frontiers in Genetics

ORIGINAL RESEARCH
published: 24 September 2020
doi: 10.3389/fgene.2020.01035

A Path-Deformation Framework for Determining Weighted Genome Rearrangement Distance

Sangeeta Bhatia[1†], *Attila Egri-Nagy*[1†], *Stuart Serdoz*[1], *Cheryl E. Praeger*[2], *Volker Gebhardt*[1] *and Andrew Francis*[1*]

[1] *Centre for Research in Mathematics and Data Science, Western Sydney University, Sydney, NSW, Australia,* [2] *School of Physics, Mathematics, and Computing, University of Western Australia, Perth, WA, Australia*

OPEN ACCESS

Edited by:
Ruriko Yoshida,
Naval Postgraduate School,
United States

Reviewed by:
David Murrugarra,
University of Kentucky, United States
Xiaoxian Tang,
Beihang University, China

****Correspondence:***
Andrew Francis
a.francis@westernsydney.edu.au

†Present address:
Sangeeta Bhatia,
MRC Center for Global Infectious Disease Analysis, School of Public Health, Imperial College London, London, United Kingdom
Attila Egri-Nagy,
Akita International University, Akita, Japan

Specialty section:
This article was submitted to Evolutionary and Population Genetics, a section of the journal Frontiers in Genetics

Received: *30 May 2020*
Accepted: *11 August 2020*
Published: *24 September 2020*

Citation:
Bhatia S, Egri-Nagy A, Serdoz S, Praeger CE, Gebhardt V and Francis A (2020) A Path-Deformation Framework for Determining Weighted Genome Rearrangement Distance. Front. Genet. 11:1035. doi: 10.3389/fgene.2020.01035

Measuring the distance between two bacterial genomes under the inversion process is usually done by assuming all inversions to occur with equal probability. Recently, an approach to calculating inversion distance using group theory was introduced, and is effective for the model in which only very short inversions occur. In this paper, we show how to use the group-theoretic framework to establish minimal distance for any weighting on the set of inversions, generalizing previous approaches. To do this we use the theory of rewriting systems for groups, and exploit the Knuth–Bendix algorithm, the first time this theory has been introduced into genome rearrangement problems. The central idea of the approach is to use existing group theoretic methods to find an initial path between two genomes in genome space (for instance using only short inversions), and then to deform this path to optimality using a confluent system of rewriting rules generated by the Knuth–Bendix algorithm.

Keywords: genome rearrangement, inversion, group theory, Knuth-Bendix algorithm, rewriting systems

1. INTRODUCTION

Large scale changes in the arrangement of genes within a chromosome abound in biology and are key agents of sequence evolution (Belda et al., 2005; Beckmann et al., 2007). The differences in the order of genes along a chromosome were used as a phylogenetic marker as early as 1938 (Dobzhansky and Sturtevant, 1938) when Dobzhansky used them to determine different strains of *Drosophila melanogaster*. *Inversions* of chromosomal fragments are believed to be the main type of rearrangement event in bacterial genomes (Belda et al., 2005).

The first formalization of the problem of determining rearrangement distance between gene arrangements was done by Watterson et al. (1982). A number of methods have been proposed since then to determine the distance between genome arrangements in terms of a single rearrangement operator or a combination of rearrangement operators. In addition to inversions, researchers have considered translocations of chromosomal fragments (Bafna and Pevzner, 1995, 1998; Yin and Zhu, 2013), fission/fusion of chromosomes, duplication of sequences (Chaudhuri et al., 2006), deletion/insertion, and a combination of these different operators (Yancopoulos et al., 2005). These choices of allowable operations constitute *models of rearrangement*, in which the genomes in the data are assumed to only change according to specific rearrangement operators being considered.

The rearrangement distance between a pair of genomes is usually defined as the minimal number of events from the set of allowed operations required to transform one of the genomes in the pair into the other. For instance, in determining inversion distance between two genomes, the set of legal operations consists of all possible inversions on a gene sequence. Initial solutions in the case of inversions involve finding the smallest number of inversion events between two genomes and the distance was the count of the events. Thus, each inversion event carries the same weight. If the weight assigned to an inversion event represents the probability of that event, then a model where all events have the same weight can be thought of as finding distances under the uniform distribution. This model is used in the Hannenhalli and Pevzner (1999) approach, which draws a graph based on the genomes and calculates the minimal distance as a function of features of the graph (for example the number of cycles and paths). These methods are simple and fast and have been implemented in software for use by the research community (Sankoff et al., 1991; Tesler, 2002; Shao et al., 2014).

As pointed out above, an implicit assumption underlying most of these methods is that all rearrangement operators included in the model are equally probable and thus are given the same weight in the rearrangement distance. An inversion model that lies at the other extreme is one that allows only very short inversions. A group-theoretic model for sorting circular permutations using inversions acting on two adjacent regions was described by Egri-Nagy et al. (2014b). In a similar vein, Galvão et al. (2015) presented an approximation algorithm for sorting signed permutations by only length 2 reversals while Chen and Skiena (1996) gave a characterization of linear and circular permutations that can be sorted by only length k reversals, for a fixed k.

The biological evidence, however, points somewhere between these two extremes. For example, focusing only on the evidence related to inversions, several studies have suggested that inversions of a short chromosomal fragment are more frequent than that of longer fragments (Eisen et al., 2000; Seoighe et al., 2000; Lefebvre et al., 2003; Darling et al., 2008). Similarly, Seoighe et al. (2000) found a high prevalence of short inversions in the yeast genome. They observed that the conservation of a small neighborhood of genes, without absolute conservation of order or orientation, suggests that small DNA inversions have contributed significantly to the evolution of ascomycete genomes. In an analysis of four pairs of related bacterial genomes, Lefebvre et al. (2003) report an over-representation of short inversions, especially those involving a single gene, in comparison with a random inversion model. Analysis of the genome of *Y. Pestis* has also found that all inversions were shorter than expected under a neutral model (Darling et al., 2008).

In view of this information, a natural extension to the definition of rearrangement distance that allows for assigning weights (derived from empirical information) to the rearrangement operators, and calculates the minimal *weighted* distance between genome arrangements, might be a better approximation of the underlying biology. Thus, it would be useful to have a method to determine weighted inversion distance, where the use of an inversion operator can be penalized based on the number of regions it affects, or where the different operators in a model may be weighted based on type.

In fact, one of the first algorithms for determining rearrangement distance, proposed in Sankoff (1992) and Sankoff et al. (1992), is in principle capable of assigning weights for inversions and transpositions. An approximation algorithm for sorting a permutation under a particular class of length sensitive cost models, where the cost function is additive i.e., $f(x)+f(y) = f(x+y)$ was presented in Pinter and Skiena (2002). This approach has been generalized to a wider class of cost functions (Bender et al., 2008). This work also improved the bounds on the cost for sorting using an additive cost function. Further pursuing this line of inquiry, Swidan et al. (2004) extended the results for signed permutations as well as circular permutations.

In this paper, we present a flexible group-theoretic framework that can be used to determine the weighted rearrangement distance for any model of genome rearrangement in which the rearrangements allowed are invertible. Thus, the framework we propose is applicable to models involving inversions and translocations, but not, for instance, insertions and deletions. The present work is based on the group theoretic approach of Egri-Nagy et al. (2014b) and Francis (2014). Throughout the paper, we will focus on determining the minimal weighted reversal distance.

1.1. Overview of the Framework Introduced in This Paper

The central idea of the method we propose in this paper is "path deformation" in genome space. The genome space is the collection of all possible genomes. A path in the genome space is a sequence of genomes where consecutive elements are connected through a single rearrangement operator, and the weight of a path is the sum of the weights of the operators along the path. The minimal weighted distance between two genomes is then the minimal weight of all paths between the genomes.

To find the minimal weighted distance between two genomes, we start by constructing a path between them. At the same time, we have also constructed a library of rules in this space. These rules consist of alternate paths, or shortcuts, for a number of small paths. We scan the existing path for any subpaths that could be replaced by a shortcut from our library, generating a new, shorter path. In this way, the existing path is deformed into a new path which is of lower weight than the original path (although it might still not be the least weighted path). Successive iterations of the deformation step should ideally lead us to an optimal path (this is guaranteed only in certain circumstances described below).

The library of "rewriting rules" in itself is easy to generate, given a group defined by a presentation (generators and relations, defined in section 3). The relations, together with the weighting functions, can be transformed to give a set of rewriting rules. It is also not too difficult to construct an initial path between the two genomes which can be edited using the rules in the library, at least for some models of genome rearrangement. However it is not clear at the outset in what sequence to apply the rules in such a way that one is guaranteed to end with a minimal weight path from one genome to another. This is where the theory of rewriting systems is used.

A rewriting system that is guaranteed to produce a minimal expression, regardless of the sequence in which the rewriting rules are applied, is called a *confluent* rewriting system. In this paper, we use the *Knuth-Bendix algorithm* to transform our rewriting system into a confluent system and use it to construct a minimal weighted path between two genomes given an existing path between them (Knuth and Bendix, 1983).

The Knuth–Bendix algorithm is a heuristic whose termination may be affected by the ordering of the generators, which is not an intrinsic property of the input (a group presentation), but rather a choice made when applying the method. Thus, the input determines neither the result nor the running time of the algorithm, which means its complexity is not defined. On the other hand, a confluent rewriting system, once obtained, provides a simple algorithm that quickly finds a globally minimal weighted distance between two genomes.

Note that while this process obtains the global minimal weighted distance, and indeed a path that realizes this distance, the path itself is not necessarily unique: several distinct paths through the genome space may attain the globally minimal weighted distance (see Clark et al., 2019 for a detailed discussion of this).

We begin the paper by introducing the group based inversion model, and formalizing the notion of weighted distance in section 2. As a preliminary to the discussion about rewriting systems, we briefly discuss group presentations in section 3 and Cayley graphs (section 4). This is followed by a discussion of rewriting systems and their properties in section 5. In section 6, with two concrete examples of rewriting systems, we use weighted distances to draw some phlyogenies. We close off with a discussion of the strengths and limitations of the present work and some directions for future research (section 7).

2. GROUP THEORETIC INVERSION SYSTEMS

The notion of an *inversion system* was formalized in Egri-Nagy et al. (2014b). Since our work uses much of this language, we briefly summarize the key concepts in this section followed by an extension to a weighted inversion system.

2.1. Genomes as Permutations and Inversion as an Action

A chromosome is represented as a map from a set of positions $\mathbf{n} = \{1, 2, \ldots, n\}$ to a set of regions X, usually also labeled with the integers $\mathbf{n} = \{1, 2, \ldots, n\}$. If we denote the chromosome map by π, we can write the arrangement in two-line notation as:

$$\pi = \begin{pmatrix} 1 & 2 & \cdots & n \\ \pi_1 & \pi_2 & \cdots & \pi_n \end{pmatrix}.$$

where π_i is the region in the i'th position. The top row in this view represents the n positions on the chromosome and the bottom row represents the set of regions.

An unsigned inversion operator $t_{i,j}$ (with $1 \le i < j \le n$) in this paradigm is a map from positions to positions. When the genome is modeled as a map from positions to regions and a rearrangement operator is a bijection on the set of positions, we require that the rearrangement operator act first on a position and then we map the new position to a region using the genome map, and so the function composition is from left to right. For a detailed discussion of right and left actions (see Bhatia et al., 2018). The inversion operator $t_{i,j}$ maps π as follows:

$$\begin{aligned} t_{i,j}\pi &= t_{i,j}\begin{pmatrix} \cdots & i & i+1 & \cdots & j & \cdots \\ \cdots & \pi_i & \pi_{i+1} & \cdots & \pi_j & \cdots \end{pmatrix} \\ &= \begin{pmatrix} \cdots & i & i+1 & \cdots & j & \cdots \\ \cdots & \pi_j & \pi_{j-1} & \cdots & \pi_i & \cdots \end{pmatrix} \end{aligned}$$

Thus the inversion operator $t_{i,j}$ flipping regions in positions i to j can be written in cycle notation as follows:

$$t_{i,j} := \begin{cases} (i,j)(i+1,j-1)\ldots(\frac{i+j}{2}-1, \frac{i+j}{2}+1) & \text{if } j-i \text{ is even,} \\ (i,j)(i+1,j-1)\ldots(\frac{i+j-1}{2}, \frac{i+j+1}{2}) & \text{if } j-i \text{ is odd.} \end{cases}$$

For example, $t_{1,4} = (1,4)(2,3)$, $t_{1,5} = (1,5)(2,4)$, and $t_{1,6} = (1,6)(2,5)(3,4)$.

Given genomes π and π', and a sequence of k inversion operations $t_{i_1,j_1}, \ldots, t_{i_k,j_k}$ that transform π into π' when applied in order with t_{i_1,j_1} first, we write

$$t_{i_k,j_k} \cdots t_{i_1,j_1}\pi = \pi'.$$

Since π is a bijective map from the set of positions to regions, π^{-1} is well-defined and we can compose with π^{-1}, to give

$$t_{i_k,j_k} \cdots t_{i_1,j_1} = \pi'\pi^{-1}.$$

Now $\pi'\pi^{-1}$ is a bijective function from positions to positions and therefore an element of the symmetric group on n objects, S_n. Thus, the problem of determining a sequence of inversion operations that transforms π into π' is equivalent to the problem of expressing the group element $\pi'\pi^{-1}$ as a product of the group elements corresponding to the rearrangement operators.

2.2. Inversion Systems

An *inversion system* is defined as a tuple $(G, \mathcal{I})$ where G is the group of permutations and $\mathcal{I}$ is a set of inversions such that G is "generated by" $\mathcal{I}$, written $G = \langle \mathcal{I} \rangle$. In other words, every permutation in G is expressible as a product of elements of $\mathcal{I}$.

In general, if we have a subset $S \subset G$ of non-trivial elements from G, then a word over S is a finite sequence of elements of S. In this paper, we will assume that S is closed under the operation of taking inverses i.e., for all $s \in S$, we have $s^{-1} \in S$. We use S^* to represent the (infinite) set of all words over S.

If S generates G, then there is a natural map $\Gamma : S^* \to G$ that sends a word $w = [s_1, s_2, \cdots, s_k]$ to the group element $g = s_1 s_2 \cdots s_k$. The brackets in w are used to emphasize that a word is an ordered sequence of elements of S and to distinguish the sequence from the product $s_1 s_2 \cdots s_k$. The set S^* also contains the empty sequence which maps to the identity element of G. The length of a group element g with respect to the generating set S is the smallest $r \in \mathbb{N}$ such that there is some element $w \in S^*$, say $w = [s_1, \ldots, s_r]$, such that $\Gamma(w) = g$, that is,

$$s_1 s_2 \cdots s_r = g.$$

The inversion distance between permutations π_1 and π_2 is the length of the group element $\pi_2\pi_1^{-1}$ in the inversion system $(G, \mathcal{I})$. For details of inversion systems, the reader is referred to Egri-Nagy et al. (2014b).

2.3. Weighted Length

The notion of the length of a group element can be extended to the *weighted* length of a group element. Suppose the elements of S are assigned (positive) weights. The weighted length of a word $w = [s_1, s_2, \ldots, s_k]$ in S^* is the sum of the weights of the s_i where i runs from 1 through k. The weighted length of a group element g is obtained by taking an infimum over the set of all words in S^* that map to g.

Definition 2.1 (Weighted length)**.** *Let S be a set of generators of a group G. Let ω be a bounded function $\omega : S \to \mathbb{R}^+$. The weighted length of a (non-identity) group element $g \in G$ is defined as*

$$\ell_{S,\omega}(g) := \inf \left\{ \sum_{i=1}^{t} \omega(s_i) \;\middle|\; s_1 s_2 \cdots s_t = g, s_i \in S, t \in \mathbb{N} \right\}.$$

The weighted length of the identity element e of G is 0*.*

We define a *weighted inversion system* to be a 3-tuple $(G, \mathcal{I}, \omega)$ where $G = \langle \mathcal{I} \rangle$ as before and $\omega : \mathcal{I} \to \mathbb{R}^+$.

3. GROUP PRESENTATIONS

We will make use of the important notion of group presentations, from group theory. A *group presentation* is an abstract description of a group G in terms of a generating set S and set of relations $\mathcal{R}$ among the generators. Following Coxeter and Moser (1980, Chapter 1), these are defined as follows.

Definition 3.1 (Group Presentation)**.** *Let G be a group and let e be the identity element of G. A* presentation $\langle S \mid \mathcal{R} \rangle$ *for G consists of a generating set $S \subseteq G$ and a set of words $\mathcal{R} \subseteq S^*$ such that*

$$\Gamma(R_i) = e \quad \textit{for all } R_i \in \mathcal{R},$$

and for $w \in S^*$, if $\Gamma(w) = e$ then w is an algebraic consequence *of the words in $\mathcal{R}$ and the group axioms.*

That is, w is the same as the word we get by one or more of the following algebraic transformations : replace any occurrence of $R_i \in \mathcal{R}$ in w by the empty word; and replace any occurrence of gg^{-1} or $g^{-1}g$ in w by the empty word for any $g \in S$.

The elements of $\mathcal{R}$ are called relators. A group presentation may also be written as $\langle S \mid u_i = v_i, \quad i \in \mathbf{I} \rangle$ where $u_i, v_i \in S^*$ as before and $\mathbf{I}$ is an indexing set. An equation of the form $u = v$ in S^* is referred to as a *relation*. The relation $u = v$ is equivalent to the relator uv^{-1} as $u = v \iff uv^{-1} = e$ where v^{-1} is the inverse of v in S^*. It is worth noting at this point that both a relator and a relation can be thought of as an element of $S^* \times S^*$ as $(R_i, \emptyset)$ and (u, v), respectively. We make use of this formulation later in section 5.

A group G can have many different generating sets and consequently many presentations.

For example, a presentation for the symmetric group S_n with the generating set $S = \{s_i \mid s_i = (i, i+1), 1 \leq i < n\}$ consists of the relations:

$$\begin{aligned} s_i^2 &= e \quad \forall 1 \leq i < n \\ s_i s_j &= s_j s_i \quad \text{if } |i - j| > 1 \\ s_i s_{i+1} s_i &= s_{i+1} s_i s_{i+1} \quad 1 \leq i < n-1. \end{aligned}$$

This is known as the Coxeter presentation (Humphreys, 1992). In particular for S_4, with the generating set $\{s_1, s_2, s_3\}$, we have the following set of relations

$$\begin{array}{lll} (R1)\ s_1^2 = e & (R2)\ s_2^2 = e & (R3)\ s_3^2 = e \\ (R4)\ s_1s_2s_1 = s_2s_1s_2 & (R5)\ s_2s_3s_2 = s_3s_2s_3 & (R6)\ s_1s_3 = s_3s_1. \end{array}$$

The word $w = [s_2, s_3, s_2, s_1, s_3, s_1, s_2, s_3]$ satisfies $s_2s_3s_2s_1s_3s_1s_2s_3 = e$ in S_4, meaning $w = e$ is an algebraic consequence of the group axioms and the relations in the presentation of S_4. This can be seen by rewriting w using the relations in the presentation and the group axioms, for example as follows.

$$\begin{aligned} s_2s_3s_2s_1s_3s_1s_2s_3 &= s_2s_3s_2s_1s_3s_1(s_3s_3)s_2s_3 && [R3, ge = g] \\ &= s_2s_3s_2(s_1s_3s_1s_3)s_3s_2s_3 && [R6, ge = g] \\ &= s_2s_3s_2s_3s_2s_3 && [R5, R2 \text{ and } R3] \\ &= e. \end{aligned}$$

The above example suggests how the relations might be developed into a set of rewriting rules and the process of rewriting carried out in a systematic manner. In section 5, we will formalize the notion of such a rewriting system and discuss the properties that make a rewriting system effective.

4. WORDS ON A CAYLEY GRAPH

Another useful way to understand relations and rewriting of words in groups is through a Cayley graph. For a group G and a generating set $S \subseteq G$, the Cayley graph $\mathcal{C}(G, S)$ of G with respect to S is a directed graph that has a vertex for each element of G. There is an edge between vertices g and h if $gh^{-1} \in S$. That is, there is an edge labeled s between g and h if there is some $s \in S$ such that $sh = g$.

The labels on the edges in a path from vertex h to g in $\mathcal{C}(G, S)$ give a word w in S^* such that $\Gamma(w) = gh^{-1}$. Recall that Γ maps a word in S^* to an element in G. The length of a group element g is the length of a shortest path between the identity vertex e and g. If the edges of this graph are assigned weights, we can talk about the weighted path length between two vertices. In particular, a path in $\mathcal{C}(G, S)$ from a vertex g to itself concatenates to give a word w that represents e. Thus, relators from the group are represented by closed paths (loops) in the Cayley graph. Since the Cayley graph of a group is vertex-transitive, any node in the graph may be fixed as the identity vertex.

In the context of genome rearrangement models, a permutation is a genome arrangement. The generating set is the set of allowed rearrangements under this model. For instance, when inversions are considered to be the only allowed rearrangements, then the generating set is $S = \mathcal{I}$. The set of all genome arrangements is the genome space which corresponds to the vertex set of the Cayley graph $\mathcal{C}(G, \mathcal{I})$.

The process of rewriting words using relators is equivalent to deforming a path in a Cayley graph using loops to identify "shortcuts." As we have seen, a word in S^* that equals e can be rewritten using the relators in the group presentation and the

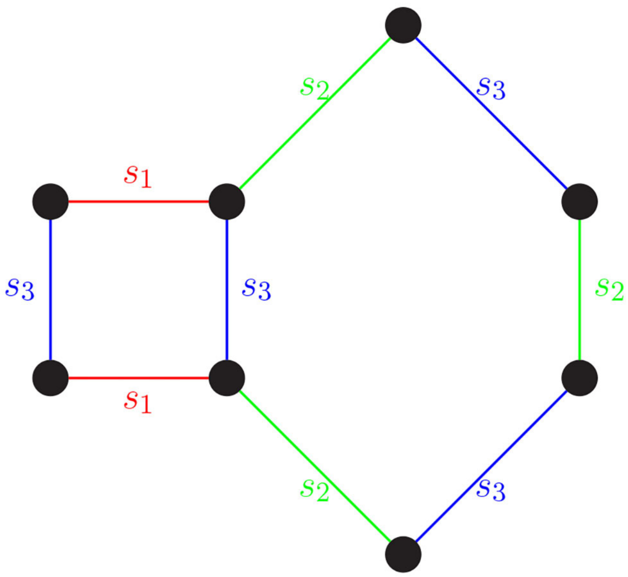

FIGURE 1 | A word in S^* that equals e can be rewritten using the relators in the group presentation and the group axioms. For instance, the word $s_1s_2s_3s_2s_3s_2s_1s_3 = e$ in S_4. Walking along an edge is equivalent to multiplying by a generator (edge label). Starting in the top left corner and tracing the word clockwise, we get the word $s_1s_2s_3s_2s_3s_2s_1s_3$. The closed path $s_1s_2s_3s_2s_3s_2s_1s_3$ is constructed using the relators $s_1s_3s_1s_3$ and $s_2s_3s_2s_3s_2s_3$.

group axioms. On a Cayley graph, this can be understood as a closed path being constructed using the closed paths in the presentation as building blocks (see **Figure 1**).

5. REWRITING SYSTEMS

As discussed earlier, a set of relators $\mathcal{R}$ is a subset of $S^* \times S^*$ and thus, $\mathcal{R}$ defines a binary relation on S^*. We write $\mathcal{R}^*$ for the reflexive transitive closure of $\mathcal{R}$. This relation is made compatible with the multiplication in S^* as follows:

> if $(l, r) \in \mathcal{R}$, then for words u, v in S^*, we say that
> *ulv rewrites to urv*.

If we impose the constraint that $\mathcal{R}$ be antisymmetric, (i.e., $(l, r) \in \mathcal{R} \implies (r, l) \notin \mathcal{R}$), then the reduction process becomes directional. In this case, $\mathcal{R}$ is referred to as a *rewriting system.* For $(l, r) \in \mathcal{R}$, we will write $l \rightarrow r$ and refer to l as the left side of the rule and r as its right side. If $(x, y) \in \mathcal{R}^*$, this means that x can be reduced to (rewritten as) y using the rules in $\mathcal{R}$. We will write this as $x \underset{\mathcal{R}^*}{\rightarrow} y$.

A word $w \in S^*$ is said to be *reducible* with respect to $\mathcal{R}$ if there is some $z \in S^*$ such that $w \underset{\mathcal{R}}{\rightarrow} z$. If no such z exists, then w is said to be *irreducible* with respect to $\mathcal{R}$.

$$\begin{tikzcd}
 & u \arrow[dl, "*"'] \arrow[dr, "*"] & \\
v \arrow[dr, "*"'] & & w \arrow[dl, "*"] \\
 & x &
\end{tikzcd}$$

FIGURE 2 | In a confluent rewriting system, if a word u can be reduced to the words v and w, then v and w can be reduced to some word x. Here, we use *as shorthand for R*.

In applying the rewriting rules to rewrite a word, one may have to make choices at each step. For instance, a word may contain the left sides of more than one rule in $\mathcal{R}$. For the process of rewriting to be effective, we need to ensure that a given word can be reduced to a unique irreducible word. In addition to this, an essential requirement is that this irreducible representative can be obtained by the application of rewriting rules in a finite number of steps. Formally, we talk about *confluence* and *termination* of a rewriting system.

Definition 5.1 (Terminating rewriting system)**.** *A rewriting system $\mathcal{R}$ over S^* is said to be* terminating *if there is no infinite sequence of words $w_i \in S^*$ such that $w_0 \rightarrow w_1 \rightarrow \dots w_k \dots$.*

Definition 5.2 (Confluence)**.** *A rewriting system $\mathcal{R}$ over S^* is said to be* confluent *if for all $u, v, w \in S^*$, if $u \underset{\mathcal{R}^*}{\rightarrow} v$ and $u \underset{\mathcal{R}^*}{\rightarrow} w$, then there exists $x \in S^*$ such that $v \underset{\mathcal{R}^*}{\rightarrow} x$ and $w \underset{\mathcal{R}^*}{\rightarrow} x$. (See **Figure 2**).*

A set of defining relations (or relators) in a presentation can be turned into a rewriting system. To ensure that the rewriting system thus created is terminating and confluent, we will need to do some more work.

5.1. Termination

The termination of a rewriting system $\mathcal{R}$ can be established by imposing a reduction order on the set S^*. A *reduction order* on S^* is a transitive relation $>$ such that for any $s, t \in S^*$,

- exactly one of the following holds: $s > t$, $t > s$ or $s = t$, and
- $s > t \implies asb > atb$, for all $a, b \in S^*$, and
- there is no infinite sequence of elements $s_0, s_1, \dots, s_i, \dots$ of S such that $s_i > s_{i+1}$.

The idea behind imposing $>$ on S^* is that if $u \rightarrow v \implies u > v$, then an infinite sequence of words w_i such that $w_i \rightarrow w_{i+1}$ induces an infinite decreasing sequence under $>$. Since the latter is not possible, $\mathcal{R}$ must terminate.

We now define a reduction order on S^*, using a weight function on S.

Definition 5.3 (Weighted Lexicographic Order)**.** *Let S be a non-empty finite set. Fix any ordering $\succ$ on the elements of S. Let $\omega : S \rightarrow \mathbb{R}^+$ be a function that assigns a positive weight to each element of S. Let $u = s_1s_2 \cdots s_k$ and $v = t_1t_2 \cdots t_l$ be in S^*. Define $u > v$ if either*

1. $\sum_{i=1}^{k} \omega(s_i) > \sum_{i=1}^{l} \omega(t_i)$, *or*
2. $\sum_{i=1}^{k} \omega(s_i) = \sum_{i=1}^{l} \omega(t_i)$ *and* $s_i \succ t_i$ *where* $i = min\{p : s_p \neq t_p\}$.

It is easy to see that the weighted lexicographic order is a reduction order.

Proposition 5.4. *For any finite set S, weighted lexicographic order is a reduction order on* S^*.

5.2. Confluence

If certain mathematical conditions are satisfied, a rewriting system can be transformed into a confluent rewriting system through a procedure due to Knuth and Bendix (1983). We will discuss the Knuth-Bendix algorithm and the properties necessary for it to return a terminating, confluent rewriting system later in this section after introducing the necessary definitions.

Definition 5.5 (Critical Pair). *Let* $\mathcal{R}$ *be a rewriting system over* S^*. *Let* $u_1a \rightarrow v_1$ *and* $au_2 \rightarrow v_2$ *be two rules in* $\mathcal{R}$ *where* $u_i, v_i, a \in S^*, a \neq e$. *That is, a non-empty suffix of the left hand side of a rule overlaps a prefix of the left hand side of another rule. Rules* $u_1a \rightarrow v_1$ *and* $au_2 \rightarrow v_2$ *are said to* overlap. *The word* $w = u_1au_2$ *can be reduced to both* v_1u_2 *and* u_1v_2. *The words* v_1u_2 *and* u_1v_2 *are said to constitute a* critical pair.

A critical pair (v_1u_2, u_1v_2) is said to be *resolved* if there exists $w \in S^*$ such that $v_1u_2 \underset{\mathcal{R}^*}{\rightarrow} w$ and $u_1v_2 \underset{\mathcal{R}^*}{\rightarrow} w$.

Theorem 5.6 ((Baader and Nipkow, 1999, Lemma 2.7.2)). *A terminating rewriting system is confluent if and only if all its critical pairs are resolved.*

The power of Theorem 5.6 derives from the fact that it allows us to ascertain the (global) confluence of a rewriting system by checking for confluence locally. This suggests a simple procedure for making a rewriting system confluent. Resolve each critical pair (u, v) by adding a rule $u \rightarrow v$ if $u > v$ and $v \rightarrow u$ otherwise. This is the gist of the Knuth-Bendix algorithm. However, we still need to ensure that this loop of adding a rule and checking if any critical pairs remain to be resolved will terminate. In fact, the Knuth-Bendix algorithm is guaranteed to terminate with a confluent and terminating rewriting system if the equivalence relation generated by $\mathcal{R}$ has finitely many equivalence classes (Holt et al., 2005, Corollary 12.21).

Definition 5.7 (Local Confluence). *Baader and Nipkow (1999) A rewriting system* $\mathcal{R}$ *over* S^* *is said to be* locally confluent *if for all* $u, v, w \in S^*$, *if* $u \underset{\mathcal{R}}{\rightarrow} v$ *and* $u \underset{\mathcal{R}}{\rightarrow} w$, *then there exists* $x \in S^*$ *such that* $v \underset{\mathcal{R}}{\rightarrow} x$ *and* $w \underset{\mathcal{R}}{\rightarrow} x$.

Note that local confluence differs from confluence (Definition 5.2) in that here, relations are from $\mathcal{R}$ rather than its reflexive transitive closure $\mathcal{R}^*$.

For a terminating and locally confluent rewriting system, each equivalence class under the closure of the relation generated by $\rightarrow$ contains a unique, irreducible element. Since each element of S^* maps to a group element, each unique, irreducible element maps to a unique group element. The number of equivalence classes in S^* must be finite if the group generated by S is finite. Thus, for a finite group, the Knuth-Bendix algorithm will give us the requisite set of rewriting rules.

The upshot of this observation is that for a genome rearrangement model, where the rearrangement operators are invertible, the Knuth-Bendix algorithm is guaranteed to generate a finite, confluent, terminating rewriting system since we are dealing with finite groups. The restriction that the operators be invertible is necessary to ensure that the operators generate a group.

In section 6, we construct rewriting systems for two different weighted inversion models and use them to find the weighted distance for genomes.

6. IMPLEMENTATION AND BIOLOGICAL EXAMPLES

The first model consists of unsigned permutations on a linear genome with 7 regions. The set of inversions $\mathcal{I}$ consists of all inversions $t_{i,j}$, for $1 \leq i < j \leq 7$, as defined in section 2. This set generates the symmetric group S_7. A simple monotonic weighting function ω is given by $\omega(t_{i,j}) := j - i$. For $1 \leq i_1 < j_1 \leq 7$ and $1 \leq i_2 < j_2 \leq 7$, let $m = m(i_1, j_1, i_2, j_2)$ be the smallest non-negative integer such that $(t_{i_1,j_1}t_{i_2,j_2})^m = 1$. A presentation for the group with this generating set is

$$G = \{\mathcal{I} \mid (t_{i_1,j_1}t_{i_2,j_2})^m,$$
$$m = m(i_1, j_1, i_2, j_2),\ 1 \leq i_1 < j_1 \leq 7,\ 1 \leq i_2 < j_2 \leq 7\}.$$

We used the software package KBMAG (Holt, 1995) to run Knuth-Bendix on this presentation. The resulting confluent rewriting system consists of 6,220 rules. KBMAG can also use the rewriting system to find a minimal representative for a given group element. The weighted distance for the group element is then defined to be the weight of the unique minimal representative.

We generated 4 random permutations in S_7 and determined the weighted distance matrix using the weight function ω, which was fed into RPhylip (Felsenstein, 1993; Revell and Chamberlain, 2014) to construct a phylogenetic tree (**Figure 3**). For the same permutations, we also constructed phylogenies with the distance matrix from GRIMM (Tesler, 2002) and the Coxeter distance matrix (also **Figure 3**). GRIMM assigns unit weight to all inversions. Coxeter generators are reversals of adjacent regions $(i, i+1)$, and so the inversion model underlying Coxeter distance assigns unit weight to reversals of adjacent regions and infinite weight to all other reversals.

The three topologies presented in **Figure 3** differ from each other in either the clustering of nodes or the branch lengths. An important point to be noted is that the weighted distance model results in the clustering $AC|BD$ while the uniform weight model (GRIMM) clusters the nodes as $AB|CD$. This difference is interesting since both the methods have the same set of inversions but different weights assigned to the generators.

Our second example deals with circular rather than linear genomes. In this case, we will use the method for returning

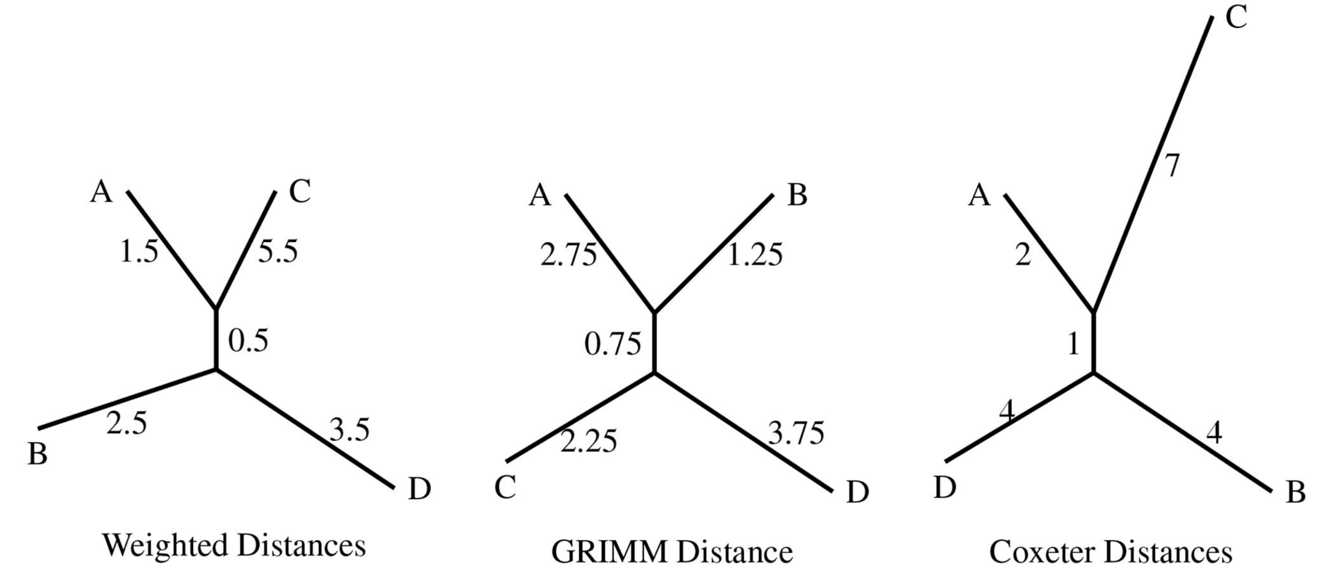

FIGURE 3 | The topologies produced with the distance matrices from the different distance algorithms as input. The four permutations are $A = (1, 4)(3, 7, 6)$, $B = (1, 3, 7, 5, 2, 6, 4)$, $C = (3, 4, 6)$, and $D = (1, 7, 6, 4, 2, 3, 5)$.

a minimal distance for adjacent inversions (Egri-Nagy et al., 2014b) to confirm the weighted distance methods in this paper are effective.

To construct the rewriting system, we used the same set of generators and relations as those in the circular inversion model presented in Egri-Nagy et al. (2014b). The generating set consists of the inversions of adjacent regions $(i, i+1)$ for $1 \leq i < n$ and the inversion $(1, n)$ that allows swap the positions n and 1. Following the notation in Egri-Nagy et al. (2014b), we denote these generators by s_i. The generating set is thus $\{s_i \mid i = 1, 2, \ldots n\}$ and the relations are:

$$\begin{aligned} s_i^2 &= e && \text{for each } i = 1, \ldots n, \\ s_i s_j &= s_j s_i && \text{if } i - j \bmod n \neq 1, \\ s_i s_{i+1} s_i &= s_{i+1} s_i s_{i+1} && \text{for each } i = 1, \ldots n-1, \textit{and} \\ s_n &= s_{n-1} s_{n-2} \ldots s_2 s_1 s_2 \cdots s_{n-2} s_{n-1}. \end{aligned}$$

All the generators are assigned unit weight as in the circular inversion model of Egri-Nagy et al. (2014b), which we will refer to as EGTF. We use this presentation as an input to KBMAG. The confluent rewriting system in this case has 6,622 rules. Once again, we generated 4 random permutations in the group and found the distances using KBMAG, GRIMM, and EGTF. The latter method has been implemented by the authors in the package BioGAP (Egri-Nagy et al., 2014a) for the software GAP (Schønert et al., 1997). Since EGTF and the method in this paper have the same generating set and the same weights, the phylogenies produced using the distance matrix from the rewriting system and that from the EGTF method should be identical, as indeed they are. The resulting phylogenies produced using RPhylip are presented in **Figure 4**.

The topologies produced by the distances derived from the rewriting system and EGTF model are the same as expected since both these methods give the exact minimal reversal distance between two circular permutations. Both the methods have been set up to factor in the rotational and reflection symmetry of a circular genome.

7. DISCUSSION AND FUTURE WORK

Researchers have recognized the need for methods to determine weighted distances in the field of genome rearrangement right from the start. Beginning with the pioneering work of Sankoff (1992) and Sankoff et al. (1992), a number of approaches have been tried. While they differ in the techniques employed, a common feature of the previous studies is that the proposed algorithms are tied to a particular model of rearrangement. The novelty of our work is that the framework presented here can be adapted to a wide variety of models. In addition, to the best of our knowledge, this work presents the first use of the theory of rewriting systems to a problem in comparative phylogenetics.

The current approach however has some limitations, which present opportunities for interesting research. For instance, the method presented here can only be used with invertible rearrangement operators. The use of other algebraic structures such as a semigroup might allow this restriction to be removed allowing more rearrangement models to be included.

Another important limitation is that the method works by distorting an existing path (in terms of the operators in the model) between two genomes into an optimal path. This is not a problem if the model includes all inversions, or all adjacent

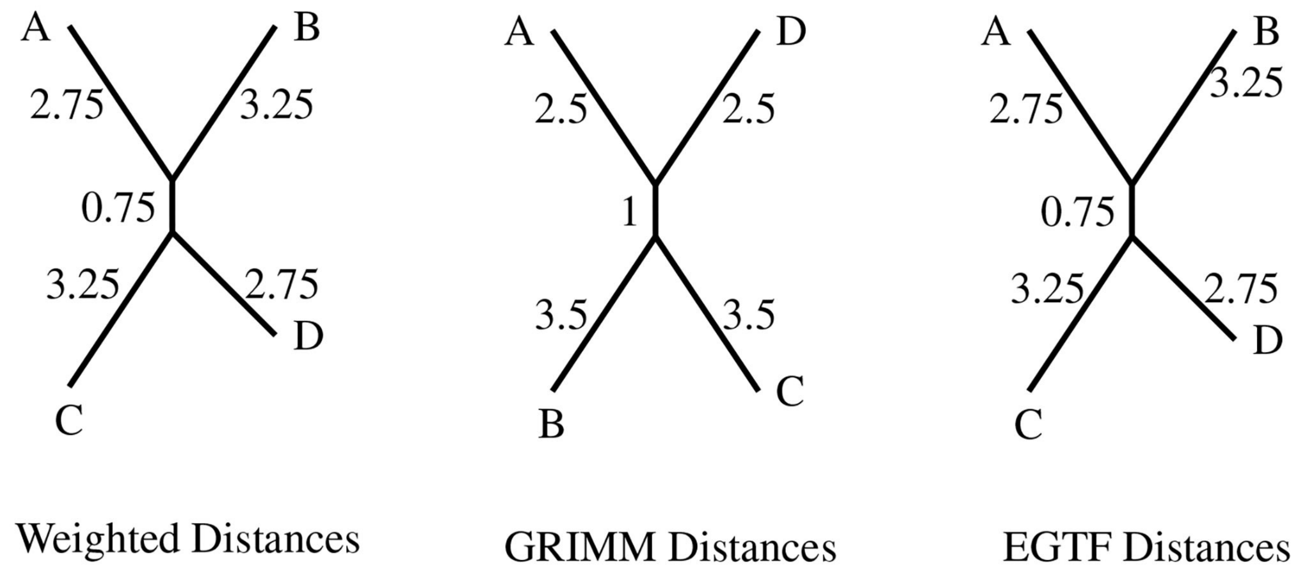

FIGURE 4 | The topologies produced for circular genomes on eight regions with the distance matrices from the different distance algorithms. The four permutations are $A = (2, 3)(6, 8)$, $B = (1, 7, 6, 8)(4, 5)$, $C = (1, 5, 6, 4, 3, 8, 7)$, and $D = (1, 2, 4)(5, 6, 7)$.

TABLE 1 | Growth in number of rules.

n	Rules
3	9
4	44
5	204
6	1049
7	6220

The size of a confluent rewriting system grows very quickly with the number of regions n. The rewriting system is for a weighted lexicographic order with weight of $t_{i,j}$ being $j - i$ for a linear genome and generators $(i, i + 1)$ for $i \in \{1, 2, \ldots, n - 1\}$.

inversions—in which case methods such as GRIMM and EGTF can provide an initial path. However, for some models, finding a path between two arbitrary genomes may be non-trivial.

Even in the case where such a path is known, for instance in the inversion model, the other deficiency at the moment is the lack of a software implementation. We have used KBMAG to derive the confluent rewriting systems. However, the use of KBMAG with finite groups is not recommended by the authors as it is optimized for infinite groups. It is not surprising therefore that for larger values of n, KBMAG cannot return a confluent rewriting system even though we know that a confluent system exists. The size of a confluent rewriting system increases very quickly with n (see **Table 1**). Thus an efficient software implementation of Knuth-Bendix optimized for finite groups and in particular for models arising from biology would be very useful.

The application of rewriting systems to a new problem also gives rise to new mathematical questions. For instance, it would be interesting to investigate the effect of the weighting function used on the size and efficiency of the rewriting system.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

SB, AF, and AE-N conceived the project. SB performed most of the analysis, including managing the code, and producing results. SB, AE-N, and SS wrote the code. CP, VG, and AE-N provided expertise and ideas on computational group theory and rewriting systems. SS performed some experiments required for the project. AF and SB led the writing of the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

AF and CP were partially supported during this research by Australian Research Council Discovery Projects DP180102215 and DP190100450, respectively.

REFERENCES

Baader, F., and Nipkow, T. (1999). *Term Rewriting and All That*. Cambridge: Cambridge University Press.

Bafna, V., and Pevzner, P. (1995). "Sorting permutations by tanspositions," in *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms* (Society for Industrial and Applied Mathematics), 614–623.

Bafna, V., and Pevzner, P. A. (1998). Sorting by transpositions. *SIAM J. Discr. Math.* 11, 224–240. doi: 10.1137/S089548019528280X

Beckmann, J. S., Estivill, X., and Antonarakis, S. E. (2007). Copy number variants and genetic traits: closer to the resolution of phenotypic to genotypic variability. *Nat. Rev. Genet.* 8:639. doi: 10.1038/nrg2149

Belda, E., Moya, A., and Silva, F. J. (2005). Genome rearrangement distances and gene order phylogeny in γ-proteobacteria. *Mol. Biol. Evol.* 22, 1456–1467. doi: 10.1093/molbev/msi134

Bender, M. A., Ge, D., He, S., Hu, H., Pinter, R. Y., Skiena, S., et al. (2008). Improved bounds on sorting by length-weighted reversals. *J. Comput. Syst. Sci.* 74, 744–774. doi: 10.1016/j.jcss.2007.08.008

Bhatia, S., Feijão, P., and Francis, A. R. (2018). Position and content paradigms in genome rearrangements: the wild and crazy world of permutations in genomics. *Bull. Math. Biol.* 80, 3227–3246. doi: 10.1007/s11538-018-0514-3

Chaudhuri, K., Chen, K., Mihaescu, R., and Rao, S. (2006). "On the tandem duplication-random loss model of genome rearrangement," in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm* (Miami, FL: Society for Industrial and Applied Mathematics), 564–570.

Chen, T., and Skiena, S. S. (1996). Sorting with fixed-length reversals. *Discr. Appl. Math.* 71, 269–295. doi: 10.1016/S0166-218X(96)00069-8

Clark, C., Egri-Nagy, A., Francis, A., Gebhardt, V. (2019). Bacterial phylogeny in the Cayley graph. *Discr. Math. Algorith. Appl.* 11:1950059. doi: 10.1142/S1793830919500599

Coxeter, H. S. M., and Moser, W. O. J. (1980). *Generators and Relations for Discrete Groups, Vol. 14*. Berlin: Springer Science & Business Media.

Darling, A. E., Miklos, I., and Ragan, M. A. (2008). Dynamics of genome rearrangement in bacterial populations. *PLoS Genet.* 4:128. doi: 10.1371/journal.pgen.1000128

Dobzhansky, T., and Sturtevant, A. H. (1938). Inversions in the chromosomes of Drosophila Pseudoobscura. *Genetics* 23:28.

Egri-Nagy, A., Francis, A. R., and Gebhardt, V. (2014a). "Bacterial genomics and computational group theory: the BioGAP package for GAP," in *Mathematical Software - ICMS 2014. ICMS 2014. Lecture Notes in Computer Science*, Vol. 8592, eds H. Hong and C. Yap (Berlin; Heidelberg: Springer). doi: 10.1007/978-3-662-44199-2_12

Egri-Nagy, A., Gebhardt, V., Tanaka, M. M., and Francis, A. R. (2014b). Group-theoretic models of the inversion process in bacterial genomes. *J. Math. Biol.* 69, 243–265. doi: 10.1007/s00285-013-0702-6

Eisen, J. A., Heidelberg, J. F., White, O., and Salzberg, S. L. (2000). Evidence for symmetric chromosomal inversions around the replication origin in bacteria. *Genome Biol.* 1, 1–11. doi: 10.1186/gb-2000-1-6-research0011

Felsenstein J. (1993). *PHYLIP (Phylogeny Inference Package)*, version 3.5 c.

Francis, A. (2014). An algebraic view of bacterial genome evolution. *J. Math. Biol.* 69, 1693–1718. doi: 10.1007/s00285-013-0747-6

Galvão, G. R., Lee, O., and Dias, Z. (2015). Sorting signed permutations by short operations. *Algorith. Mol. Biol.* 10:12. doi: 10.1186/s13015-015-0040-x

Hannenhalli, S., and Pevzner, P. A. (1999). Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *J. ACM* 46, 1–27. doi: 10.1145/300515.300516

Holt, D. F. (1995). *KBMAG—Knuth-Bendix in Monoids and Automatic Groups.* Available online at: ftp.maths.warwick.ac.uk in directory people/dfh/kbmag2.

Holt, D. F., Eick, B., and O'Brien, E. A. (2005). *Handbook of Computational Group Theory*. London: CRC Press.

Humphreys, J. E. (1992). *Reflection Groups and Coxeter Groups, Vol. 29*. Cambridge: Cambridge University Press.

Knuth, D. E., and Bendix, P. B. (1983). "Simple word problems in universal algebras," in *Automation of Reasoning*. Symbolic Computation (Artificial Intelligence), eds J. H. Siekmann and G. Wrightson (Berlin; Heidelberg: Springer). doi: 10.1007/978-3-642-81955-1_23

Lefebvre, J., El-Mabrouk, N., Tillier, E., and Sankoff, D. (2003). Detection and validation of single gene inversions. *Bioinformatics* 19, i190–i196. doi: 10.1093/bioinformatics/btg1025

Pinter, R. Y., and Skiena, S. (2002). Genomic sorting with length-weighted reversals. *Genome Inform.* 13, 103–111. doi: 10.11234/gi1990.13.103

Revell, L. J., and Chamberlain, S. A. (2014). Rphylip: an R interface for PHYLIP. *Methods Ecol. Evol.* 5, 976–981. doi: 10.1111/2041-210X.12233

Sankoff, D. (1992). "Edit distance for genome comparison based on non-local operations," in *Annual Symposium on Combinatorial Pattern Matching* (Berlin: Springer), 121–135.

Sankoff, D., Leduc, G., Antoine, N., Paquin, B., Lang, B. F., and Cedergren, R. (1992). Gene order comparisons for phylogenetic inference: evolution of the mitochondrial genome. *Proc. Natl. Acad. Sci. U.S.A.* 89, 6575–6579. doi: 10.1073/pnas.89.14.6575

Sankoff, D., Leduc, G., and Rand, D. (1991). *Derange—Minimum Weight Generation of Oriented Permutation by Block Inversions and Block Movements.* Macintosh Application, Centre des Recherches Mathematiques, Universite de Montreal.

Schønert, M., et al. (1997). *GAP – Groups, Algorithms, and Programming – Version 3 Release 4 Patchlevel 4*. Aachen: Lehrstuhl D fur Mathematik, Rheinisch Westfalische Technische Hochschule.

Seoighe, C., Federspiel, N., Jones, T., Hansen, N., Bivolarovic, V., Surzycki, R., et al. (2000) Prevalence of small inversions in yeast gene order evolution. *Proc. Natl. Acad. Sci. U.S.A.* 97, 14433–14437. doi: 10.1073/pnas.240462997

Shao, M., Lin, Y., and Moret, B. (2014). "An exact algorithm to compute the DCJ distance for genomes with duplicate genes," in *Research in Computational Molecular Biology. RECOMB 2014. Lecture Notes in Computer Science*, Vol. 8394, ed R. Sharan (Cham: Springer). doi: 10.1007/978-3-319-05269-4_22

Swidan, F., Bender, M. A., Ge, D., He, S., Hu, H., and Pinter, R. Y. (2004). "Sorting by length-weighted reversals: dealing with signs and circularity," in *Annual Symposium on Combinatorial Pattern Matching* (Berlin: Springer), 32–46.

Tesler, G. (2002). Grimm: genome rearrangements web server. *Bioinformatics* 18, 492–493. doi: 10.1093/bioinformatics/18.3.492

Watterson, G. A., Ewens, W. J., Hall, T. E., and Morgan, A. (1982). The chromosome inversion problem. *J. Theor. Biol.* 99, 1–7. doi: 10.1016/0022-5193(82)90384-8

Yancopoulos, S., Attie, O., and Friedberg, R. (2005). Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics* 21, 3340–3346. doi: 10.1093/bioinformatics/bti535

Yin, X., and Zhu, D. (2013). Sorting genomes by generalized translocations. *Theor. Comput. Sci.* 491, 127–135. doi: 10.1016/j.tcs.2013.01.004

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.