J. Math. Biol. (2015) 71:1149–1178
DOI 10.1007/s00285-014-0852-1

Mathematical Biology

Algebraic double cut and join

A group-theoretic approach to the operator on multichromosomal genomes

Sangeeta Bhatia · Attila Egri-Nagy · Andrew R. Francis

Received: 9 April 2014 / Revised: 25 November 2014 / Published online: 11 December 2014

Abstract Establishing a distance between genomes is a significant problem in computational genomics, because its solution can be used to establish evolutionary relationships including phylogeny. The "double cut and join" (DCJ) model of chromosomal rearrangement proposed by Yancopoulos et al. (Bioinformatics 21:3340–3346, 2005) has received attention as it can model inversions, translocations, fusion and fission on a multichromosomal genome that may contain both linear and circular chromosomes. In this paper, we realize the DCJ operator as a group action on the space of multichromosomal genomes. We study this group action, deriving some properties of the group and finding group-theoretic analogues for the key results in the DCJ theory.

Keywords Genome rearrangements · Double cut and join · Group theory · Inversion · Algebraic biology

Mathematics Subject Classification 20B30 · 92C42

1 Introduction

The use of genome rearrangements to estimate evolutionary distance dates back as far as Watterson et al. (1982). The novelty of this approach is that it ignores single

This research was supported by Australian Research Council grants DP130100248 and FT100100898.

S. Bhatia (✉)· A. Egri-Nagy · A. R. Francis
Centre for Research in Mathematics, University of Western Sydney, Penrith, NSW 2751, Australia
e-mail: s.bhatia@uws.edu.au

A. Egri-Nagy
e-mail: a.egri-nagy@uws.edu.au

A. R. Francis
e-mail: a.francis@uws.edu.au

nucleotide polymorphisms (SNPs) and takes genes and their relative positions (and orientation) as the fundamental unit of DNA for the purposes of distance calculations. This is particularly valuable in the case of bacterial DNA, which undergoes relatively frequent rearrangement (relative to eukaryotes, that is), but which also experiences significant horizontal, or lateral, gene transfer (HGT). The use of large-scale rearrangements to establish distance is thought to be less vulnerable to the effects of HGT than the use of SNPs (Darling et al. 2008).

The *double cut and join* (DCJ) operator, introduced by Yancopoulos et al. (2005) [see also Bergeron et al. (2006)], provided a significant breakthrough by treating a much larger family of operations acting on a more general, multichromosomal genome, and showing how distance can be expressed in a very simple formula based on features of a graph derived directly from the genome arrangements. While the DCJ operator treats all operations in its remit as equally likely, it is possible that this operator may provide a valuable base for operators that account for differences in frequency of different operations, and may potentially be specialised to genomes with specific chromosomal structure (such as a single circular chromosome).

With this potential in mind, in this paper we translate the DCJ operator into a group-theoretic setting. We show that by expressing the multi-chromosomal genome with n oriented regions as a permutation of $\{1, \ldots, 2n\}$, a DCJ operator can be defined as an action on the genome. Hence, the set of double cut and join operators generates a group acting on the entire genome space. The DCJ distance is then a path distance on the Cayley graph of this group [as described in Egri-Nagy et al. (2014)]. We show how the DCJ distance between two genomes can be obtained in a very simple way from the permutation encoding of the genomes. We obtain a formulation for the DCJ distance that is analogous to the distance formula found by Yancopoulos et al. (2005), but is expressed in terms of features of a permutation. This is derived independently of the established DCJ theory.

Over the last decade, there have been several examples of algebraic approaches to modeling biological phenomena, particularly in the genomic distance literature. While the traditional approach to the problem of finding distance between genomes is to cast them as permutations, limited use has been made of the powerful machinery that algebra provides to deal with permutations. Adopting an algebraic viewpoint might in fact reveal deep insights and lead to simplification.

A recent example of this is the work of Lu et al. (2006) who used the theory of symmetric groups to give an algorithm that gives a sorting sequence between circular genomes using fission, fusion and block interchanges. More recently, group theory has been used by the authors' group to calculate the inversion distance between circular genomes under one model (Egri-Nagy et al. 2014), and a wider algebraic framework has been proposed that includes DNA knotting (Francis 2014).

A circular genome is modeled as a cyclic permutation by Meidanis and Dias (2000), treating a genome as a permutation of the genes $a_1, a_2, \ldots, a_n$. Writing the circular genome in cycle notation as $(a_1, a_2, \ldots, a_n)$ denotes that gene a_i is adjacent on the genome to gene a_{i+1}, with gene a_n being adjacent to a_1. This approach allowed them to derive many important properties of the breakpoint graph in terms of permutation products, and to give a lower bound on the transposition distance. This work was later extended in Feijão and Meidanis (2013) to include linear chromosomes. Feijão and Meidanis (2013) model a genome as a product of disjoint 2-cycles, and present a formulation of a k-break operation as a permutation. The double cut and join model then becomes a special case of the k-break operation with $k = 2$.

The model of the genome as a product of disjoint 2-cycles and the double cut and join operation as conjugation used by Feijão and Meidanis (2013), is also employed by us in this paper. The novelty of our work lies in development of this model in a completely algebraic framework, without making use of the existing theory. This allows us to present new proofs for existing results, sometimes leading to a considerable simplification of arguments such as in the result related to counting sorting scenarios (Theorem 7.3).

This paper is organized as follows. In Sect. 2 we introduce the double cut and join model, following the exposition of Bergeron et al. (2006). Section 3 explains how we can encode a genome as a product of 2-cycles, essentially extending the concept of "adjacencies" and "telomeres" from the established theory. In Sect. 4 we give the major construction of this paper, namely the definition of the DCJ operator as a group action on the genome space. While the standard definition has several cases depending on the arrangement of the genome, this action requires just two cases. In order to find a distance formula in this model (Sect. 6), we first need to establish some results about products of involutions, covered in Sect. 5. The main result, establishing distance in this model, is given in the following theorem:

Main Theorem (Theorem 6.11) *Let G_1 and G_2 be genomes on n regions with corresponding genomic permutations π_1 and π_2. The DCJ distance between G_1 and G_2 is given by*

$$d_{DCJ}(\pi_1, \pi_2) = \frac{1}{2}(\ell_t(\pi_2\pi_1) + n_c)$$

where n_c is the number of cycles in the product $\pi_2\pi_1$ which contain two fixed points of π_1 or π_2, and ℓ_t is the transposition length.

Finally, in Sect. 7, we derive a formula for the number of optimal sorting scenarios between two genomes. That is, the number of minimal length paths in the Cayley graph of the group generated by the DCJ operators. As our work utilizes many well-known results about permutations, we have collected them in Appendix A for ease of reference. These results are stated without proofs. A complete treatment may be found in an abstract algebra text such as (Herstein 2006; Fraleigh 2003).

2 The double cut and join model

In this section we follow the notation of Bergeron et al. (2006). For a more complete introduction to the model and results, see that paper as well as Yancopoulos et al. (2005).

2.1 The genome graph

Before presenting the double cut and join operator, we first explain how multichromosomal genomes are modeled. In this model, a gene is essentially an oriented section of

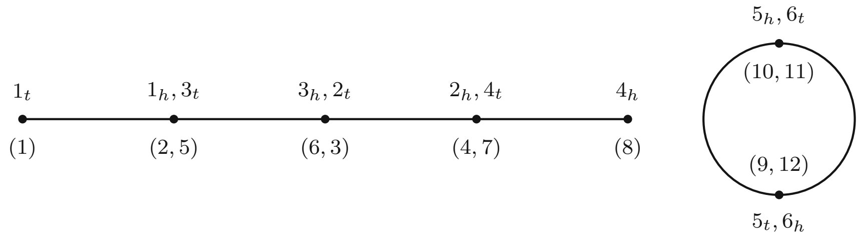

Fig. 1 The genome graph of a genome with one linear chromosome containing genes numbered 1, 2, 3 and 4, and one circular chromosome containing genes numbered 5 and 6. The vertex set of the graph is $\{\{1_t\}, \{1_h, 3_t\}, \{3_h, 2_t\}, \{2_h, 4_t\}, \{4_h\}, \{5_h, 6_t\}, \{6_h, 5_t\}\}$. Edges are drawn between extremities of the same gene. The map ϕ (Definition 3.2) maps the set of extremities $\{1_t, 1_h, 2_t, 2_h, \ldots, 6_h\}$ into $\{1, 2, 3, 4, \ldots, 12\}$, with $\phi(1_t) = 1$, $\phi(1_h) = 2$ and so on. 1_t and 4_h are telomeres, hence 1 and 8 are fixed points of the permutation π (Definition 3.3). 1_h is connected to 3_t which is captured by the 2-cycle (2, 5) in the permutation encoding. The other 2-cycles can be similarly interpreted. The above genome is thus encoded as the permutation (2, 5)(3, 6)(4, 7)(10, 11)(9, 12)

the DNA and its two ends are called its *extremities*. This of course differs from the biological meaning of the word gene and is closer to what are referred to as "conserved blocks" in the rearrangement literature [for example see Hannenhalli and Pevzner (1995), Lin and Tang (2006)]. However, for convenience we will use the words gene and region interchangeably. The extremities of the gene a are denoted a_t and a_h where the subscripts stand for tail and head respectively.

To represent a genome, considered as an arrangement of oriented genes, it is sufficient to note which extremities are adjacent on the genome. An extremity that is not adjacent to any other is the end point of a linear section of the genome and is called a *telomere*. An (unordered) pair of extremities that are adjacent on the genome is referred to as an *adjacency*. For instance, the adjacency $\{a_t, b_h\}$ indicates that the tail of gene a is adjacent to the head of gene b on the genome. Note that an extremity can be adjacent to at most one other extremity.

Thus, in this model a genome is represented by a partition of the set of extremeties of the genes into subsets of cardinality 1 (telomeres) or 2 (adjacencies). Equivalently, the genome can be viewed as a graph whose vertex set is the set of all adjacencies and telomeres and whose edges are drawn between the extremities of the same gene. Thus every vertex of a genome graph has degree one or two. Figure 1 illustrates a genome graph.

2.2 The double cut and join operator

The *double cut and join* (DCJ) operator acts on a pair of vertices of a genome graph in one of the following ways:

1. $\{p, q\}, \{r, s\}$ may be changed to $\{p, r\}, \{q, s\}$ or $\{p, s\}, \{q, r\}$,
2. $\{p, q\}, \{r\}$ may be changed to $\{p, r\}, \{q\}$ or $\{q, r\}, \{p\}$,
3. $\{p, q\}$ may be changed to $\{p\}, \{q\}$ or $\{p\}, \{q\}$ changed to $\{p, q\}$.

Depending on the vertices that it acts on, the double cut and join operator can simulate the inversion, excision and translocation of a section of the genome as well as fusion and fission of chromosomes. Figure 2 presents some examples.

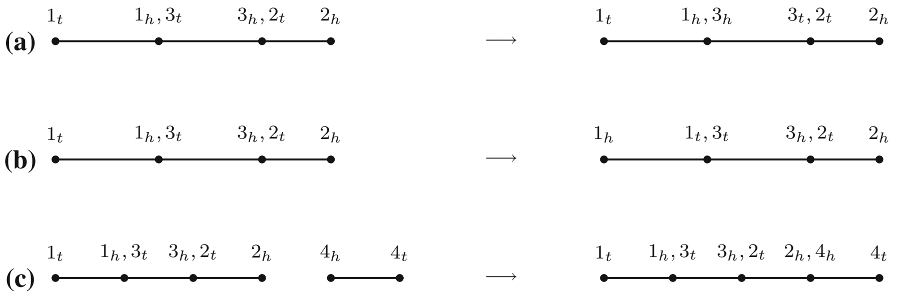

Fig. 2 **a** $\{1_h, 3_t\}$, $\{3_h, 2_t\}$ is changed to $\{1_h, 3_h\}$, $\{3_t, 2_t\}$ leading to an inversion. **b** $\{1_t\}$, $\{1_h, 3_t\}$ is changed to $\{1_h\}$, $\{1_t, 3_t\}$, another inversion. **c** $\{2_h\}$, $\{4_h\}$ is changed to $\{2_h, 4_h\}$, a fusion

2.3 The double cut and join distance

The DCJ distance between genomes G_1 and G_2 is the minimal number of DCJ operations required to change one genome into the other.

Bergeron et al. (2006) make use of a graph construct called the "adjacency graph" to determine the DCJ distance between two genomes. An *adjacency graph* $AG(G_1, G_2)$ can be drawn for any pair of genomes G_1 and G_2 defined on the same set of n genes. The vertex set of the graph is the set of all adjacencies and telomeres in G_1 and G_2. For an adjacency (or telomere) $u \in G_1$ and adjacency (or telomere) $v \in G_2$, there is an edge in $AG(G_1, G_2)$ between u and v for each gene extremity they have in common.

The vertices of the adjacency graph then have degree either one (at a telomere) or two (at an adjacency), so the graph consists of a set of cycles and a set of paths. Let c be the number of cycles and p be the number of paths of odd length in $AG(G_1, G_2)$. Bergeron et al. (2006) established that the DCJ distance between two genomes can be given in terms of these adjacency graph statistics as follows:

$$d_{DCJ}\,(G_1, G_2) = n - (c + p/2).$$

3 Genomes as permutations

We now present our reformulation of the double cut and join model. We first formalize the notion of a genome on n regions. Let $\{h, t\}$ be the extremities of a gene where h and t denote the head and tail respectively. Let $\mathbf{n}$ be the set $\{1, 2, \ldots, n\}$ enumerating the n regions.

Definition 3.1 (*Extremities*) The Cartesian product $E = \mathbf{n} \times \{h, t\}$ is the set of all extremities of n regions.

To conform to the notation used earlier in this paper and in previous literature, we will use i_h and i_t to denote the extremities (i, h) and (i, t) giving the head and tail of gene i respectively.

We define a map that assigns numeric labels to the elements of E.

Definition 3.2 (*Assignment map*) Let $\phi : E \to \mathbf{2n}$ be defined as follows:

$$\phi(i_t) = 2i - 1,$$
$$\phi(i_h) = 2i.$$

Definition 3.3 (*Genome*) A *genome* on n regions is a permutation π on the set E such that

$$\pi(i) = j \iff \pi(j) = i.$$

The above definition implies that a genomic permutation is a product of disjoint 2-cycles. The restriction in the definition of a genome captures the notion of pairing of gene extremities on a genomic strand. Therefore a 2-cycle in this formulation is an adjacency, and similarly, fixed points of a permutation are telomeres. It is important to note that at this point we use the permutation π as a static description of the genome, not as an operation, so that the 2-cycles can be considered as synonyms for unordered pairs. Furthermore, this construction with 2-cycles representing adjacencies means that the identity permutation will only arise in the trivial case in which all chromosomes in the genome contain just a single region.

As mentioned in Sect. 2, the vertex set of the genome graph consists of the adjacencies and telomeres. For every gene i, an edge is drawn between the adjacency containing i_h and i_t. In writing the genome as a permutation, the 2-cycles and the fixed points are the adjacencies and the telomeres. The assignment map ϕ tells us the correspondence between the gene extremities and the set $\mathbf{2n}$. Hence the assignment map ϕ and the genomic permutation π contain all the information that is needed to construct the genome.

Bafna and Pevzner (1993) introduced the notion of a breakpoint graph for an unsigned permutation. To extend this concept to signed permutations, they transform a signed permutation π on n elements to an unsigned permutation π' on $2n$ elements. This is done by replacing a positive integer i in π by $2i - 1$ followed by $2i$ in π' and by replacing a negative integer $-i$ in π by $2i$ followed by $2i - 1$. This transformation is precisely the labeling map ϕ. Reverse orientation of a gene on a chromosome means that the tail of the gene is present after the head of the gene. Thus in the permutation representation, a negative integer $-i$ is replaced by $2i$ (label of i_h) followed by $2i - 1$ (label of i_t).

We use cycle notation to write permutations. Thus the cycle $(i_1, i_2, i_3, \ldots, i_n)$ in a permutation α means that $\alpha(i_1) = i_2, \alpha(i_2) = i_3$ etc. and $\alpha(i_n) = i_1$ (see Appendix A). Figure 1 illustrates an example of permutation encoding of a genome on 6 regions.

A genome on n regions is a permutation of the set $\mathbf{2n}$ satisfying the constraints in Definition 3.3. Lemma 3.4 gives an expression for the number of permutations in S_{2n} satisfying this definition.

Lemma 3.4 *The number of genomes on n regions is given by*

$$\sum_{t=0}^{n} \binom{2n}{2t} (2n - 2t - 1)!! = \sum_{t=0}^{n} \binom{2n}{2t} (2t - 1)!!.$$

Table 1 The number of genomes on n regions (also the number of tableaux on $2n$ elements)

#Regions	#Genomes
1	2
2	10
3	76
4	764
5	9,496
6	140,152
7	2,390,480
8	46,206,736
9	997,313,824

Proof Let the set of all genomes on n regions be $\mathit{\Gamma}_n$. Each genome is a permutation of the $2n$ extremities E, and hence $\mathit{\Gamma}_n$ is a subset of the symmetric group S_{2n}. The cardinality of $\mathit{\Gamma}_n$ can be determined as follows.

Each genomic permutation can have an even number of fixed points, since it is a product of disjoint 2-cycles and 1-cycles acting on a set of even cardinality ($2n$). This also follows from the fact that fixed points are telomeres, and a genome must have an even number of telomeres.

Let the number of fixed points be $2t$. The remaining $2n-2t$ elements must be paired off with each other. Each such pairing of the $2n - 2t$ elements defines an involution in the symmetric group S_{2n-2t} that does not have any fixed points. An involution is an element of order 2 i.e., π is an involution if π^2 is the identity permutation.

The number of such involutions is $(2n-2t-1)!!$ (Stanley 1999, pp. 15–16) where the double factorial function is the product of odd numbers i.e. $(2k-1)!! = \prod_{i=1}^{k}(2i-1)$. Therefore the cardinality of $\mathit{\Gamma}_n$ is given by

$$|\mathit{\Gamma}_n| = \sum_{t=0}^{n}\binom{2n}{2t}(2n-2t-1)!! = \sum_{t=0}^{n}\binom{2n}{2t}(2t-1)!!.$$

□

The number of genomes is already almost a billion for 9 regions. The astute observer will note that this number is also the number of tableaux on $2n$ elements, with a correspondence given by the Robinson–Schensted algorithm [see for instance Fulton (1997)]. The first nine numbers in the sequence are shown in Table 1.

4 The DCJ operator as an action on a permutation

In this section we define an algebraic version of the DCJ operator acting on the set $\mathit{\Gamma}_n$ of genomes on n regions, and show that it is an involution. Appendix A contains a summary of some results on symmetric groups that may be useful for reference in this section and the next.

As explained in the previous section, the genome is modeled as a set of unordered pairs of gene extremities (adjacencies) and single geneextremities (telomeres). A DCJ operation as defined in Bergeron et al. (2006) swaps gene extremities between two pairs (i.e. adjacencies) or a pair and a singleton, as described in Sect. 2.2.

Hence, the possible scenarios are that the two gene extremities being swapped are: adjacent to each other on the genome; both involved in different adjacencies; one of them is a telomere and the other in an adjacency; or both of them are telomeres. When a DCJ operation acts on a pair of gene extremities that form an adjacency, it splits them, producing two telomeres, and conversely when it acts on two telomeres, it combines them into an adjacency. In the two other cases, an extremity is swapped.

Thus, in the permutation representation, the DCJ operation swapping i and j changes:

$$\begin{aligned}(i,k)(j,l) &\longrightarrow (j,k)(i,l),\\ (i,k)(j) &\longrightarrow (j,k)(i),\\ (i,j) &\longrightarrow (i)(j), \quad \text{and}\\ (i)(j) &\longrightarrow (i,j).\end{aligned}$$

With this in mind, for $i, j \in \mathbf{2n}$, and $i \neq j$ we define the double cut and join operator D_{ij} acting on the set of genomes Γ_n, $D_{ij} : \Gamma_n \to \Gamma_n$ as follows:

Definition 4.1 (*Set of fixed points*) Let π be a permutation of $\mathbf{2n}$, then the set of *fixed points* of π is defined by

$$F_\pi := \{i \mid i \in \mathbf{2n}, \pi(i) = i\}.$$

Definition 4.2 (*Algebraic double cut and join operator*) For a permutation π representing a genome, set

$$D_{ij}(\pi) := \begin{cases}(i,j)\pi & \text{if } i, j \in F_\pi \text{ or } \pi(i) = j, \text{ and}\\ (i,j)\pi(i,j) & \text{otherwise.}\end{cases}$$

Therefore, in algebraic terms, the double cut and join operators as defined above are conjugations or left actions by 2-cycle involutions. To distinguish this formulation from the standard, we will call D_{ij} the *algebraic* double cut and join operator.

The vertex set of the genome graph consists of unordered 2-tuples and singletons from the set of gene extremities E. The map ϕ simply relabels elements of the set E with the labels from the set $\mathbf{2n}$. Therefore we can consider the vertex set of the genome graph to consist of unordered 2-tuples and singletons from $\mathbf{2n}$. A genomic permutation π is a permutation on the set $\mathbf{2n}$ satisfying the constraint $\pi(i) = j \iff \pi(j) = i$.

Let ρ be the map that writes the vertex set of the genome graph G as permutation π.

$$\rho(\{i,j\}) = (i,j) \quad \text{and} \quad \rho(\{i\}) = (i).$$

Let $\mathcal{D}_{\{i,j\}}(G)$ be the DCJ operator acting on the extremities i and j of genome G and let $D_{ij}(\pi)$ be the operator acting on the permutation π. The remarks motivating

Fig. 3 Rewriting genome G as the permutation π, and acting on π by D_{ij} gives the same result as acting on G by the DCJ operator and then rewriting the result as permutation π'

$$\begin{tikzcd}
G \arrow[r, "\rho"] \arrow[d, "\mathcal{D}_{\{i,j\}}"'] & \pi \arrow[d, "D_{ij}"] \\
G' \arrow[r, "\rho"'] & \pi'
\end{tikzcd}$$

the definition of algebraic DCJ operator informally explain why we can expect the graph-theoretic and the algebraic operators to be equivalent. That is, the diagram in Fig. 3 commutes.

We now prove this statement formally.

Lemma 4.3 *For all genomes G,*

$$\rho\left(\mathcal{D}_{\{i,j\}}(G)\right) = D_{ij}\left(\rho(G)\right).$$

Proof We prove this by considering all the four cases in the definition of the operator $\mathcal{D}$ (Sect. 2.2).

Case 1. i and j are in separate 2-tuples $\{i,k\},\{j,l\} \in G$.

$$\rho\left(\mathcal{D}_{\{i,j\}}(G)\right) = \rho\left(\mathcal{D}(\{i,k\},\{j,l\})\right) = \rho\left(\{j,k\},\{i,l\}\right) = (j,k)(i,l).$$
$$D_{ij}\left(\rho(\{i,k\},\{j,l\})\right) = D_{ij}\left((i,k)(j,l)\right) = (j,k)(i,l).$$

Case 2. Exactly one of the i, j is in a 2-tuple $\{i,k\},\{j\} \in G$.

$$\rho\left(\mathcal{D}_{\{i,j\}}(G)\right) = \rho\left(\mathcal{D}(\{i,k\},\{j\})\right) = \rho(\{(j,k)\},\{i\}) = (j,k)(i).$$
$$D_{ij}\left(\rho(\{i,k\},\{j\})\right) = D_{ij}\left((i,k)(j)\right) = (j,k)(i).$$

Case 3. None of the i, j is in a 2-tuple. $\{i\},\{j\} \in G$.

$$\rho\left(\mathcal{D}_{\{i,j\}}(G)\right) = \rho\left(\mathcal{D}_{\{i,j\}}(\{i\},\{j\})\right) = \rho(\{i,j\}) = (i,j).$$
$$D_{ij}\left(\rho(\{i\},\{j\})\right) = D_{ij}\left((i)(j)\right) = (i,j).$$

Case 4. i, j are in the same 2-tuple in G $\{i,j\} \in G$

$$\rho\left(\mathcal{D}_{\{i,j\}}(G)\right) = \rho\left(\mathcal{D}_{\{i,j\}}(\{i,j\})\right) = \rho(\{i\},\{j\}) = (i)(j).$$
$$D_{ij}\left(\rho(\{i,j\})\right) = D_{ij}\left((i,j)\right) = (i)(j).$$

Thus in all cases, $\rho\left(\mathcal{D}_{\{i,j\}}(G)\right) = D_{ij}\left(\rho(G)\right)$. □

The following lemma shows that the algebraic DCJ operator is an involution.

Lemma 4.4 $D_{ij}^2(\pi) = \pi$ *for all* $\pi \in \Gamma_n$ *and* $i, j \in \mathbf{2n}$.

Proof For any permutation $\pi \in \Gamma_n$, if i, j are not both telomeres and do not form an adjacency of π, then the same holds in $D_{ij}(\pi) = (i,j)\pi(i,j)$. Similarly if i and j

are both telomeres in π then they form an adjacency in $D_{ij}(\pi)$, and if they are in an adjacency in π, they will both be telomeres in $D_{ij}(\pi)$. Thus acting by D_{ij} on $D_{ij}(\pi)$ will cause the same condition in the definition of D_{ij} to be invoked which was invoked when D_{ij} acted on π.

The operation in each case is an involution, hence $D^2_{ij}(\pi) = \pi$ for all $\pi \in \Gamma_n$. □

At this point, we make the following note about the notation employed in the remainder of this paper. Permutations are functions where the operand is written on the right. So for example, $\pi(i)$ is the permutation π acting on i. In line with this, permutation multiplication is done from right to left.

A 2-cycle is written as (i, j). We will not in general write cycles of length 1, except for emphasis.

5 Products of involutions

In this section we prove some results about products of involutions. We make use of these results in Sect. 6 to determine the DCJ distance between genomic permutations π_1 and π_2.

Lemma 5.1 *Let α and β be involutions acting on the set $\mathbf{2n} = \{1, 2, \ldots, 2n\}$. If $F_\alpha = F_\beta = \varnothing$, then*

1. *For any $i \in \mathbf{2n}$, i and $\alpha(i)$ are in different cycles of $\beta\alpha$. Similarly i and $\beta(i)$ are in different cycles of $\beta\alpha$.*
2. *$\beta\alpha$ has an even number of cycles of length k for any $k \in \mathbb{N}$.*

Proof (1) The cycle in $\beta\alpha$ containing 1 is of the form

$$\Big(1, \beta\alpha(1), \beta\alpha\beta\alpha(1), \ldots, (\beta\alpha)^k(1)\Big),$$

where k is the smallest positive integer such that $(\beta\alpha)^{k+1}(1) = 1$, therefore the length of this cycle is $k + 1$. We claim that $\alpha(1) \notin \{1, \beta\alpha(1), \ldots, (\beta\alpha)^k(1)\}$.

Suppose that $\alpha(1) = (\beta\alpha)^r(1)$ for some r. If r is even then

$$\begin{aligned}\alpha(1) &= (\beta\alpha)^r(1)\\ &= (\beta\alpha)^{(r/2)-1}\beta\alpha(\beta\alpha)^{r/2}(1).\end{aligned}$$

By multiplying on the left both sides by $(\alpha\beta)^{(r/2)-1}$, (the inverse of $(\beta\alpha)^{(r/2)-1}$), we get

$$(\alpha\beta)^{(r/2)-1}\alpha(1) = \beta\alpha(\beta\alpha)^{r/2}(1),$$

and multiplying by β yields

$$\begin{aligned}\beta(\alpha\beta)^{(r/2)-1}\alpha(1) &= \alpha(\beta\alpha)^{r/2}(1)\\ (\beta\alpha)^{r/2}(1) &= \alpha\Big((\beta\alpha)^{r/2}(1)\Big).\end{aligned}$$

In other words, $(\beta\alpha)^{r/2}(1) \in F_\alpha$, contradicting the assumption that $F_\alpha = \varnothing$. Similarly, if r is odd, we find that $\alpha(1) = (\beta\alpha)^r(1)$ implies that $(\beta\alpha)^{(r+1)/2}(1)$ is a fixed point of β, another contradiction.

Thus $\alpha(1) \notin \{1, \beta\alpha(1), \ldots, (\beta\alpha)^k(1)\}$.

(2) Write the cycle in $\beta\alpha$ containing $\alpha(1)$ as $(\alpha(1), \beta(1), \beta\alpha\beta(1), \ldots, (\beta\alpha)^s\beta(1))$, where s is the smallest positive integer such that $(\beta\alpha)^{s+1}\beta(1) = \alpha(1)$. Then multiplying both sides of the equation by $\beta(\alpha\beta)^{s+1}$ (the inverse of $(\beta\alpha)^{s+1}\beta$) we obtain

$$\begin{aligned} 1 &= \beta(\alpha\beta)^{s+1}\alpha(1) \\ &= (\beta\alpha)^{s+2}(1). \end{aligned}$$

That is, $(\beta\alpha)^{s+2}(1) = 1$. But $(\beta\alpha)^{k+1} = 1$ and minimality of k and s imply $s = k - 1$. Thus, the length of the cycle containing $\alpha(1)$ is $k + 1$, which is the same as the length of the cycle containing 1. Since the same argument holds for any $i \in \mathbf{2n}$ there will be an even number of cycles of any given length in $\beta\alpha$. □

Lemma 5.2 *Let α and β be permutations on the set* $\mathbf{2n}$ *such that α and β are involutions. Then a cycle in $\beta\alpha$ contains at most 2 points from $F_\alpha \cup F_\beta$.*

Proof If $F_\alpha \cup F_\beta = \varnothing$, then any cycle on $\beta\alpha$ contains 0 elements of $F_\alpha \cup F_\beta$ and the statement is vacuously true.

Suppose then that $F_\alpha \cup F_\beta \neq \varnothing$. Suppose $1 \in F_\alpha$; that is, $\alpha(1) = 1$. If $1 \in F_\beta$, a similar argument would apply.

The cycle containing 1 in $\beta\alpha$ is of the form

$$\Big(1, \beta\alpha(1), \beta\alpha\beta\alpha(1), \ldots, (\beta\alpha)^k(1)\Big)$$

where k is the smallest positive integer for which $(\beta\alpha)^{k+1}(1) = 1$. As in the proof of Lemma 5.1, this cycle contains $\alpha(1) = 1 = (\beta\alpha)^{k+1}(1)$. We have argued in the proof of Lemma 5.1 that if $k + 1$ is odd then

$$(\beta\alpha)^{k+1}(1) = 1 = \alpha(1) \implies (\beta\alpha)^{(k+2)/2}(1) \in F_\beta$$

and if $k + 1$ is even then

$$(\beta\alpha)^{k+1}(1) = 1 = \alpha(1) \implies (\beta\alpha)^{(k+1)/2}(1) \in F_\alpha.$$

That is, if $1 \in F_\alpha$ then the cycle containing 1 contains at least one other point from $F_\alpha \cup F_\beta$, namely $(\beta\alpha)^{(k+2)/2}(1)$ if the length of the cycle is odd, and $(\beta\alpha)^{(k+1)/2}(1)$ if it is even.

Suppose that this cycle contains another point $i \in F_\alpha \cup F_\beta$. We claim that i must be one of the points identified above. Since i is in the cycle, for some positive integer s, $i = (\beta\alpha)^s(1)$. Let s be the smallest such integer.

If $i \in F_\alpha$; that is, $\alpha((\beta\alpha)^s(1)) = (\beta\alpha)^s(1)$. We then have

$$\begin{aligned} 1 &= (\alpha\beta)^s\alpha(\beta\alpha)^s(1) \quad \text{since } ((\beta\alpha)^s)^{-1} = (\alpha\beta)^s \\ &= \alpha(\beta\alpha)^{2s}(1). \end{aligned}$$

But $\alpha(1) = 1$, so acting on both sides by α we have that $(\beta\alpha)^{2s}(1) = 1$. If $i \in F_\beta$, so that $\beta((\beta\alpha)^s(1)) = (\beta\alpha)^s(1)$, we obtain $(\beta\alpha)^{2s-1}(1) = 1$.

But since $k+1$ is the minimal integer for which $(\beta\alpha)^{k+1}(1) = 1$, and s is also minimal, it follows $2s = k+1$ or $2s-1 = k+1$ according to whether $i \in F_\alpha$ or $i \in F_\beta$. That is,

$$s = \begin{cases} (k+1)/2 & \text{if } i \in F_\alpha, \\ (k+2)/2 & \text{if } i \in F_\beta. \end{cases}$$

Hence i is one of the points identified.

The two points are the same if

$$k+1 = \frac{k+1}{2} \quad \text{or} \quad k+1 = \frac{k+2}{2}.$$

The first equation does not have any nonnegative solution. The only nonnegative integer satisfying the second condition is $k = 0$. If k is 0 i.e., $(\beta\alpha)(1) = 1$ then since $1 \in F_\alpha$, it follows that $\beta(1) = 1$ and hence 1 is a fixed point of β as well. In this case, the cycle of $\alpha\beta$ containing 1 will be of length 1 and hence contains a single point from $F_\alpha \cup F_\beta$. Thus, a cycle of $\beta\alpha$ contains no fixed points if there are no fixed points in β and α.

A cycle of length 1 contains a point i from $F_\alpha \cup F_\beta$ if i is fixed in both α and β, that is if $i \in F_\alpha \cap F_\beta$.

If $F_\alpha \cap F_\beta = \emptyset$, then a cycle of $\beta\alpha$ contains exactly two points from $F_\alpha \cup F_\beta$. □

Petersen and Tenner (2013) also investigate the nature of the product of involutions and prove similar results. Their results are stated in terms of the structure of an involution product graph.

6 Determining the DCJ distance

6.1 Subpermutations and the link to transposition distance

We define a binary relation on **2n** which will allow us to separate out the different components of a pair of genomic permutations, each of which we will then be able to sort independently of the others.

Definition 6.1 Let π_1 and π_2 be genomic permutations on n regions. That is, π_1 and π_2 are involutions on the set **2n**. Define the binary relation $\sim$ on **2n** by setting

$$i \sim j \iff (\pi_2\pi_1)^k(i) = j \text{ or } \pi_1(\pi_2\pi_1)^k(i) = j \text{ for some } k \in \mathbb{Z}.$$

The cycles of $\pi_1\pi_2$ and $\pi_2\pi_1$ are the same as sets (they are inverses of each other in S_{2n}), hence the binary relation $\sim$ defined for the pair π_1, π_2 would be the same as defined for the pair π_2, π_1. Therefore, without any ambiguity $\sim$ can be defined for an unordered pair of genomic permutations.

Lemma 6.2 *$\sim$ is an equivalence relation on* **2n**.

Proof It is easy to verify that $\sim$ is reflexive and symmetric. To establish that $\sim$ is transitive, note that since i could be related to j through either of the two relations $\pi_1(\pi_2\pi_1)^k(i) = j$ or $(\pi_2\pi_1)^k(i) = j$, there are four possible cases to be checked. For example, suppose that $\pi_1(\pi_2\pi_1)^p(i) = j$ and $(\pi_2\pi_1)^q(j) = k$. Noting that $(\pi_2\pi_1)^q\pi_1 = \pi_1(\pi_2\pi_1)^{-q}$, since π_i are involutions, we have

$$\begin{aligned} k &= (\pi_2\pi_1)^q\pi_1(\pi_2\pi_1)^p(i) \\ &= \pi_1(\pi_2\pi_1)^{p-q}(i), \end{aligned}$$

and hence $i \sim k$. The other cases can be checked similarly. □

For any $i, j \in \mathbf{2n}$, if i and j are in the same cycle of $\pi_2\pi_1$ then $j = (\pi_2\pi_1)^s(i)$ for some s and hence $i \sim j$. Also, $i \sim \pi_1(i)$ and $\pi_1(i)$ is related to all the elements in the cycle of $\pi_2\pi_1$ that contains $\pi_1(i)$. Therefore an equivalence class under $\sim$ will be the union of the cycles of $\pi_1\pi_2$ containing i and $\pi_1(i)$.

Observe that $i \sim \pi_1(i)$ and $i \sim \pi_2(i)$. Hence the partition of **2n** under $\sim$ will also partition the 2-cycles of π_1 and π_2. In what follows we would like to talk about the sub-permutations of π_1 and π_2 thus induced. Formally,

Definition 6.3 Let π_1 and π_2 be genomic permutations, and let $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r \subseteq \mathbf{2n}$ be the equivalence classes under $\sim$ defined by π_1, π_2. For $1 \leq s \leq r$ and $i = 1, 2$, the *sub-permutation* $\pi_i^{(s)}$ of π_i induced by $\sim$ is defined to be the restriction of π_i to $\mathcal{C}_s$, that is,

$$\pi_i^{(s)} := \pi_i\big|_{\mathcal{C}_s}.$$

Intuitively, we have collected in a sub-permutation all the 2-cycles that are relevant for sorting $\pi_1^{(s)}$ into $\pi_2^{(s)}$.

An example will help illustrate these definitions. Let π_1 and π_2 be the following genomic permutations on 4 regions:

$$\pi_1 = (1, 6)(2, 3)(4, 5)(7, 8), \quad \pi_2 = (1, 2)(3, 4)(5, 6).$$

The partitions of the set $\{1, 2, \ldots, 8\}$ under $\sim$ are $\mathcal{C}_1 = \{1, 2, 3, 4, 5, 6\}$, $\mathcal{C}_2 = \{7, 8\}$.

The sub-permutations $\pi_1^{(1)}$ and $\pi_1^{(2)}$ are then $\pi_1^{(1)} = (1, 6)(2, 3)(4, 5)$, $\pi_1^{(2)} = (7, 8)$. Similarly, the sub-permutations $\pi_2^{(1)}$ and $\pi_2^{(2)}$ are $\pi_2^{(1)} = (1, 2)(3, 4)(5, 6)$, $\pi_2^{(2)} = (7)(8)$ where the cycles of length 1 are written for clarity.

As remarked earlier, an equivalence class $\mathcal{C}_s$ under $\sim$ contains precisely those elements of **2n** that are contained in the cycle of $\pi_1\pi_2$ containing i and $\pi_1(i)$. Therefore as proved in Lemmas 5.1 and 5.2, the product of the sub-permutations $\pi_1^{(s)}$ and $\pi_2^{(s)}$ is either a single cycle containing one or two points from $F_{\pi_1} \cup F_{\pi_2}$ or a product of (exactly) two disjoint cycles.

Suppose the sub-permutations $\pi_1^{(s)}$ and $\pi_2^{(s)}$ are distinct. If either $\pi_1^{(s)}$ and $\pi_2^{(s)}$ are conjugate in S_{2n}, then they have the same cycle type. Hence either $\mathcal{C}_s$ contains no points from $F_{\pi_1} \cup F_{\pi_2}$, or it contains one point each from F_{π_1} and F_{π_2}. In the first case, it follows from Lemma 5.1 that $\mathcal{C}_s$ contains an even number of points. In the latter case, the cardinality of $\mathcal{C}_s$ is odd.

These observations are summarised in Corollary 6.4.

Corollary 6.4 *Let $\pi_1^{(s)}$ and $\pi_2^{(s)}$ be distinct sub-permutations induced by an equivalence class $\mathcal{C}_s$ such that $\pi_1^{(s)}$ and $\pi_2^{(s)}$ are conjugate in S_{2n}. Let $i \in \mathcal{C}_s$. The product $\pi_2^{(s)}\pi_1^{(s)}$ is given by*

$$\pi_2^{(s)}\pi_1^{(s)} = \begin{cases} (i, \pi_2\pi_1(i), \ldots, (\pi_2\pi_1)^u(i))\left(\pi_1(i), \pi_2(i), \ldots, (\pi_2\pi_1)^{u-1}\pi_2(i)\right) & \textit{if } F_{\pi_1} \cup F_{\pi_2} = \emptyset, \\ \left(i, \pi_2\pi_1(i), \ldots (\pi_2\pi_1)^{2u}(i)\right) & \textit{if } F_{\pi_1} \cup F_{\pi_2} \neq \emptyset, i \in F_{\pi_1}. \end{cases}$$

The sum of lengths of the cycles in the product is the cardinality of $\mathcal{C}_s$.

Continuing our example above, we see $\pi_1^{(1)}\pi_2^{(1)} = (1, 3, 5)(2, 6, 4)$, $\pi_1^{(2)}\pi_2^{(2)} = (7, 8)$.

Observe that $\pi_1^{(1)}\pi_2^{(1)}$ is a product of two disjoint cycles. $\pi_1^{(2)}\pi_2^{(2)}$ contains two points from F_{π_2} namely 7 and 8.

If a partition $\mathcal{C}_t$ consists of a single point say i then i is a fixed point of both π_1 and π_2, hence the DCJ distance between sub-permutations induced by $\mathcal{C}_t$ is 0.

We will determine the DCJ distance between π_1 and π_2 by determining the DCJ distance between $\pi_1^{(s)}$ and $\pi_2^{(s)}$ for $s \in \{1, 2, \ldots, r\}$.

Definition 6.5 For any permutation π, the transposition length of π denoted by $\ell_t(\pi)$ is the minimal number of transpositions needed to express π.

Since the D_{ij} operation involves multiplying a permutation with transpositions, we are interested in how multiplication by a transposition affects the tranposition length of a permutation. In fact this effect is easily stated: multiplication by a transposition changes the transposition length of a permutation by ±1.

That is,

$$\begin{aligned} \ell_t\left((i, j)\,\pi\right) &= \ell_t(\pi) \pm 1, \\ \ell_t\left(\pi(i, j)\right) &= \ell_t(\pi) \pm 1. \end{aligned} \tag{1}$$

This can be observed by noting first that a permutation can be expressed as a product of either an odd or an even number of transpositions, but not both. That is, the parity of the number of transpositions needed to write a permutation as a product is unique.

Let the transposition length of a permutation π be r. That is,

$$\pi = t_1 t_2 \ldots t_r,$$

where the t_i are transpositions.

Suppose r is odd. Then the parity of $(i, j)\pi$ is even since $(i, j)t_1t_2 \ldots t_r$ is one expression of the result as a product of transpositions, although it may not be minimal. The transposition length of $(i, j)\pi$ is also therefore even, and hence it is either $r + 1$ or $r - 1$. A similar argument follows if r is even.

In the remaining part of this section, we will show that the DCJ distance between π_1 and π_2 can be determined in terms of the transposition length of the permutation product $\pi_2\pi_1$. First of all, note that if $\pi_1 = \pi_2$ then $\pi_2\pi_1 = ()$ where $()$ is the identity permutation, hence $\ell_t(\pi_2\pi_1) = 0$.

We make the following claim regarding a lower bound on the DCJ distance between permutations π_1 and π_2.

Lemma 6.6 *Let π_1 and π_2 be genomic permutations. Then*

$$d_{DCJ}(\pi_1, \pi_2) \geq \frac{\ell_t(\pi_1\pi_2)}{2}.$$

Proof A single DCJ operation D_{ij} acts either by conjugation of π_1 by the transposition (i, j), or by multiplication of π_1 by (i, j).

If $D_{ij}(\pi_1) = (i, j)\pi_1$, then $D_{ij}(\pi_1)\pi_2 = (i, j)\pi_1\pi_2$. Hence by Eq. (1)

$$\ell_t\left(D_{ij}(\pi_1)\pi_2\right) = \ell_t(\pi_1\pi_2) \pm 1.$$

If $D_{ij}(\pi_1) = (i, j)\pi_1(i, j)$ then

$$D_{ij}(\pi_1)\pi_2 = (i, j)(\pi_1(i), \pi_1(j))\pi_1\pi_2.$$

By applying Eq. 1 twice,

$$\ell_t\left(D_{ij}(\pi_1)\pi_2\right) = \ell_t(\pi_1\pi_2) \text{ or } \ell_t(\pi_1\pi_2) \pm 2.$$

Thus a single DCJ operation on π_1 can reduce the transposition length of $\pi_1\pi_2$ by at most 2. Since $\ell_t(\pi_2\pi_1) = 0$ when $\pi_1 = \pi_2$, the DCJ distance between π_1 and π_2 must be at least $\ell_t(\pi_1\pi_2)/2$. □

6.2 DCJ distance between conjugate sub-permutations

Let the sub-permutations $\pi_1^{(s)}$ and $\pi_2^{(s)}$ be conjugate in S_{2n}. That is, there exists a $g \in S_{2n}$ such that

$$g\pi_1^{(s)}g^{-1} = \pi_2^{(s)}.$$

By writing g as a product of transpositions, we obtain a sequence of DCJ operations that transforms $\pi_1^{(s)}$ to $\pi_2^{(s)}$, each of which is conjugation by a transposition. Let $d_{DCJ}^c\left(\pi_1^{(s)}, \pi_2^{(s)}\right)$ be the minimal number of DCJ conjugation operations needed to transform $\pi_1^{(s)}$ into $\pi_2^{(s)}$. Then clearly

$$d_{DCJ}\left(\pi_1^{(s)}, \pi_2^{(s)}\right) \leq d_{DCJ}^c\left(\pi_1^{(s)}, \pi_2^{(s)}\right).$$

Theorem 6.7 *Let $\pi_1^{(s)}$ and $\pi_2^{(s)}$ be sub-permutations of genomic permutations π_1 and π_2 on n regions such that $\pi_1^{(s)}$ and $\pi_2^{(s)}$ are conjugate in S_{2n}. Then the conjugation distance $d_{DCJ}^c\left(\pi_1^{(s)}, \pi_2^{(s)}\right)$ is half the transposition length of $\pi_2^{(s)}\pi_1^{(s)}$, that is,*

$$d_{DCJ}^c\left(\pi_1^{(s)}, \pi_2^{(s)}\right) = \frac{1}{2}\ell_t\left(\pi_2^{(s)}\pi_1^{(s)}\right).$$

Proof We prove the claim by induction on $r := d_{DCJ}^c\left(\pi_1^{(s)}, \pi_2^{(s)}\right)$.

Suppose first that $d_{DCJ}^c\left(\pi_1^{(s)}, \pi_2^{(s)}\right) = 1$. Because $\pi_1^{(s)}$ and $\pi_2^{(s)}$ are conjugate, $\pi_2^{(s)}\pi_1^{(s)}$ is an even permutation, so $\ell_t\left(\pi_2^{(s)}\pi_1^{(s)}\right)$ is at least 2. Since $r = 1$, there exists a transposition $(i, j) \in S_{2n}$ such that $(i, j)\pi_1^{(s)}(i, j) = \pi_2^{(s)}$. Hence,

$$\pi_2^{(s)}\pi_1^{(s)} = (i, j)\pi_1^{(s)}(i, j)\pi_1^{(s)} = (i, j)(\pi_1^{(s)}(i), \pi_1^{(s)}(j)).$$

Therefore the transposition length of $\pi_2^{(s)}\pi_1^{(s)}$ is 2.

Assume that the hypothesis is true for all $r \in \mathbb{N}$, with $r < u$. That is, for $r < u$,

$$r = d_{DCJ}^c\left(\pi_1^{(s)}, \pi_2^{(s)}\right) = \frac{1}{2}\ell_t\left(\pi_2^{(s)}\pi_1^{(s)}\right).$$

Next suppose that $d_{DCJ}^c\left(\pi_1^{(s)}, \pi_2^{(s)}\right) = u$. That is,

$$w_u w_{u-1} \dots w_1(\pi_1^{(s)})w_1 \dots w_{u-1}w_u = \pi_2^{(s)},$$

for some transpositions $w_i \in S_{2n}$ with u minimal. Let $\pi_1' = w_{u-1} \dots w_1(\pi_1^{(s)})w_1 \dots w_{u-1}$.

Since the conjugation distance between π_1' and π_1 is $u - 1 < u$, from the induction hypothesis we know that $\ell_t\left(\pi_1'\pi_1^{(s)}\right) = 2(u-1)$. Write

$$\pi_1'\pi_1^{(s)} = t_1 t_2 \dots t_{2(u-1)}$$

for transpositions $t_i \in S_{2n}$. Since $w_u\pi_1'w_u = \pi_2^{(s)}$, we have $d_{DCJ}^c\left(\pi_1', \pi_2^{(s)}\right) = 1$, and hence $\ell_t\left(\pi_2^{(s)}\pi_1'\right) = 2$. That is, $\pi_2^{(s)}\pi_1' = v_1 v_2'$ where v_1 and v_2 are transpositions in S_{2n}.

Then

$$\pi_2^{(s)}\pi_1^{(s)} = \pi_2^{(s)}\pi_1'\pi_1'\pi_1^{(s)} = v_1 v_2 t_1 t_2 \dots t_{2(u-1)}.$$

This is a product of a permutation of transposition length $2(u-1)$ with two transpositions. The transposition length of the result will be $\ell_t\left(\pi_2^{(s)}\pi_1^{(s)}\right) \in \{2u-4, 2u-2, 2u\}$. However, if $\ell_t\left(\pi_2^{(s)}\pi_1^{(s)}\right) < 2u$ then $d_{DCJ}^c(\pi_1^{(s)}, \pi_2^{(s)}) < u$, contrary to our assumption. Hence $\ell_t(\pi_2^{(s)}\pi_1^{(s)}) = 2u$. That is,

$$d_{DCJ}^c\left(\pi_1^{(s)}, \pi_2^{(s)}\right) = u \implies \ell_t\left(\pi_2^{(s)}\pi_1^{(s)}\right) = 2u.$$

□

Putting together the lower bound for DCJ distance from Lemma 6.6 with the upper bound from Theorem 6.7, we have the following.

Corollary 6.8 *If $\pi_1^{(s)}$ and $\pi_2^{(s)}$ are conjugate sub-permutations of the genomic permutations π_1 and π_2 on n regions, then*

$$d_{DCJ}\left(\pi_1^{(s)}, \pi_2^{(s)}\right) = \frac{\ell_t\left(\pi_2^{(s)}\pi_1^{(s)}\right)}{2}.$$

6.3 Constructing a sorting element for conjugate sub-permutations

When the sub-permutations $\pi_1^{(s)}$ and $\pi_2^{(s)}$ induced by $\mathcal{C}_s$ are conjugate, an element of minimal length sorting $\pi_1^{(s)}$ into $\pi_2^{(s)}$ can easily be constructed as follows. Corollary 6.4 gives the structure of the product $\pi_2^{(s)}\pi_1^{(s)}$.

If $\pi_1^{(s)}$ and $\pi_2^{(s)}$ have no fixed points then

$$\begin{aligned}\pi_2^{(s)}\pi_1^{(s)} = &\left(1, \pi_2\pi_1(1), (\pi_2\pi_1)^2(1), \ldots, (\pi_2\pi_1)^u(1)\right)\\ &\times\left(\pi_1(1), \pi_2(1), \pi_2\pi_1\pi_2(1), \ldots, (\pi_2\pi_1)^{u-1}\pi_2(1)\right).\end{aligned} \tag{2}$$

If $\pi_1^{(s)}$ and $\pi_2^{(s)}$ contain fixed points then $\pi_2^{(s)}\pi_1^{(s)}$ is a single cycle

$$\pi_2^{(s)}\pi_1^{(s)} = \left(1, \pi_2\pi_1(1), (\pi_2\pi_1)^2(1), \ldots, (\pi_2\pi_1)^{2u}(1)\right). \tag{3}$$

Let $g = \left(1, \pi_2\pi_1(1), (\pi_2\pi_1)^2(1) \ldots, (\pi_2\pi_1)^u(1)\right)$. We claim that

$$g\pi_1^{(s)}g^{-1} = \pi_2^{(s)}.$$

If i is moved by g, then $g(i) = \pi_2\pi_1(i)$. For any 2-cycle $(i, \pi_1(i))$ in $\pi_1^{(s)}$, either i or $\pi_1(i)$ is moved by g. If $\pi_2^{(s)}\pi_1^{(s)}$ is a product of two cycles then as proved in Lemma 5.2, i and $\pi_1(i)$ are in different cycles of $\pi_2^{(s)}\pi_1^{(s)}$. Since g is precisely one of the two cycles of $\pi_2^{(s)}\pi_1^{(s)}$, it moves exactly one of i and $\pi_1(i)$.

On the other hand, if $\pi_2^{(s)}\pi_1^{(s)}$ is a single cycle as in Eq. (3), then suppose $\pi_1(1) = 1$ (if instead $\pi_2(1) = 1$, a similar argument would apply). Then,

$$\begin{aligned} 1 &= \left(\pi_2^{(s)}\pi_1^{(s)}\right)^{2u+1}(1) \\ &= \pi_2^{(s)}\left(\pi_1^{(s)}\pi_2^{(s)}\right)^{2u}(1). \end{aligned}$$

This implies that $\left(\pi_1^{(s)}\pi_2^{(s)}\right)^{2u}(1) = \pi_2^{(s)}(1)$. Now suppose $i = \left(\pi_2^{(s)}\pi_1^{(s)}\right)^{b}(1)$ for some b then $\pi_1^{(s)}(i) = \left(\pi_2^{(s)}\pi_1^{(s)}\right)^{a}(i)$ for some a, since i and $\pi_1^{(s)}(i)$ are in the same cycle. Also,

$$\begin{aligned} \pi_1^{(s)}(i) &= \pi_1^{(s)}\left(\pi_2^{(s)}\pi_1^{(s)}\right)^{b}(1) \\ &= \left(\pi_1^{(s)}\pi_2^{(s)}\right)^{b}(1) \\ &= \left(\pi_1^{(s)}\pi_2^{(s)}\right)^{-2u+b}\left(\pi_1^{(s)}\pi_2^{(s)}\right)^{2u}(1) \\ &= \left(\pi_1^{(s)}\pi_2^{(s)}\right)^{-2u+b}\pi_2^{(s)}(1) = \left(\pi_1^{(s)}\pi_2^{(s)}\right)^{-2u+b-1}(1) \\ &= \left(\pi_2^{(s)}\pi_1^{(s)}\right)^{2u-b+1}(1). \end{aligned}$$

If i is moved by g, that is if i is in the cycle $(1, \pi_2\pi_1(1), \ldots, (\pi_2\pi_1)^u(1))$, then $b \leq u$ which means that $a = 2u - b + 1 > u$ and $\pi_1^{(s)}(i)$ is not in this cycle and hence not moved by g.

If $\pi_1^{(s)}(i)$ is moved by g, that is if $\pi_1^{(s)}(i)$ is in the cycle $(1, \pi_2\pi_1(1), \ldots, (\pi_2\pi_1)^u(1))$, then $a = 2u - b + 1 \leq u$ which means that $u + 1 \leq b$ and i is not in this cycle and hence not moved by g.

Thus we have established that whether the product $\pi_2^{(s)}\pi_1^{(s)}$ is given by Eq. (2) or Eq. (3), for any $i \in \mathcal{C}_s$, g moves either i or $\pi_1^{(s)}(i)$.

So if i is moved by g, then $g(i) = \pi_2\pi_1(i)$, and since $\pi_1(i)$ is then not moved by g, $g(\pi_1(i)) = \pi_1(i)$. Consider the product $g\pi_1^{(s)}g^{-1}$. For any 2-cycle $(i, \pi_1(i))$ in $\pi_1^{(s)}$ suppose i is moved by g. We have

$$g\,(i, \pi_1(i))\,g^{-1} = (g(i), g\,(\pi_1(i))) = (\pi_2\,(\pi_1(i))\,, \pi_1(i))$$

which is the 2-cycle in $\pi_2^{(s)}$ containing $\pi_1(i)$. Thus

$$g\pi_1^{(s)}g^{-1} = \pi_2^{(s)}.$$

Since the transposition length of g is u (that is, we require u transpositions to express g as a product), and conjugation by a 2-cycle (i, j) is one D_{ij} operation, we require u DCJ operations to sort $\pi_1^{(s)}$ into $\pi_2^{(s)}$. This is exactly the DCJ distance between them. Hence g gives an optimal sorting element that is, $g\pi_1^{(s)}g^{-1} = \pi_2^{(s)}$.

As there is nothing special about the choice of 1 in this argument, the cycle

$$\left(\pi_1(1), \pi_2(1), \pi_2\pi_1\pi_2(1), \ldots, (\pi_2\pi_1)^{u-1}\pi_2(1)\right)$$

is also an optimal sorting element.

The above construction might be better understood through an example. Let π_1 and π_2 be the following genomic permutations on 6 regions:

$$\pi_1 = (1, 6)(2, 3)(4, 5), \quad \pi_2 = (1, 2)(3, 4)(5, 6).$$

Since π_1 and π_2 have the same cycle structure, we know that they are conjugate in S_{2n}. Consider the product $\pi_1\pi_2$.

$$\pi_1\pi_2 = (1, 6)(2, 3)(4, 5)(1, 2)(3, 4)(5, 6) = (1, 3, 5)(2, 6, 4).$$

Let $g = (1, 3, 5)$. Then $g^{-1} = (1, 5, 3)$.

$$g\pi_2 g^{-1} = (1, 5, 3)(1, 2)(3, 4)(5, 6)(1, 3, 5) = (1, 6)(2, 3)(4, 5) = \pi_1.$$

The above discussion is summarised in Lemma 6.9.

Lemma 6.9 *Let $\pi_1^{(s)}, \pi_2^{(s)}$ and u be as in Corollary 6.4. An element g such that $\ell_t(g) = d_{DCJ}(\pi_1^{(s)}, \pi_2^{(s)})$ that sorts $\pi_1^{(s)}$ into $\pi_2^{(s)}$ can be constructed from the product $\pi_2^{(s)}\pi_1^{(s)}$ as*

$$g = \left(1, \pi_2\pi_1(1), (\pi_2\pi_1)^2(1), \ldots, (\pi_2\pi_1)^u(1)\right) .$$

A similar construction has been given in Feijão and Meidanis (2013), where they construct the sorting element by establishing a correspondence between the connected components of adjacency graph and the permutation product.

6.4 DCJ distance between non-conjugate sub-permutations

We will now consider the case where $\pi_1^{(s)}$ and $\pi_2^{(s)}$ are not conjugate in S_{2n}.

Theorem 6.10 *Let $\pi_1^{(s)}$ and $\pi_2^{(s)}$ be non-conjugate sub-permutations of the genomic permutations π_1 and π_2 on n regions. Then*

$$d_{DCJ}\left(\pi_1^{(s)}, \pi_2^{(s)}\right) = \frac{1}{2}\left(\ell_t(\pi_2^{(s)}\pi_1^{(s)}) + 1\right) .$$

Proof We have remarked earlier that for any i in the equivalence class $\mathcal{C}_s$, $\mathcal{C}_s$ contains only the elements contained in the cycles of $\pi_1\pi_2$ containing i and $\pi_1(i)$. Hence as proved in Lemma 5.2 if the induced sub-permutations $\pi_2^{(s)}\pi_1^{(s)}$ are not identical, their product contains exactly two fixed points. Since $\pi_1^{(s)}$ and $\pi_2^{(s)}$ are not conjugate, both the fixed points belong to the same sub-permutation.

Suppose the fixed points are i_1 and i_2 and that they belong to $\pi_1^{(s)}$. Recall that if i, j are both fixed in π then $D_{ij}(\pi) = (i, j)\pi$, and hence

$$D_{i_1 i_2}\left(\pi_1^{(s)}\right) = (i_1, i_2)\pi_1^{(s)} .$$

While $\pi_1^{(s)}$ and $\pi_2^{(s)}$ are both products of 2-cycles, the number of 2-cycles in $\pi_1^{(s)}$ is one less than the number of 2-cycles of $\pi_2^{(s)}$, since it has two fixed points. Therefore $D_{i_1 i_2}(\pi_1^{(s)})$ is conjugate to $\pi_2^{(s)}$. Similarly, if the two fixed points belong to π_2 then $D_{i_1 i_2}(\pi_2^{(s)})$ is conjugate to $\pi_1^{(s)}$. As the DCJ distance is symmetric we can assume without loss of generality that the fixed points belong to $\pi_1^{(s)}$.

Let $\pi' = D_{i_1 i_2}(\pi_1^{(s)})$. From Theorem 6.7 we have that

$$d_{DCJ}\left(\pi', \pi_2^{(s)}\right) = \frac{1}{2}\ell_t\left(\pi_2^{(s)}\pi'\right) = \frac{1}{2}\ell_t\left(\pi'\pi_2^{(s)}\right) = \frac{1}{2}\ell_t\left((i_1, i_2)\pi_1^{(s)}\pi_2^{(s)}\right).$$

Since i_1 and i_2 are in the same cycle of $\pi_1^{(s)}\pi_2^{(s)}$, multiplication by (i_1, i_2) will split this cycle into two cycles, reducing the transposition length of the product by 1. Hence

$$d_{DCJ}\left(\pi', \pi_2^{(s)}\right) = \frac{1}{2}\ell_t\left((i_1, i_2)\pi_1^{(s)}\pi_2^{(s)}\right) = \frac{1}{2}\left(\ell_t\left(\pi_1^{(s)}\pi_2^{(s)}\right) - 1\right).$$

The DCJ distance between $\pi_1^{(s)}$ and π' is 1. Thus the triangle inequality gives

$$\begin{aligned} d_{DCJ}\left(\pi_1^{(s)}, \pi_2^{(s)}\right) &\leq d_{DCJ}\left(\pi_1^{(s)}, \pi'\right) + d_{DCJ}\left(\pi', \pi_2^{(s)}\right) \\ &= \frac{1}{2}\left(\ell_t\left(\pi_1^{(s)}\pi_2^{(s)}\right) - 1\right) + 1 \\ &= \frac{1}{2}\left(\ell_t\left(\pi_1^{(s)}\pi_2^{(s)}\right) + 1\right). \end{aligned}$$

At the same time we have a lower bound on the distance (Lemma 6.6), so that

$$\frac{1}{2}\ell_t\left(\pi_1^{(s)}\pi_2^{(s)}\right) \leq d_{DCJ}\left(\pi_1^{(s)}, \pi_2^{(s)}\right) \leq \frac{1}{2}\left(\ell_t\left(\pi_1^{(s)}\pi_2^{(s)}\right) + 1\right).$$

Since the DCJ distance is an integer (the number of DCJ operations), and the transposition length of $\pi_1^{(s)}\pi_2^{(s)}$ is odd, we have

$$d_{DCJ}\left(\pi_1^{(s)}, \pi_2^{(s)}\right) = \frac{1}{2}\left(\ell_t\left(\pi_1^{(s)}\pi_2^{(s)}\right) + 1\right)$$

as required. □

From Theorems 6.7 and 6.10, it is clear that the sub-permutations induced by the partitions $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r$ under $\sim$ can be sorted independently. Therefore

$$d_{DCJ}(\pi_1,\pi_2) \leq \sum_s d_{DCJ}\left(\pi_1^{(s)},\pi_2^{(s)}\right).$$

We claim that a sorting sequence that involves a DCJ operation D_{ij}, where i, j are in different partitions $\mathcal{C}_s$, cannot be shorter than a sequence that sorts each partition independently.

Let $\mathcal{C}_r$ and $\mathcal{C}_s$ be distinct equivalence classes of $\mathbf{2n}$ under $\sim$. Let $\ell_t\left(\pi_1^{(r)}\pi_2^{(r)}\right)$ be l_1 and $\ell_t\left(\pi_1^{(s)}\pi_2^{(s)}\right)$ be l_2. Then

$$\frac{1}{2}(l_1) \leq d_{DCJ}\left(\pi_1^{(r)},\pi_2^{(r)}\right) \leq \frac{1}{2}(l_1+1).$$

Similarly

$$\frac{1}{2}(l_2) \leq d_{DCJ}\left(\pi_1^{(s)},\pi_2^{(s)}\right) \leq \frac{1}{2}(l_2+1).$$

Combining these, we have

$$\frac{1}{2}(l_1+l_2) \leq d_{DCJ}(\mathcal{C}_r) + d_{DCJ}(\mathcal{C}_s) \leq \frac{1}{2}(l_1+l_2+2).$$

The action of D_{ij} on π_1 may combine the two partitions $\mathcal{C}_r$ and $\mathcal{C}_s$ into a single partition $\mathcal{C}_t$ or it may change each of the partitions $\mathcal{C}_r$ and $\mathcal{C}_s$. In the latter case, by an abuse of notation we use $\mathcal{C}_t$ to denote the union of the partitions changed by the action of D_{ij}. We wish to determine the transposition length of $\pi_1^{(t)}\pi_2^{(t)}$ in order to find the DCJ distance.

The action of D_{ij} on π_1 may be left multiplication by (i, j) or conjugation by (i, j). If it acts by left multiplicaton, so that $D_{ij}(\pi_1) = (i, j)\pi_1$, then since i and j are in different partitions and hence in different cycles of $\pi_1^{(r)}\pi_2^{(r)}\pi_1^{(s)}\pi_2^{(s)}$, multiplication by (i, j) will combine the two cycles that contain i and j. The transposition length of the product will therefore increase by 1. That is, in this case,

$$\ell_t\left(\pi_1^{(t)}\pi_2^{(t)}\right) = \ell_t\left(\pi_1^{(r)}\pi_2^{(r)}\pi_1^{(s)}\pi_2^{(s)}\right) + 1.$$

On the other hand, if D_{ij} acts by conjugation then $D_{ij}(\pi_1) = (i, j)\pi_1(i, j)$, and we have that

$$(i, j)\pi_1(i, j) = (i, j)\,(\pi_1(i), \pi_1(j))\,\pi_1.$$

The images $\pi_1(i)$ and $\pi_1(j)$ are in different cycles of $\pi_1^{(r)}\pi_2^{(r)}\pi_1^{(s)}\pi_2^{(s)}$ since i and j are in different partitions. In $x = (\pi_1(i), \pi_1(j))\,\pi_1\pi_2$, the cycles containing $\pi_1(i)$ and $\pi_1(j)$ will combine into a single cycle, increasing the length of the product by 1. Then the cycle of x containing $\pi_1(i)$ and $\pi_1(j)$ will contain either both, one, or neither of i and j.

Accordingly, multiplication by (i, j) will either split this cycle into two cycles or combine two different cycles into one cycle. Thus the transposition length may increase or decrease by 1 (from the previous step). Hence

$$\ell_t\left(\pi_1^{(t)}\pi_2^{(t)}\right) = \ell_t\left(\pi_1^{(r)}\pi_2^{(r)}\pi_1^{(s)}\pi_2^{(s)}\right),$$

or

$$\ell_t\left(\pi_1^{(t)}\pi_2^{(t)}\right) = \ell_t\left(\pi_1^{(r)}\pi_2^{(r)}\pi_1^{(s)}\pi_2^{(s)}\right) + 2.$$

In both cases $l_1 + l_2 \leq \ell_t(\mathcal{C}_t)$ and hence

$$\frac{1}{2}(l_1 + l_2) \leq d_{DCJ}\left(\pi_1^{(t)}, \pi_2^{(t)}\right).$$

Since one DCJ operation was needed to change the partition $\mathcal{C}_r$ and $\mathcal{C}_s$ into $\mathcal{C}_t$, and the DCJ distance of the sub-permutations induced by $\mathcal{C}_t$ is at least $\frac{1}{2}(l_1 + l_2)$, any sorting scenario for $\mathcal{C}_r$ and $\mathcal{C}_s$ that steps through $\mathcal{C}_t$ is of length at least $\frac{1}{2}(l_1 + l_2) + 1$. At the same time, the sum of the distances of the partitions $\mathcal{C}_r$ and $\mathcal{C}_s$ is bounded above by:

$$d_{\mathcal{C}_r} + d_{\mathcal{C}_s} \leq \frac{1}{2}(l_1 + l_2) + 1.$$

Therefore we conclude that no sequence of DCJ operations sorting π_1 into π_2 can be shorter than a sequence that sorts the sub-permutations independently.

Theorem 6.11 *Let π_1 and π_2 be genomic permutations on n regions. The DCJ distance between π_1 and π_2 is given by*

$$d_{DCJ}(\pi_1, \pi_2) = \frac{1}{2}\left(\ell_t(\pi_2\pi_1) + n_c\right)$$

where n_c is the number of cycles in the product $\pi_2\pi_1$ which contain two fixed points of F_{π_1} or F_{π_2}.

7 Counting the optimal sorting scenarios

To sort a permutation π_a into π_b means to transform π_a into π_b through a sequence of allowed operations (in this case the DCJ operation). A sorting scenario is defined as follows.

Definition 7.1 A *sorting scenario* of length k that sorts genomic permutation π_a into genomic permutation π_b is a sequence of genomic permutations

$$\{(\pi_a =)\pi_0, \pi_1, \pi_2, \ldots, \pi_{k-1}, \pi_k(= \pi_b)\},$$

such that each element of the sequence is obtained from the previous element through a single DCJ operation.

That is, a sorting scenario is the sequence of permutations we step through as π_a is sorted into π_b through DCJ operations. If the DCJ distance is d, then an optimal sorting scenario (scenario of minimal length) is a sequence of length $d + 1$. Two optimal sorting scenarios are equal if they are equal as sequences i.e., corresponding terms are equal.

The total number of optimal sorting scenarios between a pair of genomes is an interesting and important question. In constructing a phylogenetic history, the minimal distance with respect to some mutational operation is used. However such a minimal path is seldom unique. Hence as Miklós and Darling (2009) and Siepel (2002) point out, it would be more appropriate to account for and average over all possible evolutionary paths to draw meaningful statistical inferences.

Braga and Stoye (2010) extended their earlier work (Braga and Stoye 2009) to give a closed formula for the number of optimal DCJ sorting scenarios for certain instances of the problem. Ouangraoua and Bergeron (2010) also present similar results for the number of optimal sorting scenarios and establish connections between the number of sorting scenarios and other combinatorial objects such as parking functions.

Considering genomes as permutations and the DCJ as an action on a permutation allows us to count the number of optimal sorting scenarios for a subset of genomes in a straightforward manner. The subset of genomes we consider are those that are conjugate in S_{2n}. Our result is equivalent to the results obtained by the previous papers.

Let π be a genomic permutation on n regions and let $i, j \in \mathbf{2n}$. The restriction of π to the cycles containing i and j is $(i, \pi(i))\,(j, \pi(j))$. As D_{ij} acting on π only affects the cycles containing i and j,

$$\begin{aligned} D_{ij}(\pi) &= (i, j)\,(i, \pi(i))\,(j, \pi(j))\,(i, j) \\ &= (i, \pi(j))\,(j, \pi(i)) \\ &= (\pi(i), \pi(j))\,(i, \pi(i))\,(j, \pi(j))\,(\pi(i), \pi(j)) \\ &= D_{\pi(i)\pi(j)}(\pi). \end{aligned}$$

If the restriction of π to the cycles containing i and j is $(i)\,(j, \pi(j))$, then

$$D_{ij}(\pi) = (i, j)\,(i)\,(j, \pi(j))\,(i, j) = (j)\,(i, \pi(j))\,.$$

In this case there is no D_{kl} such that $D_{kl}(\pi) = D_{ij}(\pi)$. As these are the two cases where the algebraic DCJ operator acts via conjugation (see Definition 4.2), we have the following lemma.

Lemma 7.2 *Let π be a genomic permutation on n regions. Let D_{ij} and D_{kl} act on π via conjugation. If*

$$D_{ij}\,(\pi) = D_{kl}\,(\pi)\,,$$

then either $(i, j) = (k, l)$ or (i, j) and (k, l) are disjoint transpositions such that $(k, l) = (\pi(i), \pi(j))$.

Based on the characterization of the DCJ operators that act in the same way on a genomic permutation, we can easily enumerate the sorting scenarios.

Theorem 7.3 *Let π_a and π_b be genomic permutations on n regions such that π_a and π_b are conjugate in S_{2n}. If the DCJ distance $d_{DCJ}(\pi_a, \pi_b) = d$ then the number of optimal DCJ sorting scenarios sorting π_a into π_b is $(d+1)^{d-1}$.*

Proof As we have seen in the proof of Theorem 6.7, if the DCJ distance is d then we can construct an element $g \in S_{2n}$ such that g is a cycle of length $d+1$ (and consequently of transposition length d) and

$$g\pi_a g^{-1} = \pi_b.$$

Lemma 6.9 gives the construction of a cycle g such that $g\pi_a g^{-1} = \pi_b$. Let g be as in the statement of Lemma 6.9 i.e., $g = \left(1, \pi_b\pi_a(1), (\pi_b\pi_a)^2(1), \ldots, (\pi_b\pi_a)^d(1)\right)$. The number of ways to represent a cycle of length n as the product of $n-1$ transpositions (i.e., as a minimal product) is n^{n-2} (Dénes 1959). Hence g can be written as a product of transpositions in $(d+1)^{d-1}$ ways. It remains to show that

1. Each expression of g as a minimal product of transpositions corresponds to a distinct sorting scenario, and
2. There can be no other sorting scenarios. That is, if there is $h \in S_{2n}$ such that $\ell_t(h) = \ell_t(g)$ and $h\pi_a h^{-1}$, then any sorting scenario produced by h is identical to some sorting scenario produced by g.

Let $S(g)$ be the set of all expressions for g as a minimal product of transpositions. For example, if $g = (1, 3, 5)$ then $S(g) = \{(1, 5)(1, 3),\ (1, 3)(3, 5),\ (3, 5)(1, 5)\}$.

1. Claim: each expression of g as a minimal product of transpositions corresponds to a distinct sorting scenario.

From the construction of g preceding Lemma 6.9, we know that for any $i \in \mathbf{2n}$, g moves either i or $\pi_a(i)$ but not both. Suppose g moves i. Then since g fixes $\pi_a(i)$, in a minimal factorization of g as a product of transpositions, no transposition moves $\pi_a(i)$.

To observe this, note that the number of trees on d labeled vertices is $(d+1)^{d-1}$ (given by Cayley's formula). Thus there is a bijection between the $S(g)$ and the set of trees on d vertices. A minimal factorization of g into transpositions can be associated with a tree by considering a transposition (i, j) to correspond to the edge between vertices i and j. If a point $\pi_a(i)$ fixed by g is moved by some transposition in a minimal factorization of g, then the factorization must contain a cycle that would move $\pi_a(i)$ back to itself. But such a cycle would correspond to a loop in the graph corresponding to the factorization, which cannot be as the graph is a tree. Hence in a minimal factorization of g as a product of transpositions, no transposition moves $\pi_a(i)$.

Suppose $u_d u_{d-1} \ldots u_1$ and $w_d w_{d-1} \ldots w_1$ are distinct elements of $S(g)$ that produce the same sorting scenarios. We will derive a contradiction.

Let k be the lowest index such that $u_k \neq w_k$. Let $w_{k-1} \ldots w_1 = u_{k-1} \ldots u_1 = g'$ and let $g'\pi_a g'^{-1} = \pi_{k-1}$.

Let $u_k = (i, j)$. By Lemma 7.2, u_k and w_k are disjoint and $w_k = (\pi_{k-1}(i), \pi_{k-1}(j))$.

Now, u_k is a transposition in the minimal expression for g. Since u_k moves i, i is in the support of g, and $\pi_a(i)$ is not in the support of g. The support of g' is a subset of the support of g, hence $\pi_a(i)$ is not in the support of g'.

If u_k is disjoint from g', then

$$\pi_{k-1}(i) = g'\pi_a g'^{-1}(i) = \pi_a(i).$$

Hence $\pi_{k-1}(i)$ is not in the support of g.

If, on the other hand, u_k is not disjoint from g', then let i be in the support of g'.

Clearly, $g'^{-1}(i)$ is in the support of g' (it gets mapped to i by g') and hence in the support of g. Therefore, $\pi_a\left((g')^{-1}(i)\right)$ is not in the support of g (and hence not in the support of g') since for any $i \in \mathbf{2n}$, g moves either i or $\pi_a(i)$. Hence

$$\pi_{k-1}(i) = g'\pi_a g'^{-1}(i) = \pi_a\left(g'^{-1}(i)\right).$$

That is, $\pi_{k-1}(i)$ is not in the support of g.

Thus, in both cases (whether u_k is disjoint from g' or not), w_k moves an element that is not in the support of g. This contradicts the assertion that $w_d w_{d-1} \ldots w_1 = g$. Thus either $w_d w_{d-1} \ldots w_1 \notin S(g)$ or the sorting scenarios produced by distinct elements are distinct. Each expression of g as a minimal product of transpositions therefore gives a unique sorting scenario and the number of sorting scenarios is at least the cardinality of $S(g)$.

2. Claim: there are no additional sorting scenarios

Let $h \in S_{2n}$ such that $\ell_t(h) = \ell_t(g) = d$ and $h\pi_a h^{-1} = \pi_b$. Let $h = w_d w_{d-1} \ldots w_1$ be a factorization of h into transpositions. We claim that h produces the same sorting scenario as some element in $S(g)$. This will establish that the number of sorting scenarios is equal to the cardinality of $S(g)$. To prove this assertion, we first prove that there is some element $u_d \ldots u_1 \in S(g)$ such that $u_1\pi_a u_1 = w_1\pi_a w_1$.

Suppose that this is not the case. That is, no element in $S(g)$ produces a sorting scenario that has $w_1\pi_a w_1$ as the second term (the first term in all sorting scenarios is π_a). Consider the element

$$h' = u_d u_{d-1} \ldots u_1 w_1.$$

Let $w_1\pi_a w_1 = \pi_a'$. Then

$$\begin{aligned} h'\pi_a' h'^{-1} &= (u_d u_{d-1} \ldots u_1 w_1)(w_1\pi_a w_1)(w_1 u_1 \ldots w_{d-1} w_d) \\ &= (u_d u_{d-1} \ldots u_1)(\pi_a)(u_1 \ldots w_{d-1} w_d) \\ &= \pi_b. \end{aligned}$$

At the same time, the DCJ distance between π_a' and π_b is $d-1$, since $(w_d \ldots w_2)\pi_a'$ $(w_2 \ldots w_d) = \pi_b$. Hence $u_d u_{d-1} \ldots u_1 w_1$ can be written as a product of $d-1$ transpositions say $v_{d-1} \ldots v_1$. Then $v_{d-1} \ldots v_1 w_1 = g$, and is an expression of length d equal to g that is not is $S(g)$ because we have assumed that there is no element in $S(g)$

such that $u_1\pi_a u_1 = w_1\pi_a w_1$. This is a contradiction since $S(g)$ by definition contains all factorizations of g of length d. Thus there is some element in $u_d \dots u_1 \in S(g)$ such that $w_1\pi_a w_1 = u_1\pi_a u_1$.

Let $S_1(g) = \{u_d \dots u_1 \in S(g) \mid u_1\pi_a u_1 = w_1\pi_a w_1\}$.

By a similar argument we can prove that there exists some element in $S_1(g)$ such that $u_2(u_1\pi_a u_1)u_2 = w_2(u_1\pi_a u_1)w_2$. In general, let

$$\begin{aligned} S_k(g) = \{u_d \dots u_1 \in S_{k-1}(g) \mid\, & u_k(u_{k-1}\dots u_1\pi_a u_1 \dots u_{k-1})u_k \\ & = w_k(w_{k-1}\dots w_1\pi_a w_1 \dots w_{k-1})w_k\}. \end{aligned}$$

Suppose that there does not exist any element in $S_k(g)$ such that

$$u_{k+1}(u_k \dots u_1\pi_a u_1 \dots u_k)u_{k+1} = w_{k+1}(u_k \dots u_1\pi_a u_1 \dots u_k).$$

Let $u_d \dots u_1 \in S_k(g)$ and let $u_k \dots u_1\pi_a u_1 \dots u_k = \pi_a'$. Then,

$$(u_d \dots u_{k+1}w_{k+1})\left(w_{k+1}\pi_a' w_{k+1}\right)(w_{k+1}u_{k+1}\dots u_d) = \pi_b.$$

The DCJ distance between $\left(w_{k+1}\pi_a' w_{k+1}\right)$ and π_b is $d-k$. Therefore $u_d \dots u_{k+1}w_{k+1}$ can be re-written as a product of $d-k$ transpositions, say $v_{d-k}\dots v_1$. Now $v_{d-k}\dots v_1 w_{k+1}u_k \dots u_1$ is an expression of length d equal to g and prefix $u_k \dots u_1$ that is not in $S_k(g)$. This contradicts the definition of $S_k(g)$.

By repeating this argument, we can conclude that there exists some element in $S(g)$ that gives the same sorting scenario as h.

Thus the number of sorting scenarios is equal to $|S(g)| = (d+1)^{d-1}$. □

8 Conclusions and future work

The double cut and join operator has been a major step forward for the study of genome rearrangements, because it is very general, allowing numerous operations on multi-chromosomal genomes, and in addition has a very simple length formula with which to calculate genomic distance. In this paper, we have shown how this operator may be described group-theoretically, and derived a correspondingly simple length formula independently of prior results. The length formula given in Theorem 6.11 requires only the ability to write each genome as a permutation and to multiply such permutations. We are also able to provide a simple construction for an optimal sorting scenario in particular instances of the problem. Translating the model into algebra allows us to exploit established results in group theory, a field with over a century of development. The proof of Theorem 7.3 is an example of this, relying as it does on a combinatorial group theory result from the 1950s. The use of group theory to model rearrangements provides a natural context in which to study alternative assumptions about the rearrangement processes. As pointed out in Egri-Nagy et al. (2014) and Francis (2014), different assumptions about the processes gives rise to questions about length functions in different groups, or length functions with respect to different generating sets. A group-theoretic model may also provide an avenue for investigating additional operations that are not captured by the DCJ.

Appendix: Some results about symmetric groups

This paper uses some standard results on symmetric groups that we collect here for ease of reference. More details on these results can be found in many undergraduate group theory textbooks, for example Fraleigh (2003).

A permutation is a bijection from a set S to itself. S is usually taken to be a set of natural numbers $\mathbf{n} = \{1, 2, \ldots, n\}$. A permutation can be written by specifying the value of the map on all the points.

Example

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 1 & 6 & 2 & 5 \end{pmatrix}$$

is a permutation on the set {1, 2, 3, 4, 5, 6} that sends 1 to 3, 2 to 4, etc. The set of all permutations on the set $\mathbf{n}$ forms a *group* called the symmetric group and denoted by S_n.

Permutation multiplication

Since permutations are simply bijective functions, permutation multiplication is function composition. That is, to find the image of i in the product $\pi_2\pi_1$, we do $\pi_2(\pi_1(i))$.

Example Let $\pi_1 = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 3\,4\,1\,5\,2 \end{pmatrix}$ and $\pi_2 = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 2\,1\,4\,3\,5 \end{pmatrix}$. The image of 1 in the product $\pi_2\pi_1$ is $\pi_2(\pi_1(1)) = \pi_2(3) = 4$. So for each i, we have to "follow the string" – π_1 send i to j, π_2 sends j to k, so i gets sent to k by $\pi_2\pi_1$.

$$\begin{pmatrix} 1\,2\,3\,4\,5 \\ 2\,1\,4\,3\,5 \end{pmatrix}\begin{pmatrix} 1\,2\,3\,4\,5 \\ 3\,4\,1\,5\,2 \end{pmatrix} = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 4\,3\,2\,5\,1 \end{pmatrix}.$$

Inverse of a permutation

Informally, a permutation $\pi \in S_n$ scrambles the elements of $\mathbf{n}$. The inverse of π is the permutation that "undoes" the scrambling. Formally we define the *identity* permutation ι to be the permutation that maps i to i for all $i \in \mathbf{n}$.

Definition 8.1 (*Inverse*) Let $\pi \in S_n$. Then the inverse of π is the permutation π^{-1} such that

$$\pi\pi^{-1} = \iota \quad \text{and} \quad \pi\pi^{-1} = \iota.$$

If π^{-1} is the inverse of π then π is the inverse of π^{-1}. That is, $(\pi^{-1})^{-1} = \pi$. In general, $(\pi_1\pi_2)^{-1} = \pi_2^{-1}\pi_1^{-1}$.

Cycles and cycle decomposition

For a permutation $\pi \in S_n$, if we repeatedly apply π to any $i \in \mathbf{n}$,

$$i \xrightarrow{\pi} \pi(i) \xrightarrow{\pi} \pi^2(i) \ldots,$$

we must eventually (say after k steps) reach i again since $\mathbf{n}$ is a finite set. If there is some $j \in \mathbf{n}$ which does not occur in this sequence, then we can form a similar sequence for j, and keep doing this until every element of $\mathbf{n}$ occurs in some sequence.

Definition 8.2 (*Cycle*) Let $i_1, i_2, \ldots i_k$ be k distinct integers in $\mathbf{n}$. A cycle π_c written as $(i_1, i_2, \ldots, i_k)$ is a permutation in S_n defined as

$$\pi_c(i_s) := \begin{cases} i_{s+1} & \text{if } i_s \in \{i_1, i_2, \ldots i_{k-1}\} \\ i_1 & \text{if } i_s = i_k \\ i_s & \text{otherwise.} \end{cases}$$

A 2-cycle is a cycle of length 2. That is, $\pi = (i, j)$ means that $\pi(i) = j, \pi(j) = i$ and $\pi(k) = k$ if $k \neq i, j$. A cycle of length 2 is also called a *transposition*.

Two cycles are said to be disjoint if they have no elements in common.

Theorem 8.3 *Any permutation $\pi \in S_n$ can be written as a product of disjoint cycles.*

Example Let $\pi = \begin{pmatrix} 1\,2\,3\,4\,5\,6 \\ 3\,4\,1\,6\,2\,5 \end{pmatrix}$. π can be written as

$$\pi = (1, 3)(2, 4, 6, 5).$$

This way of writing a permutation is referred to as *cycle notation*. There is a unique way of writing a permutation as a product of disjoint cycles, up to the ordering of the cycles (they commute) and cyclic equivalence of each cycle (e.g. $(1, 2, 3) = (2, 3, 1) = (3, 1, 2)$). Since the sizes of the disjoint cycles will always add to n (including if necessary some 1-cycles), we can define the cycle type as follows.

Definition 8.4 (*Cycle type*) The cycle type of a permutation π is the partition $\lambda \vdash n$ whose components are the sizes of the cycles in the disjoint cycle decomposition of π.

Example The cycle type of $\pi = (1, 3)(2, 4, 6, 5)$ is (4, 2) since it has one cycle of length 2 and one cycle of length 4.

Conjugation

Definition 8.5 Let $\pi, g \in S_n$. The conjugate of π by g is defined to be the permutation $g\pi g^{-1}$, and we say that π and $g\pi g^{-1}$ are *conjugate* permutations.

Theorem 8.6 *Let π_1 and π_2 be permutations on the set* $\mathbf{n}$, *then π_1 and π_2 are conjugate in S_n if and only if they have the same cycle type.*

Permutation as product of transpositions

Theorem 8.7 *Any permutation $\pi \in S_n$ can be written as a product of transpositions.*

Example The permutation $\pi = (1,3)(2,4,6,5)$ can be written as

$$\pi = (1,3)(2,4,6,5) = (1,3)(2,5)(2,6)(2,4).$$

While the decomposition of a permutation into a product of disjoint cycles is unique, the decomposition of a permutation into a product of transpositions is not unique. However the number of transpositions used must be either always be even, or always be odd.

Theorem 8.8 *A permutation $\pi \in S_n$ can be expressed as a product of either an even number of transpositions or an odd number of transpositions, but not both.*

Definition 8.9 A permutation is said to be *even* if it can be written as a product of an even number of transpositions. Otherwise it is said to be an *odd* permutation.

References

Bafna V, Pevzner PA (1993) Genome rearrangements and sorting by reversals. In: Proceedings of 34th annual symposium on foundations of computer science. IEEE Press, New York, pp 148–157

Bergeron A, Mixtacki J, Stoye J (2006) A unifying view of genome rearrangements. Algorithms Bioinform 163–173

Braga MDV, Stoye J (2009) Counting all DCJ sorting scenarios. In: Comparative genomics. Springer, Berlin, pp 36–47

Braga MDV, Stoye J (2010) The solution space of sorting by DCJ. J Comput Biol 17(9):1145–1165

Darling AE, Miklós I, Ragan MA (2008) Dynamics of genome rearrangement in bacterial populations. PLoS Genet 4(7)

Dénes J (1959) The representation of a permutation as the product of a minimal number of transpositions and its connection with the theory of graphs. Publ Math Inst Hungar Acad Sci 4:63–70

Egri-Nagy A, Gebhardt V, Tanaka MM, Francis AR (2014) Group-theoretic models of the inversion process in bacterial genomes. J Math Biol 69(1):243–265

Feijão P, Meidanis J (2013) Extending the algebraic formalism for genome rearrangements to include linear chromosomes. IEEE/ACM Trans Comput Biol Bioinform 10(4):819–831

Fraleigh JB (2003) A first course in abstract algebra. Pearson Education India, Delhi

Francis AR (2014) An algebraic view of bacterial genome evolution. J Math Biol 69(6):1693–1718

Fulton W (1997) Young tableaux: with applications to representation theory and geometry. London Mathematical Society Student Texts. cambridge University Press, Cambridge

Hannenhalli S, Pevzner PA (1995) Transforming men into mice (polynomial algorithm for genomic distance problem). In: Proceedings of 36th annual symposium on foundations of computer science, pp 581–592

Herstein IN (2006) Topics in algebra. Wiley, New York

Lin YC, Tang CY (2006) Exposing phylogenetic relationships by genome rearrangement. Adv Comput 68:1–57

Lu CL, Huang YL, Wang TC, Chiu HT (2006) Analysis of circular genome rearrangement by fusions, fissions and block-interchanges. BMC Bioinform 7(1):295

Meidanis J, Dias Z (2000) An alternative algebraic formalism for genome rearrangements. Comparative genomics: empirical and analyitical approaches to gene order dynamics, map alignment and evolution of gene families, pp 213–223

Miklós I, Darling AE (2009) Efficient sampling of parsimonious inversion histories with application to genome rearrangement in yersinia. Genome Biol Evol

Ouangraoua A, Bergeron A (2010) Combinatorial structure of genome rearrangements scenarios. J Comput Biol 17(9):1129–1144

Petersen TK, Tenner BE (2013) How to write a permutation as a product of involutions. Integers 13

Siepel AC (2002) An algorithm to enumerate all sorting reversals. In: Proceedings of the sixth annual international conference on computational biology

Stanley RP (1999) Enumerative combinatorics, vol 2. Cambridge University Press, Cambridge

Watterson GA, Ewens WJ, Hall TE, Morgan A (1982) The chromosome inversion problem. J Theor Biol 99(1):1–7. doi:10.1016/0022-5193(82)90384-8

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