemented the five proposed algorithms and tested them unsing simulated permutations, in order to observe their performances. We generated two different permutation datasets, which we call fully-random instances (FRI) and almost random instances (ARI). Each dataset has 1,000,000 instances (π, r^π), π is a permutation with 100 elements and r^π is a sequence of 101 intergenic regions sizes.

The dataset FRI was generated in the following way: (i) let (ι, r^ι) be an initial instance, being ι with 100 elements, and each r^ι_i received a random integer $k \in [0..100]$. (ii) Generate (π, r^π) by applying w consecutive super short operations to (ι, r^ι), with randomly generated indices for both positions and intergenic sizes, always respecting the current values. We created 10,000 instances for each value of $w \in \{10, 20, 30, \ldots, 990, 1000\}$.

For Sorting by (Signed) Super Short Reversals we applied $0.8w$ 2-reversals and $0.2w$ 1-reversals, and at each step one of them was chosen at random while both were available. For Sorting by Super Short Transpositions we applied w 2-transpositions. For Sorting by (Signed) Super Short Operations we applied $0.5w$ 2-transpositions, $0.4w$ 2-reversals, and $0.1w$ 1-reversals, and at each step one of them was chosen at random while more than one were available.

The dataset ARI was generated in a similar way as FRI, but when the algorithm had to apply either a 2-reversal or a 2-transposition we randomly chose a pair among all pairs of adjacent elements that were not an inversion. Since $w < \max\{inv(\pi), \pi \in \delta_n\} = \frac{n(n-1)}{2}$, at least one pair always exists.

Given any instance (π, r^π) from ARI created using w SSOs, we know exactly how many inversions (π, r^π) has—it is the number of 2-reversals and 2-transpositions applied. The number of inversions on instances from FRI, however, is not known, but we can compute the expected number of inversions in a permutation with n elements after k random swaps (i.e., 2-reversals and 2-transpositions) applied to the identity permutation [19]:

$$E[i_{n,k}] = \frac{n(n+1)}{4} - \frac{1}{8(n+1)^2} \sum_{i,j=0}^{n} \frac{(c_j + c_i)^2}{s_k^2 s_i^2} x_{ji}^k,$$

where $c_a = \cos\alpha_a$, $s_a = \sin\alpha_a$, $x_{ab} = 1 - \frac{4}{n}(1 - c_a c_b)$, and $\alpha_a = \frac{(2a+1)\pi}{2n+2}$.

Figures 5 and 6 show the experimental results for instances of type FRI and ARI, respectively. We show the average distance returned for each algorithm described in this paper, plus the average and maximum approximation factors calculated based on the lower bound of each problem for each instance.

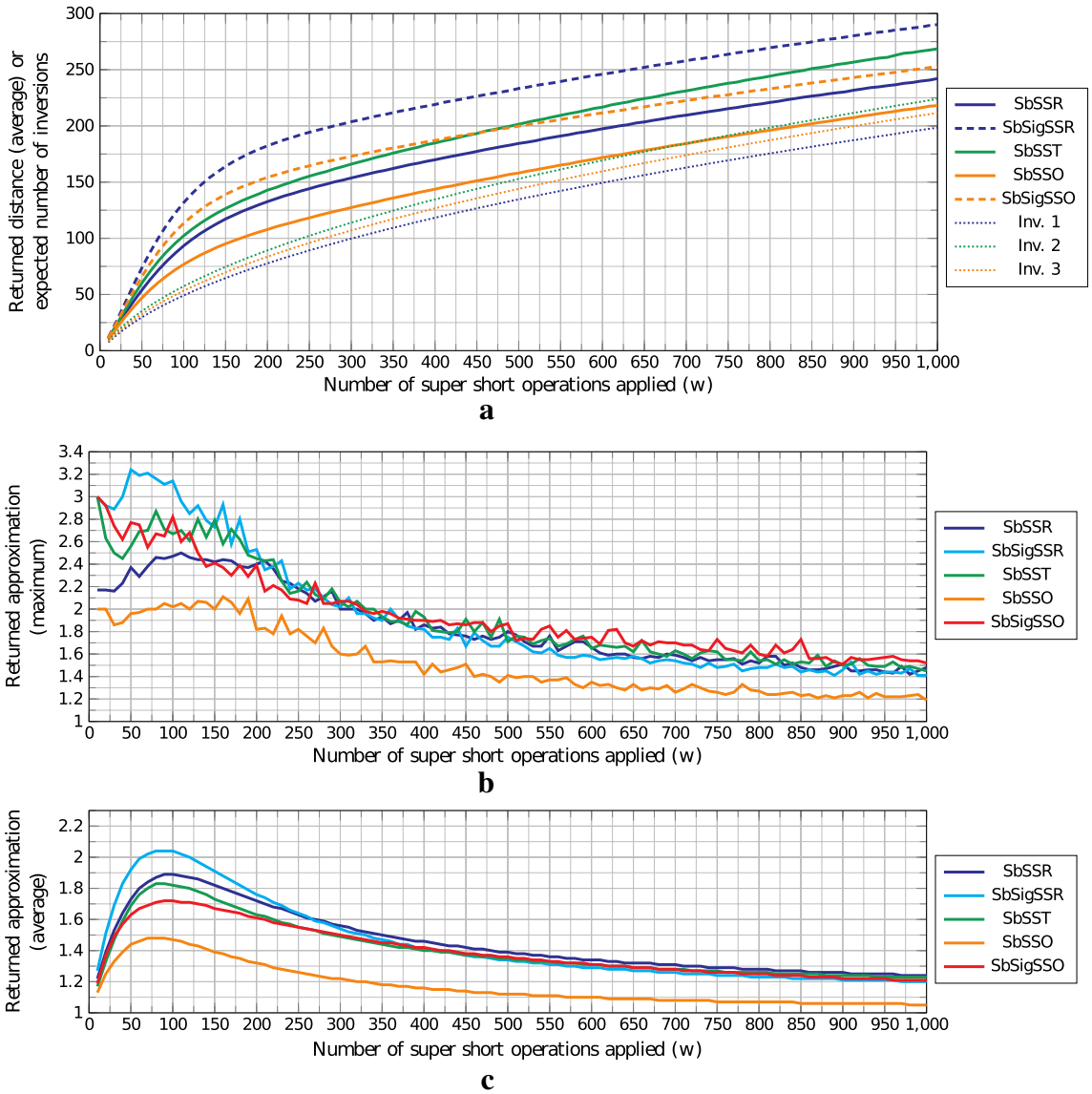

Fig. 5 **a** average returned distance, **b** maximum returned approximation, and **c** average returned approximation for instances in FRI. In **a** dotted curves represent the expected number of inversions, dashed curves represent the algorithms for signed permutations, and line curves represent the algorithms for unsigned permutations. Colors relate problems having the expected number of inversions given by the dotted line of same color. This means that SbSSR and SbSigSSR are expected to have inversions as in Inv. 1, SbSST is expected to have inversions as in Inv. 2, and SbSSO and SbSigSSO are expected to have inversions as in Inv. 3. All the approximations in **b** and **c** were calculated based on the lower bound of each problem

On Figs. 5 and 6, Algorithm 1 is denoted by SbSSR, Algorithm 4 is denoted by SbSigSSR, Algorithm 2 is denoted by SbSST, Algorithm 3 is denoted by SbSSO, and Algorithm 5 is denoted by SbSigSSO. In Fig. 5, the curve Inv. 1 represents the expected number of inversions for SbSSR and SbSigSSR, the curve Inv. 2 represents the expected number of inversions for SbSST, and the curve Inv. 3 denotes the expected number of inversions for SbSSO and SbSigSSO. These three curves were generated using the formula $E[i_{n,k}]$ described above. In Fig. 6 the curves Inv. 1, 2, and 3 follow the same idea as the curves in Fig. 5, but instead of expected number of inversions they represent the exact number of inversions.

As distance is directly related to the number of inversions, in Fig. 5a we see that, although in practice we have applied up to 1000 operations, the distance values were never greater than 300 on average—the average returned distances for each algorithm follow the trend dotted line that represents the expected number of inversions of that instances. Algorithms for signed permutations returned

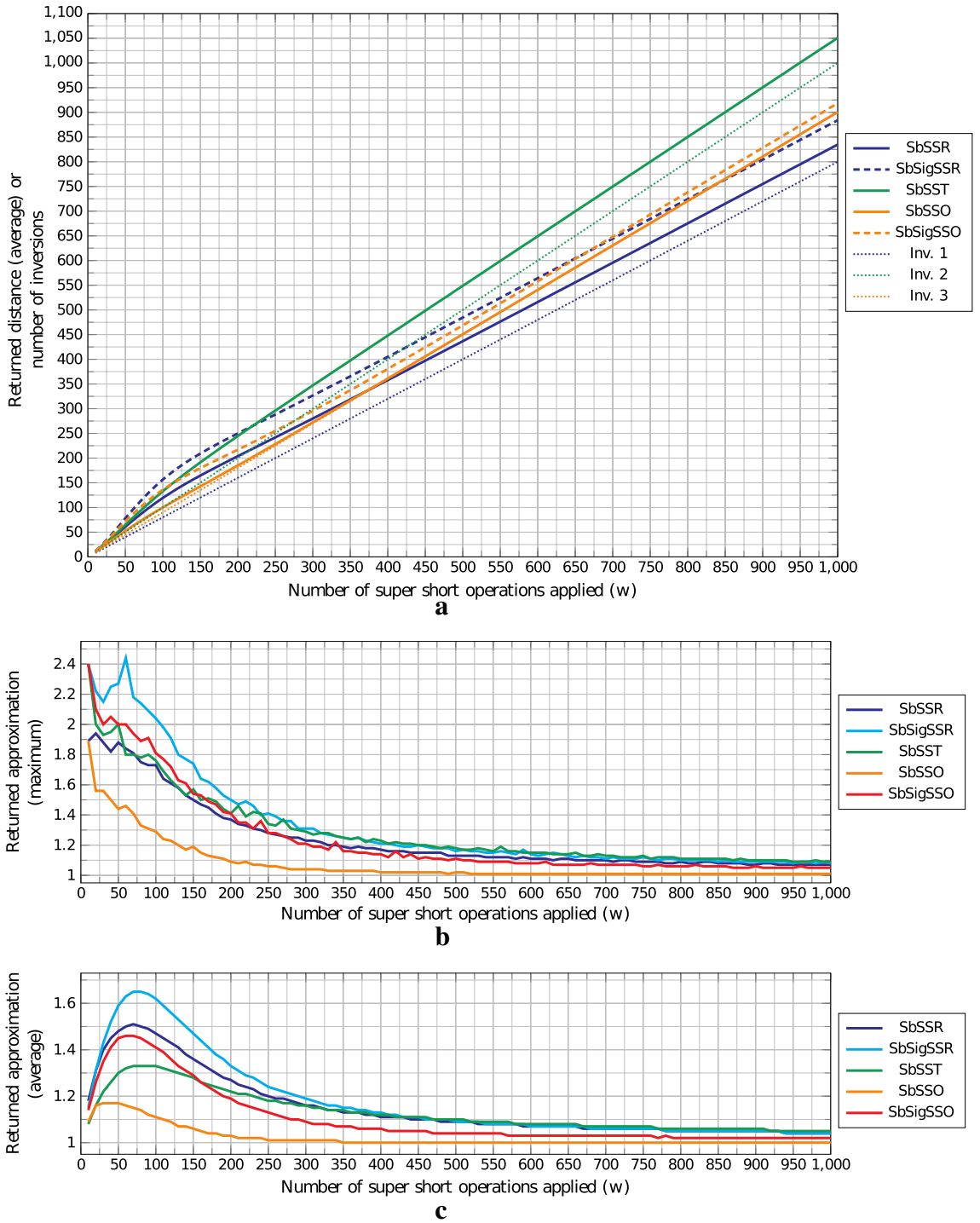

Fig. 6 **a** average returned distance, **b** maximum returned approximation, and **c** average returned approximation for instances in ARI. In **a** dotted curves represent the exact number of inversions, dashed curves represent the algorithms for signed permutations, and line curves represent the algorithms for unsigned permutations. Colors relate problems having the expected number of inversions given by the dotted line of same color. This means that SbSSR and SbSigSSR have inversions as in Inv. 1, SbSST has inversions as in Inv. 2, and SbSSO and SbSigSSO have inversions as in Inv. 3. All the approximations in **b** and **c** were calculated based on the lower bound of each problem

Table 1 Summary of the approximation factor of the approximation algorithms presented in this manuscript for general permutations (GP), permutations with $n\ell$ inversions for some $\ell \geq 1$ (ℓIP), permutations with at least n inversions (1IP), and permutations with at least $2n$ inversions (2IP)

Sorting problem	GP	ℓIP	1IP	2IP
SbSSR	3	$1+\frac{1}{\ell}$	2	1.5
SbSST	3	$1+\frac{1}{\ell}$	2	1.5
SbSSO	3	$1+\frac{1}{\ell}$	2	1.5
SbSigSSR	5	$1+\frac{2}{\ell}$	3	2
SbSigSSO	5	$1+\frac{2}{\ell}$	3	2

distances with a slightly higher value than the same algorithms for unsigned permutations, which is expected given that in addition to inversions and intergenic sizes, they also need to take care of elements with negative signs.

Concerning the approximation factors in Fig. 5b, c, it can be noted that despite the theoretical approximation factors of 3 and 5, the average approximation factors of instances in FRI were between 1 and 2.2. Furthermore, in our tests, no instance for SbSSR and SbSSO, whose theoretical approximation factors are 3, had approximation factor above 2.5, and no instance for SbSigSSR and SbSigSSO, whose theoretical approximation factors are 5, obtained approximation above 3.3 and 3, respectively.

In Fig. 6a we have another scenario where the returned distance follows the number of applied SSOs, but this behavior is due to our choice of applying only operations that do not destroy previously created inversions. One interesting thing about this figure is that distances returned by the algorithm for SSOs were very close to the number of inversions, especially when $w \geq 400$, something that did not happened on FRI. In Fig. 6b, c, we see that this dataset returned maximum and average approximations systematically better than for dataset FRI: no instance had an approximation factor greater than 2.5, and on average all algorithms have average approximation factors less than 1.7. For $w \geq 120$ (resp. $w \geq 240$), where we expect to have around n (resp. $2n$) inversions, none of the instances had approximation factor above 2 (resp. 1.5), as we expected given Lemmas 2, 4, 6, 8, and 10.

Conclusion

In this paper, we analyzed the minimum number of super short reversals and/or Super Short Transpositions needed to sort a signed or unsigned permutation π and, at the same time, transform its intergenic regions lengths r^{π} according to r^{ι}.

We defined some bounds and a graph structure that allowed us to build five algorithms (one for each considered problem) that guarantee approximation factors of 3 for unsigned permutations (using either SSRs, SSTs, or both) and 5 for signed permutations (using either SSRs or SSOs). These algorithms have better approximation factors for instances for which the number of inversions is at least n or $2n$. In the former case, it is equal to 2 for unsigned permutations (using either SSRs, SSTs, or both), and to 3 for signed permutations (using SSRs or SSOs); in the latter case it is equal to 1.5 for unsigned permutations, and to 2 for signed permutations. All of these algorithms were tested in simulated instances, showing that, on average, they behave better than their theoretical approximation factors predict.

Some questions remain open. For instance, what is the computational complexity of each of these five problems? Besides, how can we incorporate indels (insertions and deletions) of intergenic regions to these problems, to be able to compare two genomes that share the same set of genes but may differ on their total intergenic regions length?

Acknowledgements

This work was supported by the National Council for Scientific and Technological Development-CNPq (Grants 400487/2016-0, 425340/2016-3, and 140466/2018-5), the São Paulo Research Foundation-FAPESP (Grants 2013/08293-7, 2015/11937-9, 2017/12646-3, and 2017/16246-0), the Brazilian Federal Agency for the Support and Evaluation of Graduate Education-CAPES, and the CAPES/COFECUB program (Grant 831/15). We also thank the anonymous reviewers for their helpful suggestions.

Authors' contributions

First draft: ARO. Proofs: ARO, GJ, and GF. Experiments: ARO, UD, and ZD. Final manuscript: ARO, GJ, GF, UD, and ZD. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Institute of Computing, University of Campinas, Campinas, Brazil. [2] LS2N, UMR CNRS 6004, University of Nantes, Nantes, France. [3] School of Technology, University of Campinas, Limeira, Brazil.

Received: 16 February 2019 Accepted: 14 October 2019
Published online: 05 November 2019

References

1. Bafna V, Pevzner PA. Sorting by transpositions. SIAM J Discrete Math. 1998;11(2):224–40. https://doi.org/10.1137/S089548019528280X.
2. Kececioglu JD, Sankoff D. Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. Algorithmica. 1995;13:180–210. https://doi.org/10.1007/BF01188586.
3. Yancopoulos S, Attie O, Friedberg R. Efficient sorting of genomic permutations by translocation. Inversion and block interchange. Bioinformatics. 2005;21(16):3340–6. https://doi.org/10.1093/bioinformatics/bti535.
4. Hannenhalli S, Pevzner PA. Transforming men into mice (polynomial algorithm for genomic distance problem). In: Proceedings of the 36th annual symposium on foundations of computer science (FOCS'1995).

Washington, DC: IEEE Computer Society Press; 1995. https://doi.org/10.1109/SFCS.1995.492588. p. 581–92.
5. Elias I, Hartman T. A 1.375-approximation algorithm for sorting by transpositions. IEEE/ACM Trans Comput Biol Bioinform. 2006;3(4):369–79. https://doi.org/10.1109/TCBB.2006.44.
6. Fertin G, Labarre A, Rusu I, Tannier E, Vialette S. Combinatorics of genome rearrangements. Computational molecular biology. London: The MIT Press; 2009.
7. Chen T, Skiena SS. Sorting with fixed-length reversals. Discrete Appl Math. 1996;71(1–3):269–95. https://doi.org/10.1016/S0166-218X(96)00069-8.
8. Galvão GR, Lee O, Dias Z. Sorting signed permutations by short operations. Algor Mol Biol. 2015;10:12. https://doi.org/10.1186/s13015-015-0040-x.
9. Lefebvre J-F, El-Mabrouk N, Tillier ERM, Sankoff D. Detection and validation of single gene inversions. Bioinformatics. 2003;19(1):190–6. https://doi.org/10.1093/bioinformatics/btg1025.
10. Dalevi DA, Eriksen N, Eriksson K, Andersson SGE. Measuring genome divergence in bacteria: a case study using Chlamydian Data. J Mol Evol. 2002;55(1):24–36. https://doi.org/10.1007/s00239-001-0087-9.
11. Seoighe C, Federspiel N, Jones T, Hansen N, Bivolarovic V, Surzycki R, Tamse R, Komp C, Huizar L, Davis RW, Scherer S, Tait E, Shaw DJ, Harris D, Murphy L, Oliver K, Taylor K, Rajandream M-A, Barrell BG, Wolfe KH. Prevalence of small inversions in yeast gene order evolution. Proc Natl Acad Sci. 2000;97(26):14433–7. https://doi.org/10.1073/pnas.240462997.
12. McLysaght A, Seoighe C, Wolfe KH. High frequency of inversions during eukaryote gene order evolution. In: Sankoff D, Nadeau JH, editors. Comparative genomics: empirical and analytical approaches to gene order dynamics, map alignment and the evolution of gene families. New York: Springer; 2000. p. 47–58. https://doi.org/10.1007/978-94-011-4309-7_6.
13. Biller P, Guéguen L, Knibbe C, Tannier E. Breaking good: accounting for fragility of genomic regions in rearrangement distance estimation. Genome Biol Evol. 2016;8(5):1427–39. https://doi.org/10.1093/gbe/evw083.
14. Biller P, Knibbe C, Beslon G, Tannier E. Comparative genomics on artificial life. In: Beckmann A, Bienvenu L, Jonoska N, editors. Pursuit of the universal lecture notes in computer science. Cham: Springer International Publishing; 2016. p. 35–44. https://doi.org/10.1007/978-3-319-40189-8_4.
15. Fertin G, Jean G, Tannier E. Algorithms for computing the double cut and join distance on both gene order and intergenic sizes. Algor Mol Biol. 2017;12:16. https://doi.org/10.1186/s13015-017-0107-y.
16. Bulteau L, Fertin G, Tannier E. Genome rearrangements with indels in intergenes restrict the scenario space. BMC Bioinform. 2016;17(S14):225–31. https://doi.org/10.1186/s12859-016-1264-6.
17. Knuth DE. The art of computer programming, Volume 3: Sorting and searching. Reading: Addison-Wesley Publishing Company; 1998.
18. Rotem D, Urrutia J. Circular permutation graphs. Networks. 1982;12(4):429–37. https://doi.org/10.1002/net.3230120407.
19. Bousquet-Melou M. The expected number of inversions after n adjacent transpositions. Discrete Math Theor Comput Sci. 2010;12(2):65–88.

Publisher's Note