Identifying Breakpoint Median Genomes: A Branching Algorithm Approach

Poly H. da Silva
Columbia University, New York, NY, USA

Arash Jamshidpey[1]
University of California at Berkeley, Berkeley, CA, USA

David Sankoff
University of Ottawa, Ottawa, Ontario, Canada

Abstract

Genome comparison often involves quantifying dissimilarities between genomes with identical gene sets, commonly using breakpoints – points where adjacent genes in one genome are not adjacent in another. The concept of a median genome, used for comparison of multiple genomes, aims to find a genome that minimizes the total distance to all genomes in a given set. While median genomes are useful for extracting common genomic information and estimating ancestral traits, the existence of multiple divergent medians raises concerns about their accuracy in reflecting the true ancestor. The median problem is known to be NP-hard, particularly for unichromosomal genomes, and solving it becomes increasingly challenging under different genome distance models. In this work, we introduce a novel branching algorithm to efficiently find all breakpoint medians of k linear unichromosomal genomes, represented as unsigned permutations. This algorithm constructs a rooted labeled tree, where the sequence of labels along each complete ray defines a genome, providing a structured and efficient way to explore the space of candidate medians by narrowing the search to a well-defined and significantly smaller subset of the permutation space. We validate our approach with experiments on randomly generated sets of three permutations. The results show that our method successfully finds the exact medians and also identifies many near-optimal approximations. Our experiments further show that most medians lie relatively close to the input permutations, in agreement with prior theoretical results.

2012 ACM Subject Classification Mathematics of computing → Combinatorial algorithms; Mathematics of computing → Permutations and combinations; Applied computing → Computational genomics

Keywords and phrases Breakpoint distance, median genomes, phylogeny reconstruction, random permutations

Digital Object Identifier 10.4230/LIPIcs.WABI.2025.18

1 Introduction

Comparing genomes with the same syntenic blocks involves computing their dissimilarities. This dissimilarity is often quantified by identifying breakpoints, points at which genes are adjacent in one genome but not in the other. Introduced formally by Sankoff and Blanchette in 1997 [13], the total number of breakpoints serves as a metric for dissimilarity. To compare multiple genomes, we can use the concept of median. Given a set of three or more genomes $X = \{g_1, ..., g_k\}$ and a distance d, a median for the set X is a genome that minimizes the total distance function $d_T(\cdot) := \sum_{i=1}^{k} d(g_i, \cdot)$. The concept of the median was first employed by Sankoff *et al.* [14] in 1996 within the context of evolutionary gene order models. Motivated by the search for ancestral genomic information and its applications to small phylogeny problems, the median problem has since attracted significant attention [2, 3, 15, 7, 6, 16, 17].

[1] corresponding author

25th International Conference on Algorithms for Bioinformatics (WABI 2025).
Editors: Broňa Brejová and Rob Patro; Article No. 18; pp. 18:1–18:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

However, the complexity of the median problem varies with different genome distances, often proving to be NP-hard [3, 15, 7] particularly for unichromosomal genomes. For instance, the breakpoint median problem was shown to be NP-hard by Bryant [2] for linear unichromosomal genomes. Moreover, identifying a median in which adjacency sets are contained within the union of the adjacency sets of the input genomes has also been proven to be NP-hard [2]. Following the reduction of the median problem to the Traveling Salesman Problem (TSP) by Sankoff and Blanchette [13], in 2012, Boyd and Haghighi [1], using Concorde (a fast software to find TSP solutions), presented a fast algorithm to find breakpoint medians of samples of large genomes.

While median genomes aim to extract common information among given genomes and estimate ancestral characteristics, the existence of multiple medians with considerable divergence [8] raises questions about their proximity to the true ancestor or their usability in providing ancestral insights. Additionally, determining which, if any, of these medians accurately reflects ancestral traits poses a significant challenge. In fact, Zheng and Sankoff [18], Jamshidpey and Sankoff [10] and Miardan *et al.* [12] showed that median may fail to approximate the ancestor for the long-time evolution of genomes, while for genomes involved in evolution for a shorter period of time medians may approximate the true ancestor.

To address the challenge of identifying relevant medians, we propose a novel branching algorithm for efficiently finding all breakpoint medians of k linear unichromosomal genomes represented by unsigned permutations of length n. This exponential algorithm constructs a rooted labeled tree, whose sequence of labels for each ray (a shortest path connecting the root to a leaf) with length $n-1$ determines a unichromosomal genome (represented as a permutation). The set of all such unichromosomal genomes contains all medians of the k input genomes. We show that this tree construction reduces the median search space significantly compared to the full space of $n!$ permutations (see Table 3).

This paper is organized as follows. We begin by laying the foundation. In Section 2 we introduce the basic concepts of a genomic space with breakpoint distance and review some essential prior results in the literature. In Section 3, we delve into the methodology behind our branching algorithms designed to identify all medians within a given set of genomes. Subsequently, in Section 4, we provide empirical validation of our approach through a series of experiments using sets of three random permutations. A key contribution of this experimental section is that our method is able to compute the median value exactly, even in cases where it remained unknown in previous work. We examine how the median value behaves as the permutation length increases and analyze the distribution of approximate medians in the reduced search space generated by our algorithm. The results indicate that, although not all permutations in this space are true medians, a substantial proportion have total distances very close to the minimum, making them effective median approximations. We also explore how far the medians tend to be from the input permutations and find that most lie relatively close, an observation consistent with prior theoretical results [8, 9, 11, 4]. We conclude the paper with a discussion of our findings, their implications, and potential avenues for future research.

2 Breakpoint medians

We represent an unichromosomal genome by a permutation π which is a bijection on $[n] := \{1, \cdots, n\}$. In other words, a permutation π can be represented by $\pi(1), \cdots, \pi(n)$, which indicates a specific order on $[n]$. When there is no risk of ambiguity, we often write π_i instead of $\pi(i)$, and denote $\pi := \pi_1 ... \pi_n$. We define the set of adjacencies of π as

$\mathcal{A}_\pi = \{\{\pi_i, \pi_{i+1}\} \mid 1 \leq i \leq n-1\}$, where each adjacency is treated as an unordered pair. Let S_n denote the set of all permutations of length n. Given $x, y \in S_n$, we denote by $\mathcal{A}_{x,y} := \mathcal{A}_x \cap \mathcal{A}_y$ the set of all common adjacencies of x and y. For a set $X \subset S_n$, we also denote by $\mathcal{A}_X := \bigcap_{x \in X} \mathcal{A}_x$ the set of all common adjacencies of permutations in X. The breakpoint (bp) distance between x and y is define by $d(x,y) := n - 1 - |\mathcal{A}_{x,y}|$.

The breakpoint distance is neither a geodesic distance nor an edit distance, and for this reason the notion of partial geodesics was introduced by Jamshidpey *et al.* [9]. We can consider the breakpoint distance as a generalized edit distance that determines the parsimonious (shortest) paths of transforming one permutation to another, but with many missing points in the parsimonious path. In other words, in edit distances the length of every jump from a point in the parsimonious path to its closest point in the the same path is one, while in generalized edit distances such as the breakpoint distance this length may be bigger. A partial geodesic [9] between x and y is a maximal chain $x = \pi_0, \pi_1, ..., \pi_{k-1}, \pi_k = y$ in S_n such that $\sum_{i=0}^{k-1} d(\pi_i, \pi_{i+1}) = d(x,y)$. We denote by $\overline{[x,y]}$ the set of all permutations lying on partial geodesics connecting $x, y \in S_n$, and call them geodesic points of x and y.

For a set of three or more genomes $X = \{x_1, ..., x_k\}$, a breakpoint median is a genome that minimizes the total distance function $d_T(\cdot, X) := \sum_{i=1}^{k} d(x_i, \cdot)$. The minimal value of d_T is known as the median value of the set X, denoted by $\mu(X)$. The set of all breakpoint medians of X is denoted by $M(X)$.

For a set of permutations $X = \{x_1, ..., x_k\}$ in S_n for which the pairwise breakpoint distances take the maximum value $n-1$, Jamshidpey *et al.* [9] provide a necessary and sufficient condition for a permutation m to be a median of X, that is m is a median of X if and only if

$$\mathcal{A}_m \subset \bigcup_{x \in X} \mathcal{A}_x.$$

Also from [9], a permutation π is a geodesic point of two permutations x and y, and so it is a median of $\{x,y\}$, if and only if $\mathcal{A}_{x,y} \subset \pi \subset \mathcal{A}_x \cup \mathcal{A}_y$. On the other hand, we do not have a result establishing a necessary and sufficient condition for a permutation to be a median of a general set of permutations X. In fact, it is known that there may exist a median that does not contain all common adjacencies of permutations in X, i.e., there may exist a median m such that $\mathcal{A}_X \not\subseteq \mathcal{A}_m$, as the example given by Bryant [2]. However, even though there may exist medians not containing all common adjacencies of elements of X, there always exists at least one median with this property, namely, there exists at least one median m such that $\mathcal{A}_X \subset \mathcal{A}_m$ (cf. [2]). In addition, when we have a general set of permutations X, even counter-intuitively, it is not necessary that every adjacency of a median m is an adjacency of at least one of the permutations in X, that is, there may exist a median m such that

$$\mathcal{A}_m \not\subseteq \bigcup_{x \in X} \mathcal{A}_x,$$

as is shown by Bryant [2]. However, [5] provides an upper bound for the maximum number of adjacencies of a median that are not in $\bigcup_{x \in X} \mathcal{A}_x$ as stated in Theorem 1 (whose proof is provided in Appendix A). Before the statement of Theorem 1, we need the following notation. Denote by $\mathcal{P}(S)$ the set of all subsets of a set or space S. Let $X = \{x_1, ..., x_k\} \subset S_n$ and let $\mathcal{B}_X^X = \mathcal{B}_{x_1,...,x_k}^X := \mathcal{A}_{x_1,...,x_k}$. Then, for any $j = 1, \cdots, k$, let

$$\mathcal{B}_{x_1,...,x_{j-1},x_{j+1},...x_k}^X := \mathcal{A}_{x_1,...,x_{j-1},x_{j+1},...x_k} \setminus \mathcal{B}_{x_1,...,x_k}^X$$

Continuing this, for any $i_1, \cdots, i_r \in [n]$ and $U = \{x_{i_1}, ..., x_{i_r}\} \subset X$, we set

$$\mathcal{B}_U^X = \mathcal{B}_{x_{i_1},\ldots,x_{i_r}}^X := \mathcal{A}_U \setminus (\bigcup_{U \subsetneq V} \mathcal{B}_V^X).$$

In other words, $\mathcal{B}_U^X$ includes all adjacencies that are common in every $x \in U$, but missing from every $y \in X \setminus U$. We have the following theorem.

▶ **Theorem 1** ([5]). *Let $X = \{x_1, ..., x_k\} \subset S_n$ be such that*

$$d_T(x_k, X) = \min_{i=1\ldots k} d_T(x_i, X),$$

and let $m \in M(X)$. Then

$$|\mathcal{A}_m \setminus (\bigcup_{i=1}^{k} \mathcal{A}_{x_i})| \leq \mathcal{O}_n(X) := \sum_{r=2}^{k-1} (r-1) \sum_{1 \leq i_1 < \ldots < i_r < k} |\mathcal{B}_{x_{i_1},\ldots,x_{i_r}}^X|. \tag{1}$$

In particular, for $k = 3$, for any $m \in M(X)$

$$|\mathcal{A}_m \setminus \bigcup_{i=1}^{3} \mathcal{A}_{x_i}| \leq \mathcal{O}_n(\{x_1, x_2, x_3\}) := |\mathcal{B}_{x_1,x_2}^X|.$$

▶ Remark 2. Note that the theorem makes use of the upper bound $d_T(m, X) \leq d_T(x_k, X)$, for any $m \in M(X)$. In particular, for $x = 3$, $d_T(x_3, X) = \min_{i=1,2,3} d_T(x_i, X)$ is equivalent to $d(x_1, x_2) = \max_{i,j} d(x_i, x_j)$, which itself is equivalent to $|\mathcal{B}_{x_1,x_2}^X| = \min_{i \neq j} |\mathcal{B}_{x_i,x_j}^X|$. In this case, $\mathcal{O}_n(X) = |\mathcal{B}_{x_1,x_2}^X| = \min_{i \neq j} |\mathcal{B}_{x_i,x_j}^X|$ implies that the upper bound is the number of adjacencies common in the pair of farthest genomes, i.e. x_1, x_2, which are missing from x_3.

This upper bound significantly restricts the median search space, and by making use of it, we develop an algorithm to find all breakpoint medians of a general set of permutations.

We first analyze exponential algorithms that construct specific rooted labeled trees, where each ray (a shortest path from the root to a leaf) of length $n-1$ corresponds to a permutation determined by the sequence of labels along the path. The set of all such label sequences includes all medians, thereby significantly reducing the search space. Specifically, the new median search space consists of the set of all leaves of these trees. While the volume of this new search space is exponential, it is negligible compared to the size of the permutation group of length n.

3 An algorithm to find medians

To describe our algorithms, we first define the neighbors of a point (i.e., a number representing a syntenic block or gene) with respect to a given set of permutations. Specifically, for $X = \{x_1, ..., x_k\} \subset S_n$ and $i = 1, \cdots n$, we define

$$\mathcal{N}_X(i) = \mathcal{N}_{x_1,\ldots,x_k}(i) = \{j : \{i, j\} \in \bigcup_{l=1}^{k} \mathcal{A}_{x_l}\}.$$

Note that for each i, $1 \leq |\mathcal{N}_X(i)| \leq 2k$. The equality $|\mathcal{N}_X(i)| = 1$ holds when i satisfies both of the following conditions: i is either the first or last number in each permutation x_l, for $1 \leq l \leq k$; and i is an extremity of an adjacency in $\mathcal{A}_X$. On the other hand, the equality $|\mathcal{N}_X(i)| = 2k$ holds when i satisfies both of the following conditions: i is neither the first nor the last number of any permutation x_l, for $1 \leq l \leq k$; and i is not an extremity of an adjacency in $\mathcal{A}_{x_l,x_p}$, for any $l \neq p$. If X is such that $d(x_l, x_p) = n-1$ for any $l \neq p$, then $k \leq |\mathcal{N}_X(i)| \leq 2k$.

Our main goal in this paper is to find all medians for a given set of permutations $X \subset S_n$. To achieve this, we construct a family of labeled rooted trees of height $n-1$ with the following properties: Each vertex v of the tree is assigned a label, denoted by $\ell(v)$, which is a number between 1 and n. In order for two vertices, u and v, to be connected by an edge, it is necessary that $\ell(v) \in \mathcal{N}_X(\ell(u))$. Furthermore, for each path of length $n-1$ from the root to a leaf, the sequence of labels along the path forms a permutation y satisfying certain conditions. In particular, the labels of the root and leaf determine the first and last numbers in y, respectively, i.e., y_1 and y_n. We refer to y as a permutation given by a leaf.

In the rest of this paper, we first present an algorithm in Section 3.1 for constructing trees in which every permutation y given by a leaf satisfies

$$\mathcal{A}_y \subset \bigcup_{x \in X} \mathcal{A}_x.$$

In this case, if the breakpoint distance between every pair of permutations in X attains the maximum value $n-1$, then from Jamshidpey *et al.* [9], any permutation y given by a leaf at level $n-1$ is a median of X. Consequently, the algorithm finds all medians of X.

Next, in Section 3.2, we construct trees where every permutation y given by a leaf satisfies

$$\mathcal{A}_X \subset \mathcal{A}_y \subset \bigcup_{x \in X} \mathcal{A}_x.$$

In this case, if the upper bound given in (1) is zero – a weaker condition than requiring all pairwise distances in X to be maximal – then at least one of the permutations given by a leaf of the tree is a median of X (cf. [2]). This allows us to determine the median value within a relatively smaller search space.

Finally, in Section 3.3, we introduce a modification of the algorithm from Section 3.1, providing additional flexibility to identify all medians of a general set of permutations. This is achieved by allowing permutations to contain a limited number of adjacencies not present in $\bigcup_{x \in X} \mathcal{A}_x$. The upper bound in (1) ensures that all medians of X are represented among the leaves of the tree constructed by this flexible algorithm.

3.1 Finding all medians of permutations with maximum pairwise distance to each other

Let id denote the identity permutation in S_n, and let $x \in S_n$ be a permutation such that $d(id, x) = n-1$. We first describe the algorithm for the case of two permutations, id and x, and later extend it to $k > 2$ permutations.

For each $i = 1, \ldots, n$, we construct a tree whose root is labeled by i. We denote the root of this tree by $\varnothing$. The root $\varnothing$ has $|\mathcal{N}_{id,x}(i)|$ children, denoted by $\varnothing 1, \varnothing 2, \ldots, \varnothing|\mathcal{N}_{id,x}(i)|$. The label of each child is a number in $\mathcal{N}_{id,x}(i)$, such that if $j \neq j'$, then $\ell(\varnothing j) \neq \ell(\varnothing j')$. In other words, there is a bijection between the set $\{\ell(\varnothing r) \mid 1 \leq r \leq |\mathcal{N}_{id,x}(i)|\}$ and $\mathcal{N}_{id,x}(i)$. By convention, we fix this bijection so that $\ell(\varnothing r)$ is an increasing function of r; in particular, $\ell(\varnothing 1)$ and $\ell(\varnothing|\mathcal{N}_{id,x}(i)|)$ are the smallest and largest numbers in $\mathcal{N}_{id,x}(i)$, respectively.

Each vertex $\varnothing j_1$, for $1 \leq j_1 \leq |\mathcal{N}_{id,x}(i)|$, has $|\mathcal{N}_{id,x}(\ell(\varnothing j_1)) \setminus \{i\}|$ children, denoted by $\varnothing j_1 j_2$, where $1 \leq j_2 \leq |\mathcal{N}_{id,x}(\ell(\varnothing j_1)) \setminus \{i\}|$, with $\ell(\varnothing j_1 j_2) \in \mathcal{N}_{id,x}(\ell(\varnothing j_1)) \setminus \{i\}$. Moreover, if $j_2 \neq j_2'$, then $\ell(\varnothing j_1 j_2) \neq \ell(\varnothing j_1 j_2')$. Continuing this process, the parent of a vertex $\varnothing j_1 j_2 \ldots j_{l-1} j_l$ at level l is the vertex $\varnothing j_1 j_2 \ldots j_{l-1}$. If

$$\mathcal{N}_{id,x}(\ell(\varnothing j_1 j_2 \ldots j_{l-1} j_l)) \setminus \{\ell(\varnothing), \ell(\varnothing j_1), \ldots, \ell(\varnothing j_1 j_2 \ldots j_{l-1})\} \neq \emptyset,$$

then its children are $\varnothing j_1 j_2 \dots j_{l-1} j_l j_{l+1}$, for

$$1 \leq j_{l+1} \leq |\mathcal{N}_{id,x}(\ell(\varnothing j_1 j_2 \dots j_{l-1} j_l)) \setminus \{\ell(\varnothing), \ell(\varnothing j_1), \dots, \ell(\varnothing j_1 j_2 \dots j_{l-1})\}|,$$

where $\ell(\varnothing j_1 j_2 ... j_{l-1} j_l j_{l+1}) \in \mathcal{N}_{id,x}(\ell(\varnothing j_1 j_2 ... j_{l-1} j_l)) \setminus \{\ell(\varnothing), \ell(\varnothing j_1), ..., \ell(\varnothing j_1 j_2 ... j_{l-1})\}$.

Again, if $j_{l+1} \neq j'_{l+1}$, then $\ell(\varnothing j_1 j_2 \dots j_l j_{l+1}) \neq \ell(\varnothing j_1 j_2 \dots j_l j'_{l+1})$. Since this is a finite process, it results in a labeled tree for each i as the label of the root, with $1 \leq i \leq n$.

More precisely, the sequence of labels along every $(\varnothing, u)$-path, where u is a leaf at level $n-1$, represents a permutation y such that $\mathcal{A}_y \subset \mathcal{A}_{id} \cup \mathcal{A}_x$. Since $\mathcal{A}_{id} \cap \mathcal{A}_x = \emptyset$, we have $y \in \overline{[id, x]}$, meaning that we can identify all geodesic points of id and x when $d(id, x) = n - 1$. Furthermore, the number of permutations in $\overline{[id, x]}$ is equal to the number of $(\varnothing, u)$-paths of length $n - 1$ in all n trees. An example is illustrated in Figure 1.

For each i, $1 \leq i \leq n$, we denote by $\mathcal{T}^i_{id,x}$ the tree constructed as above, where the root is labeled by i. We also define $\mathcal{T}_{id,x} := \{\mathcal{T}^i_{id,x} \mid 1 \leq i \leq n\}$ as the set of all these n trees.

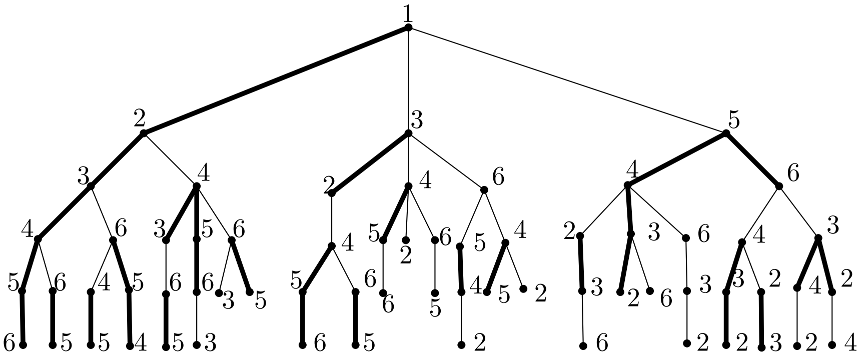

Figure 1 The representation of the tree $\mathcal{T}^1_{id,x}$, for $x = 246315$, with its labels. In this example we have $\mathcal{N}_{id,x}(1) = \{2, 3, 5\}$, $\mathcal{N}_{id,x}(2) = \{1, 3, 4\}$, $\mathcal{N}_{id,x}(3) = \{1, 2, 4, 6\}$, $\mathcal{N}_{id,x}(4) = \{2, 3, 5, 6\}$, $\mathcal{N}_{id,x}(5) = \{1, 4, 6\}$ and $\mathcal{N}_{id,x}(6) = \{3, 4, 5\}$. Also, $d(id, x) = n - 1 = 5$, and so each path from the root to a leaf in level 5 constitutes a permutation in $\overline{[id, x]}$. The list of permutations in $\overline{[id, x]}$ given by this tree is: $id = 123456$, 123465, 123645, 123654, 124365, 124563, 132456, 132465, 136542, 154236, 154632, 156432, 156423, 156342 and 156324. These are all the permutations in $\overline{[id, x]}$ that start at 1. The bold edges represent the adjacencies of id and the other edges represent the adjacencies of x.

More generally, let $X = \{x_1, ..., x_k\} \subset S_n$ be a set of permutations. Following the same steps just replacing $\mathcal{N}_{id,x}(i)$ with $\mathcal{N}_{x_1,...,x_k}(i)$, we can construct n labeled rooted trees, $\mathcal{T}^i_X$, such that the sequence of labels along each $(\varnothing, u)-$path, where u is a leaf at level $n-1$, forms a permutation y satisfying $\mathcal{A}_y \subset \bigcup_{l=1}^{k} \mathcal{A}_{x_l}$. Therefore, if $X = \{x_1, ..., x_k\} \subset S_n$ satisfies $d(x_l, x_p) = n - 1$ for any $l \neq p$, then the set of all permutations given by leaves at level $n - 1$ in the trees $\mathcal{T}^i_X$, for $i = 1, ..., n$, is exactly the set of all medians of X. We denote

$$\mathcal{T}_X := \{\mathcal{T}^i_X; 1 \leq i \leq n\}$$

and let $Y(\mathcal{T}^i_X)$ be the set of all permutations $y \in S_n$ that are given by a leaf of $\mathcal{T}^i_X$ at level $n - 1$. Moreover, let

$$Y(\mathcal{T}_X) := \bigcup_{i=1}^{n} Y(\mathcal{T}^i_X).$$

Each vertex of a tree $\mathcal{T}^i_X$ is a sequence $\varnothing j_1 j_2 ... j_l$ where each j_i, $i = 1, .., l$, is a number between 1 and $2|X|$. To construct a child vertex and its label from its parent and the parent's label, we define the following operation. Given a sequence of symbols $u = u_1 \dots u_l$ (e.g., numbers) and a symbol r, we define the operation $u \oplus r$ as a new sequence of symbols $u \oplus r := u_1 ... u_l r$. We emphasize that in the above algorithm $u_1 = \varnothing$ and $u_2, .., u_l$ and r are natural numbers in $\{1, ..., 2|X|\}$. For each fixed tree of $\mathcal{T}_X$, we denote by $T_u(\mathcal{T}_X^i) = T_u$ the ordered sequence of labels assigned to the vertices along the $(\varnothing, u)-$path for a vertex u in $\mathcal{T}_X^i$. Observe that T_u is a sequence of digits, where each digit is between 1 and n, and all digits are distinct. We denote by $dig(T_u)$ the set of labels appearing in T_u, and let $\mathcal{N}_X(T_u) := \mathcal{N}_X(\ell(u))$. Additionally, we define $L_j(\mathcal{T}_X^i) = L_j$ to be the set of all vertices of the tree $\mathcal{T}_X^i$ at level j, for $0 \leq j \leq n-1$, considering the root at level zero. Using these notations, the tree construction process for $k \geq 2$ permutations is described in Algorithm 1. These notations will also be used in the subsequent sections.

Note that we can also view the tree construction in suffix-tree terms, as follows. Each tree $\mathcal{T}_X^i$ has root label i, and every internal node that spells a prefix $(i, u_1, \ldots, u_j)$ branches to a child u_{j+1} if and only if u_{j+1} is adjacent to u_j in at least one genome in X, and u_{j+1} has not yet appeared in the prefix.

Algorithm 1 Gives permutations y such that $\mathcal{A}_y \subset \bigcup_{x \in X} \mathcal{A}_x$.

```
Data: $X = \{x_1, ..., x_k\} \subset S_n$.
Result: Permutations $y$ such that $\mathcal{A}_y \subset \bigcup_{l=1}^{k} \mathcal{A}_{x_l}$.
for $i = 1, ..., n$ do
    for $w = 0, ..., n$ do
        $L_w \longleftarrow \emptyset$
    $L_0 \longleftarrow \{\varnothing\}$
    $T_\varnothing \longleftarrow i$
    for $j = 1, ..., n-1$ do
        for $u \in L_{j-1}$ do
            $r \longleftarrow 0$
            for $k \in \mathcal{N}_X(T_u)$ do
                if $k \notin dig(T_u)$ then
                    $r \longleftarrow r + 1$
                    $L_j \longleftarrow L_j \cup \{u \oplus r\}$
                    $T_{u \oplus r} \longleftarrow T_u \oplus k$
    for $u \in L_{n-1}$ do
        print $T_u$
```

▶ Remark 3 (Finding permutations with maximum distance of a set X). Given $X = \{x_1, ..., x_k\} \subset S_n$, denote by $\overline{\mathcal{N}}_X(i) = \overline{\mathcal{N}}_{x_1,...,x_k}(i) := [n] \setminus \mathcal{N}_X(i)$ the complement set of $\mathcal{N}_X(i)$, for $1 \leq i \leq n$. Note that, if we replace $\mathcal{N}_X(\cdot)$ by $\overline{\mathcal{N}}_X(\cdot)$ in Algorithm 1, we obtain all permutations with maximum distance from X, i.e, we find all permutations y such that $d(y, x_i) = n - 1$, for $1 \leq i \leq n$.

3.2 Finding all geodesic points for a general set of permutations

A segment s of a set of adjacencies $I \subset \mathcal{A}_\pi$, for $\pi \in S_n$, is a maximal set of consecutive adjacencies of I, i.e. it is a set

$$s = \{\{\pi(r), \pi(r+1)\}, \{\pi(r+1), \pi(r+2)\}, \cdots, \{\pi(r+k-1), \pi(r+k)\}\} \subset I$$

such that $\{\pi(r-1), \pi(r)\}, \{\pi(r+k), \pi(r+k+1)\} \notin I$, for $r > 1$ and $r+k < n$. We often denote s by $\|\pi(r), \cdots, \pi(r+k)\|$, and write $s \hat{\in} I$. We say that $Int(s) := \{\pi(r+1), \cdots, \pi(r+k-1)\}$ are the internal points of s, and $End(s) := \{\pi(r), \pi(r+k)\}$ are the end points of s. Generalizing the idea, the internal and end points of $I \subset \mathcal{A}_\pi$ are defined by

$$Int(I) := \bigcup_{s \hat{\in} I} Int(s), \quad End(I) := \bigcup_{s \hat{\in} I} End(s).$$

Note that the above definitions do not depend on a specific choice of π, that is, the definitions remain intact if we replace π by any π' for which $I \subset \mathcal{A}_{\pi'}$.

Now consider the case where $x \in S_n$ satisfies $d(id, x) < n-1$, that is, $\mathcal{A}_{id,x} \neq \emptyset$. We can apply a similar idea as in the case of maximum distance, but now with some restrictions. From [9], a permutation $y \in \overline{[id, x]}$ if and only if $\mathcal{A}_{id,x} \subset \mathcal{A}_y \subset \mathcal{A}_{id} \cup \mathcal{A}_x$. As a result, if $s = \|n_0, ..., n_l\|$ is a segment of $\mathcal{A}_{id,x}$, then the ordered sequence of digits $n_0...n_l$ must appear in the ordered sequence of labels of the $(\varnothing, u)$-paths with length $n-1$. In order for this to hold, first note that no internal point of $\mathcal{A}_{id,x}$ can be a label of the root. In fact, if $i \in Int(\mathcal{A}_{id,x})$, then there exist j and j' with $\{i, j\}$ and $\{i, j'\}$ in $\mathcal{A}_{id,x}$. Therefore, if i is the label of the root, any permutation y given by a leaf at the level $n-1$ will contain either $\{i, j\}$ or $\{i, j'\}$ (but not both), and thus cannot satisfy $\mathcal{A}_{id,x} \subset \mathcal{A}_y$. This implies that if $i \in Int(\mathcal{A}_{id,x})$, then i can only be a label of an internal vertex of the tree. Moreover, since $|\mathcal{N}_{id,x}(i)| = 2$, the vertex of the label equal to i will have exactly one child. Therefore, for any segment $s = \|n_0, ..., n_l\| \hat{\in} \mathcal{A}_{id,x}$, either n_0 or n_l should appear before $n_1, \ldots, n_{l-1}$ in T_u for any leaf u. To ensure the condition $\mathcal{A}_{id,x} \subset \mathcal{A}_y$, it follows that each segment $s = \|n_0, ..., n_l\| \hat{\in} \mathcal{A}_{id,x}$, if a vertex v has label $\ell(v) = n_0$ and n_l is not in T_v (or the opposite, $\ell(v) = n_l$ and n_0 is not in T_v), then v must have exactly one child $v \oplus 1$ with label $\ell(v \oplus 1) = n_1$ (or $\ell(v \oplus 1) = n_{l-1}$).

To describe the tree construction process, for a given segment $s \hat{\in} \mathcal{A}_X$ and $j \in End(s)$, we denote by $\bar{j}$ the other end point of s and by j^* the unique point (number) such that adjacency $\{j, j^*\} \in s$. In the case where $d(id, x) < n-1$, for each $i \in [n] \setminus Int(\mathcal{A}_{id,x})$ we construct a rooted tree $\overline{\mathcal{T}}^i_{id,x}$ with the root label i. At each level l, a vertex $\varnothing j_1...j_{l-1}j_l$ is a child of $\varnothing j_1...j_{l-1}$. Now, if $\mathcal{N}_{id,x}(\ell(\varnothing j_1...j_{l-1}j_l)) \setminus \{\ell(\varnothing), \ell(\varnothing j_1), ..., \ell(\varnothing j_1 j_2...j_{l-1})\} \neq \emptyset$ then $\varnothing j_1...j_{l-1}j_l$ has children defined as follows. If $\ell(j_l) \in End(s)$ and $\overline{\ell(j_l)} \notin dig(T_{\varnothing j_1...j_l})$, for some segment $s \hat{\in} \mathcal{A}_{id,x}$, then $\varnothing j_1...j_l$ has exactly one child $\varnothing j_1...j_l 1$ with label $\ell(\varnothing j_1...j_l 1) = \ell(\varnothing j_1...j_{l-1}j_l)^*$. Otherwise, its children are $\varnothing j_1...j_l j_{l+1}$, for

$$1 \leq j_{l+1} \leq |\mathcal{N}_{id,x}(\ell(\varnothing j_1...j_{l-1}j_l)) \setminus \{\ell(\varnothing), \ell(\varnothing j_1), ..., \ell(\varnothing j_1 j_2...j_{l-1})\}|,$$

where $\ell(\varnothing j_1 j_2...j_{l-1}j_l j_{l+1}) \in \mathcal{N}_{id,x}(\ell(\varnothing j_1...j_{l-1}j_l)) \setminus \{\ell(\varnothing), \ell(\varnothing j_1), ..., \ell(\varnothing j_1 j_2...j_{l-1})\}$, in the same way that if $j_{l+1} \neq j'_{l+1}$, then $\ell(\varnothing j_1 j_2...j_l j_{l+1}) \neq \ell(\varnothing j_1 j_2...j_l j'_{l+1})$. After a finite number of steps, we construct $|[n] \setminus Int(I_{id,x})|$ trees such that for each leaf u at the level $n-1$, T_u gives a permutation y satisfying $\mathcal{A}_{id,x} \subset \mathcal{A}_y \subset \mathcal{A}_{id} \cup \mathcal{A}_x$.

We can generalize this idea to a set of k permutations $X = \{x_1, ..., x_k\} \subset S_n$. Following the same steps, just replacing $\mathcal{N}_{id,x}(i)$ with $\mathcal{N}_X(i)$ and $\mathcal{A}_{id,x}$ with $\mathcal{A}_X$, we construct $|[n] \setminus Int(\mathcal{A}_X)|$ labeled rooted trees $\overline{\mathcal{T}}^i_X$, such that for each leaf u at the level $n-1$, the sequence T_u corresponds to a permutation y with $\mathcal{A}_X \subset \mathcal{A}_y \subset \bigcup_{l=1}^{k} \mathcal{A}_{x_l}$. Denote by

$$\overline{\mathcal{T}}_X = \{\overline{\mathcal{T}}^i_X; i \in [n] \setminus Int(\mathcal{A}_X)\}$$

and define $Y(\overline{\mathcal{T}}^i_X)$ to be the set of all permutations $y \in S_n$ given by a leaf of $\overline{\mathcal{T}}^i_X$, and let

$$Y(\overline{\mathcal{T}}_X) := \bigcup_{i \in [n] \setminus Int(\mathcal{A}_X)} Y(\overline{\mathcal{T}}^i_X).$$

For a set X where the upper bound in (1) is equal to zero (recall that this condition is weaker than requiring all permutations in X to be at maximum pairwise distance), from [2], there exists at least one $y \in Y(\overline{\mathcal{T}}_X)$ that is a median of X. More precisely, in this case, any $y' \in Y(\overline{\mathcal{T}}_X)$ such that

$$\sum_{l=1}^{k} d(x_l, y') = \min_{y \in Y(\overline{\mathcal{T}}_X)} \left(\sum_{l=1}^{k} d(x_l, y) \right),$$

is a median of X. Thus, in addition to finding some medians of X, this algorithm also finds the median value of X efficiently. The tree construction process is described in Algorithm 2.

Algorithm 2 Gives all permutations y such that $\mathcal{A}_{x_1,\ldots,x_k} \subset \mathcal{A}_y \subset \bigcup_{l=1}^{k} \mathcal{A}_{x_l}$.

```
Data: $X = \{x_1, ..., x_k\} \subset S_n$.
Result: Permutations $y$ such that $\mathcal{A}_{x_1,\ldots,x_k} \subset \mathcal{A}_y \subset \bigcup_{l=1}^{k} \mathcal{A}_{x_l}$.
for $i \in \{1, ..., n\} \setminus Int(\mathcal{A}_X)$ do
    for $w = 0, ..., n$ do
        $L_w \longleftarrow \emptyset$
    $L_0 \longleftarrow \{\varnothing\}$
    $T_\varnothing \longleftarrow i$
    for $j = 1, ..., n-1$ do
        for $u \in L_{j-1}$ do
            $r \longleftarrow 0$
            if $\ell(u) \in End(\mathcal{A}_X)$ and $\overline{\ell(u)} \notin dig(T_u)$ then
                $r \longleftarrow r + 1$
                $L_j \longleftarrow L_j \cup \{u \oplus r\}$
                $T_{u \oplus r} \longleftarrow T_u \oplus \ell(u)^*$
            else
                for $k \in \mathcal{N}_X(T_u)$ do
                    if $k \notin dig(T_u)$ then
                        $r \longleftarrow r + 1$
                        $L_j \longleftarrow L_j \cup \{u \oplus r\}$
                        $T_{u \oplus r} \longleftarrow T_u \oplus k$
    for $u \in L_{n-1}$ do
        print $T_u$
```

Note that, if $X = \{x_1, ..., x_k\}$ is a set of permutations such that $d(x_l, x_p) = n - 1$, for any $l \neq p$, then $\mathcal{A}_X = \emptyset$. Therefore, in Algorithm 2, the condition

$$\ell(u) \in End(\mathcal{A}_X) \text{ and } \overline{\ell(u)} \notin dig(T_u)$$

does not hold, and hence the algorithm proceeds directly to the“*else*” branch. In this case, Algorithm 2 yields exactly the same output as Algorithm 1. Furthermore, for a general set of permutations $X = \{x_1, ..., x_k\}$, the tree $\mathcal{T}_X^i$ produced by Algorithm 1 contains, as a subgraph, the tree $\overline{\mathcal{T}}_X^i$ generated by Algorithm 2, for all $i \in [n] \setminus Int(\mathcal{A}_X)$. The main properties of these subtrees are:

- No internal point of $\mathcal{A}_X$ can be used as the label of the root, as previously noted;
- For any path starting at the root, once the path reaches one of the end points of a segment $s\hat{\in}\mathcal{A}_X$, say j, the path continues without branching until it reaches the other end point of s, namely $\overline{j}$.

3.3 An algorithm to find all medians of a general set of permutations

As seen in Theorem 1, $\mathcal{O}_n(X)$, for $X \in S_n$, is an upper bound for the number of adjacencies of any median $m \in M(X)$ outside $\cup_{x\in X}\mathcal{A}_x$. To apply this result to k independent random permutations, namely $\xi_1, ..., \xi_k \in S_n$, recall that a sequence of random variables $(Z_n)_{n\in\mathbb{Z}_+}$ converges in probability to a random variable Z, as n goes to infinity, if for any $\varepsilon > 0$, $\mathbb{P}(|Z_n - Z| > \varepsilon) \to 0$. We know that $\mathcal{O}_n(\{\xi_1, ..., \xi_k\})$ is very small, with high probability. More explicitly, from [5], we know that

$$\frac{\mathcal{O}_n(X)}{a_n} \to 0,\ n \to \infty,$$

in probability, for any sequence $(a_n)_{n\in\mathbb{N}}$ diverging to ∞, such that $a_n/n \to 0$, as $n \to \infty$. Therefore, if we consider the flexibility of using $\mathcal{O}_n(X)$ adjacencies out of $\cup_{x\in X}\mathcal{A}_x$ in Algorithm 1, then we obtain all permutations y with at most $\mathcal{O}_n(X)$ adjacencies out of $\cup_{x\in X}\mathcal{A}_x$, which we call $\mathcal{O}_n(X)-$freedom permutations. These permutations include all medians of X with high probability. More generally, for a non-negative integer $\alpha \geq 0$, we say a permutation π is $\alpha-$freedom with respect to $X \subset S_n$, if $|\mathcal{A}_\pi \setminus \cup_{x\in X}\mathcal{A}_x| \leq \alpha$. In this section, we extend our algorithm to construct $\alpha-$freedom medians of $X \subset S_n$ for $\alpha = \mathcal{O}_n(X)$, i.e. the medians of X that include at most α adjacencies out of $\cup_{x\in X}\mathcal{A}_x$.

Let $X = \{x_1, ..., x_k\} \subset S_n$ be a set of permutations such that $\mathcal{O}_n(X) \neq 0$. For every $i = 1, ..., n$, we construct a tree with a root labeled by i. We denote by $\varnothing$ the root of this tree. Now for each vertex of a tree we add a new parameter, namely, for each vertex u we assign a number τ_u, with $0 \leq \tau_u \leq \mathcal{O}_n(X)$, that determines the number of children of vertex u in the tree and the number of adjacencies that are not in $\cup_{x\in X}\mathcal{A}_x$ and appear in T_u, in the following way: if $\tau_u \neq 0$ then u has $n - |dig(T_u)|$ children, i.e., we construct $n - |dig(T_u)|$ sequences of labels by adding to the T_u all possible numbers j, from 1 to n, that did not appear in T_u, and so we add the adjacency $\{\ell(u), j\}$ for each permutation y that is being constructed from the sequence of labels, which also includes adjacencies that are not in $\cup_{x\in X}\mathcal{A}_x$. If $\tau_u = 0$ then any descendent vertex v of u has $\tau_v = 0$ and u has the same number of children given by Algorithm 1, which is $|\mathcal{N}_X(\ell(u)) \setminus dig(T_u)|$. So in this case, T_u already contain $\mathcal{O}_n(X)$ adjacencies out of $\cup_{x\in X}\mathcal{A}_x$. For the root we assign $\tau_\varnothing = \mathcal{O}_n(X)$. So the root has $n-1$ children, called $\varnothing 1$, $\varnothing 2$, ..., $\varnothing(n-1)$, with $\ell(\varnothing j) = j$, for $j < i$, and $\ell(\varnothing j) = j + 1$, for $j \geq i$. We assign $\tau_{\varnothing j} = \tau_\varnothing - 1$ if $\ell(\varnothing j) \notin \mathcal{N}_X(i)$, or $\tau_\varnothing = \tau_{\varnothing j}$ if $\ell(\varnothing j) \in \mathcal{N}_X(i)$. For each vertex $\varnothing j$, if $\tau_{\varnothing j} \neq 0$ then $\varnothing j$ has $n-2$ children, called $\varnothing jj'$, for $1 \leq j' \leq n-2$, with $\ell(\varnothing jj') \in [n] \setminus \{\ell(\varnothing), \ell(\varnothing j)\}$ such that there is a bijection between set $\{\ell(\varnothing jj') : 1 \leq j' \leq n-2\}$ and $[n] \setminus \{\ell(\varnothing), \ell(\varnothing j)\}$. If $\ell(\varnothing jj') \notin \mathcal{N}_X(\ell(\varnothing j))$ then $\tau_{\varnothing jj'} = \tau_{\varnothing j} - 1$, and if $\ell(\varnothing jj') \in \mathcal{N}_X(\ell(\varnothing j))$ then $\tau_{\varnothing jj'} = \tau_{\varnothing j}$. On the other hand, if $\tau_{\varnothing j} = 0$, then $\varnothing j$ has $|\mathcal{N}_X(\ell(\varnothing j)) \setminus \{i\}|$ children, namely $\varnothing jj'$, for $1 \leq j' \leq |\mathcal{N}_X(\ell(\varnothing j)) \setminus \{i\}|$ with $\tau_{\varnothing jj'} = 0$ and $\ell(\varnothing jj') \in \mathcal{N}_X(\ell(\varnothing j)) \setminus \{i\}$ in the way that if $j' \neq j''$, then $\ell(\varnothing jj') \neq \ell(\varnothing jj'')$. Continuing this process, the parent of a vertex $\varnothing j_1 j_2 ... j_{l-1} j_l$, in level l is the vertex $\varnothing j_1 j_2 ... j_{l-1}$. If $\tau_{\varnothing j_1 j_2 ... j_{l-1} j_l} \neq 0$, then $\varnothing j_1 j_2 ... j_{l-1} j_l$ has $n - |dig(T_{\varnothing j_1 j_2 ... j_{l-1} j_l})|$ children, called $\varnothing j_1 j_2 ... j_l j_{l+1}$, with $\ell(\varnothing j_1 j_2 ... j_{l-1} j_l j_{l+1}) \in [n] \setminus dig(T_{\varnothing j_1 j_2 ... j_{l-1} j_l})$ such that there is a bijection between set

$$\{\ell(\varnothing j_1 j_2 ... j_l j_{l+1}) : 1 \leq j_{l+1} \leq n - |dig(T_{\varnothing j_1 j_2 ... j_{l-1} j_l})|\}$$

and $[n] \setminus dig(T_{\varnothing j_1 j_2 ... j_{l-1} j_l})$. If $\ell(\varnothing j_1 j_2 ... j_l j_{l+1}) \notin \mathcal{N}_X(\ell(\varnothing j_1 j_2 ... j_l))$ then $\tau_{\varnothing j_1 j_2 ... j_l j_{l+1}} = \tau_{\varnothing j_1 j_2 ... j_l} - 1$, and if $\ell(\varnothing j_1 j_2 ... j_l j_{l+1}) \in \mathcal{N}_X(\ell(\varnothing j_1 j_2 ... j_l))$ then $\tau_{\varnothing j_1 j_2 ... j_l j_{l+1}} = \tau_{\varnothing j_1 j_2 ... j_l}$. Now, in the case that $\tau_{\varnothing j_1 j_2 ... j_l} = 0$, the children of $\varnothing j_1 j_2 ... j_l$ are labeled by $\mathcal{N}_X(\ell(\varnothing j_1 j_2 ... j_l)) \setminus dig(T_{\varnothing j_1 j_2 ... j_l})$, as in Algorithm 1. After a finite number of steps, we construct the tree denoted by $\mathcal{T}^i_{X,\mathcal{O}}$ (or $\mathcal{T}^i_{X,\alpha}$ for general $\alpha \geq 0$) such that each permutation given by a leaf in the level $n-1$ is an $\mathcal{O}_n(X)-$freedom permutation ($\alpha-$freedom permutation, respectively). We denote $\mathcal{T}_{X,\mathcal{O}} := \{\mathcal{T}^i_{X,\mathcal{O}} : \ 1 \leq i \leq n\}$, and $\mathcal{T}_{X,\alpha} := \{\mathcal{T}^i_{X,\alpha} : \ 1 \leq i \leq n\}$. We also let $Y(\mathcal{T}^i_{X,\mathcal{O}})$ be the set of all permutations $y \in S_n$ that are given by a leaf of $\mathcal{T}^i_{X,\mathcal{O}}$ in the level $n-1$, and let $Y(\mathcal{T}_{X,\mathcal{O}}) := \cup_{i=1}^n Y(\mathcal{T}^i_{X,\mathcal{O}})$. The definitions of $Y(\mathcal{T}^i_{X,\alpha})$, and $Y(\mathcal{T}_{X,\alpha})$ are similar. The construction of such trees is described in the following Algorithm 3, for general $\alpha \geq 0$.

Not only does Algorithm 3 give all $\mathcal{O}_n(X)-$freedom permutations but also for each possible permutation in the level $n-1$, the parameter τ indicates the exact number of adjacencies of the permutation from outside of $\cup_{x \in X} \mathcal{A}_x$, e.g., if $\tau_u = i$ then $(\mathcal{O}_n(X) - i)$ adjacencies are from outside in T_u. The trees constructed from Algorithm 3 have as subtrees the trees given by Algorithm 1, considering the same set of permutations. An example is given in Figure 2.

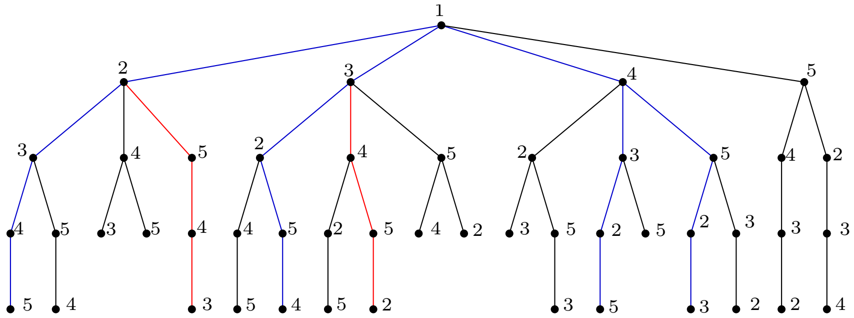

Figure 2 Representation of $\mathcal{T}^1_{X,\mathcal{O}}$, for $X = \{id = 12345, 52341, 23145\}$, where $\mathcal{O}_5(X) = 1$. The subtree induced by the blue edges is $\overline{\mathcal{T}}^1_X$ and the subtree induced by the blue and red edges is $\mathcal{T}^1_X$. The median value of X is $\mu(X) = 4$ and 14523 is the unique median given by the tree $\mathcal{T}^1_{X,\mathcal{O}}$ which is different from the input permutations. In this example, all medians given by the tree $\mathcal{T}^1_{X,\mathcal{O}}$ are actually in the subtree $\overline{\mathcal{T}}^1_X$. Also, 13254 is an example of a permutation in the set $\{y \in S_n; \mathcal{A}_X \subset \mathcal{A}_y \subset \bigcup_{x \in X} \mathcal{A}_x\}$ that is not a median for X.

4 Experimental results

For each n from 6 to 15, we performed 100 independent runs of Algorithm 3 on a set $X = \{x_1, x_2, x_3\} \in S_n$ of three permutations, where $x_1 = $ id and x_2, x_3 are randomly generated such that $\mathcal{O}_n(X) \leq 3$. For each n, we compute the mean of the normalized median value. As shown in Figure 3, the mean of the normalized median value increases with n, and we expect it to approach 2 as $n \to \infty$, which is in accordance with the last Theorem in [9].

Although not all the permutations in $Y(\mathcal{T}_{X,\mathcal{O}})$ are medians, we find that non-median permutations in $Y(\mathcal{T}_{X,\mathcal{O}})$ often have total distances close to the median value, indicating that they serve as good approximations. To formalize this, we define

$$K_j(\mathcal{T}_{X,\mathcal{O}}) := \{y \in Y(\mathcal{T}_{X,\mathcal{O}}); \sum_{x \in X} d(y,x) - \mu(X) = j\},$$

Algorithm 3 α-freedom permutations w.r.t. X.

```
Data: $X = \{x_1, ..., x_k\} \subset S_n$.
Result: $\alpha$-freedom permutations w.r.t. $X$
for $i = 1, ..., n$ do
    for $w = 0, ..., n$ do
        $L_w \longleftarrow \emptyset$
    $L_0 \longleftarrow \{\varnothing\}$
    $T_\varnothing \longleftarrow i$
    $\tau_\varnothing \longleftarrow \alpha$
    for $j = 1, ..., n-1$ do
        for $u \in L_{j-1}$ do
            $r \longleftarrow 0$
            if $\tau_u > 0$ then
                for $k = 1, ..., n$ do
                    if $k \notin dig(T_u)$ then
                        $r \longleftarrow r + 1$
                        $L_j \longleftarrow L_j \cup \{u \oplus r\}$
                        $T_{u \oplus r} \longleftarrow T_u \oplus k$
                        if $k \notin \mathcal{N}_X(T_u)$ then
                            $\tau_{u \oplus r} \longleftarrow \tau_u - 1$
                        else
                            $\tau_{u \oplus r} \longleftarrow \tau_u$
            else
                for $k \in \mathcal{N}_X(T_u)$ do
                    if $k \notin dig(T_u)$ then
                        $r \longleftarrow r + 1$
                        $L_j \longleftarrow L_j \cup \{u \oplus r\}$
                        $T_{u \oplus r} \longleftarrow T_u \oplus k$
    for $u \in L_{n-1}$ do
        print $T_u$
```

and compute the mean of the proportion $|K_j(\mathcal{T}_{X,\mathcal{O}})|/|Y(\mathcal{T}_{X,\mathcal{O}})|$ over 100 runs for each $6 \leq n \leq 15$. Note that $K_0(\mathcal{T}_{X,\mathcal{O}}) = M(X)$ and for small $j > 0$ we can consider the permutations in $K_j(\mathcal{T}_{X,\mathcal{O}})$ as approximate medians, since the total distance is close to the minimum total distance.

Figure 4 shows that a significant portion of permutations in $Y(\mathcal{T}_{X,\mathcal{O}})$ have total distances concentrated near the minimum, indicating that while most are not exact medians, many are close approximations.

In fact, across all tested values of n, the union $K_0 \cup K_1 \cup K_2$ consistently contains over 33% of $Y(\mathcal{T}_{X,\mathcal{O}})$, confirming the abundance of near-optimal solutions in the reduced space. For example, at $n = 6$, this set accounts for 61.9% of candidates; for $n = 12$, it still covers 36.4% despite the increase in size. Table 1 summarizes these proportions numerically for selected values of n.

To analyze the proportion of medians far from the input set, we denote by $M_i := \{m \in M(X); d(m, x_k) \geq i$, for $x_k \in X\}$. Note that $M_0 = M(X)$, $M_i \subset M_l$ for $l < i$, and M_i is empty set for $i > 2n/3$. Figure 5 shows the mean of the ratio of $|M_i|/|M(X)|$, for $6 \leq n \leq 15$.

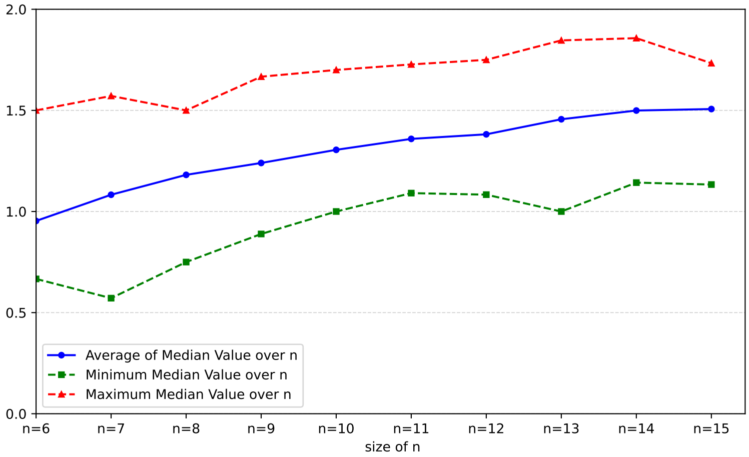

Figure 3 The blue line represents the mean normalized median value for sets of three permutations $\{id, x_2, x_3\}$, where x_2 and x_3 are randomly and independently sampled (also independently for each run) such that $\mathcal{O}_n(\{id, x_1, x_2\}) \leq 3$. The red and green lines indicate the minimum and maximum normalized median values observed across 100 independent runs of Algorithm 3, for each genome size $n = 6, ..., 15$.

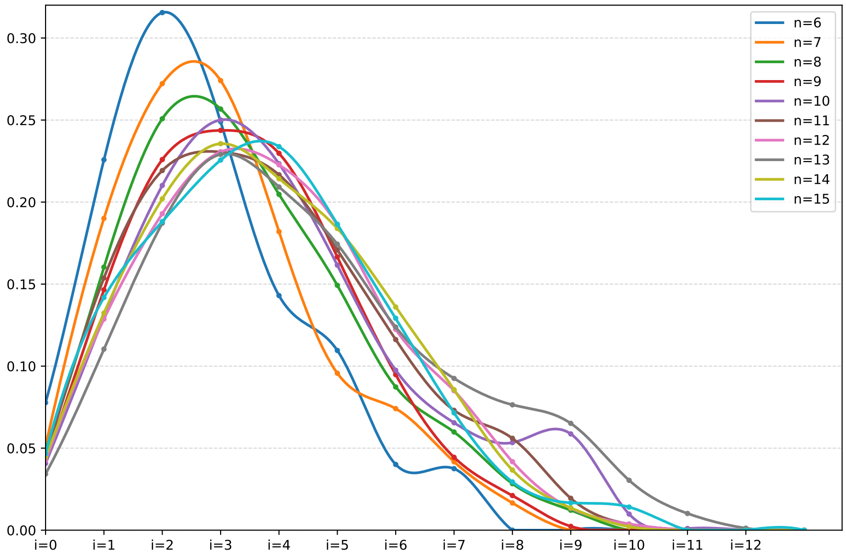

Figure 4 The mean of $|K_i(\mathcal{T}_{X,\mathcal{O}})|/|Y(\mathcal{T}_{X,\mathcal{O}})|$, for each $6 \leq n \leq 15$.

The results indicate that the proportion of medians far from all input permutations decreases rapidly, consistent with the observations and conjectures of Haghighi and Sankoff [8]. For example, when $n = 12$, over 91% of medians are within distance 3 of all inputs, and fewer than 0.8% exceed distance 6. This illustrates the general trend that most medians tend to remain close to at least one input genome.

However, as n increases, the number of medians that lie far from all inputs also grows. Table 2 reports the proportion of medians lying in M_i for values of i near $\left\lfloor \frac{2n}{3} \right\rfloor$, which corresponds to the breakpoint distance of a "midpoint" genome – that is, one that draws approximately one-third of its adjacencies from each of the three input genomes. As expected,

Table 1 Proportion of permutations in $K_0 \cup K_1 \cup K_2$ over $Y(\mathcal{T}_{X,\mathcal{O}})$ for selected values of n.

n	$\lvert K_0\rvert$	$\lvert K_1\rvert$	$\lvert K_2\rvert$	Total proportion (%)
6	7.77%	22.58%	31.55%	61.9%
8	4.91%	16.03%	25.08%	46.0%
10	4.07%	13.04%	21.01%	38.1%
12	4.25%	12.87%	19.29%	36.4%
14	4.46%	13.23%	20.20%	37.9%

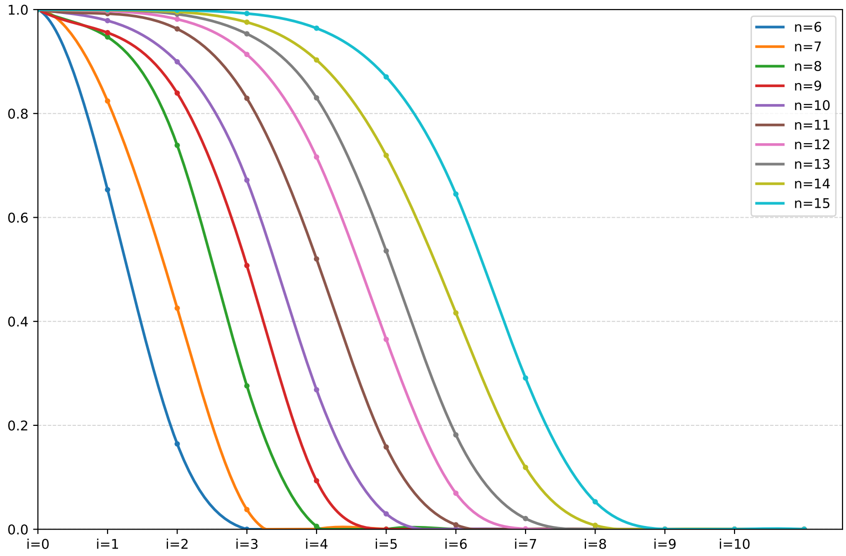

Figure 5 The mean of $|M_i|/|M(X)|$, for each $6 \leq n \leq 15$.

the proportion of medians with distance at least $\lfloor \frac{2n}{3} \rfloor$ from all inputs is either zero or negligible across all tested values of n, reflecting the rarity of truly equidistant medians. Still, for slightly smaller values such as $\lfloor \frac{2n}{3} \rfloor - 1$, $\lfloor \frac{2n}{3} \rfloor - 2$, or $\lfloor \frac{2n}{3} \rfloor - 3$, the proportion increases noticeably. For instance, when $n = 14$, the set M_6, consisting of medians at distance at least 6 from all three inputs, contains more than 11% of all medians, and M_5 contains over 41%. These medians are still far from each input genome – at least 5 breakpoints away – yet appear with consistent frequency, indicating a non-negligible presence near the midpoint region as n increases.

To quantify the algorithm's efficiency, we compare the size of the reduced space to $n!$. Table 3 demonstrates that the number of permutations explored by Algorithm 3 represents only a tiny fraction of S_n, yet suffices to find all exact and many near-optimal medians. For instance, when $n = 15$, the number of candidate medians generated by the algorithm – i.e., the search space – is less than 0.003% of the full $15! \approx 1.31 \times 10^{12}$ permutations.

Although Algorithm 3 was run with $\mathcal{O}_n(X) \leq 3$, allowing up to three adjacencies outside the union of the input adjacencies, we observed that such instances were extremely rare – and when they occurred, each involved only a single external adjacency. For example, at $n = 6$, only 0.04 medians per run (roughly 0.35% of all medians) included one adjacency not present in the union of the inputs. At $n = 12$, the mean was 0.48 per run (under 0.015%). For all other values of $n \leq 15$, there was no external adjacency. These results indicate that

Table 2 Mean proportion of medians in M_i for values of i near the midpoint distance $\left\lfloor \frac{2n}{3} \right\rfloor$.

n	$i = \left\lfloor \frac{2n}{3} \right\rfloor$	$i = \left\lfloor \frac{2n}{3} \right\rfloor - 1$	$i = \left\lfloor \frac{2n}{3} \right\rfloor - 2$	$i = \left\lfloor \frac{2n}{3} \right\rfloor - 3 \geq 4$
7	0	3.8%	42.54%	-
8	0	0.57%	27.59%	-
9	0	9.34%	50.74%	-
10	0	2.97%	26.85%	-
11	0	15.83%	52.02%	-
12	0	0.08%	6.94%	36.54%
13	0	2.06%	18.16%	53.56%
14	0	0.76%	11.90%	41.63%
15	0	0	5.29%	29.11%

Table 3 Reduction in search space by Algorithm 3 for selected values of n (with $\mathcal{O}_n(X) \leq 3$).

n	Total candidates	Total medians	$n!$	$\frac{\text{candidates}}{n!}$ (%)
6	180.94	11.46	720	25.13%
8	2981.10	82.86	40320	7.40%
10	24824.90	513.54	3628800	0.68%
12	353921.52	3387.82	4.79×10^8	0.07%
14	1882425.04	23815.54	8.72×10^{10}	0.002%
15	36659,718	48372.52	1.31×10^{12}	0.0028%

nearly all medians are already covered when we allow zero-freedom, that is, when every adjacency is drawn from the input genomes. In practice, therefore, we can use Algorithm 1, which corresponds to the zero-freedom version of Algorithm 3, to recover most of the medians while achieving substantial speed-ups. When no adjacency is taken from outside the union $\cup_{x \in X} \mathcal{A}_x$, the algorithm completes around 0.75 seconds and uses approximately 19.26 MB of memory for $n = 10$; about 40 seconds and 104.16 MB for $n = 13$; and around 3.5 minutes and 289.94 MB for $n = 15$ (mean runtime and memory usage over 5 runs, measured on a 2.3 GHz quad-core Intel Core i7 machine with 32 GB RAM).

Finally, although it is known that, given a set of genomes X, there may exist medians that do not contain all adjacencies in $\mathcal{A}_X$, we verified that for the input sets tested ($6 \leq n \leq 15$), all medians returned by Algorithm 3 contained the full set of common adjacencies $\mathcal{A}_X$ shared by the input genomes. As a result, Algorithm 3 produced the same set of medians as Algorithm 2 on all tested instances.

5 Conclusion

In this paper, we introduced a novel algorithmic framework to find all breakpoint medians of a given set of linear unsigned genomes. Unlike previous methods – which reduce the breakpoint median problem to an instance of the Traveling Salesman Problem (TSP) and return only a single median – our approach is based on the construction of rooted, labeled trees that allow us to find all medians, along with a substantial number of near-medians. Each path of length $n - 1$ from the root to a leaf encodes a unique permutation, and the tree structure is designed to efficiently capture the combinatorial space in which medians reside.

This structural strategy provides a new perspective on the median problem. It not only allows us to find all medians in exponential time, but also to systematically explore a constrained and meaningful subset of the permutation space. This is particularly valuable for comparative genomics, where the goal is often to infer an ancestral genome that minimizes evolutionary distance to the observed genomes. Having access to the entire set of medians makes it possible to evaluate and compare them based on additional biological or statistical criteria, such as similarity to known ancestral features or consistency with gene orientation and synteny.

From a theoretical point of view, we demonstrated that our method finds the exact median value, even in cases where prior methods could not. Experimentally, we showed that the number of candidate permutations generated by our trees is a vanishingly small fraction of the full symmetric group (e.g., less than 0.0028% of S_{15}), yet this restricted space reliably captures all medians and a large portion of near-optimal solutions. In particular, we found that a substantial fraction of permutations in the output tree fall into $K_0 \cup K_1 \cup K_2$, indicating that many are either exact or high-quality approximate medians. We also observed that even when allowing up to three adjacencies outside the input set, the inclusion of such external adjacencies was extremely rare, often occurring in fewer than 1% of medians.

Finally, we investigated how far medians tend to lie from all inputs using the M_i decomposition. While truly equidistant medians are rare, we found that a non-negligible proportion of medians are located near the theoretical midpoint region. Moreover, we observed that most medians are relatively close to the input permutations, an observation that aligns with theoretical results in the literature [8, 9, 4]. This suggests a layered structure in the space of medians that could be exploited for further biological modeling and inference.

While our work focuses on the breakpoint median problem for unsigned unichromosomal genomes, the algorithm and underlying methodology are not limited to this setting. The core tree-based construction and median search strategy naturally extend to more general models, including signed permutations and multichromosomal genomes. Overall, our method not only offers a new algorithmic contribution but also opens up a range of possibilities for deeper combinatorial and biological analysis of breakpoint medians and their role in gene order phylogeny.

References

1 Sylvia Boyd and Maryam Haghighi. A fast method for large-scale multichromosomal breakpoint median problems. *Journal of Bioinformatics and Computational Biology*, 10(01):1240008, 2012. doi:10.1142/S0219720012400082.

2 David Bryant. The complexity of the breakpoint median problem. *Centre de recherches mathematiques*, 1998.

3 Alberto Caprara. The reversal median problem. *INFORMS Journal on Computing*, 15(1):93–113, 2003. doi:10.1287/IJOC.15.1.93.15155.

4 Poly H da Silva, Arash Jamshidpey, and David Sankoff. Sampling gene adjacencies and geodesic points of random genomes. In *RECOMB International Workshop on Comparative Genomics*, pages 189–210. Springer, 2024. doi:10.1007/978-3-031-58072-7_10.

5 Poly H da Silva, Arash Jamshidpey, and David Sankoff. On the number of breakpoint medians of random genomes. *preprint (submitted)*, 2025.

6 Pedro Feijão and João Meidanis. SCJ: a variant of breakpoint distance for which sorting, genome median and genome halving problems are easy. In *International Workshop on Algorithms in Bioinformatics*, pages 85–96. Springer, 2009. doi:10.1007/978-3-642-04241-6_8.

7 G Fertin, A Labarre, I Rusu, E Tannier, and S Vialette. *Combinatorics of genome rearrangements.* The MIT Press, 2009.

8 Maryam Haghighi and David Sankoff. Medians seek the corners, and other conjectures. *BMC Bioinformatics*, 13(19):S5, 2012. doi:10.1186/1471-2105-13-S19-S5.

9 Arash Jamshidpey, Aryo Jamshidpey, and David Sankoff. Sets of medians in the non-geodesic pseudometric space of unsigned genomes with breakpoints. *BMC Genomics*, 15(6):S3, 2014.

10 Arash Jamshidpey and David Sankoff. Phase change for the accuracy of the median value in estimating divergence time. *BMC Bioinformatics*, 14(15):S7, 2013. doi:10.1186/1471-2105-14-S15-S7.

11 Caroline Anne Larlee, Chunfang Zheng, and David Sankoff. Near-medians that avoid the corners; a combinatorial probability approach. *BMC Genomics*, 15(6):S1, 2014.

12 Mona Meghdari Miardan, Arash Jamshidpey, and David Sankoff. Escape from parsimony of a double-cut-and-join genome evolution process. *Journal of Computational Biology*, 30(2):118–130, 2023. doi:10.1089/CMB.2021.0468.

13 David Sankoff and Mathieu Blanchette. The median problem for breakpoints in comparative genomics. *Computing and Combinatorics*, pages 251–263, 1997. doi:10.1007/BFB0045092.

14 David Sankoff, Gopalakrishnan Sundaram, and John Kececioglu. Steiner points in the space of genome rearrangements. *International Journal of Foundations of Computer Science*, 7(01):1–9, 1996. doi:10.1142/S0129054196000026.

15 Eric Tannier, Chunfang Zheng, and David Sankoff. Multichromosomal median and halving problems under different genomic distances. *BMC Bioinformatics*, 10(1):120, 2009. doi:10.1186/1471-2105-10-120.

16 Andrew Wei Xu. The median problems on linear multichromosomal genomes: Graph representation and fast exact solutions. *Journal of Computational Biology*, 17(9):1195–1211, 2010. doi:10.1089/CMB.2010.0106.

17 João Paulo Pereira Zanetti, Priscila Biller, and João Meidanis. Median approximations for genomes modeled as matrices. *Bulletin of Mathematical Biology*, 78:786–814, 2016.

18 Chunfang Zheng and David Sankoff. On the pathgroups approach to rapid small phylogeny. *BMC Bioinformatics*, 12(1):S4, 2011. doi:10.1186/1471-2105-12-S1-S4.

A Proof of Theorem 1

Below, we include the proof of Theorem 1, as presented in [5].

Proof. For a permutation π and $r \leq k$, let $\bar{\varepsilon}^X_{i_1,\dots,i_r}(\pi) := |\mathcal{A}_\pi \cap \mathcal{B}^X_{x_{i_1},\dots x_{i_r}}|$. To ease the notation, we let $\mathcal{B}_{i_1,\cdots,i_\ell} = \mathcal{B}_{x_{i_1},\dots,x_{i_\ell}}$. Let $\eta = |\mathcal{A}_m \setminus \cup_{i=1}^k \mathcal{A}_{x_i}|$. Then

$$\eta + \sum_{r=1}^{k} \sum_{1 \leq i_1 < \dots < i_r \leq k} \bar{\varepsilon}^X_{i_1,\dots,i_r}(m) = n - 1.$$

As m is a median of X, we have

$$\begin{aligned}
&d_T(m, X) = k(n-1) - \sum_{r=1}^{k} [r \sum_{1 \leq i_1 < \dots < i_r \leq k} \bar{\varepsilon}^X_{i_1,\dots,i_r}(m)] \\
&= (k-1)(n-1) + \eta - \sum_{r=2}^{k} [(r-1) \sum_{1 \leq i_1 < \dots < i_r \leq k} \bar{\varepsilon}^X_{i_1,\dots,i_r}(m)] \\
&\leq d_T(x_k, X) = (k-1)(n-1) - (\sum_{1 \leq i_1 < k} |\mathcal{B}^X_{i_1,k}| + 2 \sum_{1 \leq i_1 < i_2 < k} |\mathcal{B}^X_{i_1,i_2,k}| \\
&\qquad + \dots + (k-2) \sum_{1 \leq i_1 < \dots < i_{k-2} < k} |\mathcal{B}^X_{i_1,\dots,i_{k-2},k}| + (k-1)|\mathcal{B}^X_{1,\dots,k}|).
\end{aligned}$$

Hence,

$$\begin{aligned}
&\eta \leq \\
&(\sum_{r=2}^{k}(r-1)\sum_{1\leq i_1<\ldots<i_r\leq k}\bar{\varepsilon}^{X}_{i_1,\ldots,i_r}(m)) - (\sum_{r=2}^{k}(r-1)\sum_{1\leq i_1<\ldots<i_{r-1}<k}|\mathcal{B}^{X}_{i_1,\ldots,i_{r-1},k}|) \\
&\qquad \leq \sum_{r=2}^{k-1}(r-1)\sum_{1\leq i_1<\ldots<i_r<k}|\mathcal{B}^{X}_{i_1,\ldots,i_r}|, \qquad (2)
\end{aligned}$$

where the last inequality holds because $\bar{\varepsilon}^{X}_{i_1,\ldots,i_r}(m) \leq |\mathcal{B}^{X}_{i_1,\ldots,i_r}|$, for any $r \leq k$ and $1 \leq i_1 < \ldots < i_r \leq k$. ◀