BMC Bioinformatics

PROCEEDINGS Open Access

Assembling contigs in draft genomes using reversals and block-interchanges

Chi-Long Li, Kun-Tze Chen, Chin Lung Lu*

From RECOMB-seq: Third Annual Recomb Satellite Workshop on Massively Parallel Sequencing
Beijing, China. 11-12 April 2013

Abstract

The techniques of next generation sequencing allow an increasing number of draft genomes to be produced rapidly in a decreasing cost. However, these draft genomes usually are just partially sequenced as collections of unassembled contigs, which cannot be used directly by currently existing algorithms for studying their genome rearrangements and phylogeny reconstruction. In this work, we study the one-sided block (or contig) ordering problem with weighted reversal and block-interchange distance. Given a partially assembled genome π and a completely assembled genome σ, the problem is to find an optimal ordering to assemble (i.e., order and orient) the contigs of π such that the rearrangement distance measured by reversals and block-interchanges (also called generalized transpositions) with the weight ratio 1:2 between the assembled contigs of π and σ is minimized. In addition to genome rearrangements and phylogeny reconstruction, the one-sided block ordering problem particularly has a useful application in genome resequencing, because its algorithms can be used to assemble the contigs of a draft genome π based on a reference genome σ. By using permutation groups, we design an efficient algorithm to solve this one-sided block ordering problem in $\mathcal{O}(\delta n)$ time, where n is the number of genes or markers and δ is the number of used reversals and block-interchanges. We also show that the assembly of the partially assembled genome can be done in $\mathcal{O}(n)$ time and its weighted rearrangement distance from the completely assembled genome can be calculated in advance in $\mathcal{O}(n)$ time. Finally, we have implemented our algorithm into a program and used some simulated datasets to compare its accuracy performance to a currently existing similar tool, called SIS that was implemented by a heuristic algorithm that considers only reversals, on assembling the contigs in draft genomes based on their reference genomes. Our experimental results have shown that the accuracy performance of our program is better than that of SIS, when the number of reversals and transpositions involved in the rearrangement events between the complete genomes of π and σ is increased. In particular, if there are more transpositions involved in the rearrangement events, then the gap of accuracy performance between our program and SIS is increasing.

Background

The techniques of next generation sequencing have greatly advanced in the past decade [1-3], which allows an increasing number of draft genomes to be produced rapidly in a decreasing cost. Usually, these draft genomes are partially sequenced, leading to their published genomes as collections of unassembled contigs (short for contiguous fragments). These draft genomes in contig form, however, can not be used immediately in some bioinformatics applications, such as the study of genome rearrangements, which requires the completely assembled genomes to calculate their rearrangement distances [4]. To adequately address this issue, Gaul and Blanchette [5] introduced and studied the so-called block ordering problem defined as follows. Given two partially assembled genomes, with each representing as an unordered set of blocks, the *block ordering problem* is to assemble (i.e., order and orient) the blocks of the two genomes such that the distance of genome rearrangements between the two assembled genomes is minimized. The blocks mentioned above are the contigs,

* Correspondence: cllu@cs.nthu.edu.tw
Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan

each of which can be represented by an ordered list of genes or markers. In their work [5], Gaul and Blanchette proposed a linear-time algorithm to solve the block ordering problem if the problem is further simplified to maximize the number of cycles in the breakpoint graph corresponding to the assembled genomes. The rationale behind this modification is totally based on a result obtained by Bourque and Pevzner [6], showing that the reversal distance between two assembled genomes can be approximated well by maximizing the number of cycles in their corresponding breakpoint graph. Actually, in addition to the number of cycles, the number of hurdles, as well as the presence of a fortress or not, is also important and needed for determining the actual reversal distance [7]. Therefore, it is still a challenge to efficiently solve the block ordering problem by optimizing the true rearrangement distance.

In the literature, many different kinds of genome rearrangements have been extensively studied [4], such as reversal (also called inversion), transposition and block-interchange (also called generalized transposition), translocation, fusion and fission. Reversal affects a segment on a chromosome by reversing this segment as well as exchanging its strands. Transposition rearranges a chromosome by interchanging its two adjacent and nonoverlapping segments. Block-interchange is a generalized transposition that exchanges two nonoverlapping but not necessarily adjacent segments on a chromosome. Translocation acts on two chromosomes by exchanging their the end fragments. Fusion is a special translocation that joins two chromosomes into one and fission is also a special translocation that splits a chromosome into two. In this study, we consider a variant of the block ordering problem, in which one of the two input genomes is still partially assembled but the other is completely assembled, with optimizing the genome rearrangement distance measured by weighted reversals and block-interchanges, whose weights are 1 and 2, respectively. For distinguishing this special block ordering problem from the original one, we call it as *one-sided block* (*or contig*) *ordering problem*. In fact, an efficient algorithm to solve the one-sided block ordering problem has a useful application in genome resequencing [8,9], because the reference genome for resequencing organisms can serve as the completely assembled genome in the one-sided block ordering problem and the contigs of partially assembled resequencing genome can then be assembled together into one or several scaffolds based on the reference genome. From this respect, the one-sided block ordering problem can be considered as a kind of *contig scaffolding* (*or assembly*) *problem* that aims to use genome rearrangements to create contig scaffolds for a draft genome based on a reference genome.

Currently, several contig scaffolding tools based on the reference genomes have been developed, such as Projector 2 [10], OSLay [11], ABACAS [12], Mauve Aligner [13], fillScaffolds [14], r2cat [15] and SIS [16]. Among these contig scaffolding tools, both SIS and fillScaffolds use the concept of genome rearrangements to generate contig scaffolds for a draft genome. SIS deals with only reversals, while in addition to reversals, fillScaffolds considers other rearrangements, such as transpositions and translocations (including fissions and fusions). Basically, SIS was dedicated to creating the contig scaffolds for prokaryotic draft genomes by heuristically searching for their inversion signatures, where an *inversion signature* is a pair of adjacent genes or markers appearing along a contig such that they form a breakpoint and are also located in different transcriptional strands. As for fillScaffolds, it used the traditional technique of breakpoint graphs to assemble the contigs of draft genomes. In the study by Dias and colleagues [16], they have used real prokaryotic draft genomes to demonstrate that SIS had the best overall accuracy performance when compared to the other tools we mentioned above.

In this study, we utilize permutation groups in algebra, instead of the breakpoint graphs used by Gaul and Blanchette [5], to design an efficient algorithm, whose time complexity is $\mathcal{O}(\delta n)$, for solving the one-sided block ordering problem with weighted reversal and block-interchange distance, where n is the number of genes or markers and δ is the number of reversals and block-interchanges used to transform the assembly of the partially assembled genome (i.e., draft genome) into the completely assembled genome (i.e., reference genome). In particular, we also show that the assembly of the partially assembled genome can be done in $\mathcal{O}(n)$ time and its weighted reversal and block-interchange distance from the completely assembled genome can be calculated in advance in $\mathcal{O}(n)$ time. In addition, we have implemented our algorithm into a program and used some simulated datasets to compare its accuracy performance to SIS on assembling the contigs in the draft genomes based on their reference genomes. Our experimental results have shown that the averaged normalized contig mis-join errors of our program are lower than those of SIS, when the number of reversals and transpositions involved in the rearrangement events between the complete genomes of the partially and completely assembled organisms is increased. In particular, if there are more transpositions involved in the rearrangement events, then the gap of accuracy performance between our program and SIS is increasing.

Preliminaries

One-sided block ordering problem

In the following, we dedicate ourselves to linear, unichromosomal genomes. With a slight modification, however, our algorithmic result can still apply to circular, uni-chromosomal genomes, or to multi-chromosomal genomes with linear or circular chromosomes in a chromosome-by-chromosome manner. Once completely assembled, a uni-chromosomal genome can be represented by a signed permutation of n integers between 1 and n, with each integer representing a gene or marker and its associated sign indicating the strandedness of the corresponding gene or marker. If the genome is partially assembled, then it will be represented by an unordered set of blocks, where a block B of size k, denoted by $B = [b_1, b_2, ..., b_k]$, is an ordered list of k signed integers. Let $\bar{B} = [-b_k, -b_{k-1}, \ldots, -b_1]$ denote the *reverse* of B. Given an unordered set of m blocks, say $\mathcal{B} = \{B_1, B_2, \ldots, B_m\}$, corresponding to a partially assembled genome, an *ordering* (or *assembly*) of $\mathcal{B}$ is an ordered list of m blocks in which each block B_i or its reverse $\overline{B_i}$ appears exactly once, where $1 \leq i \leq m$. For instance, suppose that $\mathcal{B} = \{B_1, B_2, B_3\} = \{[1, 4], [3, 2], [-5, 6]\}$. Then (B_1, B_3, B_2) = ([1, 4], [-5, 6], [3, 2]) and $(B_1, -B_3, B_2)$ = ([1, 4], [-6, 5], [3, 2]) are two orderings of $\mathcal{B}$. Basically, each ordering of $\mathcal{B}$ *induces* (or *defines*) a signed permutation of size n, which is obtained by concatenating the blocks in this ordered list. For instance, the ordering (B_1, B_3, B_2) in the above exemplified $\mathcal{B}$ induces the signed permutation (1, 4, -5, 6, 3, 2), which simply is denoted by $B_1 \odot B_3 \odot B_2$. Clearly, the permutation induced by an ordering of $\mathcal{B}$ corresponds to an assembly of the blocks in $\mathcal{B}$. Now, the one-sided block ordering problem we study in this paper is formally defined as follows:

One-sided block ordering problem with reversal and block-interchange distance

Input: A partially assembled genome π and a completely assembled genome σ.

Output: Find an ordering of π such that the rearrangement distance measured by reversals and block-interchanges with the weight ratio 1:2 between the permutation induced by the ordering of π and σ is minimized.

As discussed in our previous study [17], it is biologically meaningful to assign twice the weight to block-interchanges than to reversals, due to the observation from the biological data that transpositions occur with about half the frequency of reversals [18].

Permutation groups

Permutation groups have been proven to be a very useful tool in the studies of genome rearrangements [17]. Below, we recall some useful definitions, notations and properties borrowed form our previous work [17]. Basically, given a set E = {1, 2, ..., n}, a *permutation* is defined to be a one-to-one function from E into itself and usually expressed as a product of cycles in the study of genome rearrangements. For instance, π = (1)(3, 2) is a product of two cycles to represent a permutation of E = {1, 2, 3} and means that $\pi(1) = 1$, $\pi(2) = 3$ and $\pi(3) = 2$. The elements in a cycle can be arranged in any cyclic order and hence the cycle (3, 2) in the permutation π exemplified above can be rewritten as (2, 3). Moreover, if the cycles in a permutation are all disjoint (i.e., no common element in any two cycles), then the product of these cycles is called the *cycle decomposition* of the permutation. In fact, a permutation in the cycle decomposition can be used to model a genome containing several circular chromosomes, with each disjoint cycle representing a circular chromosome. Notice that in the rest of this article, we say "cycle in a permutation" to mean "cycle in the cycle decomposition of this permutation" for simplicity, unless otherwise specified. A cycle with k elements is further called a *k-cycle*. In convention, the 1-cycles in a permutation are not written explicitly since their elements are *fixed* in the permutation. For instance, the above exemplified permutation π can be written as π = (2, 3). If the cycles in a permutation are all 1-cycles, then this permutation is called an *identify permutation* and denoted by **1**. Suppose that α and β are two permutations of E. Then their product $\alpha\beta$, also called their *composition*, defines a permutation of E satisfying $\alpha\beta(x) = \alpha(\beta(x))$ for all $x \in E$. If both α and β are disjoint, then $\alpha\beta = \beta\alpha$. If $\alpha\beta = \mathbf{1}$, then α is called the *inverse* of β, denoted by β^{-1}, and vice versa. Moreover, the *conjugation* of β by α, denoted by $\alpha \cdot \beta$, is defined to be the permutation $\alpha\beta\alpha^{-1}$. It can be verified that if $y = \beta(x)$, then $\alpha(y) = \alpha\beta(x) = \alpha\beta\alpha^{-1}\alpha(x) = \alpha \cdot \beta(\alpha(x))$. Hence, $\alpha \cdot \beta$ can be obtained from β by just changing its element x with $\alpha(x)$. In other words, if $\beta = (b_1, b_2, ..., b_k)$, then $\alpha \cdot \beta = (\alpha(b_1), \alpha(b_2), ..., \alpha(b_k))$.

It is a fact that every permutation can be expressed into a product of 2-cycles, in which 1-cycles are still written implicitly. Given a permutation α of E, its *norm*, denoted by $||\alpha||$, is defined to be the minimum number, say k, such that α can be expressed as a product of k 2-cycles. In the cycle decomposition of α, let $n_c(\alpha)$ denote the number of its disjoint cycles, notably including the 1-cycles not written explicitly. Given two permutations α and β of E, α is said to *divide* β, denoted by $\alpha|\beta$, if and only if $||\beta\alpha^{-1}|| = ||\beta|| - ||\alpha||$. In our previous work [17], it has been shown that $||\alpha|| = |E| - n_c(\alpha)$ and for any k elements in E, say $a_1, a_2, ..., a_k$, they all appear in a cycle of α in the ordering of $a_1, a_2, ..., a_k$ if and only if $(a_1, a_2, ..., a_k) \mid \alpha$.

Let $\alpha = (a_1, a_2)$ be a 2-cycle and β be an arbitrary permutation of E. If $\alpha|\beta$, that is, both a_1 and a_2 appear in the same cycle of β, then the composition $\alpha\beta$, as well as $\beta\alpha$, has the effect of fission by breaking this cycle into two smaller cycles. For instance, let α = (1, 3) and β = (1, 2, 3, 4). Then $\alpha|\beta$, since both 1 and 3 are in the cycle (1, 2, 3, 4), and $\alpha\beta$ = (1, 2)(3, 4) and $\beta\alpha$ = (4, 1)(2, 3). On the other hand, if $\alpha \nmid \beta$, that is, a_1 and a_2 appear in different cycles of β, then $\alpha\beta$, as well as $\beta\alpha$, has the effect of fusion by joining the two cycles into a bigger cycle. For example, if α = (1, 3) and β = (1, 2)(3, 4), then $\alpha \nmid \beta$ and, as a result, $\alpha\beta$ = (1, 2, 3, 4) and $\beta\alpha$ = (2, 1, 4, 3).

A model for representing DNA molecules

As mentioned before, a permutation in the form of the cycle decomposition can be used to model a genome containing multiple chromosomes (or a chromosome with multiple contigs), with each cycle representing a chromosome (or contig). To facilitate modelling the rearrangement of reversals using the permutation groups, however, we need to use two cycles to represent a chromosome, with one cycle representing a strand of the chromosome and the other representing the complementary strand. For this purpose, we first let E = {-1, 1, -2, 2, ..., -*n*, -*n*} and Γ = (1, -1)(2, -2) ... (*n*, -*n*). We then use an *admissible* cycle, which is a cycle containing no *i* and its opposite -*i* simultaneously for some $i \in E$, to represent a chromosomal strand, say π^+, and use $\pi^- = \Gamma \cdot (\pi^+)^{-1}$, which is the *reverse complement* of π^+, to represent the opposite strand of π^+. As demonstrated in our previous work [17], it is useful to represent a double stranded chromosome π by the product of its two strands π^+ and π^-, that is, $\pi = \pi^+\pi^- = \pi^-\pi^+$, because a reversal (respectively, block-interchange) acting on this DNA molecule can be mimicked by multiplying two (respectively, four) 2-cycles with π, as described in the following lemmas.

Lemma 1 ([17]) *Let* $\pi = \pi^+\pi^-$ *denote a double stranded DNA and let x and y be two elements in E. If* $(x, y) \nmid \pi$, *that is, x and y are in the different strands of* π, *then the effect of* $(\pi\Gamma(y), \pi\Gamma(x))(x, y)\pi$ *is a reversal acting on* π.

Lemma 2 ([17]) *Let* $\pi = \pi^+\pi^-$ *denote a double stranded DNA and let u, v, x and y be four elements in E. If* $(x, u, y, v)|\pi$, *that is, x, u, y and v appear in the same strand of* π *in this order, then the effect of* $(\pi\Gamma(v), \pi\Gamma(u))\ (\pi\Gamma(y), \pi\Gamma(x))\ (u, v)(x, y)\pi$ *is a block-interchange acting on* π.

Moreover, as described in the following lemma, we have shown in [17] that given two different DNA molecules π and σ, every cycle α in (the cycle decomposition of) $\sigma\pi^{-1}$ always has a *mate* cycle $(\pi\Gamma) \cdot \alpha^{-1}$ that also appears in $\sigma\pi^{-1}$. In fact, α and $(\pi\Gamma) \cdot \alpha^{-1}$ in $\sigma\pi^{-1}$ are each other's mate cycle.

Lemma 3 ([17]) *Let* π *and* σ *be two different double-stranded DNA molecules. If* α *is a cycle in* $\sigma\pi^{-1}$, *then* $(\pi\Gamma) \cdot \alpha^{-1}$ *is also a cycle in* $\sigma\pi^{-1}$.

An efficient algorithm for the one-sided block ordering problem

To clarify our algorithm, we start with defining some notations. Let α denote an arbitrary linear DNA molecule (or contig). As mentioned previously, it is represented by the product of its two strands α^+ and α^-, that is, $\alpha = \alpha^+\alpha^-$. If α contains *k* genes (or markers), we also denote its α^+ by ($\alpha^+[1]$, $\alpha^+[2]$, ..., $\alpha^+[k]$), where $\alpha^+[i]$ is the *i*-th gene in α, and its α^- by (α^- [1], $\alpha^-[2]$, ..., $\alpha^-[k]$). By convention, $\alpha^+[1]$ and $\alpha^-[1]$ are called as *tails* of α. Let $\pi = \pi_1\pi_2 \ldots \pi_m$ be a linear, uni-chromosomal genome that is partially assembled into *m* contigs π_1, π_2, ..., π_m, each with n_i genes, and σ = (1, 2, ..., *n*) be a linear, uni-chromosomal genome that is assembled completely. Let $C = \{c_k = n + k + 1: 0 \le k \le 2m - 1\} \cup \{-c_k = -n - k - 1: 0 \le k \le 2m - 1\}$ be a set of 4*m* distinct integers, called *caps*, which are different from those genes in *E*. Let $\widehat{E} = E \cup C$ and $\widehat{\Gamma} = (1, -1)(2, -2) \ldots (n + 2m, -n - 2m)$. For the purpose of designing our algorithm later, we add four caps $c_{2(i-1)}, c_{2(i-1)+1}, -c_{2(i-1)}$ and $-c_{2(i-1)+1}$ to the ends of each contig π_i, where $1 \le i \le m$, leading to a capping contig $\hat{\pi}_i$ with $\hat{\pi}_i^+[1] = c_{2(i-1)}$, $\hat{\pi}_i^+[j] = \pi_i^+[j - 1]$, for $2 \le j \le n_i + 1$, $\hat{\pi}_i^+[n_i + 2] = c_{2(i-1)+1}$, $\hat{\pi}_i^-[1] = \hat{\Gamma}(c_{2(i-1)+1})$, $\hat{\pi}_i^-[j] = \pi_i^-[j - 1]$ for $2 \le j \le n_i + 1$, and $\hat{\pi}_i^-[n_i + 2] = \hat{\Gamma}(c_{2(i-1)})$. Moreover, we insert *m*-1 dummy contigs without any genes (i.e., null contigs) σ_2, σ_3, ..., σ_m into σ, where the original contig in σ becomes σ_1 now, and add four caps $c_{2(i-1)}$, $c_{2(i-1)+1}$, $-c_{2(i-1)}$ and $-c_{2(i-1)+1}$ to the ends of each contig σ_i to obtain a capping contig $\hat{\sigma}_i$, where $\hat{\sigma}_i^+[1] = c_{2(i-1)}, \hat{\sigma}_i^+[j] = \sigma_i^+[j - 1]$ for $2 \le j \le n_i + 1, \hat{\sigma}_i^+[n_i + 2] = c_{2(i-1)+1}$, $\hat{\sigma}_i^-[1] = \hat{\Gamma}(c_{2(i-1)+1}), \hat{\sigma}_i^-[j] = \sigma_i^-[j - 1]$ for $2 \le j \le n_i + 1$, and $\hat{\sigma}_i^-[n_i + 2] = \hat{\Gamma}(c_{2(i-1)})$. Notice that the purpose of adding caps to the ends of the contigs is to serve as delimiters when we use permutation groups to model translocations of multiple contigs later. We denote the capping π and σ by $\widehat{\pi}$ and $\widehat{\sigma}$, respectively. To distinguish the four caps in a capping contig, say $\widehat{\pi}_i$, we call the left caps $\hat{\pi}_i^+[1]$ and $\hat{\pi}_i^-[1]$ as 5' caps and the right caps $\hat{\pi}_i^+[n_i + 2]$ and $\hat{\pi}_i^-[n_i + 2]$ as 3' caps.

Given an integer *x* in $\widehat{E}$ that is contained in a contig $\alpha = \alpha^+\alpha^-$ with *k* genes (or markers), we define a function **char**$(x, \hat{\alpha})$ below to represent the character of *x* in the capping contig $\hat{\alpha} = \hat{\alpha}^+\hat{\alpha}^-$ that is obtained by adding four caps from *C* to the ends of α.

$$\operatorname{char}(x,\ \hat{\alpha}) = \begin{cases} \text{C5,} & \text{if } x = \hat{\alpha}^+[1] \text{ or } x = \hat{\alpha}^-[1] \\ & \quad \text{(that is, } x \text{ is a } 5' \text{ cap in } \hat{\alpha}\text{).} \\ \text{C3,} & \text{if } k \neq 0 \text{ and } (x = \hat{\alpha}^+[k+2] \text{ or } x = \hat{\alpha}^-[k+2]) \\ & \quad \text{(that is, } \alpha \text{ is not null and } x \text{ is a } 3' \text{ cap in } \hat{\alpha}\text{).} \\ \text{N3,} & \text{if } k = 0 \text{ and } (x = \hat{\alpha}^+[k+2] \text{ or } x = \hat{\alpha}^-[k+2]) \\ & \quad \text{(that is, } \alpha \text{ is null and } x \text{ is the } 3' \text{ cap in } \hat{\alpha}\text{).} \\ \text{T,} & \text{if } k \neq 0 \text{ and } (x = \hat{\alpha}^+[2] \text{ or } x = \hat{\alpha}^-[2]) \\ & \quad \text{(that is, } \alpha \text{ is not null and } x \text{ is a tail in } \alpha\text{).} \\ \text{O,} & \text{otherwise.} \end{cases}$$

In addition, we define $5\mathrm{cap}(x,\ \hat{\alpha})$ to be the 5' cap in the strand of $\hat{\alpha}$ that contains *x*. For convenience, we extend the definitions above from the capping contig to the capping genome. For instance, given a capping genome, say $\hat{\pi}$, char$(x, \hat{\pi})$ denotes the character of x in a capping contig $\hat{\pi}_i$ of $\hat{\pi}$ that contains x, and 5cap$(x, \hat{\pi})$ denotes the 5' cap of the strand in $\hat{\pi}_i$ containing x, that is, char$(x, \hat{\pi})$ = char$(x, \hat{\pi}_i)$ and 5cap$(x, \hat{\pi})$ = 5cap$(x, \hat{\pi}_i)$. In our previous work [17], we have shown the following lemma.

Lemma 4 ([17]) *For a capping genome* $\hat{\pi}$ *and* $x \in \hat{E}$, *if* char $(x, \hat{\pi})$ = C3 *(respectively, T), then* char$(\hat{\pi}\widehat{\Gamma}(x), \hat{\pi})$ *is T (respectively, C3) and if* char$(x, \hat{\pi})$ = *O* (*respectively, N3 and C5), then* char$(\hat{\pi}\widehat{\Gamma}(x), \hat{\pi})$ *is O (respectively, N3 and C5).*

Basically, we design our algorithm to solve the one-sided block ordering problem by dealing with the contigs of the capping genome $\hat{\pi}$ as if they were linear chromosomes. Let $c_1 = (x, y)$ and $c_2 = (u, v)$ be two 2-cycles with character pairs of (non-C5, non-C5) and (C5, C5), respectively, and let $c'_1 = (\hat{\pi}\widehat{\Gamma}(y), \hat{\pi}\widehat{\Gamma}(x))$ and $c'_2 = (\hat{\pi}\widehat{\Gamma}(v), \hat{\pi}\widehat{\Gamma}(u))$. Notice that the character pair of c'_2 is (C5, C5) by Lemma 4. In our previous study [17], we have proven that performing a translocation τ on $\hat{\pi}$ can be mimicked by the composition of $c'_2c'_1c_2c_1\hat{\pi}$ (i.e., $\tau = c'_2c'_1c_2c_1$), if $(x, u) \mid \hat{\pi}, (y, v) \mid \hat{\pi}, (x, y) \nmid \hat{\pi}$ and $(x, \hat{\Gamma}(y)) \nmid \hat{\pi}$ (i.e., x and u, as well as y and v, lie in the same contig stand in $\hat{\pi}$, but x and y appear in the different contigs in $\hat{\pi}$). Moreover, if the character pair of c_1 is in CEpair = {(C3, C3), (C3, N3), (T, T), (T, N3), (N3, N3)}, then τ acts on $\hat{\pi}$ by exchanging the two caps of some contig in $\hat{\pi}$ with the two caps of another contig and, as a result, leaves the original genome π unaffected. Notice that the character pair of c'_1 also belongs to CEpair and that of c'_2 is (C5, C5) according to Lemma 4. Furthermore, if c_1 is a 2-cycle of character pair (T, C3) (respectively, (O, N3)), then performing τ on $\hat{\pi}$ becomes a fusion (respectively, fission) to act on π. Hence, we have the following lemma, where it can be verified that $\hat{\pi}\widehat{\Gamma}(5\text{cap}(x, \hat{\pi})) = 5\text{cap}(\hat{\pi}\widehat{\Gamma}(x), \hat{\pi})$ and and $\hat{\pi}\widehat{\Gamma}(5\text{cap}(y, \hat{\pi})) = 5\text{cap}(\hat{\pi}\widehat{\Gamma}(y), \hat{\pi})$.

Lemma 5 ([17]) *Let* $c_1 = (x, y)$ *denote a 2-cycle with* char $(x, \hat{\pi}) = T$ *and* char $(y, \hat{\pi})$ = C3 , *and let* $c'_1 = \left(\hat{\pi}\widehat{\Gamma}\left(y\right), \hat{\pi}\widehat{\Gamma}\left(x\right)\right)$, $c'_1 = \left(\hat{\pi}\widehat{\Gamma}\left(y\right), \hat{\pi}\widehat{\Gamma}\left(x\right)\right)$ *and* $\hat{\pi}\widehat{\Gamma}(5\text{cap}(x, \hat{\pi})))$, $\hat{\pi}\widehat{\Gamma}(5\text{cap}(x, \hat{\pi})))$. *If* $(x, y) \nmid \hat{\pi}$ *and* $(x, \widehat{\Gamma}(y)) \nmid \hat{\pi}$, *then the effect of* $c'_2c'_1c_2c_1\hat{\pi}$ *is a fusion that acts on π by concatenating the contig containing y with the contig containing x.*

It is not hard to see that the permutation induced by an ordering of the uncapped genome π can be considered as the result of applying consecutive m - 1 fusions to the m contigs in π. Based on the above discussion, it can be realized that our purpose is to find m - 1 translocations to act on $\hat{\pi}$ such that their rearrangement effects on the original π are m - 1 fusions and the genome rearrangement distance measured by weighted reversals and block-interchanges between the resulting assembly of the contigs in π and σ is minimum. In Algorithm 1 below, we describe our algorithm for efficiently solving the one-sided block ordering problem, where reversals are weighted one and block-interchanges are weighted two. Basically, we try to derive m - 1 fusions from $\hat{\sigma}\hat{\pi}^{-1}$ to act on π in Algorithm 1.

```
Algorithm 1
Input: A partially assembled, linear, uni-chromosomal genome $\pi = \pi_1\pi_2 ... \pi_m$ and a completely assembled, linear, uni-chromosomal genome $\sigma = \sigma_1$.
Output: An optimally assembled genome of $\pi$, denoted by assembly($\pi$), and the weighted reversal and block-interchange distance $\Delta(\pi, \sigma)$ between assembly($\pi$) and $\sigma$.
1: Add $m$ - 1 null contigs $\sigma_2, \sigma_3, \ldots, \sigma_m$ into $\sigma$ such that $\sigma = \sigma_1\sigma_2 \ldots \sigma_m$.
   Obtain $\hat{\pi} = \hat{\pi}_1\hat{\pi}_2 \ldots \hat{\pi}_m$ and $\hat{\sigma} = \hat{\sigma}_1\hat{\sigma}_2 \ldots \hat{\sigma}_m$ by capping $\pi$ and $\sigma$.
2: Compute $\hat{\sigma}\hat{\pi}^{-1}$ and $\hat{\pi}\widehat{\Gamma}$.
3: /* To perform cap exchanges */

    Let $i = 0$.
    while there are $x$ and $y$ in a cycle of $\hat{\sigma}\hat{\pi}^{-1}$ such that (char($x, \hat{\pi}$), char($y, \hat{\pi}$)) $\in$ CEpair do
        Let $i = i + 1$.
        Find $x$ and $y$ in a cycle of $\hat{\sigma}\hat{\pi}^{-1}$ with (char($x, \hat{\pi}$), char($y, \hat{\pi}$)) $\in$ CEpair.
        Let $\chi_i = (\hat{\pi}\widehat{\Gamma}(5\text{cap}(y, \hat{\pi})), \hat{\pi}\widehat{\Gamma}(5\text{cap}(x, \hat{\pi})))(\hat{\pi}\widehat{\Gamma}(y), 5\text{cap}(y, \hat{\pi}))(x, y), 5\text{cap}(y, \hat{\pi}))(x, y)$.
        Calculate new $\hat{\pi} = \chi_i\hat{\pi}$, new $\hat{\pi}\widehat{\Gamma} = \chi_i\hat{\pi}\widehat{\Gamma}$ and new $\hat{\sigma}\hat{\pi}^{-1} = \hat{\sigma}\hat{\pi}^{-1}\chi_i^{-1}$.
    end while

4: /* To find consecutive m - 1 fusions */

    Let $i = 0$.
    while there are two adjacent elements $x$ and $y$ in a cycle of $\hat{\sigma}\hat{\pi}^{-1}$ such that (char($x, \hat{\pi}$), char($y, \hat{\pi}$)) = (T, C3) and $(x, y) \nmid \hat{\pi}$ do
        Let $i = i + 1$.
        Find two adjacent elements $x$ and $y$ in a cycle of $\hat{\sigma}\hat{\pi}^{-1}$ such that (char($x, \hat{\pi}$), char($y, \hat{\pi}$)) = (T, C3) and $(x, y) \nmid \hat{\pi}$.
        Let $\tau_i = (\hat{\pi}\widehat{\Gamma}(5\text{cap}(y, \hat{\pi})), \hat{\pi}\widehat{\Gamma}(5\text{cap}(x, \hat{\pi})))(\hat{\pi}\widehat{\Gamma}(y), 5\text{cap}(y, \hat{\pi}))(x, y), 5\text{cap}(y, \hat{\pi}))(x, y)$.
        Calculate new $\hat{\pi} = \tau_i\hat{\pi}$, new $\hat{\pi}\widehat{\Gamma} = \tau_i\hat{\pi}\widehat{\Gamma}$ and new $\hat{\sigma}\hat{\pi}^{-1} = \hat{\sigma}\hat{\pi}^{-1}\tau_i^{-1}$.
    end while
    while $i < m - 1$ do
        Let $i = i + 1$.
        Find two adjacent elements $x$ and $y$ in a cycle of $\hat{\sigma}\hat{\pi}^{-1}$ such that $(\text{char}(x, \hat{\pi}), \text{char}(y, \hat{\pi})) = (\text{T}, \text{C3})$ and $(x, y)|\hat{\pi}$.
        Find the strand of a different contig in $\hat{\pi}$ with at least a non-cap integer and its 3' cap, say $z$, different from $y$.
        Let $\tau_i = (\hat{\pi}\widehat{\Gamma}(z), \hat{\pi}\widehat{\Gamma}(x))(\hat{\pi}\widehat{\Gamma}(z), \hat{\pi}\widehat{\Gamma}(y))(y, z)(x, z)$.
        Calculate new $\hat{\pi} = \tau_i\hat{\pi}$, new $\hat{\pi}\widehat{\Gamma} = \tau_i\hat{\pi}\widehat{\Gamma}$ and new $\hat{\sigma}\hat{\pi}^{-1} = \hat{\sigma}\hat{\pi}^{-1}\tau_i^{-1}$.
    end while
    Let $assembly(\pi)$ denote the assembled contig in current $\hat{\pi}$ whose caps are removed.

5: /* To find reversals */

    Let $n_\gamma = 0$.
    while there are two adjacent elements $x$ and $y$ in a cycle of $\hat{\sigma}\hat{\pi}^{-1}$ such that $(x, \widehat{\Gamma}(y))|\hat{\pi}$ do
        Let $n_\gamma = n_\gamma + 1$.
        Find two adjacent elements $x$ and $y$ in a cycle of $\hat{\sigma}\hat{\pi}^{-1}$ such that $(x, \widehat{\Gamma}(y))|\hat{\pi}$.
        Let $\gamma_{n_\gamma} = (\hat{\pi}\widehat{\Gamma}(y), \hat{\pi}\widehat{\Gamma}(x))(x, y)$.
        Calculate new $\hat{\pi} = \gamma_{n_\gamma}\hat{\pi}$, new $\hat{\pi}\widehat{\Gamma} = \gamma_{n_\gamma}\hat{\pi}\widehat{\Gamma}$ and new $\hat{\sigma}\hat{\pi}^{-1} = \hat{\sigma}\hat{\pi}^{-1}\gamma_{n_\gamma}^{-1}$.
    end while

6: /* To find block-interchanges */

    Let $n_\beta = 0$.
    while $\hat{\sigma}\hat{\pi}^{-1} \neq 1$ do
        Let $n_\beta = n_\beta + 1$.
        Choose any two adjacent elements $x$ and $y$ in a cycle of $\hat{\sigma}\hat{\pi}^{-1}$.
        Find two adjacent integers $u$ and $v$ in a cycle of $\hat{\sigma}\hat{\pi}^{-1}(x, y)$ such that $(u, v) \nmid (x, y)\hat{\pi}$.
        Let $\beta_\delta = (\hat{\pi}\widehat{\Gamma}(v), \hat{\pi}\widehat{\Gamma}(u))(\hat{\pi}\widehat{\Gamma}(y), \hat{\pi}\widehat{\Gamma}(x))(u, v)(x, y)$.
        Calculate new $\hat{\pi} = \beta_{n_\beta}\hat{\pi}$, new $\hat{\pi}\widehat{\Gamma} = \gamma_{n_\beta}\hat{\pi}\widehat{\Gamma}$ and new $\hat{\sigma}\hat{\pi}^{-1} = \hat{\sigma}\hat{\pi}^{-1}\beta_{n_\beta}^{-1}$.
    end while

7: Output $assembly(\pi)$ and $\Delta(\pi, \sigma) = n_\gamma + 2n_\beta$.
```

Below, we consider an example to clarify Algorithm 1. Let π = {[1, 4], [-5, 6], [3, 2]} and σ = {[1, 2, ..., 6]} be the input linear, uni-chromosomal genomes of Algorithm 1. In our algorithm, these two genomes will be further represented by π = (1, 4)(-4, -1)(-5, 6)(-6, 5)(3, 2)(-2, -3) and σ = (1, 2, ..., 6)(-6, -5, ..., -1). First of all, we add two null contigs into σ and cap all the contigs in π and σ in a way such that $\hat{\pi} = (7,\ 1,\ 4,\ 8)(-8,\ -4,\ -1,\ -7)(9,\ -5,\ 6,\ 10)(-10,\ -6,\ 5,\ -9)(11,\ 3,\ 2,\ 12)(-12,\ -2,\ -3,\ -11)$ and $\hat{\sigma} = (7,\ 1,\ 2,\ \ldots,\ 6,\ 8)(-8,\ -6,\ -5,\ \ldots,\ -1,\ -7)(9,\ 10)(-10,\ -9)(11,\ 12)(-12,\ -11)$. Next, we compute $\hat{\sigma}\hat{\pi}^{-1} = (2,\ 4)(-1,\ -3)(3,\ 12)(-2,\ -11)(5,\ -5,\ 10,\ 8)(-4,\ -6,\ -9,\ 6)$. It can be found that 10 and 8 are in a cycle of current $\hat{\sigma}\hat{\pi}^{-1}$ with $(\text{char}\,(10,\ \hat{\pi}),\ \text{char}(8,\ \hat{\pi})) = (\text{C3},\ \text{C3}) \in \text{CEpair}$. We perform a cap exchange on $\hat{\pi}$ by multiplying $(\hat{\pi}\widehat{\Gamma}(5\text{cap}(8,\ \hat{\pi})),\ (\hat{\pi}\widehat{\Gamma}(5\text{cap}(8,\ \hat{\pi})),\ \hat{\pi}\widehat{\Gamma}(5\text{cap}(10,\ \hat{\pi})))(\hat{\pi}\widehat{\Gamma}(8),\ 5\text{cap}(8,\ \hat{\pi}))(10,\ 8) = (-8,\ -10)(-4,\ -6)(9,7)(10,8)$ $5\text{cap}(8,\ \hat{\pi}))(10,\ 8) = (-8,\ -10)(-4,\ -6)(9,7)(10,8)$ with $\hat{\pi}$, resulting in new $\hat{\pi} = (7,\ 1,\ 4,\ 10)(-10,\ -4,\ -1,\ -7)(9,\ -5,\ 6,\ 8)(-8,\ -6,\ 5,\ -9)(11,\ 3,\ 2,\ 12)(-12,\ -2,\ -3,\ -11)$. In addition, we have new $\hat{\sigma}\hat{\pi}^{-1} = (2,\ 4)(-1,\ -3)(3,\ 12)(-2,\ -11)(5,\ -5,\ 10)(-4,\ -9,\ 6)(9,\ 7)(-10,\ -8)$. It can be observed that -5 and 10 are in the same cycle of $\hat{\sigma}\hat{\pi}^{-1}$ with satisfying that $\text{char}(--5,\ \hat{\pi}) = \text{T}$, $\text{char}(10,\ \hat{\pi}) = \text{C3}$ and $(-5,\ 10) \nmid \hat{\pi}$ (since -5 and 10 are in different contigs in current $\hat{\pi}$). Therefore, we perform a fusion on $\hat{\pi}$, by multiplying $\hat{\pi}\widehat{\Gamma}(5\text{cap}(-5,\ \hat{\pi})))(\hat{\pi}\widehat{\Gamma}(10),\ \hat{\pi}\widehat{\Gamma}(5\text{cap}(-5,\ \hat{\pi})))(\hat{\pi}\widehat{\Gamma}(10),\ 5\text{cap}(10,\ \hat{\pi}))(-5,\ 10) =,$ $5\text{cap}(10,\ \hat{\pi}))(-5,\ 10) = (-10,\ -8)(-4,\ -9)(9,7)(-5,10)$ with $\hat{\pi}$, to obtain new $\hat{\pi} = (7,\ 1,\ 4,\ -5,\ 6,\ 8)(-8,\ -6,\ 5,\ -4,\ -1,\ -7)(9,10)(-10,\ -9)(11,\ 3,\ 2,\ 12)(-12,\ -2,\ -3\ -11)$. Moreover, we have new $\hat{\sigma}\hat{\pi}^{-1} = (2,\ 4)(-1,\ -3)(3,\ 12)(-2,\ -11)(5,\ -5)(-4,\ 6)$, in which 3 and 12 form a (T, C3) pair but they belong to the same contig strand in $\hat{\pi}$, that is, $(3,\ 12)|\hat{\pi}$. In this case, $\hat{\pi}$ has a contig strand (7, 1, 4, -5, 6, 8) whose 3' cap is 8 that is different from 12. Hence, we multiply $(\hat{\pi}\widehat{\Gamma}(8),\ \hat{\pi}\widehat{\Gamma}(3))(\hat{\pi}\widehat{\Gamma}(8),$ $\hat{\pi}\widehat{\Gamma}(12))(12,\ 8)(3,\ 8) = (-6,\ -11)(-6,\ -2)(12,\ 8)(3,\ 8)$ with $\hat{\pi}$ to obtain new $\hat{\pi} = (7,\ 1,\ 4,\ -5,\ 6,\ 3,\ 2,\ 8)(-8,\ -2,\ -3,\ -6,\ 5,\ -4,\ -1,\ -7)(9,\ 10)(-10,\ -9)(11,\ 12)(-12,\ -11)$ and new $\hat{\sigma}\hat{\pi}^{-1} = (2,\ 4)(-1,\ -3)(5,\ -5)(-4,\ 6)(3,\ 8)(-2,\ -6)$. Notice that -4 and 6 are adjacent in a cycle of current $\hat{\sigma}\hat{\pi}^{-1}$ and they are in different strands in current $\hat{\pi}$ since $(-4,\ \widehat{\Gamma}(6))|\hat{\pi}$. Thus, we can find a reversal, which is $(\hat{\pi}\widehat{\Gamma}(6),\ \hat{\pi}\widehat{\Gamma}(-4))(-4,\ 6) = (5,\ -5)(-4, 6)$, from $\hat{\sigma}\hat{\pi}^{-1}$ to transform $\hat{\pi}$ into (7, 1, 4, 5, 6, 3, 2, 8) (-8, -2, -3, -6, -5, -4, -1, -7) (9, 10) (-10, -9) (11, 12) (-12, -11). After that, we have new $\hat{\sigma}\hat{\pi}^{-1} = (2,\ 4)(-1,\ -3)(3,\ 8)(-2,\ -6)$, which can serve as a block-interchange to further transform $\hat{\pi}$ into (7, 1, 2, 3, 4, 5, 6, 8)(-8, -6, -5, -4, -3, -2, -1, -7) (9, 10) (-10, -9) (11, 12) (-12, -11), which is equal to $\hat{\sigma}$. As a result, we obtain an ordering ([1,4], [-5, 6], [3,2]) of π whose induced permutation [1,4] ⊙ [-5, 6] ⊙ [3,2] = (1, 4, -5, 6, 3, 2) can be transformed into the permutation (1, 2, ..., 6) of σ using a reversal and a block-interchange (i.e., $\Delta(\pi,\ \sigma) = 3$).

Actually, after running the step 3 of Algorithm 1, it can be verified according to the capping of π and σ and Lemma 3 that for any two adjacent elements x and y in a cycle of $\hat{\pi}\hat{\sigma}^{-1}$ with $(\text{char}(x,\ \hat{\pi}),\ \text{char}(y,\ \hat{\pi})) = (\text{T},\ \text{C3})$, if $(x,\ y) \nmid \hat{\pi}$, then $(x, \widehat{\Gamma}(y)) \nmid \hat{\pi}$. Moreover, the operation $\tau_i = (\hat{\pi}\widehat{\Gamma}(z),\ \hat{\pi}\widehat{\Gamma}(x))(\hat{\pi}\widehat{\Gamma}(z),\ \hat{\pi}\widehat{\Gamma}(y))(y,\ z)(x,\ z)$ used in the step 4 of Algorithm 1 acts on $\hat{\pi}$ still as a fusion of π, as explained as follows. Notice that $(x,\ y)|\hat{\pi}$, meaning that x and y are in the same cycle of $\hat{\pi}$ and hence $5\text{cap}(x, \hat{\pi}) = 5\text{cap}(y, \hat{\pi})$. It can be verified that $(5\text{cap}(y, \hat{\pi}), 5\text{cap}(z, \hat{\pi})) = 1$, $5\text{cap}(z, \hat{\pi})) = 1$. Since $(x, y)|\hat{\pi}$, we have $(\hat{\pi}\widehat{\Gamma}(y), \hat{\pi}\widehat{\Gamma}(x))|\hat{\pi}$ and hence $(5\text{cap}(\hat{\pi}\widehat{\Gamma}(z), \hat{\pi})$, $5\text{cap}(\hat{\pi}\widehat{\Gamma}(y), \hat{\pi}))(5\text{cap}(\hat{\pi}\widehat{\Gamma}(z), \hat{\pi})$, $5\text{cap}(\hat{\pi}\hat{\Gamma}(x), \hat{\pi})) = 1$. It is not hard to see that $(\hat{\pi}\widehat{\Gamma}(z), \hat{\pi}\widehat{\Gamma}(x))(\hat{\pi}\widehat{\Gamma}(z), \hat{\pi}\widehat{\Gamma}(y)) = (\hat{\pi}\widehat{\Gamma}(x), \hat{\pi}\widehat{\Gamma}(y))(\hat{\pi}\widehat{\Gamma}(z), \hat{\pi}\widehat{\Gamma}(x))$. Thus, τ_i can be rewritten as $\tau_i = \alpha_2\alpha_1$, where $\alpha_1 = (5\text{cap}(\hat{\pi}\widehat{\Gamma}(z), \hat{\pi})$, $5\text{cap}(z, \hat{\pi}))(x, z)$, $5\text{cap}(z, \hat{\pi}))(x, z)$ and $\alpha_2 = (5\text{cap}(\hat{\pi}\widehat{\Gamma}(z), \hat{\pi})$, $5\text{cap}(z, \hat{\pi}))(y, z)$, $5\text{cap}(z, \hat{\pi}))(y, z)$. It can be verified that $\alpha_2 = (5\text{cap}(\alpha_1\hat{\pi}\widehat{\Gamma}(z), \alpha_1\hat{\pi})$, $5\text{cap}(\alpha_1\hat{\pi}\widehat{\Gamma}(y), \alpha_1\hat{\pi}))(\alpha_1\hat{\pi}\widehat{\Gamma}(z), \alpha_1\hat{\pi}\widehat{\Gamma}(y))(5\text{cap}(z, \alpha_1\hat{\pi})$, $5\text{cap}(y, \alpha_1\hat{\pi}))(y, z)$. By Lemma 5, as well as the previous discussion, it can be realized that a_1 acts on $\hat{\pi}$ as a fusion of π and α_2 continues to act on $\alpha_1\hat{\pi}$ as a cap exchange. As a result, the rearrangement effect of acting τ_i on $\hat{\pi}$ is still equivalent to a fusion acting on π. The above discussion indicates that a fusion to π can be mimicked by a translocation τ, which acts on $\hat{\pi}$ as a fusion of π, followed by zero or more translocations acting on $\tau\hat{\pi}$ as cap exchanges.

In the following, we prove the correctness of Algorithm 1. Initially, it is not hard to see that all the 5' caps are fixed in $\hat{\sigma}\hat{\pi}^{-1}$ and $\text{char}(x, \hat{\pi}) \neq \text{N3}$ for all $x \in \widehat{E}$. For any element $x \in \widehat{E}$ with $\text{char}(x, \hat{\pi}_i) = \text{T}$, where $1 \leq i \leq m$, if $\hat{\pi}^{-1}(x) \neq \hat{\sigma}_1^{+}[1]$ and $\hat{\pi}^{-1}(x) \neq \hat{\sigma}_1^{-}[1]$, that is, the 5' cap of $\hat{\pi}_i$ is not equal to that of $\hat{\sigma}_1$, then the character of $\hat{\sigma}\hat{\pi}^{-1}(x)$ in $\hat{\pi}$ must be C3. If any cycle in $\hat{\sigma}\hat{\pi}^{-1}$ contains any two elements x and y with the same character (either T or C3) in $\hat{\pi}$, then we can extract two 2-cycles $c_1 = (x, y)$ and $c'_1 = (\hat{\pi}\widehat{\Gamma}(y), \hat{\pi}\widehat{\Gamma}(x))$ from two mate cycles in $\hat{\sigma}\hat{\pi}^{-1}$ and multiply $c'_2c'_1c_2c_1$ with $\hat{\pi}$ to exchange the caps of the contigs containing x and y, respectively, in $\hat{\pi}$, where $c_2 = (5\text{cap}(x, \hat{\pi}), 5\text{cap}(y, \hat{\pi}))$ and $c_{2'} = (\hat{\pi}\widehat{\Gamma}(5\text{cap}(y, \hat{\pi})), \hat{\pi}\widehat{\Gamma}(5\text{cap}(x, \hat{\pi})))$. This is the job to be performed in the step 3 in Algorithm 1. Moreover, after finishing the cap exchanges in the step 3, each cycle in the remaining $\hat{\sigma}\hat{\pi}^{-1}$ has at most one element with T character and at most one element with C3 character. In other words, after running the step 3, there are at least 2(m-1) cycles in the resulting $\hat{\sigma}\hat{\pi}^{-1}$ such that each such a cycle contains exactly one element, say x, with $(x, \hat{\pi}) = \text{T}$ and exactly one element, say y, with $\text{char}(y, \hat{\pi}) = \text{C3}$, and $\hat{\sigma}\hat{\pi}^{-1}(x) = y$. In this case, we can further derive 2(m - 1) 2-cycles from these cycles in $\hat{\sigma}\hat{\pi}^{-1}$ with each 2-cycle having a character pair of (T, C3). Intriguingly, we shall show below that these 2(m-1) 2-cycles with character pair (T, C3), denoted by $f_1, f'_1, \ldots, f_{m-1}, f'_{m-1}$, can be used to obtain an optimal ordering of π such that the weighted reversal and block-interchange distance between the permutation induced by this ordering of π and σ is minimum. In fact, f_k and f'_k, where 1 ≤ k ≤ m - 1, are derived from two mate cycles in $\hat{\sigma}\hat{\pi}^{-1}$ and hence we call them as *mate* 2-cycles below. Moreover, if $f_k = (x, y)$, then $f'_k = (\hat{\pi}\widehat{\Gamma}(y), \hat{\pi}\widehat{\Gamma}(x))$.

For 1 ≤ k ≤ m - 1, we simply let $f_k = (x_k, y_k)$, where $\text{char}(x_k, \hat{\pi}) = \text{T}$ and $\text{char}(y_k, \hat{\pi}) = \text{C3}$. Then $f'_k = (\hat{\pi}\widehat{\Gamma}(y_k), \hat{\pi}\widehat{\Gamma}(x_k))$. As mentioned previously, the permutation induced by an ordering of π can be mimicked by performing m - 1 consecutive fusions on π that has m contigs initially. According to Lemma 5 and our previous discussion, if $f_k \nmid \hat{\pi}$, where 1 ≤ k ≤ m - 1, then $g'_kf'_kg_kf_k$ can be applied to $\hat{\pi}$ to function as a fusion of two contigs in π, where $g_k = (5\text{cap}(x_k, \hat{\pi}), 5\text{cap}(y_k, \hat{\pi}))$ and $g'_k = (\hat{\pi}\widehat{\Gamma}(5\text{cap}(y, \hat{\pi})), \hat{\pi}\widehat{\Gamma}(5\text{cap}(x, \hat{\pi})))$. Notice that g_k and g'_k are mate 2-cycles. However, not all $f_1, f_2, \ldots, f_{m-1}$ cannot divide $\hat{\pi}$. Suppose that only the first λ 2-cycles $f_1, f_2, \ldots, f_\lambda$ cannot divide $\hat{\pi}$, where 0 ≤ λ ≤ m - 1, that is, $f_k \nmid \hat{\pi}$ for 1 ≤ k ≤ λ, but $f_k|\hat{\pi}$ for λ + 1 ≤ k ≤ m - 1. In this situation, we shall show below that we still can use $f_1, f_2, \ldots, f_{m-1}$, as well as their mate 2-cycles, to derive an optimal ordering of π, as we did in the step 4 in Algorithm 1.

Recall that the 5' caps are all fixed in the beginning $\hat{\sigma}\hat{\pi}^{-1}$ (before the step 3 in Algorithm 1). As mentioned before, for any translocation used to perform on $\hat{\pi}$, it can be expressed as four 2-cycles, two with (non-C5, non-C5) character pair and the others with (C5, C5). It can be verified that during the process of the step 3, no two elements x and y with char $(x, \hat{\pi})$ = C5 but char $(y, \hat{\pi}) \neq \text{C5}$ can be found in a cycle of the $\hat{\sigma}\hat{\pi}^{-1}$[17], that is, C5 and non-C5 elements are not mixed together in the same cycle of $\hat{\sigma}\hat{\pi}^{-1}$. Actually, this property still continues to be asserted when we later perform any translocation on $\hat{\pi}$ to function as a fusion of π. Let us now pay attention on those cycles in $\hat{\sigma}\hat{\pi}^{-1}$ with only non-C5 elements and temporarily denote the composition of these cycles by $\phi(\hat{\sigma}\hat{\pi}^{-1})$. If we still can find any two elements x and y from a cycle in $\phi(\hat{\sigma}\hat{\pi}^{-1})$ such that $(\hat{\pi}\widehat{\Gamma}(5\text{cap}(y, \hat{\pi})), \hat{\pi}\widehat{\Gamma}(5\text{cap}(x, \hat{\pi})))(\hat{\pi}\widehat{\Gamma}(y), \hat{\pi}\widehat{\Gamma}(x))(5\text{cap}(x, \hat{\pi})$, $5\text{cap}(y, \hat{\pi}))(x, y)$ is an exchange of caps when applying it to $\hat{\pi}$, then we apply this cap exchange to $\hat{\pi}$ until we cannot find any one from $\phi(\hat{\sigma}\hat{\pi}^{-1})$. Finally, we denote such a $\phi(\hat{\sigma}\hat{\pi}^{-1})$ without any cap exchange by $\psi(\hat{\sigma}\hat{\pi}^{-1})$. Basically, $\psi(\hat{\sigma}\hat{\pi}^{-1})$ can be considered as a permutation of $E' = E \cup \{-c_{2i}, c_{2i+1} \ : \ 0 \leq i \leq m-1\}$ and hence its norm $||\psi(\hat{\sigma}\hat{\pi}^{-1})||$ is equal to $|E'| - \ n_c(\psi(\hat{\sigma}\hat{\pi}^{-1}))$ according to the formula we mentioned before.

Lemma 6 Let $\tau = (\hat{\pi}\widehat{\Gamma}(5\text{cap}(y, \hat{\pi}))$, $\hat{\pi}\widehat{\Gamma}(5\text{cap}(x, \hat{\pi})))(\hat{\pi}\widehat{\Gamma}(y), \hat{\pi}\widehat{\Gamma}(x))(5\text{cap}(x, \hat{\pi})$, $5\text{cap}(y, \hat{\pi}))(x, y)$ *be a fusion to act on* π, *where*

$\text{char}(x, \hat{\pi}) = T$ *and* $\text{char}(y, \hat{\pi}) = C3$. *Then* $||\psi(\hat{\sigma}\hat{\pi}^{-1})|| - ||\psi(\hat{\sigma}\hat{\pi}^{-1}\tau^{-1})|| \in \{-2, 0, 2\}$.

Proof. For simplicity, it is assumed that we cannot find any cap exchange from $\hat{\sigma}\hat{\pi}^{-1}$ to perform on $\hat{\pi}$. We then consider the following two cases.

Case 1: Suppose that $(x, y)|\hat{\sigma}\hat{\pi}^{-1}$, that is, both x and y lie in the same cycle, say α, in $\hat{\sigma}\hat{\pi}^{-1}$. Without loss of generality, let $\alpha = (a_1, a_2, \ldots, a_i \equiv x, \ldots, a_j \equiv y)$. Then α can be expressed as $\alpha = \alpha_1\alpha_2(x, y)$, where $\alpha_1 = (a_1, ..., a_i\})$ and $\alpha_2 = (\mathrm{a}_{i+1}, ..., a_j)$. Let α' denote the mate cycle of α in $\hat{\sigma}\hat{\pi}^{-1}$, that is, $\alpha' = (\hat{\pi}\widehat{\Gamma}(a_j), \ldots, \hat{\pi}\widehat{\Gamma}(a_i), \ldots, \hat{\pi}\widehat{\Gamma}(a_2), \hat{\pi}\widehat{\Gamma}(a_1))$. Then it can be expressed as $\alpha' = \alpha'_1\alpha'_2(\hat{\pi}\widehat{\Gamma}(y), \hat{\pi}\widehat{\Gamma}(x))$, where $\alpha'_1 = (\hat{\pi}\widehat{\Gamma}(a_{i-1}), \ldots, \hat{\pi}\widehat{\Gamma}(a_1), \hat{\pi}\widehat{\Gamma}(a_j))$ and $\alpha'_2 = (\hat{\pi}\widehat{\Gamma}(a_{j-1}), \hat{\pi}\widehat{\Gamma}(a_{j-2}), \ldots, \hat{\pi}\widehat{\Gamma}(a_i))$. Clearly, after applying τ to $\hat{\pi}$, the cycle α becomes two disjoint cycles α_1 and α_2 in $\hat{\sigma}\hat{\pi}^{-1}\tau^{-1}$ and α' becomes two disjoint α'_1 and α'_2. It means that $n_c(\psi(\hat{\sigma}\hat{\pi}^{-1}\tau^{-1})) = n_c(\psi(\hat{\sigma}\hat{\pi}^{-1})) + 2$ and hence $||\psi(\hat{\sigma}\hat{\pi}^{-1})|| - ||\psi(\hat{\sigma}\hat{\pi}^{-1}\tau^{-1})|| = 2$.

Case 2: Suppose that $(x, y) \nmid \hat{\sigma}\hat{\pi}^{-1}$, that is, x and y lie in two different cycles, say α_1 and α_2, in $\hat{\sigma}\hat{\pi}^{-1}$. In this case, $\hat{\pi}\widehat{\Gamma}(x)$ and $\hat{\pi}\widehat{\Gamma}(y)$ also are in two different cycles, say α'_1 and α'_2, that are the mate cycles of α_1 and α_2, respectively, in $\hat{\sigma}\hat{\pi}^{-1}$. By Lemma 4, char $(\hat{\pi}\widehat{\Gamma}(x), \hat{\pi}) = \mathrm{C3}$ and char $(\hat{\pi}\widehat{\Gamma}(y), \hat{\pi}) = \mathrm{T}$. Then performing τ on $\hat{\pi}$ leads α_1 and α_2 to be joined together into a cycle, say α, in $\hat{\sigma}\hat{\pi}^{-1}\tau^{-1}$ and α'_1 and α'_2 to be joined into another cycle, say α'. If α_1 and α_2, as well as α'_1 and α'_2, does not contain both T and C3 elements simultaneously, then $n_c(\psi(\hat{\sigma}\hat{\pi}^{-1}\tau^{-1})) = n_c(\psi(\hat{\sigma}\hat{\pi}^{-1})) - 2$ and hence $||\psi(\hat{\sigma}\hat{\pi}^{-1})|| - ||\psi(\hat{\sigma}\hat{\pi}^{-1}\tau^{-1})|| = -2$. If exactly one of α_1 and α_2, as well as exactly one of α'_1 and α'_2, contains both T and C3 elements simultaneously, then joining α_1 and α_2 will also change char $(x, \hat{\pi})$ from T to O and char $(y, \hat{\pi})$ from C3 to N3, and joining α'_1 and α'_2 will change char $(\hat{\pi}\widehat{\Gamma}(x), \hat{\pi})$ from C3 to N3 and char $(\hat{\pi}\widehat{\Gamma}(y), \hat{\pi})$ from T to O. Therefore, the cycle α, as well as α', contains a C3 (or T) element and an N3 element. In this case, we can use these four elements, along with their corresponding 5' caps in $\hat{\pi}$, as a cap exchange to perform on $\hat{\pi}$, resulting in that each of the cycles α and α' is divided into two smaller ones in new $\hat{\sigma}\hat{\pi}^{-1}$. As a result, $n_c(\psi(\hat{\sigma}\hat{\pi}^{-1}\tau^{-1})) = n_c(\psi(\hat{\sigma}\hat{\pi}^{-1}))$ and hence $||\psi(\hat{\sigma}\hat{\pi}^{-1})|| - ||\psi(\hat{\sigma}\hat{\pi}^{-1}\tau^{-1})|| = 0$. Suppose that both α_1 and α_2, as well as both α'_1 and α'_2, contain T and C3 elements at the same time. Then, after applying τ to $\hat{\pi}$, one of the above two T elements becomes an O element in new $\hat{\pi}$, leading to α, as well as α', containing only a T element, along with a C3 element and an N3 element. Next, we can use the T and N3 elements (or the C3 and N3 elements) in α and α' and their corresponding 5' caps in $\hat{\pi}$ to exchange the caps of $\hat{\pi}$. After that, α, as well as α', is divided into two cycles in the new $\hat{\sigma}\hat{\pi}^{-1}$ and, consequently, $n_c(\psi(\hat{\sigma}\hat{\pi}^{-1}\tau^{-1})) = n_c(\psi(\hat{\sigma}\hat{\pi}^{-1}))$ and hence $||\psi(\hat{\sigma}\hat{\pi}^{-1})|| - ||\psi(\hat{\sigma}\hat{\pi}^{-1}\tau^{-1})|| = 0$.

Notice that if $\hat{\pi} = \hat{\sigma}$, then $||\psi(\hat{\sigma}\hat{\pi}^{-1})|| = 0$. According to Lemmas 5 and 6, any translocation τ that acts on $\hat{\pi}$ as a fusion of π decreases the norm $||\psi(\hat{\sigma}\hat{\pi}^{-1})||$ at most by two. Hence, we call τ as a *good* fusion of π if $||\psi(\hat{\sigma}\hat{\pi}^{-1})|| - ||\psi(\hat{\sigma}\hat{\pi}^{-1}\tau^{-1})|| = 2$. By the discussion in the proof of Lemma 6, we have the following corollary.

Corollary 1 *Let* $\tau = (\hat{\pi}\widehat{\Gamma}(5\text{cap}(y, \hat{\pi})), \hat{\pi}\widehat{\Gamma}(5\text{cap}(x, \hat{\pi})))(\hat{\pi}\widehat{\Gamma}(y), 5\text{cap}(y, \hat{\pi}))(x, y), 5\text{cap}(y, \hat{\pi}))(x, y)$ *be a fusion to act on* π, *where* char $(x, \hat{\pi}) = T$ *and* char $(y, \hat{\pi}) = \mathrm{C3}$. *If* $(x, y)|\hat{\sigma}\hat{\pi}^{-1}$, *then* τ *is a good fusion to perform on* π.

According to Corollary 1, it can be realized that f_k, as well as its mate 2-cycle f'_k, can derive a good fusion to act on π, where $1 \leq k \leq \lambda$. If $\lambda = m - 1$, then performing the m - 1 fusions on π, as we did in Algorithm 1, corresponds to an optimal ordering of π such that the weighted reversal and block-interchange distance between the assembly of π and σ is minimum. If $\lambda < m$ - 1, then we show below that the fusions of m - 1 contigs in π performed by our algorithm utilizing $f_1, f_2, ..., f_{m-1}$ is still optimal.

Lemma 7 *Let* $\tau_1, \tau_2, \ldots, \tau_{m-1}$ *be any sequence of* m - 1 *translocations that act on* $\hat{\pi}$ *as fusions to assemble* m - 1 *contigs in* π. *Let* $\hat{\omega}_k$ *be the genome obtained by performing* τ_k *and zero or more following cap exchanges on* $\hat{\omega}_{k-1}$ *such that no more cap exchange can be derived from* $\hat{\sigma}\hat{\omega}_k^{-1}$, *where* $\hat{\omega}_0 = \hat{\pi}$ *and* $1 \leq k \leq m - 1$. *Then* $||\psi(\hat{\sigma}\hat{\omega}_0^{-1})|| - ||\psi(\hat{\sigma}\hat{\omega}_{m-1}^{-1})|| \leq 2\lambda$.

Proof. For simplicity, we assume that in the beginning, no cap exchange can be derived from $\hat{\sigma}\hat{\omega}_0^{-1}$ to act on $\hat{\omega}_0$. Let ω_k denote the genome obtained from $\hat{\omega}_k$ by removing its caps, where $1 \leq k \leq m$ - 1. By Lemma 6, $||\psi(\hat{\sigma}\hat{\omega}_{k-1}^{-1})|| - ||\psi(\hat{\sigma}\hat{\omega}_k^{-1})|| \in \{-2, 0, 2\}$ and by Corollary 1, $||\psi(\hat{\sigma}\hat{\omega}_{k-1}^{-1})|| - ||\psi(\hat{\sigma}\hat{\omega}_k^{-1})|| = 2$ if τ_k is a good fusion to ω_{k-1}. In fact, there are at most λ translocations from $\tau_1, \tau_2, \ldots, \tau_{m-1}$ that are good fusions. The reason is as follows. As mentioned before, we can obtain 2λ 2-cycles $f_1, f'_1, \ldots, f_\lambda, f'_\lambda$ from $\hat{\sigma}\hat{\pi}^{-1}$ that can derive λ good fusions to act on π, say $\tau_1, \tau_2, \ldots, \tau_\lambda$, as well as 2 ($m$ - λ - 1) other 2-cycles $f_{\lambda+1}, f'_{\lambda+1}, \ldots, f_{m-1}, f'_{m-1}$ that cannot derive any good fusions to act on π since their T and C3 elements lie in the same contig strand in $\hat{\pi}$. If we can further extract two 2-cycles, say f and its mate 2-cycle f, from $\hat{\sigma}\hat{\pi}^{-1}$ that can derive a good fusion, say

τ, to act on π, then the C3 elements in both f and f must locate at a contig whose T elements are in some f_k and f'_k, respectively, where $1 \leq k \leq \lambda$. This implies that the good fusion τ cannot act on $\hat{\pi}$ together with $\tau_1, \tau_2, \ldots, \tau_\lambda$ at the same time, since they will assemble a circular contig that is not allowed. Now, we suppose that $\tau_1, \tau_2, \ldots, \tau_{m-1}$ are the fusions obtained by the step 4 of Algorithm 1. Clearly, for $1 \leq k \leq \lambda$, $||\psi(\hat{\sigma}\hat{\omega}_{k-1}^{-1})|| - ||\psi(\hat{\sigma}\hat{\omega}_k^{-1})|| = 2$ since τ_k is a good fusion to ω_{k-1}. Moreover, for $\lambda + 1 \leq k \leq m - 1$, $||\psi(\hat{\sigma}\hat{\omega}_k^{-1})|| - ||\psi(\hat{\sigma}\hat{\omega}_{k-1}^{-1})|| = 0$, due to the following reason. According to Algorithm 1, we have $\tau_k = (\hat{\omega}_{k-1}\widehat{\Gamma}(z_k)$, $\hat{\omega}_{k-1}\widehat{\Gamma}(y_k))(y_k, z_k)(x_k, z_k)$, $\hat{\omega}_{k-1}\widehat{\Gamma}(y_k))(y_k, z_k)(x_k, z_k)$, which actually equals to $(\hat{\omega}_{k-1}\widehat{\Gamma}(x_k)$, $\hat{\omega}_{k-1}\widehat{\Gamma}(z_k)$, $\hat{\omega}_{k-1}\widehat{\Gamma}(y_k))(y_k, z_k, x_k)$. Moreover, we have $\psi(\hat{\sigma}\hat{\omega}_k^{-1}) = \psi(\hat{\sigma}\hat{\omega}_{k-1}^{-1})\tau_k^{-1}$, in which the composition of $(x_k, y_k)(y_k, z_k, x_k)^{-1}$ equals to (x_k, z_k) and the composition of $(\hat{\omega}_{k-1}\widehat{\Gamma}(y_k)$, $\hat{\omega}_{k-1}\widehat{\Gamma}(z_k)$, $\hat{\omega}_{k-1}\widehat{\Gamma}(z_k)$, $\hat{\omega}_{k-1}\widehat{\Gamma}(y_k))^{-1}$ equals to $(\hat{\omega}_{k-1}\widehat{\Gamma}(y_k), \hat{\omega}_{k-1}\widehat{\Gamma}(z_k))$. Recall that $f_k = (x_k, y_k)$ and $f_{k'} = (\hat{\omega}_{k-1}\widehat{\Gamma}(y), \hat{\omega}_{k-1}\widehat{\Gamma}(x))$, both of which are extracted from two mate cycles in $\psi(\hat{\sigma}\hat{\omega}_{k-1}^{-1})$. According to the above discussion, both y_k and $\hat{\pi}\widehat{\Gamma}(x_k)$ will be fixed in $\psi(\hat{\sigma}\hat{\omega}_k^{-1})$, thus increasing the number of cycles by two. However, the 2-cycle (x_k, z_k) will further join other two cycles respectively containing x_k and z_k together into one cycle and $(\hat{\pi}\widehat{\Gamma}(y_k), \hat{\pi}\widehat{\Gamma}(z_k))$ will join another two cycles respectively containing $\hat{\pi}\widehat{\Gamma}(y_k)$ and $\hat{\pi}\widehat{\Gamma}(z_k)$ together into one cycle, thus decreasing the number of cycles by two. As a result, $n_c(\psi(\hat{\sigma}\hat{\omega}_k^{-1})) = n_c(\psi(\hat{\sigma}\hat{\omega}_{k-1}^{-1}))$. Therefore, we have $||\psi(\hat{\sigma}\hat{\omega}_0^{-1})|| - ||\psi(\hat{\sigma}\hat{\omega}_{m-1}^{-1})|| \leq 2\lambda$ for the (m -1) fusions obtained by the step 4 of Algorithm 1. In fact, to let $||\psi(\hat{\sigma}\hat{\omega}_0^{-1})|| - ||\psi(\hat{\sigma}\hat{\omega}_{m-1}^{-1})|| > 2\lambda$ happen, there must be a translocation τ_i that acts on $\hat{\omega}_{i-1}$ as a fusion of ω_{i-1} satisfying either (1) $||\psi(\hat{\sigma}\hat{\omega}_{i-1}^{-1})|| - ||\psi(\hat{\sigma}\hat{\omega}_i^{-1})|| = 0$, the number of good fusions newly created by τ_i and its following cap exchanges minus that of good fusions currently destroyed by τ_i and the following cap exchanges is greater than or equal to one, and the total available good fusions can assemble more contigs than before, or (2) $||\psi(\hat{\sigma}\hat{\omega}_{i-1}^{-1})|| - ||\psi(\hat{\sigma}\hat{\omega}_i^{-1})|| = -2$, the number of good fusions created by τ_i and its following cap exchanges minus that of the currently destroyed good fusions is greater than or equal to two, and the total good fusions can assemble more contigs than before. However, we show below that no such a translocation τ_i exits. Let $\tau_i = (\hat{\omega}_{i-1}\widehat{\Gamma}(5\text{cap}(y, \hat{\omega}_{i-1}))$, $\hat{\omega}_{i-1}\widehat{\Gamma}(5\text{cap}(x, \hat{\omega}_{i-1})))(\hat{\omega}_{i-1}\widehat{\Gamma}(y), \hat{\omega}_{i-1}\widehat{\Gamma}(x))(5\text{cap}(x, \hat{\omega}_{i-1})$, $5\text{cap}(y, \hat{\omega}_{i-1}))(x, y)$ be a fusion (but not a good one) to ω_{i-1}, where char $(x, \hat{\omega}_{i-1}) = \text{T}$ and char $(y, \hat{\omega}_{i-1}) = \text{C3}$. According to Corollary 1, we have $(x, y) \nmid \hat{\sigma}\hat{\omega}_{i-1}^{-1}$, that is, x and y are in different cycles of $\hat{\sigma}\hat{\omega}_{i-1}^{-1}$. Moreover, char $(x, \tau_i\hat{\omega}_{i-1}) = \text{O}$ and char $(y, \tau_i\hat{\omega}_{i-1}) = \text{N3}$ after applying τ_i to $\hat{\omega}_{i-1}$. Below, we consider two cases.

Case 1: Suppose that there is a 2-cycle $f_j = (x_j, y_j)$ such that $x_j = x$, where $1 \leq j \leq m - 1$, char $(x_j, \hat{\omega}_{i-1}) = \text{T}$ and char $(y_j, \hat{\omega}_{i-1}) = \text{C3}$. For simplifying our discussion, we assume that f_j is disjoint from the other cycles in $\psi(\hat{\sigma}\hat{\omega}_{i-1}^{-1})$ and y is in the cycle $\alpha = (a_1, a_2, \ldots, a_h \equiv y)$ of $\psi(\hat{\sigma}\hat{\omega}_{i-1}^{-1})$. Then in $\psi(\hat{\sigma}\hat{\omega}_{i-1}^{-1})\tau_i^{-1}$, the cycles f_j and α are joined into a cycle $\beta = (a_1, a_2, \ldots, a_{h-1}, y, y_j, x)$, which can be expressed as $\gamma(y, y_j)$, where $\gamma = (a_1, a_2, \ldots, a_{h-1}, y, x)$, char $(y, \tau_i\hat{\omega}_{i-1}) = \text{N3}$ and char $(y_j, \tau_i\hat{\omega}_{i-1}) = \text{C3}$. According to Lemma 3, there is a cycle $\beta' = (\tau_i\hat{\omega}_{i-1}\widehat{\Gamma}) \cdot \beta^{-1}$. that is the mate cycle of β in $\psi(\hat{\sigma}\hat{\omega}_{i-1}^{-1})\tau_i^{-1}$. In other words, we can extract $c_1 = (y, y_j)$ from β and $c'_1 = (\tau_i\hat{\omega}_{i-1}\widehat{\Gamma}(y_j), \tau_i\hat{\omega}_{i-1}\widehat{\Gamma}(y))$ from β', and then apply $\tau'_i = c'_2c'_1c_2c_1$ to $\tau_i\hat{\omega}_{i-1}$ as a cap exchange, where $c_2 = (5\text{cap}(y, \tau_i\hat{\omega}_{i-1}), 5\text{cap}(y_j, \tau_i\hat{\omega}_{i-1}))$ and $\tau_i\hat{\omega}_{i-1}\widehat{\Gamma}(5\text{cap}(y, \tau_i\hat{\omega}_{i-1})))$, $\tau_i\hat{\omega}_{i-1}\widehat{\Gamma}(5\text{cap}(y, \tau_i\hat{\omega}_{i-1})))$, since the character pair (C3, N3) of (y_j, y) belongs to CEpair. After that, y_j, as well as $\tau_i\hat{\omega}_{i-1}\hat{\Gamma}(y)$, will be fixed in the resulting $\psi(\hat{\sigma}\hat{\omega}_i^{-1})$ and char $(y, \hat{\omega}_i)$ will become C3. As a result, $n_c(\psi(\hat{\sigma}\hat{\omega}_i^{-1})) = n_c(\psi(\hat{\sigma}\hat{\omega}_{i-1}^{-1}))$ and hence $||\psi(\hat{\sigma}\hat{\omega}_{i-1}^{-1})|| - ||\psi(\hat{\sigma}\hat{\omega}_i^{-1})|| = 0$. According to the above discussion, if $j \leq \lambda$, that is, f_j can be used to derive a good fusion to ω_{i-1}, then this good fusion will be destroyed when we perform $\tau'_i\tau_i$ on $\hat{\omega}_{i-1}$. On the other hand, if char $(a_{h-1}, \hat{\omega}_{i-1}) = \text{T}$ and $(a_{h-1}, y) = f_k$ with $k \leq \lambda$, that is, f_k can also be used to derive a good fusion to ω_{i-1}, then this good fusion will be destroyed after applying $\tau'_i\tau_i$ to $\hat{\omega}_{i-1}$, since f_k will become a 2-cycle with character pair of (T, N3) in the resulting $\hat{\omega}_i$. Based on the above discussion, the number of good fusions newly created by τ_i and τ'_i minus that of good fusions currently destroyed by τ_i and τ'_i must be less than or equal to zero.

Case 2: Suppose that there is no $f_j = (x_j, y_j)$ such that $x_j = x$, where $1 \leq j \leq m - 1$, char $(x_j, \hat{\omega}_{i-1}) = \text{T}$ and char $(y_j, \hat{\omega}_{i-1}) = \text{C3}$. Let α_1 denote the cycle containing x and α_2 denote the cycle containing y in $\hat{\sigma}\hat{\omega}_{i-1}^{-1}$. Also let α'_1 and α'_2 be the mate cycles of α_1 and α_2, respectively, in $\hat{\sigma}\hat{\omega}_{i-1}^{-1}$. Note that after applying τ_i to $\hat{\omega}_{i-1}$, the cycles α_1 and α_2 will be merged into a single cycle, say α, in $\hat{\sigma}\hat{\omega}_{i-1}^{-1}\tau_i^{-1}$ and α'_1 and α'_2 will be merged into a single cycle, say α'. Moreover, the characters of x and y

in $\tau_i\hat{\omega}_{i-1}$ will become O and N3, respectively. As discussed in the proof of Lemma 6, if both α_1 and α_2, as well as both α_1' and α_2', do not contain T and C3 elements simultaneously, then $||\psi(\hat{\sigma}\hat{\omega}_{i-1}^{-1})|| - ||\psi(\hat{\sigma}\hat{\omega}_i^{-1})|| = -2$. In this case, it can be verified that no existing good fusion is destroyed by τ_i and no new good fusion is created by τ_i. In other words, the number of the increased good fusions minus that of the destroyed good fusions is zero. If at least one of α_1 and α_2, as well as at least one of α_1' and α_2', has both T and C3 elements at the same time, then $||\psi(\hat{\sigma}\hat{\omega}_{i-1}^{-1})|| - ||\psi(\hat{\sigma}\hat{\omega}_i^{-1})|| = 0$ according to the discussion in the proof of Lemma 6. Now suppose that α_1 has no C3 element. Then T and C3 elements in α_2 can form a 2-cycle that equals to some $f_k = (x_k, y_k)$, where $1 \leq k \leq m-1$ and $y_k = y$. After applying τ_i to $\hat{\omega}_{i-1}$, the T element x from α_1 becomes an O element in α and the C3 element y from α_2 becomes an N3 element in α. We can continue to extract (x_k, y), which is now a 2-cycle of (T, N3), from α and act $\tau_i' = c_2'c_1'c_2c_1$ on $\tau_i\hat{\omega}_{i-1}$ as a cap exchange, where $c_2 = (5\text{cap}(x_k, \tau_i\hat{\omega}_{i-1})$, $c_2 = (5\text{cap}(x_k, \tau_i\hat{\omega}_{i-1}), 5\text{cap}(y, \tau_i\hat{\omega}_{i-1}))$, $c_1' = (\tau_i\hat{\omega}_{i-1}\hat{\Gamma}(y), \tau_i\hat{\omega}_{i-1}\hat{\Gamma}(x_k))$ and $c_2' = (\tau_i\hat{\omega}_{i-1}\hat{\Gamma}(5\text{cap}(y, \tau_i\hat{\omega}_{i-1}))$, $\tau_i\hat{\omega}_{i-1}\hat{\Gamma}(5\text{cap}(x_k, \tau_i\hat{\omega}_{i-1})))$. Clearly, no new good fusion is created in this case and one existing good fusion derived by f_k will be destroyed if $k \leq \lambda$. Therefore, the number of the increased good fusions minus that of the destroyed good fusions is less than or equal to zero. Suppose that α_1 contains both T and C3 elements, where we denote the C3 element by z for convenience. Then x and z can form a 2-cycle of (T, C3) pair, which can derive a good fusion $\tau = c_2'c_1'c_2c_1$ to $\hat{\omega}_{i-1}$ if $(x, z) \nmid \hat{\omega}_{i-1}$, where $c_2 = (5\text{cap}(x, \hat{\omega}_{i-1})$, $c_2 = (5\text{cap}(x, \hat{\omega}_{i-1}), 5\text{cap}(z, \hat{\omega}_{i-1}))$, $c_1' = (\hat{\omega}_{i-1}\hat{\Gamma}(z), \hat{\omega}_{i-1}\hat{\Gamma}(x))$ and $c_2' = (\hat{\omega}_{i-1}\hat{\Gamma}(5\text{cap}(z, \hat{\omega}_{i-1}))$, $\hat{\omega}_{i-1}\hat{\Gamma}(5\text{cap}(x, \hat{\omega}_{i-1})))$. If $(x, z) \nmid \hat{\omega}_{i-1}$, then, as mentioned previously, τ cannot work together with λ other good fusions derived by $f_1, f_2, \ldots, f_\lambda$ at the same time, since they will assemble a circular contig that is not allowed. For the case in which α_2 contains no T element, it is not hard to see that no new good fusion will be created and no existing good fusion will be destroyed when performing τ_i and its following cap exchange on ω_{i-1}, resulting in that the number of the created good fusions minus that of the destroyed good fusions is zero. We now assume that α_2 contains a T element, say w, and a C3 element y. Then w and y are adjacent in α_2 and (w, y) equals to some f_k, where $1 \leq k \leq m-1$. After applying τ_i to $\hat{\omega}_{i-1}, \alpha$ has a C3 element z, a T element w and an N3 element y. Then a 2-cycle (w, y) can be extracted from α such that $\tau_i' = c_2'c_1'c_2c_1$ can further perform on $\tau_i\hat{\omega}_{i-1}$ as a cap exchange, where $c_1 = (w, y)$, $c_2 = (5\text{cap}(w, \tau_i\hat{\omega}_{i-1})$, $\tau_i\hat{\omega}_{i-1}\hat{\Gamma}(5\text{cap}(w, \tau_i\hat{\omega}_{i-1})))$, $c_1' = (\tau_i\hat{\omega}_{i-1}\hat{\Gamma}(y), \tau_i\hat{\omega}_{i-1}\hat{\Gamma}(w))$ and $\tau_i\hat{\omega}_{i-1}\hat{\Gamma}(5\text{cap}(w, \tau_i\hat{\omega}_{i-1})))$, $\tau_i\hat{\omega}_{i-1}\hat{\Gamma}(5\text{cap}(w, \tau_i\hat{\omega}_{i-1})))$. Hence, if $k \leq \lambda$, then the good fusion derived by f_k will be destroyed by τ_i and τ_i'. However, the remaining α still contains a C3 element z and a T element w, which can derive a good fusion, say τ'. Hence, the number of the increased good fusions minus that of the destroyed good fusions is zero. On the other hand, if k > λ, then no exiting good fusion is destroyed by τ_i and τ_i'. In this case, the number of the increased good fusions minus that of the destroyed good fusions is equal to one. However, it can be verified that τ' cannot work with λ other good fusions derived by $f_1, f_2, \ldots, f_\lambda$, because they will produce a circular contig that is not allowed. In other words, no more contigs can be assembled after performing τ_i and τ_i' on $\hat{\omega}_{i-1}$.

According to the above discussion, we can conclude that $||\psi(\hat{\sigma}\hat{\omega}_0^{-1})|| - ||\psi(\hat{\sigma}\hat{\omega}_{m-1}^{-1})|| \leq 2\lambda$. □

Based on Lemma 7, as well as the discussion in its proof, the m - 1 fusions derived by Algorithm 1 correspond to an optimal ordering of π with an induced permutation *assembly* (π) such that the weighted rearrangement distance $\Delta(\pi, \sigma)$ between *assembly* (π) and σ is minimized. The obtained rearrangement distance $\Delta(\pi, \sigma)$ is calculated based on the algorithm in our previous study [17], and is equal to $\frac{||\hat{\sigma}\hat{\pi}^{-1}||}{2}$, where $\hat{\pi}$ is the genome obtained by performing the cap exchanges and m - 1 fusions on the initial capping of π, as done in the steps 3 and 4 in Algorithm 1, respectively. The total time complexity of Algorithm 1 is $\mathcal{O}(\delta n)$, where δ is the number of reversals and block-interchanges used to transform *assembly* (π) into σ. The reason is as follows. Since $m \leq n$, the cost of the step 1 for capping the input genomes is $\mathcal{O}(n)$. The computation of $\hat{\sigma}\hat{\pi}^{-1}$ in the step 2 still can be done in $\mathcal{O}(n)$ time. Recall that after running the step 3, each cycle in $\hat{\sigma}\hat{\pi}^{-1}$ has at most a T element and at most a C3 element. Totally, there are $2m$T elements and $2m$ C3 elements in the cycles of $\hat{\sigma}\hat{\pi}^{-1}$. Moreover, deriving two 2-cycles to serve as a cap exchange from two long mate cycles in $\hat{\sigma}\hat{\pi}^{-1}$ will divide these two long cycles into four smaller cycles. Hence, there are $\mathcal{O}(n)$ cap exchanges to be performed in the step 3, which totally cost $\mathcal{O}(n)$ time since each cap exchange needs only constant time. The step 4 assembles m contigs by utilizing 2(m - 1) 2-cycle $f_1, f_1', \ldots, f_{m-1}, f_{m-1}'$, which can be derived in advance from $\hat{\sigma}\hat{\pi}^{-1}$ in $\mathcal{O}(n)$ time. Since each fusion requires only constant time, the cost of the step 4 is $\mathcal{O}(m+n)$, which is equal to $\mathcal{O}(n)$. As to the steps 5 and 6, they can be done in $\mathcal{O}(\delta n)$ time in total, since there are totally δ iterations to find the reversals and block-interchanges and the time complexity of each iteration is dominated by the cost of finding a reversal or block-interchange that is $\mathcal{O}(n)$ time. Notice that although Algorithm 1 we described above is dedicated to linear, uni-chromosomal genomes, it can still be applied to circular, uni-chromosomal genomes, or to multi-chromosomal genomes with linear or circular chromosomes in a way of chromosome by chromosome.

Theorem 1 *Given a partially assembled genome π and a completely assembled genome σ, the one-sided block ordering problem can be solved in $O(\delta n)$ time and the weighted rearrangement distance between the permutation assembly(π) induced by the optimal ordering of π and σ is $\frac{||\hat{\sigma}\hat{\pi}^{-1}||}{2}$ that can be computed in $O(n)$ time, where $\hat{\pi}$ is the capping genome of π with the cap exchanges and m - 1 fusions being done, $\hat{\sigma}$ is the capping genome of σ, n is the number of genes or markers, and δ is the number of reversals and block-interchanges used to transform assembly(π) into σ.*

As mentioned in the introduction, any algorithm to solve the one-sided block ordering problem can be used to assemble (i.e., order and orient) the contigs in a draft genome based on a reference genome, if we denote this draft genome as π and use the reference genome as σ. For this application, our Algorithm 1 can finish its job just in $\mathcal{O}(n)$ time, because it does not need to do the steps 5 and 6 in this situation.

Experimental results

We have implemented Algorithm 1 as mentioned in the previous section into a program and also compared its accuracy performance to SIS on assembling the contigs of partially assembled genomes using some simulated datasets of linear, uni-chromosomal genomes. For this purpose, we compared the permutation induced by an assembly algorithm for a partially assembled genome

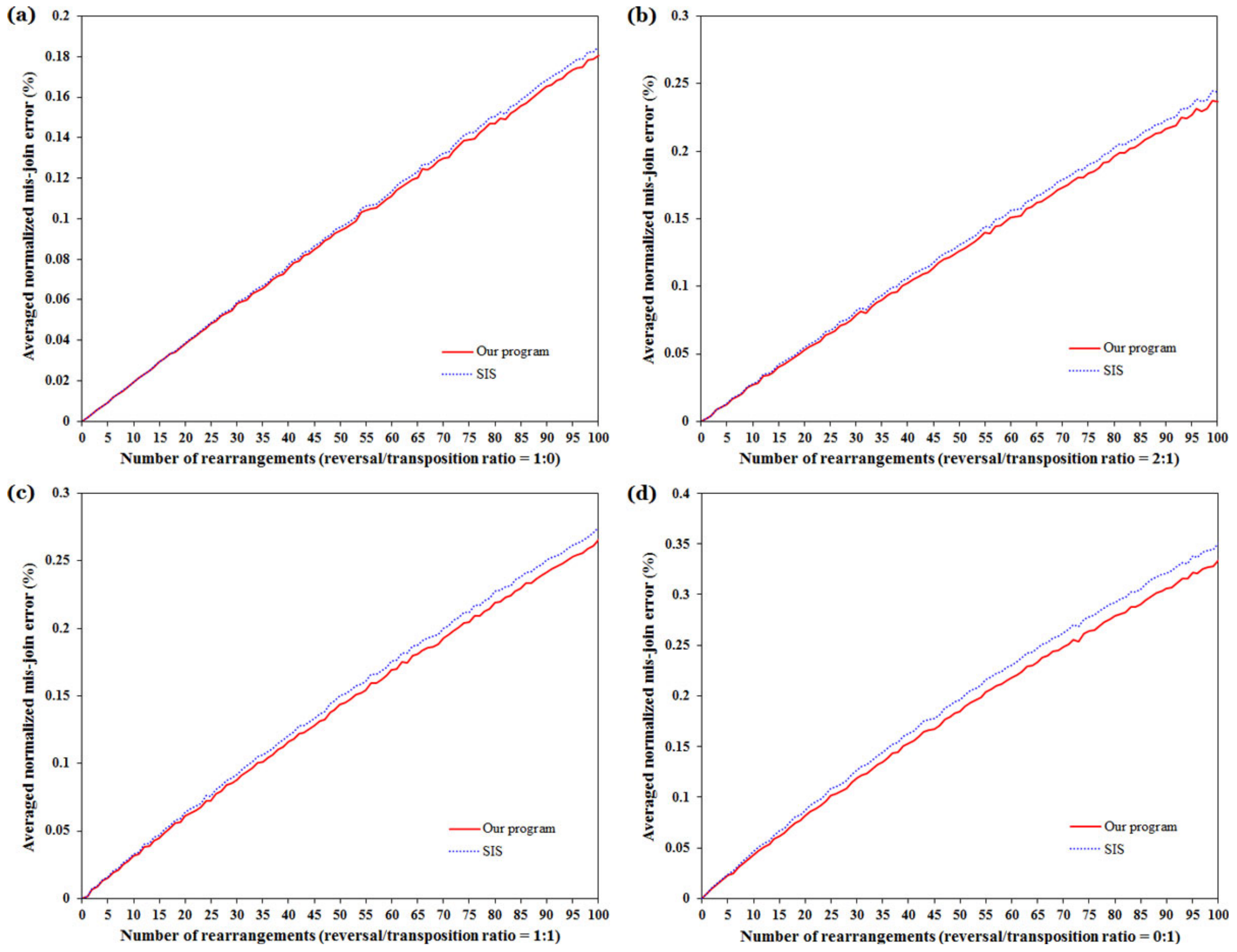

Figure 1 Comparison of accuracy performance between our program and SIS on simulated datasets with different ratios of the involved reversals and transpositions.

with its actual permutation by counting the number of breakpoints between them, where each breakpoint corresponds to an error of incorrectly joining two contigs (i.e., a mis-join error) caused by the assembly algorithm. This breakpoint number is then normalized by the number of contigs minus ρ to represent a fraction of incorrect contig joins, where $\rho = 1$ if the chromosome is linear; otherwise, $\rho = 0$. Each of partially assembled genomes with single linear chromosome in our simulated datasets was prepared and tested as follows. First, we generated the reference genome $\sigma = (1, 2, ..., n)$ with a linear chromosome of n genes, where n varies from 50 to 1000 with in the step of 50, and performed δ random rearrangement events (reversals and/or transpositions) on s to obtain a permutation of a linear, uni-chromosomal genome π', where δ varies from zero to 100 in the step of 1. Among the δ rearrangement events in our simulations, we used four different occurrence ratios to randomly generate reversals and transpositions: (1) 1:0, (2) 2:1, (3) 1:1 and (4) 0:1. Next, the genome π' is randomly fragmented into m contigs of various sizes to simulate the partially assembled genome π, where m varies from 50 to 500 with step 50. Finally, for each choice of n, m, δ and reversal/transposition ratio, we repeated the experiments 10 times and compared our program with SIS using their averaged normalized mis-join errors. As shown in Figure 1, the averaged normalized contig mis-join errors of our program are lower than those of SIS for all simulated datasets when the number of the involved reversals and transpositions is increased. In particular, if there are more transpositions involved in the rearrangement events, then the gap of accuracy performance between our program and SIS is increasing. The main reason may be due to the fact that our program can deal with both reversals and block-interchanges (including transpositions as a special case), while SIS considers only reversals without taking into account transpositions.

Conclusions

In this study, we introduced and studied the one-sided block/contig problem with optimizing the weighted reversal and block-interchange distance, which particularly has a useful application in genome resequencing. We finally designed an efficient algorithm to solve this problem in $\mathcal{O}(\delta n)$ time, where n is the number of genes or markers and d is the number of used reversals and block-interchanges. In addition, we showed that the assembly of the partially assembled genome can be done in $\mathcal{O}(n)$ time and its weighted rearrangement distance from the completely assembled genome can be calculated in advance in $\mathcal{O}(n)$ time. Finally, our simulation results showed that the accuracy performance of our program is better than that of the currently existing tool SIS when the number of the involved reversals and transpositions is increased. Moreover, the gap of this accuracy performance between our program and SIS is increasing, if there are more transpositions involved in the rearrangement events.

Authors' contributions

Corresponding author CLL conceived of the study, designed and analyzed the algorithm, and drafted the manuscript. The other authors CLL and KTC participated in the development of the program, as well as in the simulated experiments and their result discussion. The authors wish it to be known that the first two authors CLL and KTC contributed equally to this work and should be considered co-first authors. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work was supported in part by National Science Council of Republic of China under grant NSC100-2221-E-007-129-MY3.

Declarations

The publication costs for this article were funded by National Science Council of Republic of China under grant NSC100-2221-E-007-129-MY3. This article has been published as part of *BMC Bioinformatics* Volume 14 Supplement 5, 2013: Proceedings of the Third Annual RECOMB Satellite Workshop on Massively Parallel Sequencing (RECOMB-seq 2013). The full contents of the supplement are available online at http://www.biomedcentral.com/bmcbioinformatics/supplements/14/S5.

Published: 10 April 2013

References

1. Shendure J, Ji HL: **Next-generation DNA sequencing.** *Nature Biotechnology* 2008, **26**:1135-1145.
2. Mardis ER: **The impact of next-generation sequencing technology on genetics.** *Trends in Genetics* 2008, **24**:133-141.
3. Metzker ML: **Sequencing technologies - the next generation.** *Nature Reviews Genetics* 2010, **11**:31-46.
4. Fertin G, Labarre A, Rusu I, Tannier E, Vialette S: *Combinatorics of Genome Rearrangements* Cambridge, Massachusetts: The MIT Press; 2009.
5. Gaul E, Blanchette M: **Ordering partially assembled genomes using gene arrangements.** *Lecture Notes in Computer Science* 2006, **4205**:113-128.
6. Bourque G, Pevzner PA: **Genome-scale evolution: reconstructing gene orders in the ancestral species.** *Genome Research* 2002, **12**:26-36.
7. Hannenhalli S, Pevzner PA: **Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals.** *Journal of the ACM* 1999, **46**:1-27.
8. Bentley DR: **Whole-genome re-sequencing.** *Current Opinion in Genetics and Development* 2006, **16**:545-552.
9. Koboldt DC, Ding L, Mardis ER, Wilson RK: **Challenges of sequencing human genomes.** *Briefings in Bioinformatics* 2010, **11**:484-498.
10. van Hijum SAFT, Zomer AL, Kuipers OP, Kok J: **Projector 2: contig mapping for efficient gap-closure of prokaryotic genome sequence assemblies.** *Nucleic Acids Research* 2005, **33**:W560-W566.
11. Richter DC, Schuster SC, Huson DH: **OSLay: optimal syntenic layout of unfinished assemblies.** *Bioinformatics* 2007, **23**:1573-1579.
12. Assefa S, Keane TM, Otto TD, Newbold C, Berriman M: **ABACAS: algorithm-based automatic contiguation of assembled sequences.** *Bioinformatics* 2009, **25**:1968-1969.
13. Rissman AI, Mau B, Biehl BS, Darling AE, Glasner JD, Perna NT: **Reordering contigs of draft genomes using the Mauve Aligner.** *Bioinformatics* 2009, **25**:2071-2073.
14. Muñoz A, Zheng CF, Zhu QA, Albert VA, Rounsley S, Sankoff D: **Scaffold filling, contig fusion and comparative gene order inference.** *BMC Bioinformatics* 2010, **11**:304.

15. Husemann P, Stoye J: **r2cat: synteny plots and comparative assembly.** *Bioinformatics* 2010, **26**:570-571.
16. Dias Z, Dias U, Setubal JC: **SIS: a program to generate draft genome sequence scaffolds for prokaryotes.** *BMC Bioinformatics* 2012, **13**:96.
17. Huang YL, Lu CL: **Sorting by reversals, generalized transpositions, and translocations using permutation groups.** *Journal of Computational Biology* 2010, **17**:685-705.
18. Blanchette M, Kunisawa T, Sankoff D: **Parametric genome rearrangement.** *Gene* 1996, **172**:GC11-GC17.

doi:10.1186/1471-2105-14-S5-S9
Cite this article as: Li *et al.*: Assembling contigs in draft genomes using reversals and block-interchanges. *BMC Bioinformatics* 2013 **14**(Suppl 5):S9.