Median clouds and a fast transposition median solver

Niklas Eriksen

FPSAC 2009, Hagenberg, Austria DMTCS proc. **AK**, 2009, 373–384

Median clouds and a fast transposition median solver

Niklas Eriksen[1]

[1] ner<at>chalmers.se, *Mathematical Sciences, Göteborg University and Chalmers University of Technology, S-412 96 Göteborg, Sweden*

Abstract. The median problem seeks a permutation whose total distance to a given set of permutations (the base set) is minimal. This is an important problem in comparative genomics and has been studied for several distance measures such as reversals. The transposition distance is less relevant biologically, but it has been shown that it behaves similarly to the most important biological distances, and can thus give important information on their properties. We have derived an algorithm which solves the transposition median problem, giving all transposition medians (the median cloud). We show that our algorithm can be modified to accept median clouds as elements in the base set and briefly discuss the new concept of median iterates (medians of medians) and limit medians, that is the limit of this iterate.

Résumé. Le problème de la médiane est de trouver une permutation dont la distance totale à un ensemble donné de permutations (l´ensemble de base) est minimale. C'est un problème important en génomique comparative et il a été étudié pour certaines mesures de distance. La distance de transposition n'est pas directement liée à la biologie, mais il a été démontré que son comportement est similaire à celui des distances biologiques essentielles, et elle peut donc donner des indications sur leurs propriétés. Nous construisons un algorithme qui résoud le problème de la médiane pour la transposition, et donne toutes les transpositions médianes (le nuage des médianes). Nous démontrons que notre algorithme peut être modifié pour admettre des nuages de médianes comme éléments de l´ensemble de base et introduisons le concept de médianes itérées (médianes de médianes) et de médianes limites, c-à-d de limites de ces itérations.

Keywords: median, transposition, reversal, DCJ, median cloud

1 Introduction

The median problem in comparative genomics calls for a permutation such that the total distance to a given set S of permutations is minimised. Using the permutations in S as models for some species' genomes, by regarding the genome as a permutation of the genes therein, the median permutation is an approximation of the gene order of these species' closest ancestor. Using median computations, biologists can infer phylogenetic trees, which show how different species are related (8; 2; 7; 1).

The gene order typically changes in a species by **reversals**, where a segment is taken out and inserted backwards at the same place (changing for instance 1234567 to 1543267), **block transpositions**, where a segment is taken out and inserted, possibly backwards, at another place (changing 1234567 to 1456237,

for instance), or **Double Cut and Join** (DCJ), which generalise reversals to genomes with several chromosomes, noting that a reversal can be seen as cutting the genome in two places and then putting it together again). Usually one also attaches a sign (+/-) to each gene, changing the sign of every gene in a reversed segment to indicate that the reading directions of these genes have been flipped. Distances are measured in the number of operations (reversals, block transposition, DCJ or combinations) needed to transform one permutation into another. There are also simpler distances, such as the number of elements in one permutation which are followed by different elements in the two permutations under comparison. Such positions are called **breakpoints**.

Depending on which distance measure we use, the median problem may be easy or hard. However, for all these distances, including the simple breakpoint distance, the median problem is NP-hard, see (4; 9; 11; 10) and references in the latter. Thus, variations on this problem which could shed some light on how to simplify it are most welcome.

In this paper, we consider the median problem under the usual transposition distance (exchanging positions of any two elements). While this operation has no relevance in genomic development, the distance function behaves very similarly to the reversal and the DCJ distances for signed genomes (6), which both take the number of genes, subtract the number of cycles and then add some more terms which for most permutations are zero (3). Studying the transposition median would therefore be regarded as a somewhat simpler version of the reversal median and the DCJ median.

We give a branch and bound algorithm which computes the transposition median. This algorithm resembles algorithms for the reversal median (4) and the DCJ median (12; 11) and has a comparable running time. We conjecture that the transposition median problem is NP-hard as well and expect that this can be proved by using the same techniques as Caprara, but it does not seem trivial to change his proof for undirected graphs into a similar proof for directed graphs.

Interestingly, this algorithm gives all transposition medians. Previous studies of the transposition median have explained why transposition medians, and medians in general, are not unique when the base set is fairly separated (5). We now consider the entire set of medians, here called the **median cloud**, and try to extract more information from it than we would get from any single median. We also revise the algorithm to accept median clouds in the base set, instead of only permutations.

First, we consider what happens when we compute the median of a median cloud. There are reasons to believe that the second median would be closer to the true ancestor, and this is also the case, even though the difference is not large. Iterating *ad infinitum*, we obtain the limit median, in case of convergence. We give some results on the appearance of the limit median.

Second, we give an example on how median clouds can be used to enhance computations of inner nodes in a given phylogeny. We show that methods based on median cloud in general outperform methods based on a single genome.

There are good reasons to believe that median solvers for other distances (breakpoints, reversals, DCJ) can be extended to compute median clouds and accepting them in their base sets. We are thus confident that our results will improve on biologically relevant median computations.

2 Background and definitions

Let $S = \{\pi_1, \pi_2, \ldots, \pi_k\}$, $\pi_i \in \mathfrak{S}_n$, be a set of permutations called the **base set**. We will use both one line notation (for example $\pi = 3412$) and cycle notation ($\pi = (13)(24)$). Unless otherwise stated, k is the number of elements in S and n is the length of the permutations. Given any distance function $d(\cdot, \cdot)$

between two permutations, the distance between a permutation $\pi \in \mathfrak{S}_n$ and S is defined to be

$$d(\pi, S) = \sum_i d(\pi, \pi_i)$$

and a **median** is any $\mu \in \mathfrak{S}_n$ which minimises $d(\mu, S)$. The set of medians is denoted $M(S)$ and we let $d(S) = d(\mu, S)$ for $\mu \in M(S)$. The choice of distance measure $d(\cdot, \cdot)$ gives rise to several interesting median problems; in this article we focus on the **transposition median problem** (TMP).

It is well known that the following bounds for $d(S)$ hold under any metric distance.

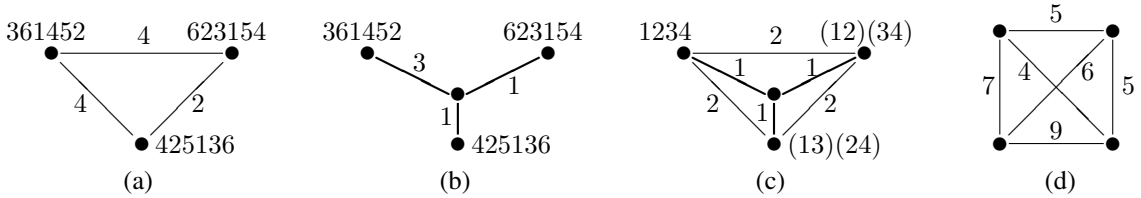
Lemma 2.1 *For any distance measure $d(\cdot, \cdot)$, the median distance $d(S)$ for $S = \{\pi_1, \ldots, \pi_k\}$ is bounded by*

$$\frac{\sum_{i<j} d(\pi_i, \pi_j)}{k-1} \leq d(S) \leq \min_i \sum_j d(\pi_i, \pi_j).$$

Proof: For the lower bound, we note that by the triangle inequality, $d(\pi_i, \pi_j) \leq d(\mu, \pi_i) + d(\mu, \pi_j)$, and hence $\sum_{i<j} d(\pi_i, \pi_j) \leq (k-1) \sum_i d(\mu, \pi_i)$. The upper bound is the minimum of $d(\pi_i, S)$. □

We note that the upper bound gives a $(2 - 2/k)$-approximation of $d(S)$, and hence the median problem is trivial for $k \leq 2$, as expected. In addition, since the transposition distance changes parity for every transposition applied, we can always assign edge lengths in the tree with three (or less) genomes as leaves and a single inner node, which attains the lower limit without breaking the triangle inequality. However, we can not always find a median which attains the lower bound. For $k \geq 4$, the lower limit is only rarely realisable in the tree without breaking the triangle inequality.

Example 2.1 *Consider the three permutations in (a) with given transposition distances. The tree which attains the lower limit can be found in (b). In this case, the unique median which attains the lower limit is $\mu = 423156$. On the other hand, the base set S in (c) has $d_{\mathrm{trp}}(S)$ strictly larger than the lower limit; in fact, $M(S) = S$, giving $d_{\mathrm{trp}}(S) = 4$, while the lower limit is 3. In (d), with the distances given, the lower limit of 12 is clearly not attainable; indeed, the top and bottom edges demand $d_{\mathrm{trp}}(S) \geq 14$.*



In the following, the term **graph** refers to edge coloured directed graphs $G = (V(G), E(G))$, unless otherwise stated. An edge from v_1 to v_2 of colour j is denoted $(v_1 \xrightarrow{j} v_2)$. By $\deg_{\mathrm{in}}(G, v, j) = |\{u : (u \xrightarrow{j} v)\}|$ and $\deg_{\mathrm{out}}(G, v, j) = |\{u : (v \xrightarrow{j} u)\}|$ we denote the in/out-degree of colour j at vertex v. The number of edges from u to v in G is denoted $|(u \longrightarrow v)|_G$, suppressing G if no confusion can arise. An **alternating path** with colours c_1 and c_2 in a graph G is a sequence of vertices $v_1, v_2, \ldots, v_{2m}$ such that G contains edges $(v_{2i-1} \xrightarrow{c_1} v_{2i})$ coloured c_1 for $1 \leq i \leq m$ and edges $(v_{2i} \xrightarrow{c_2} v_{2i+1})$ coloured c_2

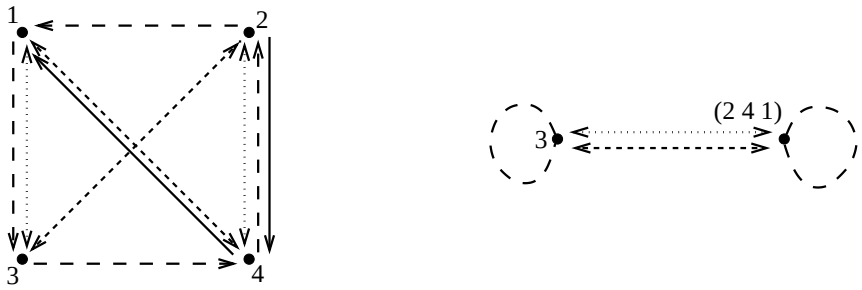


Fig. 1: A cycle graph $G(S, \mu_2)$ with $S = \{3142, 3412, 4321\}$ and $\mu_2 = \cdot 4 \cdot 1$, where a dot at position i indicates that $i \notin A_2$, and its reduced graph $\mathrm{red}(G)$.

for $1 \leq i \leq m-1$, and $j = i + 2a \Rightarrow v_i \neq v_j$ for $a > 0$. A **maximal alternating path** is an alternating path which cannot be elongated, and an **alternating cycle** is an alternating path with $v_{2m-1} = v_1$.

Let μ_b be a partial permutation with b values, that is μ_b is an injective map from $A_b \subseteq [n]$ to $[n]$ with $|A_b| = b$. Our algorithm will give a sequence $\{\mu_1, \mu_2, \ldots, \mu_n\}$ such that $A_1 \subset A_2 \subset \cdots \subset A_n$ and $\mu_c(j) = \mu_b(j)$ for $c \leq b$ and $j \in A_c$. We thus tacitly assume that if $\mu_b(u) = v$, then $\mu_j(u) = v$ for all $j \geq b$. Any μ_n fulfilling this criterion for a given μ_b is called a **completion** of μ_b.

The **cycle graph** of S, $G = G(S, \mu_b)$, is a graph on n vertices labelled $1, \ldots, n$, with $(v_1 \xrightarrow{j} v_2)$ if $\pi_j(v_1) = v_2$. It corresponds to the breakpoint graph which is often considered when studying reversal distance problems, but has directed edges instead of undirected. The cycle graph also contains b edges of colour $k+1$, from here on called **black**, which indicate the inverse of the partial permutation μ_b: we have $(v_1 \xrightarrow{k+1} v_2)$ if $\mu_b(v_2) = v_1$. We may conclude that for all $v \in V(G)$ we have $\deg_{\mathrm{in}}(G, v, j) = \deg_{\mathrm{out}}(G, v, j) = 1$ for $1 \leq j \leq k$, and also $\deg_{\mathrm{in}}(G, v, k+1) \leq 1$ and $\deg_{\mathrm{out}}(G, v, k+1) \leq 1$.

Given a cycle graph $G = G(S, \mu_b)$ with b black edges, the **reduced cycle graph** $G' = \mathrm{red}(G)$ is a graph defined as follows. For each maximal path $(v_1 \xrightarrow{k+1} v_2 \xrightarrow{k+1} \cdots \xrightarrow{k+1} v_m)$ in G, we get the vertex $(v_1 v_2 \ldots v_m)$ in G'. For each maximal alternating path $(v_1 \xrightarrow{j} v_2 \xrightarrow{k+1} \cdots \xrightarrow{j} v_{2m})$ in G, we add the edge $(v_1 \ldots) \xrightarrow{j} (\ldots v_{2m})$ in G'. We note that since the alternating path is maximal, there is no black edge going to v_1; hence v_1 is the first vertex in the black path giving the vertex $(v_1 \ldots) \in V(G')$, and similarly v_{2m} is the last vertex in its black path. We thus observe that the reduced cycle graph $G' = \mathrm{red}(G)$ is a cycle graph on $n - b$ vertices.

Example 2.2 *Consider the cycle graph G to the left in Figure 1, with $k = 3$ and two black edges. With black edges $(2 \xrightarrow{4} 4 \xrightarrow{4} 1)$, we get the vertex $(2\ 4\ 1)$ in $\mathrm{red}(G)$ to the right in the figure. With the long dashes as colour 1, we get the maximal alternating paths $(3 \xrightarrow{1} 4 \xrightarrow{4} 1 \xrightarrow{1} 3)$ and $(2 \xrightarrow{1} 1)$ in G, giving edges $(3 \xrightarrow{1} 3)$ and $((2\ 4\ 1) \xrightarrow{1} (2\ 4\ 1))$ in $\mathrm{red}(G)$.*

3 Efficient bounds on $d_{\mathrm{trp}}(\mu, S)$ and optimal assignments

The transposition distance between any two permutations $\sigma, \tau \in \mathfrak{S}_n$ is easy to compute using this classical theorem.

Theorem 3.1 *The transposition distance between $\sigma \in \mathfrak{S}_n$ and $\tau \in \mathfrak{S}_n$ is given by*

$$d_{\mathrm{trp}}(\sigma, \tau) = n - c(\sigma^{-1}\tau),$$

where $c(\pi)$ is the number of cycles in π.

The standard proof uses the fact that any transposition $(a\ b)$ will either merge the two cycles in $\sigma^{-1}\tau$ containing a and b, respectively, or split the cycle containing both a and b. In addition, $\sigma^{-1}\tau$ has n cycles if and only if $\sigma = \tau$.

The distance between π_i and π_j in S can of course be computed directly from $G(S, \mu_0)$. Each cycle in $\pi_i^{-1}\pi_j$ corresponds to an alternating cycle with colours i and j, provided that all edges $(v_1 \xrightarrow{\ i\ } v_2)$ are flipped into $(v_2 \xrightarrow{\ i\ } v_1)$. This alternating cycle may also be written $(v_1 \xleftarrow{\ i\ } v_2 \xrightarrow{\ j\ } v_3 \xleftarrow{\ i\ } \cdots \xrightarrow{\ j\ } v_1)$. We thus have $c(\pi_i^{-1}\pi_j) = c(G, i, j)$, where $c(G, i, j)$ is the number of alternating cycles with colours i and j in G, provided that the edges coloured i are flipped.

Similarly, the distance between the black coloured median $\mu = \mu_n$ and any base permutation π_i is given by the number of alternating cycles with colours $k+1$ and i, not flipping any edges since the black edges give μ^{-1}. We use $c(G, i)$ to denote this quantity. But we can also say something about this distance given only μ_b. To this end, we let $p(G, i)$ be the number of maximal alternating paths and cycles with colours $k+1$ and i in G.

Lemma 3.2 *Given a cycle graph $G = G(S, \mu_b)$, the transposition distance between any completion μ of μ_b and $\pi_i \in S$ satisfies*

$$d_{\mathrm{trp}}(\mu, \pi_i) \geq n - p(G, i).$$

Proof: The lemma is clearly true for $b = 0$, since $p(G(S, \mu_0), i) = n$, and for $b = n$, since $p(G(S, \mu_n), i) = c(G(S, \mu_n), i)$. But it is also clear that $p(G(S, \mu_{b-1}), i) - p(G(S, \mu_b), i) \in \{0, 1\}$, since adding a black edge will either turn a path into a cycle or unite two paths. □

Combining the previous lemma with the upper and lower bounds of Lemma 2.1, we get strong bounds on $d_{\mathrm{trp}}(\mu, S)$, given μ_b.

Lemma 3.3 *Let $S = \{\pi_1, \pi_2, \ldots, \pi_k\}$, and let $G = G(S, \mu_b)$ and $G' = \mathrm{red}(G)$. For any completion μ of μ_b, we have*

$$\frac{\sum_{i<j}((n-b) - c(G', i, j))}{k-1} \leq d_{\mathrm{trp}}(\mu, S) - \sum_i (n - p(G, i)) \leq \min_i \sum_j ((n-b) - c(G', i, j)).$$

Proof: It follows from Lemma 3.2 that $d_{\mathrm{trp}}(\mu, S) - \sum(n - p(G, i)) \geq 0$. This quantity is obtained by adding black edges to $G(S, \mu_b)$, or equivalently to $G' = \mathrm{red}(G(S, \mu_b))$. Since G' is a cycle graph, we can invoke Lemma 2.1, and this lemma follows. □

Example 3.1 *Returning to Figure 1, the transposition median distance of $G(S, \mu_0)$ is bounded by $(3 + 1 + 2)/2 \leq d_{\mathrm{trp}}(\mu, S) \leq (1 + 2)$, that is $3 \leq d_{\mathrm{trp}}(\mu, S) \leq 3$, and thus one permutation in the base set (the one marked with dots) actually gives a median. For $G(S, \mu_2)$, we have $(1 + 1 + 0)/2 \leq d_{\mathrm{trp}}(\mu, S) - (1 + 1 + 1) \leq 1 + 0$. Thus, any completion of the given μ_2 gives $d_{\mathrm{trp}}(\mu, S) \geq 4$. We have made at least one bad choice among the black edges.*

We can now make a couple of observations of the influence an added black edge has on the lower limit of $d_{\mathrm{trp}}(\mu, S)$.

Lemma 3.4 *Assume we set $\mu_{b+1}(v_2) = v_1$, that is we add the black edge $(v_1 \overset{k+1}{\longrightarrow} v_2)$ to $G(S, \mu_b)$, obtaining $G(S, \mu_{b+1})$. Then,*

$$\sum_i (p(G(S,\mu_b), i) - p(G(S,\mu_{b+1}), i)) = k - |((v_2 \ldots) \longrightarrow (\ldots v_1))|_{\mathrm{red}(G(S,\mu_b))}.$$

Proof: If there is an edge $((v_2 \ldots) \overset{i}{\longrightarrow} (\ldots v_1))$ in $\mathrm{red}(G(S, \mu_b))$, adding the edge will close an alternating path in $G(S, \mu_b)$ into a cycle in $G(S, \mu_{b+1})$, thus not changing p. Otherwise, two alternating paths in $G(S, \mu_b)$ will be united, reducing p by one. □

Lemma 3.5 *If the edges $((v_2 \ldots) \overset{c_1}{\longrightarrow} (\ldots v_1))$ and $((v_2 \ldots) \overset{c_2}{\longrightarrow} (\ldots v_1))$ both belong to $E(\mathrm{red}(G(S, \mu_b)))$, then letting $\mu_{b+1}(v_2) = v_1$ gives $c(\mathrm{red}(G(S, \mu_b)), c_1, c_2) - c(\mathrm{red}(G(S, \mu_{b+1})), c_1, c_2) = 1$.*

Proof: The alternating cycle $((v_2 \ldots) \overset{c_1}{\longrightarrow} (\ldots v_1) \overset{c_2}{\longleftarrow} (v_2 \ldots))$ in $\mathrm{red}(G(S, \mu_b))$ will have disappeared in $\mathrm{red}(G(S, \mu_{b+1}))$ and no other alternating cycles with colours c_1 and c_2 are affected. □

Lemma 3.6 *If $((v_2 \ldots) \overset{c_1}{\longrightarrow} (\ldots v_1)) \in E(\mathrm{red}(G(S, \mu_b)))$, but $((v_2 \ldots) \overset{c_2}{\longrightarrow} (\ldots v_1)) \notin E(\mathrm{red}(G(S, \mu_b)))$, then letting $\mu_{b+1}(v_2) = v_1$ does not change the number of cycles, that is $c(\mathrm{red}(G(S, \mu_b)), c_1, c_2) - c(\mathrm{red}(G(S, \mu_{b+1})), c_1, c_2) = 0$.*

Proof: With $u_1 = (\ldots v_1)$ and $u_2 = (v_2 \ldots)$, the alternating cycle $(u_0 \overset{c_2}{\longrightarrow} u_1 \overset{c_1}{\longleftarrow} u_2 \overset{c_2}{\longrightarrow} u_3 \overset{c_1}{\longleftarrow} \ldots \overset{c_1}{\longleftarrow} u_0)$ will be reduced to $(u_0 \overset{c_2}{\longrightarrow} u_3 \overset{c_1}{\longleftarrow} \ldots \overset{c_1}{\longleftarrow} u_0)$. All other alternating cycles are untouched. □

Lemma 3.7 *Let $u_1 = (\ldots v_1)$ and $u_2 = (v_2 \ldots)$. Assume that $(u_0 \overset{c_1}{\longrightarrow} u_1)$ and $(u_2 \overset{c_2}{\longrightarrow} u_3)$, where $u_0 \neq u_2, u_1 \neq u_3$, belong to $E(\mathrm{red}(G(S, \mu_b)))$, which gives that neither $(u_2 \overset{c_1}{\longrightarrow} u_1)$ nor $(u_2 \overset{c_2}{\longrightarrow} u_1)$ belong to $E(\mathrm{red}(G(S, \mu_b)))$. Then, letting $\mu_{b+1}(v_2) = v_1$ implies $c(\mathrm{red}(G(S, \mu_b)), c_1, c_2) - c(\mathrm{red}(G(S, \mu_{b+1})), c_1, c_2) = -1$ if $(u_0 \overset{c_1}{\longrightarrow} u_1 \overset{c_2}{\longleftarrow} u_4 \overset{c_1}{\longrightarrow} \cdots \overset{c_1}{\longrightarrow} u_3 \overset{c_2}{\longleftarrow} u_2 \overset{c_1}{\longrightarrow} u_5 \overset{c_2}{\longleftarrow} \cdots \overset{c_2}{\longleftarrow} u_0)$ is an alternating cycle of $\mathrm{red}(G(S, \mu_b))$ and 1 otherwise.*

Proof: If the alternating cycle $(u_0 \overset{c_1}{\longrightarrow} u_1 \overset{c_2}{\longleftarrow} u_4 \overset{c_1}{\longrightarrow} \cdots \overset{c_1}{\longrightarrow} u_3 \overset{c_2}{\longleftarrow} u_2 \overset{c_1}{\longrightarrow} u_5 \overset{c_2}{\longleftarrow} \cdots \overset{c_2}{\longleftarrow} u_0)$ exists, it will be split in two, namely $(u_0 \overset{c_1}{\longrightarrow} u_5 \overset{c_2}{\longleftarrow} \cdots \overset{c_2}{\longleftarrow} u_0)$ and $(u_4 \overset{c_2}{\longrightarrow} u_3 \overset{c_1}{\longleftarrow} \cdots \overset{c_1}{\longleftarrow} u_4)$. Otherwise, we have the two alternating cycles $(u_0 \overset{c_1}{\longrightarrow} u_1 \overset{c_2}{\longleftarrow} u_4 \overset{c_1}{\longrightarrow} \cdots \overset{c_2}{\longleftarrow} u_0)$ and $(u_5 \overset{c_1}{\longleftarrow} u_2 \overset{c_2}{\longrightarrow} u_3 \overset{c_1}{\longleftarrow} \cdots \overset{c_2}{\longrightarrow} u_3)$, which unite into $(u_0 \overset{c_1}{\longrightarrow} u_5 \overset{c_2}{\longleftarrow} \cdots \overset{c_1}{\longrightarrow} u_3 \overset{c_2}{\longleftarrow} u_4 \overset{c_1}{\longrightarrow} \cdots \overset{c_2}{\longleftarrow} u_0)$. Remaining alternating cycles are untouched. □

We are now way on our way to find $M(S)$. Using the above lemmata, we can control the lower limit of $d_{\mathrm{trp}}(S)$ as we add edges to μ_b.

Theorem 3.8 *Let $G = G(S, \mu_b)$ and $G' = \mathrm{red}(G)$. For $u_1 = (\dots v_1)$ and $u_2 = (v_2 \dots)$, assume that $j = |(u_2 \longrightarrow u_1)|_{G'}$ and that there are m alternating cycles in colours $1 \le c_1 < c_2 \le k$ with an odd number of edges between u_1 and u_2. Then, letting $\mu_{b+1}(v_2) = v_1$ will increase the lower limit of $d_{\mathrm{trp}}(S)$,*

$$\frac{\sum_{c_1<c_2}((n-b) - c(G', c_1, c_2))}{k-1} + \sum_{c_1}(n - p(G, c_1))$$

by

$$\delta(v_1, v_2) = \frac{2}{k-1}\left(\binom{k-j}{2} - m\right).$$

In particular, for $k = 3$, the integral lower limit stays unchanged for $j \ge 2$, increases by at most 1 for $j = 1$ and at most 3 for $j = 0$.

Proof: It is clear from Lemma 3.4 that $\sum(n - p(G, c_1))$ increases with $k - j$. Next, consider colour pairs $1 \le c_1 < c_2 \le k$. If $(u_2 \xrightarrow{c_1} u_1)$ and $(u_2 \xrightarrow{c_2} u_1)$ are both present in $E(G')$, Lemma 3.5 gives that $((n-b) - c(G', c_1, c_2))$ does not change. If only one of these edges is present, $((n-b) - c(G', c_1, c_2))$ decreases by 1 (Lemma 3.6). Finally, Lemma 3.7 says that if none of the edges are present, $((n-b) - c(G', c_1, c_2))$ decreases by 2 if the cycle passes both u_1 and u_2 with an odd number of edges in between, and stays unchanged otherwise.

Summing up, we get that the bound increases with

$$\delta(v_1, v_2) = (k-j) - \frac{(k-j)j + 2m}{k-1} = \frac{(k-j)^2 - (k-j) - 2m}{k-1} = \frac{2}{k-1}\left(\binom{k-j}{2} - m\right).$$

□

It is not obvious that adding an edge which does not increase the lower bound is optimal. In fact, it is not even true. However, there are some black edges which are guaranteed to be optimal.

Theorem 3.9 *Assume that μ_b can be completed to all medians in $M(S)$. If $|((v_2 \dots) \longrightarrow (\dots v_1))|_{\mathrm{red}(G(S,\mu_b))} > k/2$, then $\mu(v_2) = v_1$ for all $\mu \in M(S)$. If $|((v_2 \dots) \longrightarrow (\dots v_1))|_{\mathrm{red}(G(S,\mu_b))} = k/2$, then $\mu(v_2) = v_1$ for some $\mu \in M(S)$.*

Proof: Assume that a median μ has $\mu^{-1}(v_1) = v_3 \neq v_2$. If $((v_2 \dots) \xrightarrow{c_1} (\dots v_1)) \in E(\mathrm{red}(G(S, \mu)))$, then the alternating cycle $(v_2 \xrightarrow{c_1} \dots \xrightarrow{c_1} v_1 \xrightarrow{k+1} v_3 \xleftarrow{c_1} \dots \xrightarrow{k+1} v_2)$ will split if v_1 is redirected to v_2. Hence, $c(G(S, \mu \circ (v_2\ v_3))) - c(G(S, \mu), c_1) = 1$, and summing over all colours we get $d_{\mathrm{trp}}(\mu \circ (v_2\ v_3), S) < d_{\mathrm{trp}}(\mu, S)$, contradicting the fact that μ is a median. Similarly, if $|((v_2 \dots) \longrightarrow (\dots v_1))| = k/2$, we obtain a median $\mu \circ (v_2\ v_3)$ which satisfies $(\mu \circ (v_2\ v_3))(v_2) = \mu(v_3) = v_1$. □

4 A median solver

Based on the theorems in the previous section, we have devised a transposition median solver which gives all medians, that is $M(S)$, for any S. We start with $\mu_0 = 0$ and then make a depth first search through the space of μ_b. At any node μ_b in the search tree, if $|((v_2 \dots) \longrightarrow (\dots v_1))|_{\mathrm{red}(G(S,\mu_b))} > k/2$ then $\mu_{b+1}(v_2) = v_1$ is optimal. Otherwise, we search all subtrees of μ_b, stopping as soon as the lower

bound on that subtree rises above the lowest value on $d_{\mathrm{trp}}(\mu, S)$ found so far. A formal description of the algorithm Median can be found in Algorithm 1 (last page).

Algorithm 1 can of course be improved upon. For instance, to achieve a more effective pruning, we compute the increase of $d_{\mathrm{trp}}(\mu, S)$ for each assignment $\mu_{b+1}(v_2) = v_1$, where $(\dots v_1), (v_2 \dots) \in \mathrm{red}(G(S, \mu))$. Keeping v_2 fixed, it is clear that $d_{\mathrm{trp}}(\mu, S) \geq d_{\mathrm{trp}}(\mu_b, S) + \min_{v_1} \delta(v_1, v_2)$, since we are free to assign $\mu_{b+1}(v_2)$ at any stage. Similarly, keeping v_1 fixed, we have $d_{\mathrm{trp}}(\mu, S) \geq d_{\mathrm{trp}}(\mu_b, S) + \min_{v_2} \delta(v_1, v_2)$. This leads to more effective pruning. Our implementation in Matlab is available upon request. The speed of this implementation is comparable to the DCJ median solver by Xu (11).

5 Median clouds

Given the set of medians $M(S)$, there are some parts which are common to all medians and some parts which vary more or less between the medians. If a median is chosen at random from this set, the choice of the parts which vary between medians will be impossible to distinguish from the parts which are common between all medians, and they will probably effect later computations using this median. To minimise this effect, we would like to keep as much information as possible about $M(S)$ instead of just choosing a single median.

Since our median solver gives the complete median cloud $M(S)$, we would like to keep this cloud and use it in further calculation. In those calculations, this median cloud should play the role of a single permutation in a base set. How can we revise the median solver to accept such base sets, that is to compute the median of a base set of sets, $S = (S_1, S_2, \dots, S_k)$?

One method which seems tempting is to take the permutation matrices A_j of all permutations in each set S_i and compute their arithmetical mean, $\sum A_j/|S_i|$. However, since the algorithm requires not only the extent of which a set S_i maps v_1 to v_2, but also the alternating cycle structure, we lose too much information in this process. Instead, we need to consider each pair $\pi_1 \in S_i$ and $\pi_2 \in S_j$ separately.

To be more precise, we give each permutation $\pi \in S_i$ weight $w(\pi)$ such that $\sum_{\pi \in S_i} w(\pi) = 1$. Usually, $w(\pi) = |S_i|^{-1}$ will do. Then, it is easy to see that if we define

$$d^w(\mu, S) = \sum_i \sum_{\pi \in S_i} d(\mu, \pi) w(\pi),$$

a lower transposition median distance limit of any completion of μ_b is given by

$$\frac{\sum_{1 \leq i < j \leq k} \sum_{\pi \in S_i, \sigma \in S_j} ((n-b) - c(G', c_1, c_2)) w(\pi) w(\sigma)}{k-1} + \sum_i \sum_{\pi \in S_i} (n - p(G, c_1)) w(\pi).$$

We can thus use Algorithm 1 almost unchanged. We note, however, that the running time is proportional to $\max_i |S_i|^2$ in the worst case and median sets $M(S)$ grow fast when we scatter the base set. However, pruning may be more effective when median distances are given rational numbers instead of integers.

6 Limit medians

Medians and median clouds are often used to estimate the ancestor of three or more contemporary species. Medians are approximations of the ancestor and should "surround" the ancestor. This leads us to compute

the median of a median cloud, which could improve on the estimate of the ancestor, although not on the distance $d(\mu, S)$.

Definition 6.1 *Given a base set S, the kth* **median iterate** *of S is $M^k(S)$, where $M^k(S) = M(M^{k-1}(S))$. If the limit*

$$M^\infty(S) = \lim_{k\to\infty} M^k(S)$$

exists, we say that $M^\infty(S)$ is a **limit median**.

It is obvious that $M^\infty(S) = M(S)$ if $M(S)$ is a singleton. But what can be said if $|M(S)| > 1$?

Proposition 6.1 *If $S = \{\mathrm{id}, (1\ 2\ \ldots\ m)\}$, then $M(S)$ contains all permutations $\pi \in \mathfrak{S}_n$ such that $d_{\mathrm{trp}}(\pi, \mathrm{id}) + d_{\mathrm{trp}}(\pi, (1\ 2\ \ldots\ m)) = m - 1$.*

Proof: The assertion is given directly by the triangle inequality, since the set contains all permutation on a shortest path from id to $(1\ 2\ \ldots\ m)$. □

We conjecture based on extensive calculations that for $S = \{\mathrm{id}, (1\ 2\ \ldots\ m)\}$, $M^2(S) = S$. If this holds, $M^k(S)$ is periodic with period 2.

Proposition 6.2 *With $S = \{\mathrm{id}, (1\ 2\ \ldots\ n)\}$, we have*

$$|\{\pi \in M(S) : d_{\mathrm{trp}}(\pi, \mathrm{id}) = k\}| = N(n, k+1) = \frac{\binom{n}{k}\binom{n-1}{k}}{k+1} = \frac{\binom{n}{k}\binom{n}{k+1}}{n},$$

where $N(n,k)$ are the Narayana numbers. Hence, $|M(S)| = C_n$, the nth Catalan number.

Proof: The Narayana numbers $N(n, k+1)$ count the number of Dyck paths of length n with $n-k$ peaks. Extend each peak into a mountain, that is continue the steps $(1,1)$ and $(1,-1)$ which constitute the peak until they cut the x-axis. Draw left parenthesis at positions where the left mountain sides cut the line $y = 1/2$ and right parenthesis where the right mountain sides cut the same line. Then, inserting the numbers $j \in [n]$ at positions $2j-1$ gives the permutations in $M(S)$ with $n-k$ cycles, given that we recursively interpret the expression $(a\ \ldots\ b\ (c\ \ldots\ d)\ f \ldots\ g)$ as $(a\ \ldots\ b\ f\ \ldots\ g)(c\ \ldots\ d)$. □

The following proposition follows directly from independence of disjoint cycles.

Proposition 6.3 *If $S = \{\mathrm{id}, \pi\}$ and the cycles in π are given by $\pi = c_1 c_2 \ldots c_m$, each $\tau \in M(S)$ can be written as a product of permutations $\tau_1\tau_2\ldots\tau_m$ such that $\tau_j \in M(\{\mathrm{id}, c_j\})$.*

For all base sets S we have looked at, the sequence $M^k(S)$ has either had a limit or been eventually periodic with period 2. In fact, we have yet to discover a base set S such that $M^4(S) \neq M^2(S)$.

7 Computing ancestral permutations

Median clouds can be used to facilitate median computations in a given phylogeny in two different ways. First, previously computed inner nodes are used to compute the remaining inner nodes, and these computations may be improved on by using median clouds instead of just a single median. Second, if the inner node we seek to approximate with a median has three edges leading to several leaves in each direction, we can take the leaves of each direction and merge them into a cloud, instead of choosing one of these

leaves at random. To test these approaches, we have made simulations and compared different methods for approximating the inner nodes of a known tree from the leaves.

Consider the phylogenetic tree in Figure 2. Given edge lengths, we simulate leaf permutations using transpositions chosen randomly and independently with uniform distribution. We then use five different median methods to estimate the inner nodes as closely as possible. Thus, we get indications on the quality of the methods. In particular, we wish to examine if median clouds can be used to enhance our abilities to find the inner nodes.

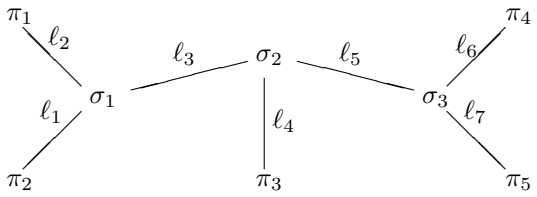


Fig. 2: The phylogenetic tree of $\pi_1, \ldots, \pi_5$.

The five methods used are the following: First, we compute the median of three leaves. This can be done in three ways for σ_1 and σ_3, and four ways for σ_2. Second, we use medians computed with the first method; for instance, we approximate σ_1 with $\tau \in M(\pi_1, \pi_2, \mu)$, where $\mu \in M(\pi_1, \pi_3, \pi_4))$. Third, we compute medians using all leaves on each side of the median. For instance, we approximate σ_1 with $\tau \in M(\pi_1, \pi_2, \{\pi_3, \pi_4, \pi_5\})$. The fourth method is similar to the second, except that we use the median clouds from the third method instead of a single median from the first. Fifth, for comparison we use the inner nodes, that is σ_1 is approximated by $M(\pi_1, \pi_2, \sigma_2)$. This gives a lower limit on the error we can achieve using information only on the leaves.

Tab. 1: Comparing five methods for estimating permutations at inner nodes in the phylogenetic tree in Figure 2. Edge lengths are as below and $n = 40$. In the table, mean distances to the correct inner node are given, summing both over 500 simulations and over all ways to compute the inner node using the respective methods. We note that the results improve significantly as we refine the methods. We should add that the first two methods can be improved upon in the case where edge lengths are as different as in the second row by always choosing the closest leaf on each side, but the fourth method is still somewhat better.

Edge lengths	First	Second	Third	Fourth	Fifth
$(7, 7, 7, 14, 7, 7, 7)$	4.2	2.7	2.1	1.9	0.8
$(15, 3, 4, 15, 4, 4, 12)$	4.0	2.3	1.6	1.4	0.5

The five methods are compared in Table 1. We find that the third and fourth methods constitute a significant improvement over the first two, both with similar edge length and a mixture of long and short edges. In the second case, choosing closely related permutations improves on the mean results, but the third and fourth methods are still better even in this extreme case.

8 Open problems

Our results leave several open problems. Which sets are medians clouds for some base set? Which sets are limit clouds for some base set? Are there base sets whose median sequence $M^k(S)$ is not periodic, or has a longer period than 2? What kind of regularities and symmetries can we expect to find in a limit cloud? All these questions are also interesting under other distances, for example reversals.

We are also anxious to see if median clouds can be incorporated into median computations under other distances, such as breakpoints, reversals and DCJ. In addition, a proof that the transposition median problem is, as conjectured, NP-complete (or even better, a polynomial time solver) would of course be welcomed.

References

[1] William Arndt and Jijun Tang. Improving reversal median computation using commuting reversals and cycle information. *Journal of Computational Biology*, 15:1079–1092, 2008.

[2] Guillaume Bourque and Pavel Pevzner. Genome-scale evolution: Reconstructing gene orders in the ancerstral species. *Genome Research*, 12:26–36, 2002.

[3] Alberto Caprara. Sorting permutations by reversals and eulerian cycle decompositions. *SIAM Journal of Discrete Mathematics*, 12:91–110, 1999.

[4] Alberto Caprara. The reversal median problem. *INFORMS Journal on Computing*, 15:93–113, 2003.

[5] Niklas Eriksen. Reversal and transposition medians. *Theoretical Computer Science*, 374:111–126, 2007.

[6] Niklas Eriksen and Axel Hultman. Estimating the expected reversal distance after a fixed number of reversals. *Advances in Applied Mathematics*, 32:439–453, 2004.

[7] Bret Larget, Donald Simon, Joseph Kadane, and Deborah Sweet. A bayesian analysis of metazoan mitochondrial genome arrangements. *Molecular Biology and Evolution*, 22:486–495, 2005.

[8] Bernard Moret, David Bader, Stacia Wyman, Tandy Warnow, and Mi Yan. A new implementation and detailed study of breakpoint analysis. In *Proceedings of the Pacific Symposium of Biocomputing (PSB 01)*, 2001.

[9] Itsik Pe'er and Ron Shamir. The median problems for breakpoints are np-complete. *Electronic Colloquium on Computational Complexity*, TR98-071, 1998.

[10] Eric Tannier, Chunfang Zheng, and David Sankoff. Multichromosomal median and halving problems under different genomic distance. In Keith Crandall and Jens Lagergren, editors, *Proceedings of the Workshop on Algorithms in Bioinformatics, WABI 2008*, volume 5251 of *LNBI*, pages 1–13. Springer, 2008.

[11] Andrew Wei Xu. A fast and exact algorithm for the median of three problem—a graph decomposition approach. In Craig Nelson and Stéphane Vialette, editors, *Proceedings of RECOMB Comparative Genomics, RECOMB-CG 2008*, volume 5267 of *LNBI*, pages 184–197. Springer, 2008.

[12] Andrew Wei Xu and David Sankoff. Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem. In Keith Crandall and Jens Lagergren, editors, *Proceedings of the Workshop on Algorithms in Bioinformatics, WABI 2008*, volume 5251 of *LNBI*, pages 25–37. Springer, 2008.

Algorithm 1: Median: A simplified branch-and-bound algorithm for finding $M(S)$ under the transposition distance. It is called with $\mu = (0, 0, \ldots, 0)$ and $B = \infty$. The ApplyOptimal algorithm iteratively applies all majority rule assignments according to Theorem 3.9.

Data: S, μ, B
Result: $M(S)$, B
$\mu \leftarrow$ ApplyOptimal(S, μ);
if $\mu \in \mathfrak{S}_n$ **then**
    if $d(\mu, S) < B$ **then**
        $M(S) \leftarrow \{\mu\}$;
        $B \leftarrow d(\mu, S)$;
    else if $d(\mu, S) = B$ **then**
        $M(S) \leftarrow M(S) \cup \{\mu\}$;
    end
else
    $e \leftarrow \min\{j : j \notin \mu\}$;
    foreach i *such that* $\mu(i) = 0$ **do**
        $\mu(i) \leftarrow e$;
        if $d(\mu, S) \leq B$ **then**
            $(M(S), B) \leftarrow$ Median(S, μ, B);
        end
    end
end