involve
a journal of mathematics

Using ciliate operations to construct chromosome phylogenies

Jacob L. Herlin, Anna Nelson and Marion Scheepers

msp

2016 vol. 9, no. 1

INVOLVE 9:1 (2016)
dx.doi.org/10.2140/involve.2016.9.1

Using ciliate operations to construct chromosome phylogenies

Jacob L. Herlin, Anna Nelson and Marion Scheepers

(Communicated by Joseph A. Gallian)

Whole genome sequencing has revealed several examples where genomes of different species are related by permutation. The number of certain types of rearrangements needed to transform one permuted list into another can measure the distance between such lists. Using an algorithm based on three basic DNA editing operations suggested by a model for ciliate micronuclear decryption, this study defines the distance between two permutations to be the number of ciliate operations the algorithm performs during such a transformation. Combining well-known clustering methods with this distance function enables one to construct corresponding phylogenies. These ideas are illustrated by exploring the phylogenetic relationships among the chromosomes of eight fruit fly (*Drosophila*) species, using the well-known UPGMA algorithm on the distance function provided by the ciliate operations.

Over evolutionary time, “local” DNA editing events such as nucleotide substitutions, deletions or insertions diversify the set of DNA sequences present in organisms. Results of whole genome sequencing suggest that also “global” DNA editing events diversify these DNA sequences.

Consider two species S_1 and S_2 with a common ancestor whose genome was organized over n linear chromosomes. A gene G of the ancestor was inherited as gene G_1 by species S_1 and as gene G_2 by species S_2. G_1 and G_2 are *orthologous* genes, or simply *orthologs*. Assume that the species S_1 and S_2 each also has n chromosomes, and that for each ancestral chromosome i, the orthologs of any ancestral gene on chromosome i are also in the descendant species S_1 and S_2 on the corresponding chromosome i. This assumption is known, in the context of certain

MSC2010: 05E15, 20B99, 92-08, 92D15, 92D99.
Keywords: permutations, reversals, block interchanges, fruit fly, ciliate, phylogeny.
This research was started during the Summer of 2011 when Herlin and Nelson participated in the Boise State University Mathematics REU program, funded by NSF grant DMS 1062857. We gratefully acknowledge funding by the NSF and by Boise State University.

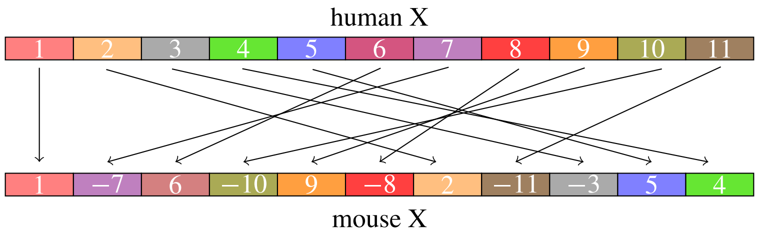

Figure 1. The permutation between 11 synteny blocks of the human and the mouse X chromosomes. A negative symbol denotes an orientation change by a 180° rotation of a synteny block. The lengths of synteny blocks are not to scale. See Figure 2 of [Pevzner and Tesler 2003].

fruit fly species, as the *Muller hypothesis*[1]. In this paper we shall assume the Muller hypothesis for our applications.

It may happen that the order in which orthologs on chromosome i appear in species S_1 is different from the order in which they appear in species S_2. In this case chromosome i in each of these two species can be partitioned into a number, say k, of *synteny blocks*[2]: a synteny block is a maximal list of adjacent orthologous genes that have the same adjacencies in the two species. In this definition of a synteny block, we permit blocks consisting of single genes. An endpoint of a synteny block is also called a *breakpoint*. Synteny blocks may have opposite orientation in two species. Thus the synteny blocks of chromosome i of species S_1 are a *signed* permutation of the corresponding synteny blocks of chromosome i of species S_2. This phenomenon is observed in several branches in the tree of life. Figure 1 illustrates the phenomenon for 11 synteny blocks of orthologous genes in the X chromosome of human and mouse.

Since the appearance of [Sturtevant and Dobzhansky 1936] and [Dobzhansky and Sturtevant 1938] on fruit fly genomes, it has been popular to use *reversals*[3] as the primary global DNA sequence editing operation to describe phylogenetic relationships among genomes. See, for example, [Bafna and Pevzner 1995; Hannenhalli and Pevzner 1999].

An insightful phylogenetic analysis that includes fine structural elements of reversals is given in [Bhutkar et al. 2008]. It addresses the question of whether reversals can occur at arbitrary locations in the genome of an organism. Certain locations,

[1] Named after H. J. Muller [1940] who observed that for the data then known for relatives of *Drosophila melanogaster*, this assumption is true even for chromosome arms.

[2] This definition of a synteny block is more restrictive than the one used in [Bhutkar et al. 2008]: The latter allows for differences in gene order up to a certain threshold, and does not allow for single gene blocks. See the section “An application to genome phylogenetics” for more information.

[3] A reversal is a rotation of a DNA segment through 180°. Reversals are also called inversions.

which would disrupt the coding region of an essential gene, would not be observed in extant organisms. Similarly, locations that negatively affect the fitness of organisms would disappear over time due to "purifying selection". Additionally, certain sequence motifs may actually promote DNA recombination that results in a genome rearrangement. For example, [Coghran and Wolfe 2002] reports a correlation between *breakpoints*[4] associated with rearrangements, and repetitive DNA. This point is also considered in [Bhutkar et al. 2008]. In the review [Hughes 2000], a similar correlation between rearrangements in bacterial genomes and repetitive DNA is discussed. These considerations suggest that genome rearrangement events that lead to the diverse genomes we observe in nature are not arbitrary, but constrained by contexts. In this paper, we explore the use of *context-directed* DNA recombination events to analyze genome rearrangements and to construct a phylogeny based on these.

In recent years, transpositions and block interchanges have also been considered as possible global DNA sequence editing operations [Bafna and Pevzner 1998; Coghran and Wolfe 2002; Mira and Meidanis 2007; Yancopoulos et al. 2005]. In a *block interchange*, two disjoint segments of a chromosome exchange locations without changing orientation. Thus, in Figure 1, synteny blocks 2 and 7 would have been a block interchange if synteny block 7 did not also undergo a reversal. A *transposition* is a special block interchange where the two segments that exchange location are adjacent. In Figure 1, synteny blocks 4 and 5 illustrate a transposition.

On page 1661 of [Bhutkar et al. 2008], in the discussion of their selection of genes to which their analysis of rearrangements in fruit fly genomes apply, the authors indicate that genes deemed to have been relocated by a transposition rather than a reversal have been explicitly removed from the analysis. Thus, the analysis of [Bhutkar et al. 2008] features reversals exclusively. On the other hand, the analysis in [Coghran and Wolfe 2002] of rearrangements in the genomes of two nematode species includes reversals, transpositions and *translocations*. A translocation occurs when segments from two different chromosomes exchange positions. In this paper, we explore only reversals and block interchanges (both constrained by contexts) in the analysis of rearrangements.

Experimental results from ciliate laboratories present us with examples of DNA editing operations that routinely occur during developmental processes in these organisms. The textbook [Ehrenfeucht et al. 2004] and the two surveys [Prescott 1994; 2000] give a good starting point for information about these "ciliate operations" and the corresponding biological background. We shall call the yet to be fully identified system in ciliates that accomplishes micronuclear decryption[5], the *ciliate decryptome*.

[4] Referring to the mouse X chromosome in Figure 1, a breakpoint is a transition point between synteny blocks that are not consecutively numbered.

[5] Some details regarding this process are given below in Section 1.

We shall illustrate how to use "ciliate operations" to deduce potential phylogenetic relationships from genome rearrangement phenomena. Previous work, including [Bafna and Pevzner 1995; Bhutkar et al. 2008; Hannenhalli and Pevzner 1999], used unconstrained reversals to deduce phylogenetic relationships. Our main ideas are to use ciliate genomic elements to model two genomes related by permutations of locations and orientations of synteny blocks, to apply the context-directed DNA operations of the ciliate decryptome to define a distance function between the relevant permuted genomes, and to then use a classical distance-based algorithm to derive phylogenies. Of the several different distance-based algorithms available, we selected the UPGMA algorithm[6].

Then we apply these ideas to chromosomes of eight species of fruit flies (*Drosophila*) to obtain a phylogeny for each of these chromosomes.

The use of ciliate operations as the basis for deriving a distance function has the attractive feature that the ciliate decryptome is programmable [Nowacki et al. 2007], and the computational steps taken by the decryptome can be monitored under laboratory conditions [Möllenbeck et al. 2008]. Thus, there are extant organisms that are poised to be employed as DNA computing devices naturally equipped to determine phylogenetic relationships among permuted genomes.

Our paper is organized as follows: In Section 1 we briefly describe ciliate nuclear duality. This duality is the basis for modeling pairs of genomes related by permutation as genetic elements of the ciliate genome. In Section 2 we briefly describe the context-directed DNA operations of the ciliate decryptome. In Section 3 we introduce and analyze the mathematical notion of a pointer list. In Section 4 we model relevant features of the ciliate decryptome's DNA operations by mathematical operations on pointer lists. In Section 5 we describe an algorithm which we call the HNS algorithm, that uses these operations on pointer lists to compute the distance between chomosomes that are related by permutation. In Section 6 we use data downloaded from flybase.org and the HNS and UPGMA algorithms to construct phylogenies over eight species for each of the fruit fly chromosomes. In the closing section, we discuss possible future directions related to this work.

1. Ciliates and nuclear duality

A ciliate is a single cell eukaryote that hosts two types of nuclei: one type, the macronucleus, contains the transcriptionally active somatic genome, while the other type, the micronucleus, contains a transcriptionally silent germline-like genome. The micronuclear genome is, in the technical sense of the word, an encrypted version of the macronuclear genome. Special events in the ciliate life cycle predictably trigger

[6] Descriptions of UPGMA can be found in Chapter 27 of [Barton et al. 2007], available online, or in the textbook [Clote and Backofen 2000].

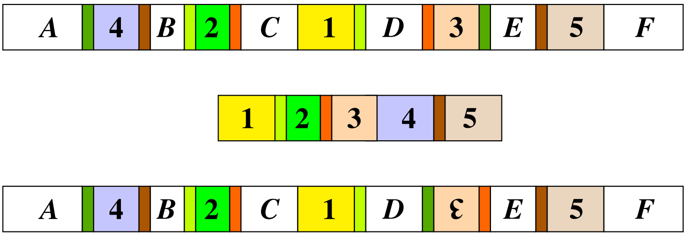

Figure 2. The top diagram depicts a possible micronuclear precursor, and the bottom diagram is another possible micronuclear precursor of the macronuclear gene in the middle diagram.

conjugation between a pair of mating-compatible cells. Conjugation results in what amounts to a Diffie–Hellman exchange[7] between two conjugants, the formation of a new micronucleus in each, and the decryption of one copy of the new micronuclear genome to establish a replacement macronuclear genome, while in each conjugant the instances of its preexisting genome are discarded. Readers interested in a thorough survey of ciliate nuclear duality could consult [Prescott 1994].

The relationship between micro- and macronuclear DNA. To describe the experimentally observed relationship between the micronuclear and macronuclear DNA molecules, consider Figure 2.

The micronuclear DNA sequences in the top and the bottom rows of Figure 2 each have three types of regions: The white blocks, labeled with letters, are called *internal eliminated sequences* (IESs). The blocks labeled with numbers are called *macronuclear destined sequences* (MDSs), while the narrow strips are called *pointers*. As the micronuclear precursors show, there are two copies of each pointer. For example, MDS 2 has a pointer on the left flank that is identical to the pointer on the right flank of MDS 1. This pointer will be called the "1-2 *pointer*". And MDS 2 has a pointer on its right flank which is identical to the pointer on the left flank of MDS 3. This pointer is called the "2-3 *pointer*". The other pointers are named similarly. Also note that MDS 1 does not have a pointer on its left flank, and MDS 5 does not have a pointer on its right flank. As MDS 3 and the pointers on its flanks show in the bottom row of Figure 2, in the micronuclear precursor, an MDS plus its flanking pointer(s), as a unit, can be in a 180-degree rotated orientation of the corresponding components in the macronuclear gene. The corresponding macronuclear sequence in the middle row of Figure 2 contains only one of each of the pointers present in its micronuclear precursor, and all the MDSs, but none

[7] A Diffie–Hellman exchange is a cryptographic protocol for secure exchange of a secret key in a hostile environment. The conjugants exchange a haploid copy of the germline genome, which is an encrypted version of the somatic genome.

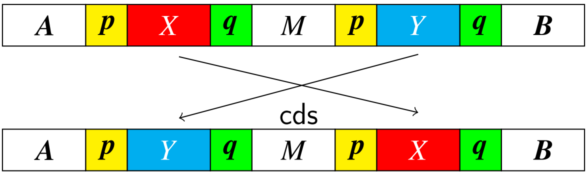

Figure 3. Context-directed block swaps: the $\boldsymbol{p} \cdots \boldsymbol{q} \cdots \boldsymbol{p} \cdots \boldsymbol{q}$ pointer context permits swapping the DNA segments X and Y.

of the IESs of the micronuclear precursor. In the macronuclear sequence, these components occur in a specific order, which we call the *canonical order*.

In shorthand, the micronuclear precursor in the top row of Figure 2 is [4, 2, 1, 3, 5], while the micronuclear precursor in the bottom row of Figure 2 is [4, 2, 1, −3, 5].

2. The ciliate DNA operations

We now turn to the actual ciliate algorithm that processes micronuclear precursors to produce their corresponding macronuclear versions. The articles [Angeleska et al. 2007; Prescott et al. 2003] propose hypotheses about biochemical processes that perform the decryption algorithm in ciliates. We do not address the biochemical foundations here.

The textbook [Ehrenfeucht et al. 2004] describes three DNA editing operations underlying this decryption process. There is experimental evidence that these three operations accomplish the decryption process. The article [Möllenbeck et al. 2008] gives experimental data about the DNA products of intermediate steps of the ciliate algorithm. We henceforth assume that the three operations that produce macronuclear molecules from their micronuclear precursors are as proposed in [Ehrenfeucht et al. 2004]: context-directed block interchanges (swaps), context-directed reversals and context-directed excisions.

Context-directed block interchanges (swaps). The top strip in Figure 3 represents a segment of DNA in a micronuclear chromosome of some ciliate. The symbols $\boldsymbol{p}$ and $\boldsymbol{q}$ denote identified pointers, while A, B, M, X, Y represent segments of DNA.

The three necessary conditions to swap segments X and Y are:

(1) X and Y both have an occurrence of each of the pointers $\boldsymbol{p}$ and $\boldsymbol{q}$ at their flanks;

(2) the pointer pair $\boldsymbol{p}, \boldsymbol{q}$ appears in the (alternating) context $\cdots \boldsymbol{p} \cdots \boldsymbol{q} \cdots \boldsymbol{p} \cdots \boldsymbol{q} \cdots$;

(3) neither occurrence of the pointer $\boldsymbol{p}$ nor of pointer $\boldsymbol{q}$ is flanked by a pair of successively numbered MDSs. For specificity consider Figure 4, where numbered blocks denote MDSs while lettered blocks denote IESs. The X of Figure 3 may be taken to be the segment $2B$ of Figure 4, while the Y of Figure 3 may be taken to be the segment $D3$ of Figure 4.

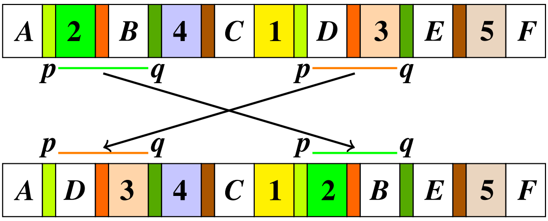

Figure 4. The top diagram depicts a possible micronuclear precursor, and the bottom diagram is the result of cds applied to the pointer pair $\boldsymbol{p} = (1, 2)$ and $\boldsymbol{q} = (3, 4)$.

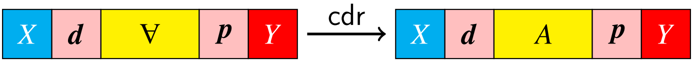

Figure 5. Context-directed reversal: the $-\boldsymbol{p} \cdots \boldsymbol{p}$ or $\boldsymbol{p} \cdots - \boldsymbol{p}$ pointer context-permits 180° rotation of flanked segment A.

Only when *all three* conditions are met is an interchange of the segments X and Y permitted. The result of this swap is depicted in the bottom strip of Figure 3. The reader may check, by comparing the bottom strips of Figures 3 and 4, that subsequent to an application of cds the contextual conditions (1) and (2) are still valid, but condition (3) is no longer met: indeed, one occurrence of each of the pointers $\boldsymbol{p}$ and $\boldsymbol{q}$ is now flanked by successively numbered MDSs.

Context-directed reversal. To describe a context-directed reversal, consider the left strip in Figure 5. It depicts a segment of DNA appearing in the micronucleus.

To rotate the yellow segment, labeled by an upside-down A, by 180°, that is, to reverse A, two necessary contextual conditions must be met:

(1) A is flanked by a pointer $\boldsymbol{p}$ and by the 180° rotation[8] of $\boldsymbol{p}$;

(2) neither occurrence of $\boldsymbol{p}$ is flanked by successively numbered MDSs. For specificity, consult Figure 6, where numbered blocks denote MDSs and lettered blocks denote IESs. The A of Figure 5 corresponds to the segment $-2C4D$ of Figure 6.

Only when *both* of these contextual requirements are met is rotation of the segment flanked by the relevant pointer context permitted. The result of this context-directed reversal is depicted by the right strip in Figure 5, and the corresponding bottom strip of Figure 6. As illustrated, subsequent to a context-directed reversal, one of the occurrences of the pointer $\boldsymbol{p}$ now has successively numbered MDSs on both flanks and no further applications of cdr are permitted to this pointer context.

[8] In text, the 180° rotation of $\boldsymbol{p}$ will be denoted $-\boldsymbol{p}$.

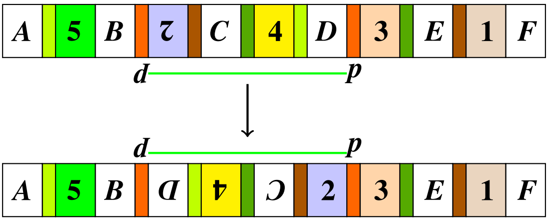

Figure 6. The top row depicts a possible micronuclear precursor. The bottom row results from cdr applied to the pointer $\boldsymbol{p} = (2, 3)$.

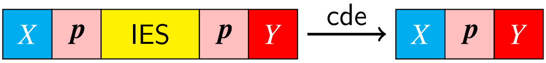

Figure 7. Context-directed excision: the IES flanked by pointer $\boldsymbol{p}$ on both sides is removed, along with one copy of $\boldsymbol{p}$.

Context-directed excision. To describe context-directed excision, consider Figure 7. In it, the pointer $\boldsymbol{p}$ flanks a DNA segment identified as an IES (the yellow segment). This context $\boldsymbol{p}\,\mathsf{IES}\,\boldsymbol{p}$ permits the excision of the IES segment plus one of the pointers, with the result of joining the DNA segments flanking the original pair of pointers, to the flanks of the remaining pointer.

Observe that context-directed block interchanges and context-directed reversals do not decrease or increase the length of the string they operate on, and they retain all the pointers. But context-directed excision, as illustrated in Figure 7, changes the pointer contexts by deleting selected pointers and IESs.

3. Pointer lists

Pointers are an essential ingredient of the three DNA editing operations. We exploit this central role of pointers by now basing our computational formalism (that mathematically models these three ciliate operations) on pointers. Towards this end, we introduce the notion of a *pointer list*[9].

Definition 1. A finite sequence $P := [x_1, \dots, x_m]$ of integers is said to be a *pointer list* if it satisfies the following six conditions:

(1) m is an even positive integer.

(2) There is a unique i with $\mu = |x_i| = \min\{|x_j| : 1 \le j \le m\}$.

(3) There is a unique j with $\lambda = |x_j| = \max\{|x_i| : 1 \le i \le m\}$.

[9] In anticipation of wider applicability of the notion of a pointer list, we give a definition that is more general than the specific instance of it that we need.

(4) For each $i \in \{1, \dots, m\}$ with $\mu < |x_i| < \lambda$, there is a unique $j \in \{1, \dots, m\} \setminus \{i\}$ such that $|x_i| = |x_j|$.

(5) For each odd $i \in \{1, \dots, m\}$, $x_i \leq x_{i+1}$ and $x_i \cdot x_{i+1} > 0$.

(6) Whenever $i \in \{1, \dots, m\}$ is odd, there is no j such that $|x_i| < |x_j| < |x_{i+1}|$ or $|x_{i+1}| < |x_j| < |x_i|$.

The following two mathematical facts are important in reasoning about ciliate operations on pointer lists.

Lemma 1. *Let* $[x_1, x_2, \dots, x_{m-1}, x_m]$ *be a pointer list. If* i *and* j *are distinct indices for which* $|x_i| = |x_j|$, *then* x_i *and* x_j *have the same sign if, and only if,* i *and* j *have distinct parity.*

Lemma 2. *If* $[x_1, x_2, \dots, x_{m-1}, x_m]$ *is a pointer list of length larger than* 4, *then at least one of the following three statements is false*:

(a) $(\forall i)(x_i \neq x_{i+1})$.

(b) $(\forall i)(\forall j)($*If* $|x_i| = |x_j|$, *then* $x_i = x_j)$.

(c) $(\forall i)(\forall j)(\forall k)(\forall \ell)($*If* $i \neq k$, $j \neq \ell$, $i < j$ *and* $x_i = x_k$ *and* $x_j = x_\ell$, *then either* $i < j < \ell < k$ *or* $i < k < j < \ell)$.

In the interest of readability, the somewhat lengthy, yet elementary, proofs of these facts are left to the reader.

Pointer lists to which we will apply the ciliate operations come about as follows: Let $\mathbb{Z}$ denote the set of integers. For a set S, the symbol ${}^{<\omega}S$ denotes the set of finite sequences with entries from S. For an integer z, we define

$$\check{z}(1) = \begin{cases} z & \text{if } z = |z|, \\ z - 1 & \text{otherwise,} \end{cases}$$

and in all cases, $\check{z}(2) = \check{z}(1) + 1$. Then define the function $\pi : {}^{<\omega}\mathbb{Z} \to {}^{<\omega}\mathbb{Z}$ by

$$\pi([z_1, \dots, z_k]) = [\check{z}_1(1), \check{z}_1(2), \dots, \check{z}_k(1), \check{z}_k(2)].$$

Thus, for example, $\pi([-1, 4, 3, 5, 2, -9, 7, 10, -8, 6])$ is the sequence

$$[-2, -1, 4, 5, 3, 4, 5, 6, 2, 3, -10, -9, 7, 8, 10, 11, -9, -8, 6, 7].$$

It can be verified that this sequence is indeed a pointer list. The following lemma captures this fact.

Lemma 3. *For each finite sequence* $M := [s_1, s_2, \dots, s_n]$ *of nonzero integers such that there is an integer* m *for which* $\{|s_i| : 1 \leq i \leq n\} = \{m + 1, \dots, m + n\}$, *the sequence* $\pi(M)$ *is a pointer list.*

The proof consists of verifying that $\pi(M)$ meets all stipulations of Definition 1.

4. The ciliate operations on pointer lists

We now introduce three special functions, cde, cdr and cds, from ${}^{<\omega}\mathbb{Z}$ to ${}^{<\omega}\mathbb{Z}$, inspired by the three ciliate operations, as follows. Let $P := [x_1, \dots, x_m]$ be a given finite sequence.

Context directed excision:

$$\mathsf{cde}(P) = \begin{cases} P & \text{if there is no } i \text{ with } x_i = x_{i+1}, \\ [x_1, \dots, x_{i-1}, x_{i+2}, \dots, x_m] & \text{for } i \text{ minimal with } x_i = x_{i+1}, \end{cases}$$

Context-directed reversal:

$$\mathsf{cdr}(P) = \begin{cases} P & \text{if there are no } i < j \text{ with } x_i = -x_j, \\ [x_1, \dots, x_{i-1}, x_i, \underline{-x_j, \dots, -x_{i+1}}, x_{j+1}, \dots, x_m] & \text{for the minimal } i \text{ with } x_i = -x_j \text{ for a } j > i. \end{cases}$$

Context-directed block swaps: We set $\mathsf{cds}(P) = P$ if there are no $i < j < k < \ell$ with $x_i = x_k$ and $x_j = x_\ell$. However, if there are $i < j < k < \ell$ with $x_i = x_k$ and $x_j = x_\ell$, then choose the least such i, and for it the least corresponding j, and define $\mathsf{cds}(P)$ to be

$$[x_1, \dots, x_i, \underline{x_k, \dots, x_\ell}, x_j, \dots, x_{k-1}, \underline{x_{i+1}, \dots, x_{j-1}}, x_{\ell+1}, \dots, x_m].$$

These three operations have now been defined on arbitrary finite sequences of integers. They behave rather well on the subset $\mathsf{PL} = \{\sigma \in {}^{<\omega}\mathbb{Z} : \sigma \text{ is a pointer list}\}$ of their domain, as stated in the next two theorems. In the interest of readability the proofs have been omitted.

Theorem 4. *If P is a pointer list of length larger than* 4, *then at least one of the following statements is true*:

(1) $\mathsf{cde}(P) \neq P$.

(2) $\mathsf{cdr}(P) \neq P$.

(3) $\mathsf{cds}(P) \neq P$.

Theorem 5 (pointer list preservation). *Let $P = [x_1, \dots, x_m]$ be a pointer list. Then each of* $\mathsf{cde}(P)$, $\mathsf{cdr}(P)$ *and* $\mathsf{cds}(P)$ *is a pointer list.*

A finite sequence σ is a *fixed point* of a function $F : {}^{<\omega}\mathbb{Z} \to {}^{<\omega}\mathbb{Z}$ if $F(\sigma) = \sigma$.

Theorem 6. *If P is a pointer list of length larger than* 4 *and not a fixed point of* $F \in \{\mathsf{cdr}, \mathsf{cds}\}$, *then $F(P)$ is not a fixed point of* cde.

5. The HNS algorithm

Call a pointer list a *destination* if it is one of the following: $[\mu, \lambda]$, $[-\lambda, -\mu]$, or for some integer z with $|z| \notin \{\lambda, \mu\}$, the pointer list is one of $[z, \lambda, \mu, z]$ or $[z, -\mu, -\lambda, z]$.

Let P be a pointer list. Letting $\mathsf{cde}^i(P)$ denote the i-th iteration of cde on P, define $e(P)$ to be the minimal value of i such that $\mathsf{cde}^{i+1}(P) = \mathsf{cde}^i(P)$. Then define $\mathsf{E}(P) = \mathsf{cde}^{e(P)}(P)$.

In the following theorem, recall that a finite sequence σ is a *fixed point* of a function $F : {}^{<\omega}\mathbb{Z} \to {}^{<\omega}\mathbb{Z}$ if $F(\sigma) = \sigma$.

Theorem 7. *For a given pointer list P_0, define the sequence $P_0, P_1, \ldots, P_i, \ldots$ so that*

$$P_{i+1} = \begin{cases} \mathsf{E}(P_i) & \textit{if } P_i \textit{ is not a } \mathsf{cde} \textit{ fixed point,} \\ \mathsf{cds}(P_i) & \textit{if } P_i \textit{ is a } \mathsf{cde}\textit{, but not a } \mathsf{cds} \textit{ fixed point,} \\ \mathsf{cdr}(P_i) & \textit{if } P_i \textit{ is a } \mathsf{cde} \textit{ and a } \mathsf{cds} \textit{ but not a } \mathsf{cdr} \textit{ fixed point.} \end{cases}$$

Then the sequence $P_0, P_1, \ldots, P_i, \ldots$ terminates in a destination.

Proof. By Theorem 5, each term in this sequence is a pointer list. By Theorem 4, as long as such a pointer list has more than four terms, it is not a fixed point of the ciliate operations. By Theorem 6, the sequence does not terminate with an application of cds or of cdr, but with an application of E. Each application of E reduces the length of a pointer list that is not a fixed point for E by a positive even number of terms. According to the definitions of the ciliate operations, the pointers with absolute value λ and μ are never excised, and thus are present in any fixed point of a ciliate operation. Thus, a fixed point consisting of only two terms necessarily consists of the terms with absolute values λ and μ. As such, a two-term result is still a pointer list by Theorem 5. Stipulation (5) of Definition 1 shows that this fixed point must be $[\mu, \lambda]$ or $[-\lambda, -\mu]$. Since applications of cde remove terms that are equal and adjacent, a four-term fixed point must contain, in addition to terms with absolute values μ and λ, two terms of equal absolute value. If these two terms have opposite sign, the pointer list is not a fixed point for cdr. Thus, these two terms must be of the same sign. But then, as the pointer list is a fixed point of cde, these two terms are not adjacent. Moreover, their absolute value is strictly between μ and λ. Now stipulation (5) of Definition 1 implies that this pointer list is one of the two remaining claimed destinations. □

Thus the following algorithm, which we call the HNS algorithm, halts:

(1) Input: A pointer list P, its length $|P|$ and integers r and s;

(2) Iteratively apply cde until a cde fixed point is reached. With each application, decrease $|P|$ by 2. Then proceed to (3).

(3) If P is a fixed point of cds, proceed to (4). Else, apply cds, increase s by 1, and return to (1).

(4) If P is a fixed point of cdr, terminate the algorithm and report the current values of P, r and s. Else, apply cdr, increase r by 1, and return to (1).

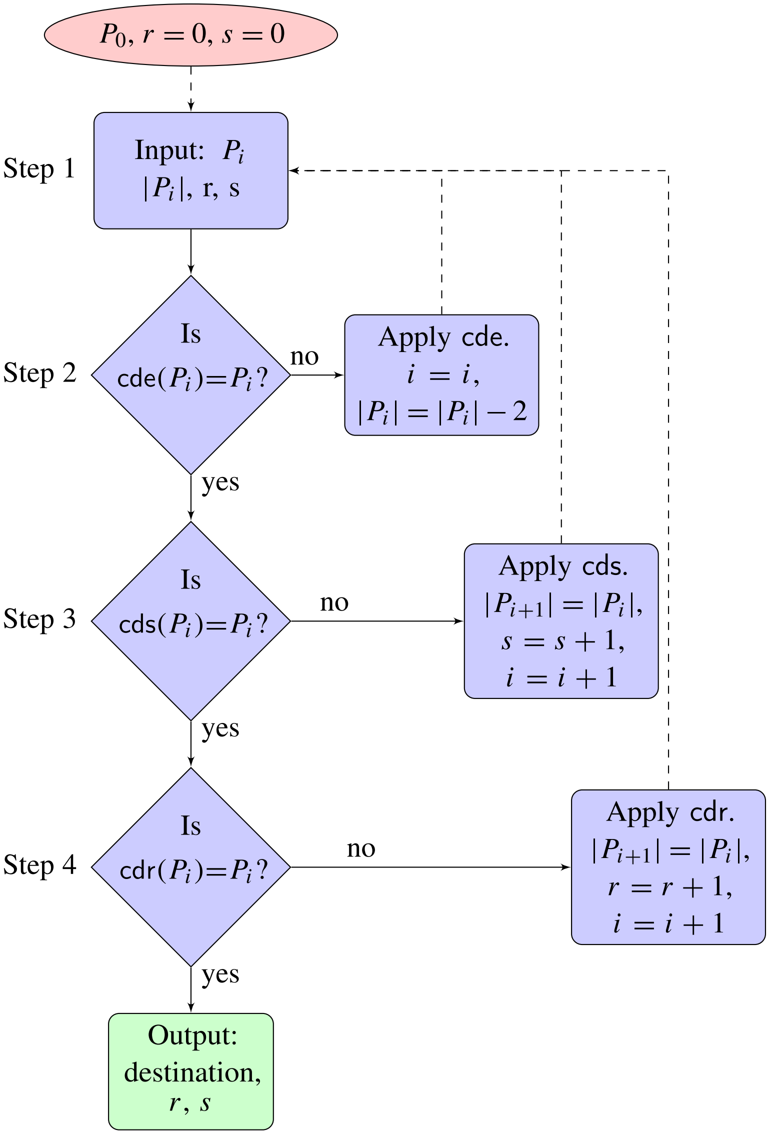

Figure 8. A flow diagram for the HNS algorithm.

Figure 8 depicts the algorithm in flow-diagram style. Let the original length of the pointer list P be denoted $|P|$.

In step (2), the algorithm examines $|P|-1$ adjacent pairs. If P is not a cde fixed point, then with the application of cde, $|P|$ decreases by 2. In this step we update the length of the resulting P with each nontrivial application of cde.

In step (3), the algorithm starts with a position $k < |P|$ and then chooses a position $\ell > k+1$ with $x_k = x_\ell$ if any. This takes at most $(|P|-1)+(|P|-2)+\cdots+2$ search steps, which is $O(|P|^2)$. If this search fails, proceed to step (4). Else, suppose a successful $k+1<\ell<|P|$ is found. Then for $k<j<\ell$, search for an $m>\ell$ with $x_m=x_\ell$. This would require at most $(\ell-k)(|P|-\ell)$ steps. If this fails, proceed to step (4). Else, execute a cds based on the found quadruple (k, j, ℓ, m), increase s by 1, and return to step (1). Step (3) is completed in $O(|P|^2)$ search steps.

In step (4), the algorithm starts with a position $k<|P|$ and then scans positions $j>k$ until it finds an $x_j=-x_k$. The worst case scenario for this search is also $(|P|-1)+(|P|-2)+\cdots+2$, or $O(|P|^2)$. If the search succeeds, the result of cdr is obtained in at most $|P|-1$ search steps. Increase r by 1, and return to step (2). Else, if the search fails, terminate the algorithm and report the current values of P, r and s.

In one cycle of executing steps until return to step (1), the worst case scenario employs at most $O(|P|^2)$ search and execution steps. For the next round, an upper bound is$O((|P|-1)^2)=O(|P|^2)$. This continues for at most $|P|/2$ rounds. Thus a global upper bound, in terms of the length of the initial pointer list, is $O(|P|^3)$.

The efficiency of this algorithm that produces from an initial pointer list a fixed point for the operations cde, cds and cdr in $O(|P|^3)$ steps can probably be improved. Additionally, this algorithm most likely does not minimize the number of steps taken, using cde, cds and cdr, to reduce a pointer list to a fixed point.

In our phylogenetic application below, any calibration of time span in terms of the number of operations required is based on the above HNS algorithm as computational standard for the calibration.

6. An application to genome phylogenetics

As illustrated in Figure 1, for organisms S_1 and S_2 there may be synteny blocks of orthologous genes on corresponding chromosomes. Choose S_1 as reference and number the synteny blocks in their $5'$ to $3'$ order of appearance on S_1's chromosome as $1, 2, 3, \ldots, n$. In species S_2, the synteny blocks of these same genes may appear in a different order, and individual synteny blocks may also appear in orientation opposite from the orientation in S_1. Write the corresponding list of numbers in their order of appearance on S_2's chromosome, making the number negative if the synteny block orientation is opposite to that in S_1. The result is a signed permutation of the list $1, 2, 3, \ldots, n$.

Now imagine that the list of synteny blocks for S_1 are the MDSs of a ciliate macronuclear gene G, while the signed permutation that represents the corresponding list of synteny blocks for S_2 is the micronuclear precursor of G. Take the number of operations the ciliate decryptome performs to convert the micronuclear precursor to its macronuclear version G as a measure of the evolutionary distance between the two chromosomes of S_1 and S_2. We used the HNS algorithm to simulate the actions of the ciliate decryptome on the set of highly permuted genomes from various species of fruit flies.

The fruit fly genome is organized in four[10] chromosomes, enumerated 1, 2, 3 and 4. These four chromosomes are traditionally divided into six so-called Muller elements. The left and right arms of chromosome 2 are each one of these Muller elements, and it is similar for chromosome 3. Chromosome 1 is the X chromosome. The correspondence of chromosomal material to Muller elements is as follows:

chromosome	1 = X	2L	2R	3L	3R	4
Muller element	*A*	*B*	*C*	*D*	*E*	*F*

The fruit fly genome has at least 13,600 confirmed genes (and counting), but is not expected to host significantly more genes. Recall that our definition of a synteny block is more restrictive than the one used in [Bhutkar et al. 2008], where "microinversions" are permitted. See, for example, Table 1 on page 1662 of [loc. cit.] for data on these more relaxed synteny blocks relative to the genome of *D. melanogaster*. Between two species, the number of synteny blocks can still be well over a thousand, as can be gleaned from Table 1 of [loc. cit.], where the more relaxed definition of synteny block actually provides a lower bound on the number of synteny blocks as defined in our paper.

According to findings of [Bhutkar et al. 2008], 95% of orthologous genes between two species are present on the same Muller element. For the species we are using, with one exception to be noted now, evidence suggests that all orthologous genes are present on the same Muller elements. Using data obtained from flybase.org, we examined the permutation structure of these for the eight species *D. melanogaster*, *D. yakuba*, *D. erecta*, *D. sechellia*, *D. mojavensis*, *D. simulans*, *D. grimshawi* and *D. virilis*. As illustrated in Figure 3 of [Bhutkar et al. 2008], there is a translocation of genes between Muller elements B and C for *D. erecta*, one of the species in our sample. Thus we combined Muller elements B and C into one computational unit (chromosome 2) for our application. Thus, we refer to the five units A, B/C, D, E and F in the remainder of this discussion.

For each of the five units we computed, using in-house developed software written in Python, the number of applications of context-directed swaps or context-directed reversals performed by the HNS algorithm to permute the synteny block order of one species to produce the corresponding synteny block order of another species. This was done with each species considered as reference species. Since HNS gives preference to block interchanges, the number of reversals in our derived data is low.

Note that although we used the full gene lists from flybase.org, using pointer lists and ciliate operations automatically reduces to performing ciliate sorting operations on synteny blocks between pairs of species.

[10] There are exceptions: see, for example, Figure 1 of [Schaeffer et al. 2008]. None of the exceptional species is considered in our paper.

From our data about the number of context-directed swaps, s, and reversals, r, we define a corresponding distance matrix by using the formula $s + r/2$. As the reader would observe from examining our data, this in fact does define a metric[11].

Then we applied the unweighted pair group method with arithmetic mean, also known as the UPGMA algorithm[12], to these metrics. We used an in-house developed MAPLE implementation of UPGMA to compute these phylogenies.

The Appendix contains the data, derived distance matrices and corresponding phylogenetic trees for the five units in Figures 9, 10, 11, 12 and 13. An entry in the format "$r : s$" in row i and column j of a table is interpreted as follows: r denotes the number of context-directed reversals (cdr operations), while s denotes the number of context-directed block interchanges (cds operations) executed by the HNS algorithm to convert the permutation of the species in row i to that of the species in column j. Thus the species in column j is the *reference species*. The total for whole genomes is given in Figure 14.

We used the timeline given in Figures 1 and 3 of [Hahn et al. 2007] to calibrate the timeline in our phylogenetic trees[13]. This calibration is a rough timeline: Our work describes evolutionary relationships among instances of a specific chromosome present in these eight species. The evolutionary timeline for a chromosome need not agree with the evolutionary timeline for speciation. According to Figures 1 and 3 of [Hahn et al. 2007], the time span from the earliest common ancestor of our species is roughly 60 million years.

Discussion

Comparison of our results in the Appendix and the results of [Bhutkar et al. 2008, Table 2] show a significant difference in the number of sorting operations, with ours typically higher. One reason for these differences lies in our definition of synteny blocks: We allow blocks consisting of a single gene, and we do not allow blocks containing different gene orders. Thus, we have a larger number of synteny blocks to be sorted, and our computations took into account all orthologous genes. This point is illustrated by comparing the number of synteny blocks for Muller element E

[11] There are strong grounds for equating the value of two reversals with that of a single swap. As computations show, the result (given in the Appendix) is a matrix that is symmetric over its diagonal. It is also evident that the number of sorting operations to sort permutation α to obtain permutation β, plus the number of sorting operations to sort permutation β to permutation γ, is no smaller than the number of sorting operations to directly sort permutation α to permutation γ. Thus, the triangle inequality holds.

[12] This is Algorithm 4.1 in [Clote and Backofen 2000]. A good exposition is also given in Chapter 27 of [Barton et al. 2007], available online at www.evolution-textbook.org.

[13] We could have used alternative timelines, such as the timelines given in the figure at the DroSpeGe website http://insects.eugenes.org/DroSpeGe/. Whichever published timeline one chooses will determine the corresponding calibration applied to our data.

computational unit	cso
A	266.75
B/C	207.75
D	247.25
E	364.25
F	8.5

Table 1. Ciliate sorting operations since most recent common ancestor of all considered species.

for *D. yakuba*, *D. sechellia* and *D. simulans* (computed relative to *D. melanogaster*) reported in [loc. cit., Table 5] with the actual number of sorting operations reported for these species (with *D. melanogaster* as reference) in our Figure 12. Moreover, whereas in [loc. cit.] the authors used unconstrained reversals as sorting operation, we used context-directed reversals. Additionally, in [loc. cit.] genes that suggest that a transposition is responsible for the rearrangement were excluded from the analysis. We included all orthologous genes since the sorting operation of context-directed swaps (block interchanges) accounts also for transpositions.

Comparison of the phylogenies in the Appendix with the phylogeny in [loc. cit., Figure 8] or with the phylogeny of sequenced species at flybase.org[14] indicate that our placement of *D. sechellia* is in all cases quite different. The placement of *D. mojavensis, D. virilis* and *D. grimshawi* relative to each other and to the other species agrees with both of these phylogenies for all but Muller elements A and E.

By using the UPGMA algorithm to construct phylogenies from distance matrices, we assumed a uniform rate of evolution for the Muller elements. Comparing these uniform rates among the different chromosomes indicates that no two individual chromosomes undergo permutations at the same rates. Our sorting data suggests the upper bounds in Table 1 on the number of ciliate sorting operations (cso) since the most recent common ancestor of all the species considered.

These numbers were computed by taking the largest ciliate sorting distance achieved between a pair of the considered species, and dividing[15] by 2 to obtain an estimate of the number of ciliate sorting operations to each species' corresponding genomic element since their most recent common ancestor.

The Muller F element has undergone remarkably few permutations in comparison with the other Muller elements. Muller element E appears to be the most susceptible to permutation, while Muller element F appears the most "resistant" to permutation. This, however, may be a biased view of susceptibility to permutation since these computational units do not harbor the same number of genes or synteny blocks. As

[14] http://flybase.org/static_pages/species/sequenced_species.html

[15] Using our hypothesis of uniform rate of evolution.

indicated in [Hochman 1971], chromosome 4 (Muller element F) is generally a very small chromosome: it may contain fewer than 100 genes (see, for example, the results regarding Muller element F for various species in [Schaeffer et al. 2008]). The other Muller elements each contains well over 1000 genes. Thus one would expect the number of rearrangements needed to sort one species' chromosome 4 gene content to that of another species to be relatively low in comparison with the other, larger, chromosomes.

Tables 5 and 6 of [Bhutkar et al. 2008] report rearrangement rates that are computed from the number of synteny blocks relative to *D. melanogaster*, the nucleotide length of the Muller element, and the estimated divergence time for the species in question. These rates assume that arbitrary reversals cause the rearrangements and thus ignore genes deemed to have been moved by other sorting mechanisms, and use a definition of synteny block that ignores certain rearrangements. In the case of our context-directed sorting operations, a more appropriate measure of "susceptibility to permutation" should probably take into account additional parameters regarding nucleotide patterns in the Muller elements. Progress in this regard would address the third[16] and fourth[17] questions raised in [Schaeffer et al. 2008, pp. 1603–1604], phrased for arbitrary reversals, and may also indicate whether context-directed reversals and block interchanges are more suitable sorting operations for phylogenetic analyses based on permutations of genomic material. Such rearrangement rates may be used as "susceptibility coefficients", measuring the susceptibility of a genomic element to rearrangement.

According to [Bhutkar et al. 2008, Figure 3], the F element of *D. willistoni* (which is not among the species we considered) has been absorbed in the E-element of *D. willistoni*. It would be interesting to "distill" the *D. willistoni* F-element from the *D. willistoni* E-element, and compare its level of permutation relative to the F-element of the eight species in our study. Establishing susceptibility coefficients may enable us to obtain from the current permutation state of the distilled *D. willistoni* F-element, and established evolutionary time distances for the fruit fly phylogeny, an estimate of when absorption of the F-element into the E-element took place.

Similarly, by separating the treatment of the B and C elements, calculating the corresponding susceptibility coefficients of these elements, and distilling the B-element components and the C-element components for *D. ananassae*, one may be able to estimate when these transpositions occurred. Figure 3 of [loc. cit.] also indicates that part of *D. pseudoobscura*'s Muller A element was transposed to its Muller E element. Susceptibility coefficients may be useful in estimating when

[16] "...how do new inversions originate?" This can be expanded to include the question of how new block interchanges originate.

[17] "...what is the molecular basis for gene arrangement polymorphism?"

this transposition occurred. An investigation of the structural properties of the chromosomes involved in these interchromosomal translocations may also reveal if any DNA motifs promote these translocations.

The differences in phylogenies for different chromosomal domains in the considered species suggest the possibility of inferring from Mendelian inheritance hypotheses and diploidy of the fruit fly genomes, interbreeding among ancestor species that would produce the observed chromosomal configurations.

We relied on the UPGMA algorithm for constructing our phylogenies. Other clustering techniques such as neighbor joining, or several other algorithms, as, for example, in [Clote and Backofen 2000], may reveal finer details than the technique applied here.

While using ciliate operations to compute the permutation-based distances between pairs of species, we found permutations which are not reducible to each other by ciliate operations. In contrast to the case for unrestricted block interchanges and unrestricted reversals, not all permutations are invertible by context-directed block interchanges and reversals. When our algorithm terminates with a destination of length 4 instead of 2, this indicates that the two permutations involved in the distance measure require an additional transposition to complete the transformation. Though we have not done so in our current paper, the fact of noninvertibility by ciliate decryptome operations could be taken as an additional parameter in measuring evolutionary distance. Instead, in this paper we counted this additional transposition needed at the end as a single step towards the distance. An argument can be made that the necessity of this additional transposition should be accounted for more significantly in computing evolutionary distance. It also raises the question of determining an easily applicable characterization of permutations that are invertible by constrained block interchanges or reversals. The problem of mathematically characterizing permutations that are invertible by context-directed operations has been solved in subsequent work [Adamyk et al. $\geq$ 2016].

Finally, although the HNS algorithm finds in polynomial time the data needed to construct a distance matrix, we do not propose that this algorithm finds optimal data in the following sense: when one permutation can be transformed to another by means of context-directed reversals and block interchanges, what is the least number of these operations needed for such a transformation? The answer for context-directed block interchanges has been obtained in [Adamyk et al. $\geq$ 2016]. The minimal number of operations may depend on strategic sorting decisions made while sorting a permutation. One may inquire whether certain permutations require less strategic decision making in order to obtain a successful sorting. The permutations requiring the least number of strategic decisions for context-directed block interchanges have been characterized in [Anderson et al. $\geq$ 2016], but a complete answer is currently not known.

Appendix: The distance matrices underlying the application of UPGMA to the five chromosomes of eight fruit fly species

	D. vir	*D. gri*	*D. sim*	*D. moj*	*D. mel*	*D. ere*	*D. yak*	*D. sec*
D. vir		32:431	38:463	31:438	33:403	40:426	29:414	35:514
D. gri	26:434		36:446	35:430	36:381	45:404	40:391	45:504
D. sim	36:464	34:447		35:460	8:268	19:311	21:282	26:505
D. moj	29:439	37:429	41:457		41:407	34:434	36:422	37:515
D. mel	37:401	40:379	6:269	45:405		1:171	35:93	19:482
D. ere	36:428	43:405	29:306	42:430	3:170		25:182	28:499
D. yak	43:407	40:391	31:277	50:415	11:105	25:182		29:481
D. sec	43:510	39:507	20:508	39:514	7:488	22:502	17:487	

	D. vir	*D. gri*	*D. sim*	*D. moj*	*D. mel*	*D. ere*	*D. yak*	*D. sec*
D. vir	0.0	447.0	482.0	453.5	419.5	446.0	428.5	531.5
D. gri	447.0	0.0	464.0	447.5	399.0	426.5	411.0	526.5
D. sim	482.0	464.0	0.0	477.5	272.0	320.5	292.5	518.0
D. moj	453.5	447.5	477.5	0.0	427.5	451.0	440.0	533.5
D. mel	419.5	399.0	272.0	427.5	0.0	171.5	110.5	491.5
D. ere	446.0	426.5	320.5	451.0	171.5	0.0	194.5	513.0
D. yak	428.5	411.0	292.5	440.0	110.5	194.5	0.0	495.5
D. sec	531.5	526.5	518.0	533.5	491.5	513.0	495.5	0.0

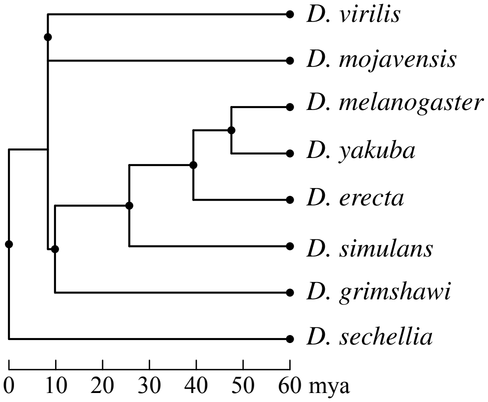

Figure 9. Data, distance matrix and resulting phylogeny for the Muller *A*-element.

	D. vir	*D. gri*	*D. sim*	*D. moj*	*D. mel*	*D. ere*	*D. yak*	*D. sec*
D. vir		62:255	44:290	31:161	41:262	51:324	63:332	38:375
D. gri	56:258		43:318	38:187	52:280	62:342	50:370	45:393
D. sim	48:288	49:315		53:205	2: 95	24:188	16:223	9:256
D. moj	67:143	32:190	55:204		49:173	51:254	48:285	47:319
D. mel	59:253	58:277	8: 92	49:173		19:159	101:145	3:229
D. ere	45:327	44:351	14:193	57:251	11:163		9:249	32:275
D. yak	49:339	42:374	14:224	44:287	7:192	15:246		34:286
D. sec	52:368	49:391	7:257	41:322	3:229	38:272	46:280	

	D. vir	*D. gri*	*D. sim*	*D. moj*	*D. mel*	*D. ere*	*D. yak*	*D. sec*
D. vir		286	312	176.5	282.5	349.5	363.5	394
D. gri	286		339.5	206	306	373	395	415.5
D. sim	312	339.5		231.5	96	200	331	260.5
D. moj	176.5	206	231.5		197.5	279.5	309	342.5
D. mel	282.5	306	96	197.5		168.5	195.5	230.5
D. ere	349.5	373	200	279.5	168.5		253.5	291
D. yak	363.5	395	331	309	195.5	253.5		303
D. sec	394	415.5	260.5	342.5	230.5	291	303	

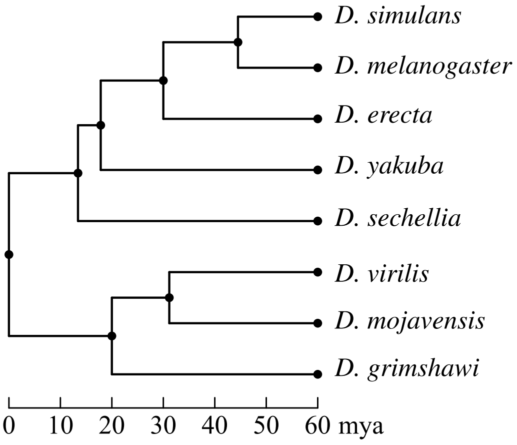

Figure 10. Data, distance matrix and resulting phylogeny for the Muller B/C-element.

	D. vir	*D. gri*	*D. sim*	*D. moj*	*D. mel*	*D. ere*	*D. yak*	*D. sec*
D. vir		21:124	60:193	27:113	69:175	58:160	53:231	60:450
D. gri	27:121		51:210	29:154	52:187	56:174	56:244	59:460
D. sim	68:189	59:206		65:219	2: 69	5: 56	10:129	2:390
D. moj	23:115	23:157	59:222		53:214	59:192	55:257	51:469
D. mel	69:175	62:182	2: 69	81:200		8: 35	10:109	0:388
D. ere	66:156	58:173	7: 55	67:188	10: 34		12: 79	90:337
D. yak	59:228	64:240	26:121	71:249	14:107	18: 76		12:416
D. sec	54:453	55:462	2:390	49:470	0:388	4:380	12:416	

	D. vir	*D. gri*	*D. sim*	*D. moj*	*D. mel*	*D. ere*	*D. yak*	*D. sec*
D. vir	0	134.5	223	126.5	209.5	189	257.5	480
D. gri	134.5	0	235.5	168.5	213	202	272	489.5
D. sim	223	235.5	0	251.5	70	58.5	134	391
D. moj	126.5	168.5	251.5	0	240.5	221.5	284.5	494.5
D. mel	209.5	213	70	240.5	0	39	114	388
D. ere	189	202	58.5	221.5	39	0	85	382
D. yak	257.5	272	134	284.5	114	85	0	422
D. sec	480	489.5	391	494.5	388	382	422	0

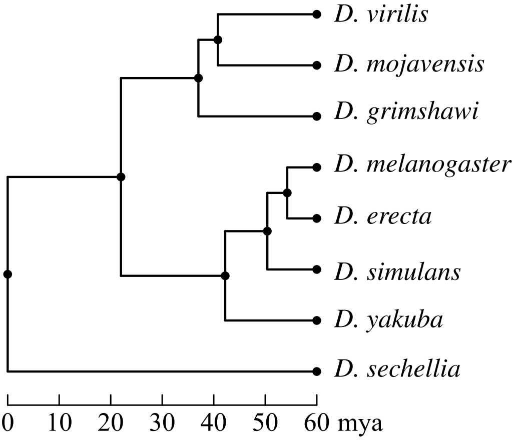

Figure 11. Data, distance matrix and resulting phylogeny for the Muller *D*-element.

	D. vir	*D. gri*	*D. sim*	*D. moj*	*D. mel*	*D. ere*	*D. yak*	*D. sec*
D. vir		47:634	40:451	27:340	40:432	46:551	42:436	41:598
D. gri	47:634		47:616	25:549	46:602	55:664	54:603	47:705
D. sim	52:445	57:611		45:213	8: 71	142:241	13: 75	14:347
D. moj	89:309	53:535	39:216		39:185	43:401	31:194	45:446
D. mel	44:430	54:598	8: 71	39:185		196:196	7: 38	21:334
D. ere	50:549	55:664	10:307	39:403	6:291		12:291	38:428
D. yak	54:430	62:599	19: 72	43:188	15: 34	8:293		23:334
D. sec	51:593	53:702	14:347	39:449	5:342	38:428	9:341	

	D. vir	*D. gri*	*D. sim*	*D. moj*	*D. mel*	*D. ere*	*D. yak*	*D. sec*
D. vir		657.5	471	353.5	452	574	457	618.5
D. gri	657.5		639.5	561.5	625	691.5	630	728.5
D. sim	471	639.5		235.5	75	312	81.5	354
D. moj	353.5	561.5	235.5		204.5	422.5	209.5	468.5
D. mel	452	625	75	204.5		294	41.5	344.5
D. ere	574	691.5	312	422.5	294		297	447
D. yak	457	630	81.5	209.5	41.5	297		345.5
D. sec	618.5	728.5	354	468.5	344.5	447	345.5	

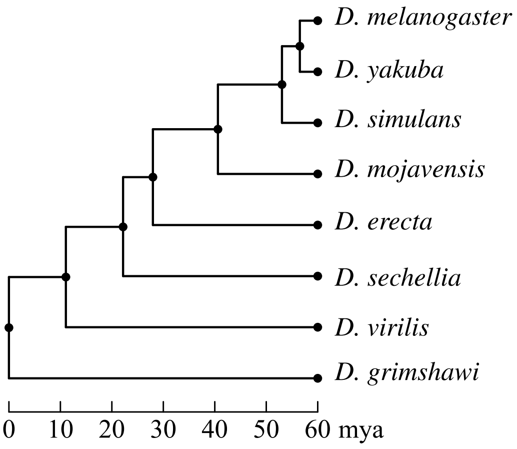

Figure 12. Data, distance matrix and resulting phylogeny for the Muller E-element.

	D. vir	*D. gri*	*D. sim*	*D. moj*	*D. mel*	*D. ere*	*D. yak*	*D. sec*
D. vir		3:5	12:8	2:1	11:6	11:6	11:6	8:13
D. gri	3:5		10:12	3:4	11:9	11:9	11:9	10:12
D. sim	8:10	8:13		10:9	4:5	4:5	4:5	4:13
D. moj	2:1	3:4	6:11		7:7	7:7	7:7	9:12
D. mel	9:7	7:11	6:4	7:7		0:0	0:0	0:12
D. ere	9:7	11:9	6:4	9:6	0:0		0:0	0:12
D. yak	9:7	7:11	6:4	7:7	0:0	0:0		0:12
D. sec	8:13	8:13	2:14	9:12	0:12	0:12	0:12	

	D. vir	*D. gri*	*D. sim*	*D. moj*	*D. mel*	*D. ere*	*D. yak*	*D. sec*
D. vir		6.5	14	2	11.5	11.5	11.5	17
D. gri	6.5		17	5.5	14.5	14.5	14.5	17
D. sim	14	17		14	7	7	7	15
D. moj	2	5.5	14		10.5	10.5	10.5	16.5
D. mel	11.5	14.5	7	10.5		0	0	12
D. ere	11.5	14.5	7	10.5	0		0	12
D. yak	11.5	14.5	7	10.5	0	0		12
D. sec	17	17	15	16.5	12	12	12	

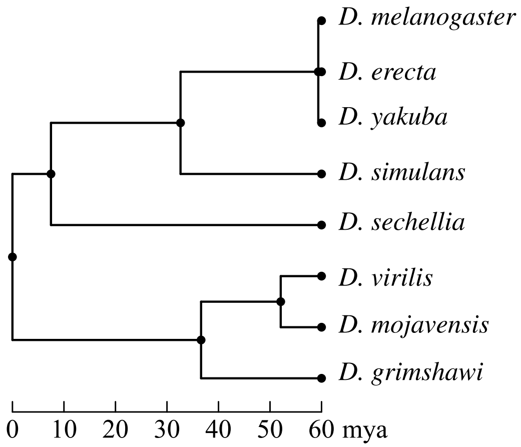

Figure 13. Data, distance matrix and resulting phylogeny for the Muller F-element.

	D. vir	*D. gri*	*D. sim*	*D. moj*	*D. mel*	*D. ere*	*D. yak*	*D. sec*
D. vir		165:1449	194:1405	118:1053	194:1278	206:1467	198:1419	182:1950
D. gri	165:1449		187:1602	130:1324	197:1459	229:1593	211:1617	206:2074
D. sim	194:1405	187:1602		208:1106	24:508	194:801	64:722	55:1511
D. moj	118:1053	130:1324	208:1106		189:986	194:1288	177:1165	189:1761
D. mel	194:1278	197:1459	24:508	189:986		224:561	153:385	43:1445
D. ere	206:1467	229:1593	194:801	194:1288	224:561		58:801	188:1551
D. yak	198:1419	211:1617	64:722	177:1165	153:385	58:801		98:1529
D. sec	182:1950	206:2074	55:1511	189:1761	43:1445	188:1551	98:1529	

	D. vir	*D. gri*	*D. sim*	*D. moj*	*D. mel*	*D. ere*	*D. yak*	*D. sec*
D. vir	0	1231.5	1502	1112	1375	1570	1518	2041
D. gri	1231.5	0	1695.5	1389	1557.5	1707.5	1722.5	2177
D. sim	1502	1695.5	0	1210	520	898	746	1538.5
D. moj	1112	1389	1210	0	1080.5	1385	1253.5	1655.5
D. mel	1375	1557.5	520	1080.5	0	673	461.5	1463.5
D. ere	1570	1707.5	898	1385	673	0	830	1645
D. yak	1518	1722.5	746	1253.5	461.5	830	0	1578
D. sec	2041	2177	1538.5	1655.5	1463.5	1645	1578	0

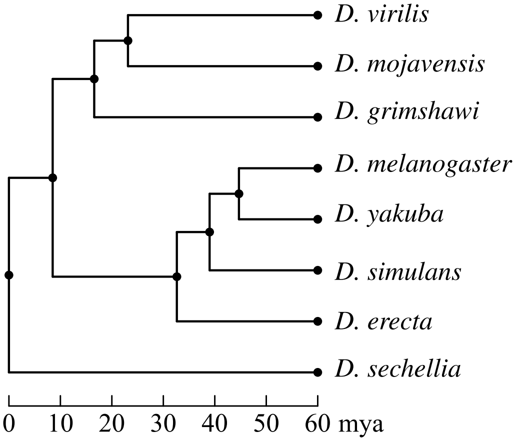

Figure 14. Data, distance matrix and resulting phylogeny for the whole genome.

Acknowledgement

We gratefully acknowledge that the advice of a very careful referee helped to greatly improve the readability of this paper.

References

[Adamyk et al. $\geq$ 2016] K. Adamyk, E. Holmes, G. Mayfield, D. J. Moritz, and M. Scheepers, "Games, genomes and graphs", preprint.

[Anderson et al. $\geq$ 2016] C. Anderson, M. Scheepers, M. Warner, and H. Wauck, "On permutations optimized for sorting by ciliate operations", preprint.

[Angeleska et al. 2007] A. Angeleska, N. Jonoska, M. Saito, and L. F. Landweber, "RNA-guided DNA assembly", *J. Theoret. Biol.* **248**:4 (2007), 706–720. MR 2899092

[Bafna and Pevzner 1995] V. Bafna and P. Pevzner, "Sorting by reversals: genome rearrangements in plant organelles and evolutionary history of the X chromosome", *Mol. Biol. Evol.* **12**:2 (1995), 239–246.

[Bafna and Pevzner 1998] V. Bafna and P. A. Pevzner, "Sorting by transpositions", *SIAM J. Discrete Math.* **11**:2 (1998), 224–240. MR 99e:05002 Zbl 0973.92014

[Barton et al. 2007] N. H. Barton, D. E. G. Briggs, J. A. Eisen, D. B. Goldstein, and N. H. Patel, *Evolution*, Cold Spring Harbor Laboratory Press, 2007.

[Bhutkar et al. 2008] A. Bhutkar, S. W. Schaeffer, S. M. Russo, M. Xu, T. F. Smith, and W. M. Gelbart, "Chromosomal rearrangement inferred from comparisons of 12 *Drosophila* genomes", *Genetics* **179**:3 (2008), 1657–1680.

[Clote and Backofen 2000] P. Clote and R. Backofen, *Computational molecular biology: an introduction*, John Wiley & Sons, Ltd., Chichester, 2000. MR 2002h:92021 Zbl 0955.92013

[Coghran and Wolfe 2002] A. Coghran and K. H. Wolfe, "Fourfold faster rate of genome rearrangement in nematodes than in *Drosophila*", *Genome Res.* **12**:6 (2002), 857–867.

[Dobzhansky and Sturtevant 1938] T. Dobzhansky and A. H. Sturtevant, "Inversions in the chromosomes of *Drosophila pseudoobscura*", *Genetics* **23**:1 (1938), 28–64.

[Ehrenfeucht et al. 2004] A. Ehrenfeucht, T. Harju, I. Petre, D. M. Prescott, and G. Rozenberg, *Computation in living cells: gene assembly in ciliates*, Springer, Berlin, 2004. Zbl 1069.68048

[Hahn et al. 2007] M. Hahn, M. Han, and S.-G. Han, "Gene family evolution across 12 *Drosophila* genomes", *PLOS Genetics* **3**:11 (2007), 2135–2146.

[Hannenhalli and Pevzner 1999] S. Hannenhalli and P. A. Pevzner, "Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals", *J. ACM* **46**:1 (1999), 1–27. MR 2000j:92013 Zbl 1064.92510

[Hochman 1971] B. Hochman, "Analysis of chromosome 4 in *Drosophila melanogaster* II: ethyl methanesulfonate induced lethals", *Genetics* **67**:2 (1971), 235–252.

[Hughes 2000] D. Hughes, "Evaluating genome dynamics: the constraints on rearrangements within bacterial genomes", *Genome Biology* **1**:6 (2000), 1–8.

[Mira and Meidanis 2007] C. Mira and J. Meidanis, "Sorting by block-interchanges and signed reversals", pp. 670–676 in *Proceedings of the Fourth International Conference on Information Technology: New Generations* (Las Vegas, NV, 2007), edited by S. Latifi, IEEE Computer Society, Los Alamitos, CA, 2007.

[Möllenbeck et al. 2008] M. Möllenbeck, Y. Zhou, A. R. O. Cavalcanti, F. Jönsson, B. P. Higgins, W.-J. Chang, S. Juranek, T. G. Doak, G. Rozenberg, H. J. Lipps, and L. F. Landweber, "The pathway to detangle a scrambled gene", *PLOS One* **3**:6 (2008), e2330.

[Muller 1940] H. J. Muller, "Bearings of the *Drosophila* work on systematics", pp. 185–268 in *The New Systematics*, edited by J. Huxley, Clarendon Press, Oxford, 1940.

[Nowacki et al. 2007] M. Nowacki, V. Vijayan, Y. Zhou, K. Schotanus, T. Doak, and L. Landweber, "RNA-mediated epigenetic programming of a genome-rearrangement pathway", *Nature* **451**:7175 (2007), 153–158.

[Pevzner and Tesler 2003] P. A. Pevzner and G. Tesler, "Genome rearrangements in mammalian evolution: lessons from human and mouse genomes", *Genome Res.* **13**:1 (2003), 37–45.

[Prescott 1994] D. M. Prescott, "The DNA of ciliated protozoa", *Microbiol. Rev.* **58**:2 (1994), 233–267.

[Prescott 2000] D. M. Prescott, "Genome gymnastics: unique modes of DNA evolution and processing in ciliates", *Nature Reviews Genetics* **1** (2000), 191–198.

[Prescott et al. 2003] D. M. Prescott, A. Ehrenfeucht, and G. Rozenberg, "Template-guided recombination for IES elimination and unscrambling of genes in stichotrichous ciliates", *J. Theoret. Biol.* **222**:3 (2003), 323–330. MR 2067536

[Schaeffer et al. 2008] S. W. Schaeffer, A. Bhutkar, B. F. McAllister, M. Matsuda, L. M. Matzkin, P. M. O'Grady, C. Rohde, V. L. S. Valente, M. Aguadé, W. W. Anderson, K. Edwards, A. C. L. Garcia, J. Goodman, J. Hartigan, E. Kataoka, R. T. Lapoint, E. R. Lozovsky, C. A. Machado, M. A. F. Noor, M. Papaceit, L. K. Reed, S. Richards, T. T. Rieger, S. M. Russo, H. Sato, C. Segarra, C. R. Smith, T. F. Smith, V. Strelets, Y. N. Tobari, Y. Tomimura, M. Wasserman, T. Watts, R. Wilson, K. Yoshida, T. A. Markow, W. M. Gelbart, and T. C. Kaufman, "Polytene chromosomal maps of 11 *Drosophila* species: the order of genomic scaffolds inferred from genetic and physical maps", *Genetics* **179**:3 (2008), 1601–1655.

[Sturtevant and Dobzhansky 1936] A. H. Sturtevant and T. Dobzhansky, "Inversions in the third chromosome of wild races of *Drosophila pseudoobscura*, and their use in the study of the history of the species", *PNAS* **22**:7 (1936), 448–450.

[Yancopoulos et al. 2005] S. Yancopoulos, O. Attie, and R. Friedberg, "Efficient sorting of genomic permutations by translocation, inversion and block interchange", *Bioinformatics* **21**:16 (2005), 3340–3346.

Received: 2013-01-23 Revised: 2014-12-11 Accepted: 2014-12-21

jlherlin@indiana.edu *Department of Mathematics, Indiana University, Rawles Hall,*
831 East 3rd Street, Bloomington, IN 47405, United States

anelson@math.utah.edu *Department of Mathematics, University of Utah, 155 S 1400 E,*
Room 233, Salt Lake City, UT 84112-0090, United States

mscheepe@boisestate.edu *Department of Mathematics, Boise State University,*
Boise, ID 83725, United States

involve

msp.org/involve

MANAGING EDITOR

Kenneth S. Berenhaut, Wake Forest University, USA, berenhks@wfu.edu

PRODUCTION

Silvio Levy, Scientific Editor

Cover: Alex Scorpan

See inside back cover or msp.org/involve for submission instructions. The subscription price for 2016 is US $160/year for the electronic version, and $215/year (+$35, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscribers address should be sent to MSP.

Involve (ISSN 1944-4184 electronic, 1944-4176 printed) at Mathematical Sciences Publishers, 798 Evans Hall #3840, c/o University of California, Berkeley, CA 94720-3840, is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

Involve peer review and production are managed by EditFlow® from Mathematical Sciences Publishers.

PUBLISHED BY

mathematical sciences publishers

nonprofit scientific publishing

http://msp.org/

involve

2016 vol. 9 no. 1