JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 30, Number 2, 2023
© Mary Ann Liebert, Inc.
Pp. 118–130
DOI: 10.1089/cmb.2021.0468

Escape from Parsimony of a Double-Cut-and-Join Genome Evolution Process

MONA MEGHDARI MIARDAN,[1] ARASH JAMSHIDPEY,[2] and DAVID SANKOFF[1]

ABSTRACT

We analyze models of genome evolution based on both restricted and unrestricted double-cut-and-join (DCJ) operations. Not only do our models allow different types of operations generated by DCJs (including reversals, translocations, transpositions, fissions, and fusions) to take different weights during the course of evolution, but they also let these weights fluctuate over time. We compare the number of operations along the evolutionary trajectory with the DCJ distance of the genome from its ancestor at each step, and determine at what point they diverge: the process escapes from parsimony. Adapting the method developed by Berestycki and Durrett, we approximate the number of cycles in the breakpoint graph of a random genome at time *t* and its ancestral genome by the number of tree components in a random graph (not necessarily an Erdös–Rényi one) constructed from the model of evolution. In both models, the process on a genome of size *n* is bound to its parsimonious estimate up to $t \approx n/2$ steps.

Keywords: breakpoint graph, double-cut-and-join, Erdös–Rényi graphs, parsimony binding, random walk.

1. INTRODUCTION

The introduction of the ''double-cut-and-join'' (DCJ) methods by Yancopoulos et al. (2005), and in a slightly different formulation by Bergeron et al. (2006), greatly invigorated the field of genome distance algorithmics. DCJ incorporated the operation of interchanging blocks of genes between arbitrary regions of a chromosome (including ''block transposition'' of neighboring regions) to the two previously studied operations of reversing of a block of genes in a chromosome (''inversion'') and interchanging the prefixes or suffixes of two chromosomes (''reciprocal translocation,'' which formally includes chromosome fusion and fission). As a result, the exact DCJ distance is easily calculated in linear time, avoiding the cumbersome work necessitated by the ingenious solution (called ''HP'') of Hannenhalli and Pevzner (1995a) and Hannenhalli and Pevzner (1995b) a decade earlier, where only reversals and reciprocal translocations were considered.

Edit distances for comparing genomes, such as DCJ and HP, minimize the number of steps, chosen from a small repertoire of operations, to transform one genome into another. A mistaken comment sometimes

[1] Department of Mathematics and Statistics, University of Ottawa, Ottawa, Canada.
[2] Department of Mathematics, Columbia University, New York, New York, USA.

heard, whether approving or critical, is that these methods assume that ''evolution takes the shortest path.'' The methods, however, make no assumption at all about the preferences of the evolutionary process; they are simply tools for inferring evolutionary history.

The minimization criterion basic to edit distances makes it a maximum parsimony method. As such it is model-free, unlike probability-based models. Nevertheless, under any reasonable model of evolution, calculations or simulations show that for a genome that has only moderately evolved, DCJ or similar methods inevitably recover the true number of steps that actually occurred (cf. Berestycki and Durrett, 2006; Jamshidpey and Sankoff, 2013). In this sense, evolution has indeed ''followed'' a most parsimonious path, but this is only an inferential result, not an evolutionary tendency. We say that the evolution is ''parsimony-bound'' during this initial period.

As a genome diverges more substantially from its initial state, however, parsimony methods will eventually find a shorter path than the one actually traversed by the evolving genome. Parsimony underestimates the length of the evolutionary trajectory. When this occurs, we can say that evolution ''escapes from parsimony.''

The nature of this escape and the point at which it happens depend on the stochastic process modeling evolution and the distance measure comparing points on the sample path to the starting point. Little mathematical work has been done relevant to this direct question, although there has been much research on probabilistic modeling of evolution using DCJ operations, some of which is reviewed in Biller et al. (2015). In the present article, we model the evolution of genomes as a time-continuous Markov jump process on the space of genomes, for which the different types of genomic operations (e.g., reversals, translocations, and transpositions) generated by the restricted (Yancopoulos et al., 2005) or unrestricted (Bergeron et al., 2006) DCJs occur at different rates, and use DCJ as a tool for tracking it.

A basic result in this field is due to Berestycki and Durrett (2006), who found that escape from parsimony for a random transposition (pairwise exchange) process does not occur before $n/2$ steps, where n is the number of genes in the genome.

2. GENOMES AND DCJ OPERATIONS

The genes or markers in a genome can be represented by different positive integers. We denote by $\mathcal{G}_{n,k}$ the space of all genomes with n specific genes (or markers), denoted by $1, \ldots, n$, exactly k linear chromosomes, and a number of circular chromosomes (possibly 0). For the purpose of this article, it is most convenient to represent a genome $G \in \mathcal{G}_{n,k}$ as an alternating graph $G = (V(G), E(G))$ on $2(n+k)$ vertices, $2n$ of which are of degree 2 and the rest are of degree 1, as in Figure 1. More precisely, we denote by $+i$ and $-i$ the extremities (head and tail) of gene (or marker) $i \in [n] := \{1, \ldots, n\}$, and denote by $\tau_j, j \in [2k]$, the telomeres of k linear chromosomes, that is,

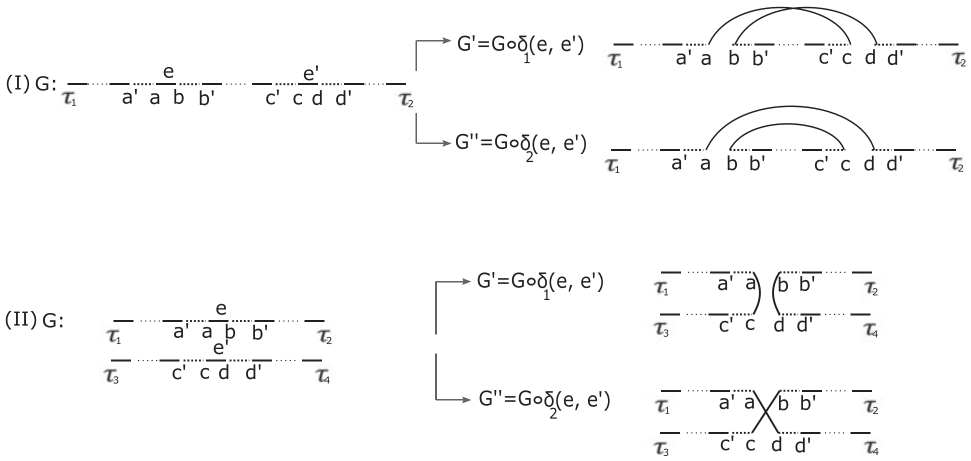

FIG. 1. DCJ operations. The dotted and solid lines represent genes and adjacencies, respectively. The DCJ operators act on the adjacencies $e = (a, b)$ and $e' = (c, d)$. DCJ, ''double-cut-and-join.'' Adjacencies e and $\acute{e}$ pertain to the same linear chromosome (I) and different linear chromosomes (II) respectively.

$$V(G) = \{\pm 1, \ldots, \pm n\} \cup \{\tau_1, \ldots, \tau_{2k}\}.$$

On the contrary, let $E_1(G)$ be the invariant for all genomes G, and define it to be the set of edges $\{-i, +i\}$, for $i \in [n]$, and let $E_2(G)$ vary for different genomes G and be defined as a perfect matching on $V(G)$. Then, $E(G)$ consists of the edges in $E_1(G)$ and $E_2(G)$, where multiple edges are allowed. For instance, any circular chromosome of size one, including only a gene (marker) i, is represented in fact by a double edge between its two extremities $-i$ and $+i$. We call $E_1(G)$ and $E_2(G)$ the set of genes and set of adjacencies, respectively, of genome G. We notice that $|E_2(G)| = n + k$, for any $G \in \mathcal{G}_{n,k}$. We suppose the edges in $E_1(G)$ and $E_2(G)$ are dotted and solid, respectively. The solid edges that are incident to the telomeric vertices $\{\tau_1, \ldots, \tau_{2k}\}$ are called telomeric edges, and those vertices in $\{\pm 1, \ldots, \pm n\}$ that are adjacent to $\{\tau_1, \ldots, \tau_{2k}\}$ are called telomeric gene extremities.

It is then clear that the graph G is a union of disjoint alternating paths (linear chromosomes) and cycles (circular chromosomes). Indeed, $\deg(+i) = \deg(-i) = 2$ and $\deg(\tau_j) = 1$, for any $i \in [n]$ and $j \in [2k]$. Note that, under this definition, a linear chromosome may be null (i.e., with no genes). In this case, the null chromosome only contains two telomeres τ_i, τ_j and their connecting edge $\{\tau_i, \tau_j\}$.

The main purpose of this article is to study some classes of DCJ Markov processes, modeling evolution of genomes, in which the probabilities of choosing *two different types* of DCJ operators to act on the current genome state are unequal. To classify the DCJ operators into two natural types, one including the reversals and the other including the nonreversals when the DCJ operator acts on two edges of the same chromosome, we need to assign an orientation to the connected components (chromosomes) of each genome in $\mathcal{G}_{n,k}$. To establish this, recall that an orientation of a graph G is a digraph whose underlying graph is G. In other words, we can assign a direction to each edge $\{u, v\} \in E(G)$ by assigning exactly one of the two possible arcs (u, v) or (v, u) to it.

An orientation of G is then a digraph obtained from G by assigning a direction to each of its edges. Clearly, there are $2^{2(n+k)}$ orientations for $G \in \mathcal{G}_{n,k}$. Let C be a cycle component, and let P be a path component of G. A direction of C, namely $\overrightarrow{C}$, is an orientation for which $\overleftarrow{C}$ is a directed cycle, that is $\deg_+(v) = \deg_-(v) = 1$, for $v \in V(C)$, where $\deg_+(v)$ and $\deg_-(v)$ denote the outdegree and indegree of v. Similarly, one can give a direction to a linear chromosome P, by choosing one of its telomeres as a starting point. To be more precise, an orientation $\overrightarrow{P}$ is called a direction for P, if it is a directed path. Hence, there are exactly two directions for each component of G. Denoting by $\kappa(G)$ the number of connected components of G, this implies that the number of directions of G is $2^{\kappa(G)}$.

Now, let x be the gene with the smallest label in a circular chromosome C (or a linear chromosome P, respectively). The standard direction is the unique direction of C (P, respectively) for which the direction of $\{-x, +x\}$ is given by arc $(-x, +x)$. Moreover, the standard direction of G is the unique orientation of that for which each component is assigned to its standard direction. We henceforth assume that each genome $G \in \mathcal{G}_{n,k}$ is furnished with its standard direction.

A DCJ operation is a genomic operation that cuts two given adjacencies (edges) of a genome and rejoins the free extremities of the genes involved in one of the two possible ways (Fig. 1). Two types of the DCJ operations are defined as follows. For $G \in \mathcal{G}_{n,k}$ and $e = (a, b)$, $e' = (c, d) \in E_2(G)$, a first-type DCJ (DCJ-1 or δ_1) operation, denoted by $\delta_1(e, e')$ is defined by $G' = G \circ \delta_1(e, e')$, where $V(G') = V(G)$, $E_1(G') = E_1(G)$, and

$$E_2(G') = E_2(G) \cup \{(a, c), (b, d)\} \backslash \{e, e'\}.$$

Similarly, a second-type DCJ (DCJ-2 or δ_2) operation on G is defined by $G'' = G \circ \delta_2(e, e')$, where $V(G'') = V(G)$, $E_1(G'') = E_1(G)$, and

$$E_2(G'') = E_2(G) \cup \{(a, d), (b, c)\} \backslash \{e, e'\},$$

as depicted in Figure 1.

Note that for e, e' taken from the same chromosome, $\delta_1(e, e')$ is an inversion (reversal) operation, as shown in Figure 1-I. From the definition, $\mathcal{G}_{n,k}$ is closed under the DCJ operations, and the DCJ distance between two genomes $G_1, G_2 \in \mathcal{G}_{n,k}$, $d(G_1, G_2)$, is the minimum number of DCJ operations required to transform G_1 into G_2 or vice versa. Equivalently, this distance can be interpreted as the length of the shortest path on the graph $\mathcal{G}_{n,k}$ with the set of vertices $\mathcal{G}_{n,k}$ where each pair of vertices are connected with an edge if their DCJ distance is 1. The formula for the DCJ distance of two genomes can be given in terms of the numbers of cycles and even paths in their breakpoint graph.

In mathematical terms, letting $T:=\{\tau_1,\ldots,\tau_{2k}\}$ and $T':=\{\tau'_1,\ldots,\tau'_{2k}\}$, the breakpoint graph of two genomes $G_1, G_2\in\mathcal{G}_{n,k}$, denoted by $BP(G_1,G_2)$, is a graph whose vertices are given by $\{\pm 1,\ldots,\pm n\}\cup T\cup T'$ and whose edges are the adjacencies of G_1 and G_2^*, that is, edges in $E_2(G_1)$ and $E_2(G_2^*)$ (multiple edges are allowed), where G_2^* is obtained from G_2 by changing the labels of its telomeric vertices τ_j to τ'_j, for $j=1,\ldots,2k$. The adjacencies of G_1 and G_2^* are usually presented in two different colors, say gray and black. We notice that every vertex of $BP(G_1,G_2)$ obtained from a telomeric vertex of G_1 or G_2^* has degree 1. The rest of the vertices of $BP(G_1,G_2)$ are all of degree 2, that is each such vertex is incident to exactly one gray edge and one black edge. Hence, $BP(G_1,G_2)$ is the disjoint union of some alternating cycles and $2k$ alternating paths (or lines). The DCJ distance between two genomes G_1 and G_2 is then given by

$$d(G_1,G_2)=n-\mathbf{C}(G_1,G_2)-\frac{\mathbf{P}_e(G_1,G_2)}{2}, \tag{1}$$

where $\mathbf{C}(G_1,G_2)$ and $\mathbf{P}_e(G_1,G_2)$ determine the number of cycles and the number of even paths in $BP(G_1,G_2)$, respectively (cf. Bergeron et al., 2006). A line is called a *TT*-line or with *TT* ends if both telomeric vertices of that line belong to *T*. Lines with *TT′* or *T′T′* ends are defined similarly. It is straightforward to see that a line is even (of even size) if and only if it has *TT′* ends.

To measure the exact displacement of a genome G_2, after performing some DCJ operators on it, with respect to another genome G_1, one needs to study the effects of different types of DCJ operators on $BP(G_1,G_2)$. To formulate this, in the same spirit of the standard direction assigned to linear and circular chromosomes of a genome in $\mathcal{G}_{n,k}$, we need to give a direction to the cycles and lines of a breakpoint graph. This can be done by determining a start and an end point for a line. More precisely, the direction of a *TT′* line is given by traversing it, starting at its *T* telomeric vertex and ending at its *T′* telomeric vertex. For *TT* or *T′T′* lines, the starting point is the telomeric vertex with a smaller index.

On the contrary, a cycle can be traversed, starting at its gene extremity with the minimum index, through the unique black edge incident to it. It is clear that the described direction of a cycle or a line of $BP(G_1,G_2)$ does not necessarily coincide with its orientation induced from the direction of edges in genomes G_1 and G_2.

The effect of a DCJ operation on the distance between two genomes can be studied as follows. Let $G_1, G_2\in\mathcal{G}_{n,k}$ and $e=[a,b]$, $e'=[c,d]\in E_2(G_2)$, where with a mild abuse of notation, by $[a,b]=[a,b]_{G_1,G_2}$, we mean that the direction of e is from a to b, in the natural direction of $BP(G_1,G_2)$, described above. Note once again that this direction may be different from the direction of $\{a,b\}$ in the genome G_2. The DCJ operation $\Delta_1(e,e')=\Delta_1^{G_1,G_2}(e,e')$ is defined on G_2 by cutting e and e' and joining a to c, and b to d. Similarly $\Delta_2(e,e')=\Delta_2^{G_1,G_2}(e,e')$ cuts e and e' and joins a to d, and b to c. Note that δ_i and Δ_i operations are different due to the different orientations they use to join vertices.

Let $\alpha_i:=d(G_1,G_2\circ\Delta_i(e,e'))-d(G_1,G_2)$. In fact, α_i measures the displacement of the genome G_2, with respect to a reference genome G_1, after a Δ_i-DCJ operation is performed on it. If e, e' are in two different cycles of $BP(G_1,G_2)$, then $\alpha_1=\alpha_2=1$, since two cycles merge following the DCJ operation on e and e', as in Figure 2-III. If both are in one cycle (Fig. 2-IV), then Δ_1 does not change the number of cycles or paths, but Δ_2 fragments it into two smaller cycles. Hence $\alpha_1=0$ and $\alpha_2=-1$. If they belong to one line (Fig. 2-V), then, similarly to the cycle case, $\alpha_1=0$. In this case, Δ_2 splits the line into a cycle and a line whose length has the same parity as that of the original line (e.g., if the original line is of an even size, then so is the size of the new line). In this case, since a new cycle is added, we get $\alpha_2=-1$. When one edge belongs to an odd line (*TT* or *T′T′* line) and the other belongs to an even line (*TT′*-line), as in Figure 2-I, clearly, after splitting and rejoining two lines together, both Δ_1 and Δ_2 operations result in an odd and an even line, which means $\alpha_1=\alpha_2=0$.

If both lines are odd, but all of their four telomeric vertices come from one genome, say with *TT* ends, then both types of DCJ operator generate two odd *TT*-lines, that is, $\alpha_1=\alpha_2=0$. If both are odd, one with *TT* ends and the other with *T′T′*-ends, then Δ_1 and Δ_2 operations generate two *TT′*-lines which are even, hence $\alpha_1=\alpha_2=-1$. For two even *TT′*-lines, the Δ_1 operation gives two odd lines, one with *TT* ends and the other with *T′T′* ends, while Δ_2 operator gives two even *TT′*-lines. Therefore, $\alpha_1=1$, $\alpha_2=0$. Finally, as shown in Figure 2-II, if e belongs to a line and e' belongs to a cycle, the cycle and line merge after implementing Δ_1 or Δ_2 operations. This gives rise to a line whose length has the same parity as that of the original one. Hence in this case $\alpha_1=\alpha_2=1$.

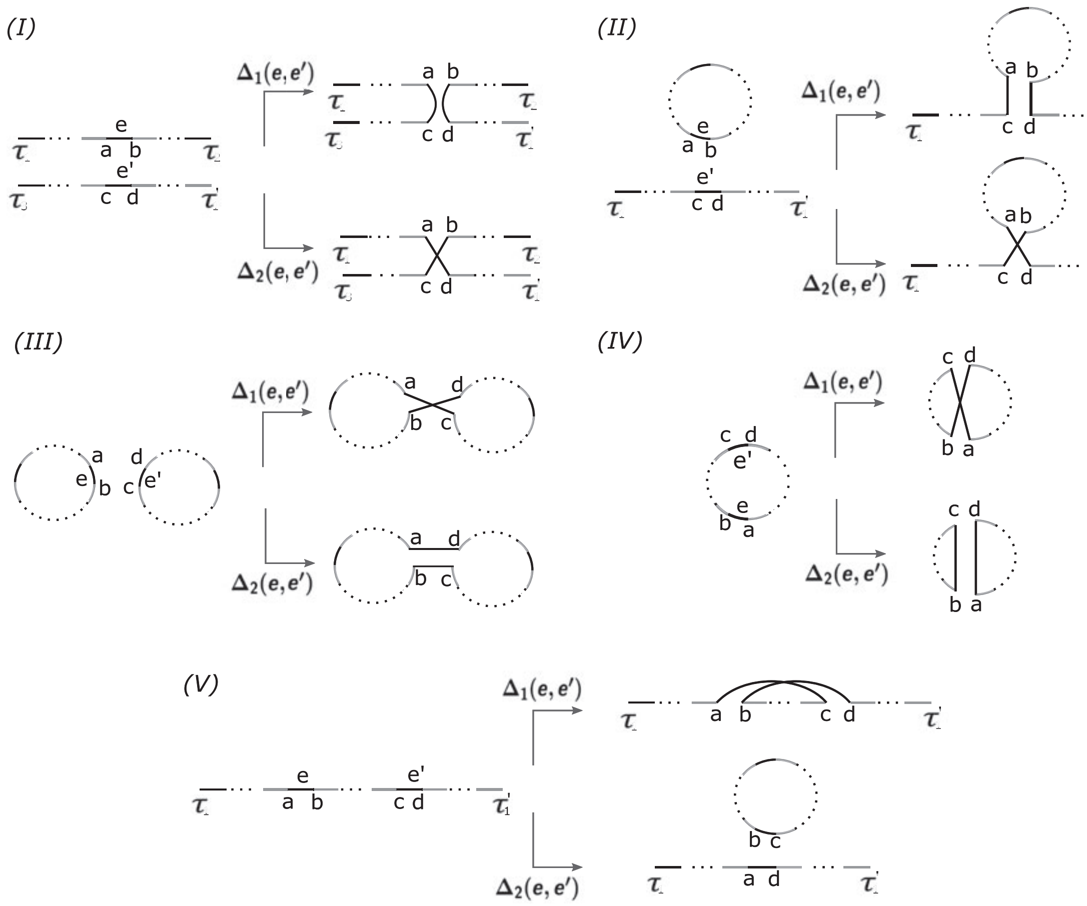

FIG. 2. Effects of DCJ operations on the components of the breakpoint graph of two genomes. The DCJ operations act on $e=[a, b]$ and $e'=[c, d]$ for an odd line and an even line (I), a cycle and a line (II), two different cycles (III), and one cycle (IV) or one line (V).

3. DCJ EVOLUTION AND DCJ PARSIMONIOUS PATHS

As mentioned in the previous section, the reversals are, in fact, DCJ-1 operations. DCJ can also generate translocations, fissions, and fusions. In addition, two consecutive DCJs may generate a genomic transposition or a block interchange operation (Yancopoulos et al., 2005; Bergeron et al., 2006; Fertin et al., 2009). Consequently, DCJs provide a rich family of mathematical operations, which model the genomic evolution of some species and their edit distances. We can model the DCJ evolution of genomes as a Markov jump process on the space of genomes $\mathcal{G}_{n,k}$. Given an initial weight $p \in [0, 1]$, let $\iota_1, \iota_2, \ldots$ be an *i.i.d.* sequence of random variables with $\mathbb{P}(\iota_k=1)=p=1-\mathbb{P}(\iota_k=2)$, for any $k \in \mathbb{N}$. The DCJ process $X^{(p)}=(X_t^{(p)})_{t\geq 0}$ starts at genome $X_0^{(p)}=G_0 \in \mathcal{G}_{n,k}$, and jumps at random times $0 \leq \tau_1 < \tau_2 < \ldots$, where $\{\tau_1, \tau_2, \ldots\}$ are points of a Poisson process on $\mathbb{R}_+$, with rate 1.

At a jump time τ_i, having $X_{\tau_i}^{(p)}=G$, we choose two different solid edges $(a, b), (c, d) \in E_2(G)$, uniformly at random, without replacement, and let the random walk jump to $G \circ \delta_{\iota_i}((a, b), (c, d))$. In other words, at each time τ_i, $X^{(p)}$ jumps to $G \circ \delta_1(e, e')$ with probability $p_{n,k} := p/\{(n+k)(n+k-1)\}$, and to $G \circ \delta_2(e, e')$ with probability $q_{n,k} := (1-p)/\{(n+k)(n+k-1)\}$, for any pair $e \neq e' \in E_2(G)$. Equivalently, $X^{(p)}$ can be considered a continuous-time-biased nearest-neighbor random walk on $\mathscr{G}_{n,k}$ starting at G_0, for which at a Poisson time τ_i, it jumps to one of its neighbors with a probability proportional to its weight where the weights of each δ_1-neighbor and each δ_2-neighbor of G are given by p and $1-p$, respectively. We can see that, for $p=1/2$, $X^{(p)}$ is a continuous-time (symmetric) simple random walk on $\mathscr{G}_{n,k}$. Hereafter, we suppress the superscript p when p is fixed and there is no risk of ambiguity.

We can generalize the above *DCJ* process as follows. Let $\mathbf{p} := (p_t)_{t\geq 0}$, $p_t \in [0, 1]$ for $t \in \mathbb{R}_+$, be a given function from $\mathbb{R}_+$ to $[0, 1]$. To obtain $X^{(\mathbf{p})}=(X_t^{(\mathbf{p})})_{t\geq 0}$ from $X^{(p)}$, we modify the above definition by giving the weight p_{τ_k} and $1-p_{\tau_k}$ to δ_1 and δ_2 operations, which can be performed on the current genome at each jump time τ_k. Then, the probability of choosing each δ_1 operation and each δ_2 operation, at time τ_k, is given by $p_{\tau_k}/\{(n+k)(n+k-1)\}$ and $(1-p_{\tau_k})/\{(n+k)(n+k-1)\}$. In fact, $X^{(p)}$ is a special case of $X^{(\mathbf{p})}$, for $p_t=p$; $t \in \mathbb{R}_+$.

Being parsimony-bound means that in transforming into another genome *B*, a genome *A* has evolved along a shortest path to *B*. This is, in fact, the fastest way that *A* can be transformed into *B* through the paths of evolution. Roughly speaking, any set of allowed genomic operations *R* on a space of genomes *S*, under which the genome space is closed, induces an edit distance ρ on *S* that, for $G_1, G_2 \in S$, is defined as the minimum number of elements of *R* needed to transform G_1 into G_2 or vice versa. Of course, we assume that this is possible in finite steps, for any pair of genomes in *S*. When a random walk $(\xi_t)_{t\geq 0}$ models the evolution of genomes under *R*-operations, evolution is parsimony-bound up to time *t*, if

$$\rho(\xi_0, \xi_t) \approx t.$$

Setting $S := \mathcal{G}_{n,k}$, and letting *R* include all DCJ operations, the simulation results in Figure 3 show that the parsimony binds the trajectory up to time *cn*, for $c \approx 0.5$. At each step, the DCJ distance is computed using UniMoG (Braga et al., 2011; Hilker et al., 2012).

Berestycki and Durrett (2006) proved that the transposition random walk is bound to parsimony up to time $n/2$. Counting the number of hurdles of a reversal random walk, and making use of their results, one can see this also holds for the reversal walk up to the same time $n/2$ (Jamshidpey and Sankoff, 2013). In this study, we make use of the Berestycki-Durrett method to show that the same result is true for the nonhomogeneous (biased) DCJ stochastic process defined above. To this end, we need a labeling mechanism that is updated at each jump of the DCJ process. The relabeling is important for constructing a random graph process whose number of components estimates the number of cycles of the $BP(X_0, X_t)$, and therefore the DCJ distance between the genomes X_0 and X_t. A labeling of a genome $G \in \mathcal{G}_{n,k}$ is a bijection from $E_2(G)$ onto $[n+k] := \{1, 2, \ldots, n+k\}$.

The labeling process $L=(L_t)_{t\geq 0}$ can be obtained from the DCJ process *X* as follows. We start from an arbitrary label $L_0=\hat{L}_0$ for X_0. From exchangeability, the results in the next theorems do not really depend on any particular choice of $\hat{L}_0$. Let *t* be a jump time such that X_t jumps to $X_{t+} := X_t \circ \delta_i(e, e')$ for $e=(a, b), e'=(c, d) \in X_t$. Let $x := \min\{a, b, c, d\}$, and let $\tilde{u}_x$ be the black edge of X_t, either *e* or *e'*, which is incident to *x*, while $\tilde{v}_x$ be the other edge, that is, $\tilde{v}_x=\{e, e'\}\backslash\tilde{u}_x$. Similarly, let u_x be the edge of X_{t+}, which is incident to *x* after performing $\delta_i(e, e')$ on X_t, and let v_x be the other new edge in X_{t+} that is created as a result of joining the two other free gene extremities. We update the labeling process *L* by

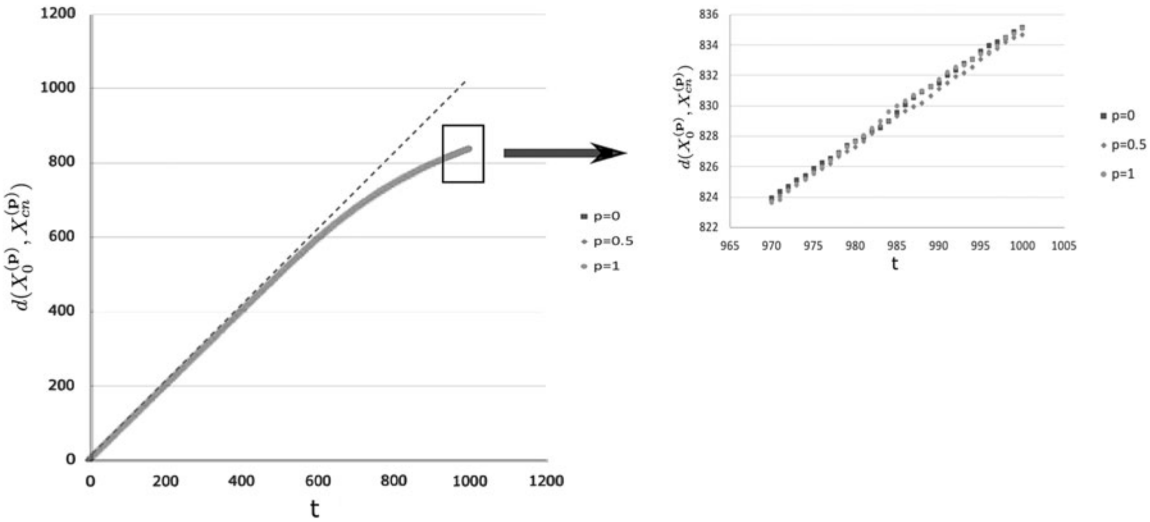

FIG. 3. Simulation for the unrestricted DCJ model illustrating parsimony binding up to time *cn*, for $c \approx 0.5$. The average DCJ distance estimates are based on 50 runs, where $p_t=p$ $(p=0,\ 0.5,\ 1)$ at a jump time *t*, if the randomly selected edges involved in the DCJ operation are in the same chromosome, and $p_t=0.5$ if the edges are in two distinct chromosomes; see Remark 2. The ancestral genome contains 1000 genes in 3 linear chromosomes of sizes 250, 350, and 400.

$$L_{t+}(w) = \begin{cases} L_t(\tilde{u}_x) & \text{if } w = u_x \\ L_t(\tilde{v}_x) & \text{if } w = v_x \\ L_t(w) & \text{if } w \notin \{u_x, v_x\}. \end{cases} \tag{2}$$

Using L and X, we can now define a random graph process $Z = (Z_t)_{t \geq 0}$ whose number of tree components estimates the number of cycles of the breakpoint graph, after a convenient rescaling. Let Z_0 be the trivial graph with vertices $1, 2, \dots, n+k$ (representing the labels) and with no edges. Let $(\lambda_1(i), \lambda_2(i))$ be the pair of labels of two uniformly random edges (without replacement) in X_{τ_i}, which is selected for a DCJ operation at time τ_i. Connect $\lambda_1(i)$ and $\lambda_2(i)$ by an edge in Z_{τ_i+}, if they are not already adjacent in Z_{τ_i}. As a matter of fact, the number of times at which the random sequence $((\lambda_1(i), \lambda_2(i)))_{i \in \mathbb{N}:\, \tau_i < cn}$ takes an ordered pair of labels $(\ell, \ell') \in [n+k]^2$, $\ell \neq \ell'$, is a Poisson random variable with mean $cn/\{(n+k)(n+k-1)\}$. Hence, the probability that each edge $\{\ell, \ell'\}$ that appears in the random graph is given by

$$1 - e^{-\frac{2cn}{(n+k)(n+k-1)}} \approx \frac{2cn}{(n+k)(n+k-1)} + O(n^{-2}).$$

Thus, letting Z^*_{cn} be the Erdös–Rényi random graph on the vertex set $[n+k]$ in which each edge is independently present with probability $2c/n$, the expected difference of number of edges in Z_{cn} and Z^*_{cn} is of order $O(1)$. As the results from Bollobás (2001; Theorem 5.12) hold for Erdös–Rényi random graphs with the parameter $2c/n$, we get

$$\mathbb{E}[\mathcal{T}_{cn}] = (1 - \gamma(c))n + O(1),$$

where $\mathcal{T}_{cn}$ denotes the number of tree components of Z_{cn}, and for $c > 0$,

$$\gamma(c) : = 1 - \frac{1}{2c} \sum_{j=1}^{\infty} \frac{j^{j-2}}{j!} (2ce^{-2c})^j.$$

In fact, $\gamma(c) = c$, for $0 \leq c \leq \frac{1}{2}$, and $\gamma(c) < c$, for $c > \frac{1}{2}$. Similarly, once again from Bollobás (2001; Theorem 5.12), we get

$$var(\mathcal{T}_{cn}) = O(n).$$

We can then easily see that the Erdös–Rényi random graph Z^*_{cn} approximates Z_{cn}. In mathematical terms, from the Chebyshev's inequality, as $n \to \infty$

$$\frac{\mathcal{T}_{cn} - (1 - \gamma(c))n}{b_n \sqrt{n}} \to 0, \text{ in probability,} \tag{3}$$

for any sequence $(b_n)_{n \in \mathbb{N}}$, with $b_n \to \infty$.

Before stating the main result of the article, we also need to count labels $\{\ell, \ell'\}$ picked at least twice by $(\{\lambda_1(i), \lambda_2(i)\})_{i:\, \tau_i < cn}$. More precisely, for any pair of labels $\{\ell, \ell'\}$, let $I_{cn}(\ell, \ell') = I_{cn}(\ell', \ell) = 1$, if the random sequence $(\{\lambda_1(i), \lambda_2(i)\})_{i:\, \tau_i < cn}$ takes $\{\ell, \ell'\}$ at least twice, and let it be equal to 0, otherwise. Letting $I_{cn} = \sum_{\ell, \ell'} I_{cn}(\ell, \ell')$, the following lemma follows immediately.

Lemma 1. *For any $c > 0$ and any sequence $(b_n)_{n \geq 1}$ with $b_n \to \infty$, we have*

$$\frac{I_{cn}}{b_n \sqrt{n}} \to 0, \text{ in probability,}$$

as $n \to \infty$.

Proof. We have

$$\mathbb{E}[I_{cn}(\ell, \ell')] = \mathbb{P}(I_{cn}(\ell, \ell') = 1) = e^{-\frac{2cn}{(n+k)(n+k-1)}} \sum_{j=2}^{\infty} \left(\frac{2cn}{(n+k)(n+k-1)} \right)^j / j! = O(n^{-2}),$$

and

$$var(I_{cn}(\ell, \ell')) = \mathbb{P}(I_{cn}(\ell, \ell') = 1) - \mathbb{P}(I_{cn}(\ell, \ell') = 1)^2 = O(n^{-2}).$$

Thus $\mathbb{E}[I_{cn}] = O(1)$ and $var(I_{cn}) = O(1)$, that means the number of edges in Z_{cn} for which the corresponding labels are taken more than once is negligible, and the lemma follows from the Chebyshev's inequality. □

Theorem 1. *Fix* $\mathbf{p} = (p_t)_{t \geq 0}$ *and* $c > 0$. *Let* $(b_n)_{n \geq 1}$ *be any sequence for which* $b_n \to \infty$, *as* $n \to \infty$. *Then, as* $n \to \infty$,

$$\frac{d(X_0^{(\mathbf{p})}, X_{cn}^{(\mathbf{p})}) - \gamma(c)n}{b_n \sqrt{n}} \to 0, \text{ in probability.} \tag{4}$$

Proof. We follow the proof of Berestycki and Durrett (2006; Theorem 3) and adapt it for the DCJ stochastic process. First, note that the number of paths of even length in $BP(X_0, X_{cn})$ is bounded by $2k$, and hence,

$$\frac{\mathbf{P}_e(X_0, X_{cn})}{2b_n\sqrt{n}} \to 0,$$

pointwise, as $n \to \infty$. Using Equation (1), it then suffices to analyze the asymptotic behavior of $\mathbf{C}(X_0, X_{cn})$. The number of cycles of size greater than $2\sqrt{n+k}$ is less than $\sqrt{n+k}$. Therefore, we only need to handle the cycles whose sizes are less than or equal to $2\sqrt{n+k}$. To this end, we say a cycle in $BP(X_0, X_{cn})$ is nonfragmented, if it is the result of a finite number of merging events, without any fragmentation in its history. Otherwise, it is called a fragmented cycle. Note that a fragmented cycle always has at least one fragmentation event (either a fragmentation on a cycle or on a line) in its history. For any $t \geq 0$, denote by $\mathcal{C}_t^{(f)}$ and $\mathcal{C}_t^{(nf)}$ the number of fragmented and nonfragmented cycles of $BP(X_0, X_t)$, respectively, with size less than or equal to $2\sqrt{n+k}$.

Clearly, $\mathcal{C}_{cn}^{(f)}$ is bounded by 2 times the number of fragmentation events, up to time cn, which results in a fragmented cycle of size at most $2\sqrt{n+k}$, that is, a fragmented cycle with at most $\sqrt{n+k}$ edges from X_{cn}. In order that a DCJ operation on two black edges e, e', located in one connected component of $BP(X_0, X_t)$, splits that component and produces a cycle of size less than $2\sqrt{n+k}$, e and e' should be in distance at most $2\sqrt{n+k}$, in $BP(X_0, X_{cn})$, from each other. At each jump time τ_i, regardless of the value of p_{τ_i}, the probability that $(\lambda_1(i), \lambda_2(i))$ takes a pair of edges with the mentioned property is bounded by $2\sqrt{n+k}/(n+k-1)$. Therefore, the number of such fragmentation events, up to time cn, is bounded by a Poisson random variable with parameter $cn \cdot 2\sqrt{n+k}/(n+k-1) \approx 2c\sqrt{n}$. This implies that, as $n \to \infty$,

$$\frac{\mathcal{C}_{cn}^{(f)}}{b_n\sqrt{n}} \to 0, \text{ in probability.}$$

We notice that for any $c \geq 0$, any tree component K in Z_{cn} represents either (i) a nonfragmented component of $BP(X_0, X_{cn})$, or (ii) those fragmented components of $BP(X_0, X_{cn})$ for which a same pair of labels for a DCJ operation is selected more than once (the number of the latter type of the components is bounded by I_{cn}). This can be easily verified by induction on the number of vertices of K. First note that the claim is clear if K is a trivial component of Z_{cn} (i.e., a component with only one vertex a), as the corresponding adjacency a in X_0 has never been chosen for any DCJ event, up to time cn, and therefore, it is not involved in any fragmentation event up to that time. For $c > 0$ and for a tree component K with $m \geq 2$ vertices, in Z_{cn}, suppose e is the last edge of K that was added at some time $s \leq cn$ to form K.

Then $K' := K \backslash \{e\}$ contains two tree components, say K_1, K_2, each of which has at most $m-1$ vertices, and so from the induction hypothesis, K_i, $i = 1, 2$, is either a type (i) or a type (ii) component of Z_{s-}. If K_1 and K_2 are both of type (i), that is, corresponding to nonfragmented components of $BP(X_0, X_{s-})$, then e corresponds to a coagulation event at time s by which two nonfragmented components of $BP(X_0, X_{s-})$ merge. So in this case, the resulting component K is of type (i) at time$s$. Now if the labels of the end points of any edge in K are chosen again for another DCJ operation, during the time interval (s, cn), then K is a type (ii) component of Z_{cn}; otherwise it remains a type (i) component of that. On the contrary, as $K_1$and K_2 are subtrees of K, if at least one of them is of type (ii) at s^-, then so is K at time cn.

Now, as the number of even paths is bounded by $2k$, the number of cycles or tree components of size greater than $2\sqrt{n+k}$ is both bounded by $\sqrt{n+k}$, and from Lemma 1, the number of labels (ℓ, ℓ') picked at least twice by $(\{\lambda_1(i), \lambda_2(i)\})_{i:\ \tau_i<cn}$ is very small, we can deduce that the tree components of Z_{cn} can approximate $\mathcal{C}_{cn}^{(nf)}$. More precisely,

$$\frac{\mathcal{T}_{cn}-\mathcal{C}_{cn}^{(nf)}}{b_n\sqrt{n}} \longrightarrow 0, \text{ in probability,}$$

as $n \longrightarrow \infty$. Therefore Equation (3) yields

$$\frac{\mathcal{C}_{cn}^{(nf)}-(1-\gamma(c))n}{b_n\sqrt{n}} \longrightarrow 0, \text{ in probability,}$$

as $n \longrightarrow \infty$, which finishes the proof. □

Remark 1. *For $c \le 0.5$, the convergence in Equation (4) implies that DCJ stochastic evolution X is bound by parsimony, since $\gamma(c)=c$.*

Remark 2. *As Theorem 1 holds for any deterministic $(p_t)_{t\ge 0}$, and its proof is indeed independent of the choice of p_t at any jump time t, it can be easily extended to more complicated stochastic processes $(p_t)_{t\ge 0}$ with values in* $[0, 1]$. *For instance p_t may be a random function of X_t or a random function of the edges selected for the DCJ operation at a jump time t. As an example, for given $p, q \in [0, 1]$, the theorem is true when at any jump time t, we define $p_t=p$ if the randomly selected edges for the DCJ are in the same chromosome, and $p_t=q$ if they are in two distinct chromosomes. The weights of the different types of genomic operations generated by DCJs can then be determined by p and q. The simulation results for $p=0, 0.5, 1$ and $q=0.5$, are given in* Figure 3.

4. RESTRICTED DCJ EVOLUTION AND PARSIMONIOUS PATHS

In the previous section, we studied the parsimony binding for the DCJ model when there is no constraint for the number of circular genomes. In other words, an arbitrary number of δ_2-operations, say r, could be consecutively performed on one or more chromosomes to produce r new circular chromosomes in the genome. In this section, we study the parsimony problem for the DCJ operations restricted to the unichromosomal linear genomes with the same genes, that is we assume that consecutive δ_1-operations (reversals) are allowed without any restriction, but on the contrary, no two consecutive δ_2-operations are allowed, which means that once an intermediate circular chromosome appears as a result of a δ_2-operation, two adjacencies (black edges), one from each chromosome, are chosen and a DCJ on them generates a unichromosomal linear genome.

Another way of looking into this is by considering a particular subgraph of $\mathcal{G}_{n,1}$, denoted by $\mathcal{G}_n^*$, induced by those vertices (genomes) in $\mathcal{G}_{n,1}$, which have at most one circular chromosome. The vertices of $\mathcal{G}_n^*$ can be partitioned into U, the unichromosomal linear genomes, and $\tilde{U}$, the genomes with exactly one linear and one circular chromosome. From the definition, it is clear that no two vertices in $\tilde{U}$ are adjacent in $\mathcal{G}_n^*$. The edges between two vertices in U represent reversals, while the ones between a vertex from U and a vertex from $\tilde{U}$ represent the nonreversal DCJ operations needed to transform a unichromosomal linear genome into a genome with one linear and one circular chromosome, and vice versa. The restricted DCJ distance d^* between two genomes $G_1, G_2 \in \mathcal{G}_n^*$ is then the shortest path between G_1 and G_2 in $\mathcal{G}_n^*$. Yancopoulos et al. (2005) showed that, in fact, for any pair of genomes $G_1, G_2 \in \mathcal{G}_n^*$,

$$d^*(G_1, G_2)=d(G_1, G_2).$$

Hence, once again, one can use the number of cycles and even paths in $BP(G_1, G_2)$, to find $d^*(G_1, G_2)$.

For $\mathbf{p}=(p_t)_{t\in\mathbb{R}_+}$, where $0 \le p_t \le 1$, $t \in \mathbb{R}_+$, we define a continuous-time-biased nearest-neighbor random walk $Y^{(\mathbf{p})}=(Y_t^{(\mathbf{p})})_{t\ge 0}$ on $\mathcal{G}_n^*$, where $Y^{(\mathbf{p})}$ jumps at Poisson times with rate 1. At each jump time t, if $Y_t^{(\mathbf{p})} \in \tilde{U}$, then it chooses one of its neighbors, uniformly at random, and jumps to it, and if $Y_t^{(\mathbf{p})} \in U$, then it jumps to each of its δ_1-neighbors (obtained as a result of a reversal DCJ) with probability $2p_t/\{n(n+1)\}$ and jumps to each of its δ_2-neighbors (obtained as a result of a nonreversal DCJ) with probability $2(1-p_t)/\{n(n+1)\}$. The case $p_t = 1$, for any $t \geq 0$, gives the reversal (inversion) random walk on the signed symmetric group of size n. It is known that the reversal random walk diverges from its origin at its maximum speed up to time $n/2$, for large n, where n is the number of genes in a unichromosomal linear genome (Berestycki and Durrett, 2006; Jamshidpey and Sankoff, 2013).

We define the labeling $\tilde{L} = (\tilde{L}_t)_{t \geq 0}$ and the random graph process $\tilde{Z} = (\tilde{Z}_t)_{t \geq 0}$, from $Y^{(\mathbf{p})}$, in the same manner as we defined L and Z from $X^{(\mathbf{p})}$. As before, we drop the superscript $\mathbf{p}$ when there is no risk of confusion. The following theorem estimates $d(Y_0, Y_{cn})$ using the number of tree components of $\tilde{Z}_{cn}$.

Theorem 2. *Let* $\mathbf{p} = (p_t)_{t \geq 0}$, *such that* $0 \leq p_t \leq 1$, $c > 0$, *and let* $(b_n)_{n \geq 1}$ *be an arbitrary sequence of real numbers such that* $b_n \to \infty$, *as* $n \to \infty$. *Then, as* $n \to \infty$,

$$\frac{d(Y_0^{(\mathbf{p})}, Y_{cn}^{(\mathbf{p})}) - n + \tilde{\mathcal{T}}_{cn}}{b_n \sqrt{n}} \to 0, \ \textit{in probability},$$

where $\tilde{\mathcal{T}}_t$ *denotes the number of tree components of* $\tilde{Z}_t$.

Proof. The proof is similar to the proof of Theorem 1. We define the number of fragmented and nonfragmented cycles of $BP(Y_0, Y_{cn})$, as we defined the similar notions for $BP(X_0, X_{cn})$, and denote by $\tilde{\mathcal{C}}_t^{(f)}$ and $\tilde{\mathcal{C}}_t^{(nf)}$, the number of fragmented and nonfragmented cycles of $BP(Y_0, Y_{cn})$ whose sizes are not greater than $2\sqrt{n+1}$. As before, there are at most 2 paths and at most $\sqrt{n}$ cycles of size greater than $2\sqrt{n+1}$ in $BP(Y_0, Y_{cn})$. Therefore, we only need to handle the asymptotic behaviors of $\tilde{\mathcal{C}}_t^{(f)}$ and $\tilde{\mathcal{C}}_t^{(nf)}$. Once again, the former is bounded by 2ζ, where ζ is a Poisson random variable with mean $2c\sqrt{n+1}$, and hence, as $n \to 0$,

$$\frac{\tilde{\mathcal{C}}_{cn}^{(f)}}{b_n \sqrt{n}} \to 0, \ \text{in probability.}$$

Now, it suffices to show, as $n \to \infty$,

$$\frac{\tilde{\mathcal{T}}_{cn} - \tilde{\mathcal{C}}_{cn}^{(nf)}}{b_n \sqrt{n}} \to 0, \ \text{in probability.}$$

To establish this, for any pair of labels $\{\ell, \ell'\}$ in $(BP(\tilde{Y}_0, \tilde{Y}_t))_{0 \leq t \leq cn}$, we define the indicator functions $\tilde{I}_{cn}(\ell, \ell') = \tilde{I}_{cn}(\ell', \ell)$ and their total sum $\tilde{I}_{cn}$, similarly to their counterparts introduced in the proof of Theorem 1. To find an upper bound for $\mathbb{E}[\tilde{I}_{cn}]$, first observe that the probability that any pair of labels $\{\ell, \ell'\}$ is picked for a DCJ at a jump time t is given by $2/\{n(n+1)\}$, if $Y_t \in U$. If $Y_t \in \tilde{U}$, given that the size of the circular chromosome is $2j$ (which means there are exactly j black edges in the circular chromosome), this probability is given by $1/\{j(n+1-j)\}$. Letting q_j be the probability that a nonreversal DCJ results in a circular chromosome of size $2j$, this means that the probability that any pair of labels $\{\ell, \ell'\}$ is picked for a DCJ, at time t, for $Y_t \in \tilde{U}$ is bounded by

$$\begin{aligned} \sum_{j=1}^{n} \frac{q_j}{j(n+1-j)} &\leq \frac{2}{n} \sum_{j=1}^{n} \frac{1}{j(n+1-j)} \\ &= \frac{2}{n(n+1)} \sum_{j=1}^{n} \left(\frac{1}{j} + \frac{1}{n+1-j} \right) \\ &= \frac{4h_n}{n(n+1)} \approx \frac{4 \ln n}{n(n+1)}, \end{aligned} \tag{5}$$

where $h_n := \sum_{j=1}^{n} 1/j$ is the harmonic number. Therefore, integrating this on $[0, cn]$ and letting $\tilde{\zeta}$ be a Poisson random variable with mean q_{cn}^* for

$$q_{cn}^* := \max\left\{ \frac{2cn}{n(n+1)}, \frac{4cnh_n}{n(n+1)} \right\} \approx \frac{4c \ln n}{n},$$

we can couple the random variables $\tilde{I}_{cn}(\ell, \ell')$ with indicator random variables $J_{cn}(\ell, \ell')$ defined for any pair of labels $\{\ell, \ell'\}$ by

$$\mathbb{P}(J_{cn}(\ell, \ell')=1)=1-\mathbb{P}(J_{cn}(\ell, \ell')=0)=\mathbb{P}(\tilde{\zeta} \geq 2),$$

such that $J_{cn}(\ell, \ell')=1$ if $\tilde{I}_{cn}(\ell, \ell')=1$ (the converse is not assumed), that is, $\tilde{I}_{cn}(\ell, \ell') \leq J_{cn}(\ell, \ell')$. We have

$$\begin{aligned}\mathbb{P}(J_{cn}(\ell, \ell')=1) &= e^{-q^*_{cn}} \sum_{j=2}^{\infty} (q^*_{cn})^j/j! \\ &= O\left(\left(\frac{\ln n}{n}\right)^2\right).\end{aligned}$$

Defining $J_{cn}=\sum_{\ell, \ell'} J_{cn}(\ell, \ell')$, we can show $J_{cn}/(b_n\sqrt{n}) \to 0$ in L^1. To see this, write

$$\begin{aligned}\mathbb{E}\left|\frac{J_{cn}}{b_n\sqrt{n}}\right| &\leq \frac{1}{b_n\sqrt{n}} \sum_{(\ell, \ell')} \mathbb{E}|J_{cn}(\ell, \ell')| \\ &= \frac{1}{b_n\sqrt{n}} \sum_{(\ell, \ell')} \mathbb{P}(J_{cn}(\ell, \ell')=1) \\ &= \frac{n(n-1)}{b_n\sqrt{n}} O\left(\left(\frac{\ln n}{n}\right)^2\right) = O\left(\frac{(\ln n)^2}{b_n\sqrt{n}}\right),\end{aligned}$$

implying $J_{cn}/(b_n\sqrt{n}) \to 0$ in L^1 and probability, as $n \to \infty$. Therefore, we get

$$\frac{\tilde{I}_{cn}}{b_n\sqrt{n}} \to 0, \quad in\ L^1 \text{ and probability,}$$

as $n \to \infty$, since $0 \leq \tilde{I}_{cn} \leq J_{cn}$. The same lines of arguments as those in the proof of Theorem 1 finish the proof. □

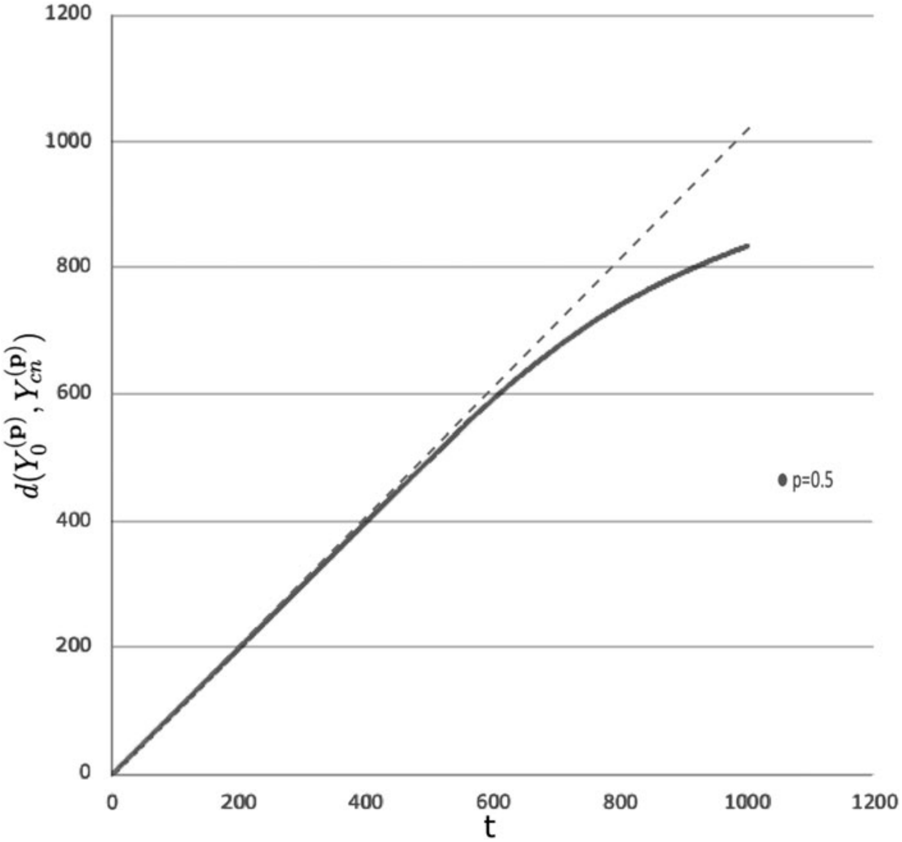

FIG. 4. Simulation for the restricted DCJ model illustrating parsimony binding for restricted DCJ up to time cn, for $c \approx 0.5$. The average DCJ distance estimates are based on 100 runs, for 1000 genes; $p=0.5$.

Remark 3. *Note that $\tilde{Z}_{cn}$ is not an Erdö́s–Ré́nyi random graph. Therefore, an exact analysis of the number of tree components of $\tilde{Z}_{cn}$ is not easy and needs further investigation.*

We have seen that the number of cycles in $BP(Y_0, Y_{cn})$ can be estimated by the number of tree components of the random graph $\tilde{Z}_{cn}$. This provides a method to see to what extent evolution by the restricted DCJ process is bound by parsimony. Of course, this could also be checked by simulating Y and deriving the DCJ distance between Y_{cn} and Y_0, using the existing linear-time algorithms for computing the DCJ distance (cf. Yancopoulos et al., 2005; Bergeron et al., 2006). However, to make use of the method described for the Erdös–Rényi random graph, we investigated the maximum time up to which evolution is bound to parsimony by counting the number of tree components in $\tilde{Z}_{cn}$. The simulation study summarized in Figure 4 shows that the binding holds up to time $n/2$, for large n. The DCJ distance is computed using UniMog.

5. DISCUSSION

Following ideas formulated in Jamshidpey and Sankoff (2013), we have adapted the analysis of Berestycki and Durrett (2006), originally developed for mathematical transposition random walk on permutations, to the biologically relevant DCJ operations. We have shown that as n gets large, the parsimonious DCJ inference algorithms accurately reflect the number of evolutionary steps traced by evolutionary processes as long as this remains less than $n/2$. Parsimony binding and the escape from parsimony are illustrated by simulation experiments. Similar results have been discussed by Biller et al. (2015) and Alexeev and Alekseyev (2017).

Our work required somewhat different techniques for restricted (Yancopoulos et al., 2005) and unrestricted (Berestycki and Durrett, 2006) DCJ frameworks. We also introduced differential weighting to the different genomic operations generated by DCJs (e.g., reversals, translocations), weights that can fluctuate over time. We established explicit relationships between the number of tree components of certain random graphs (including Erdös–Rényi graphs in some cases) and the number of components of the breakpoint graph, explicitly using the Berestycki–Durrett theory,

AUTHORS' CONTRIBUTIONS

M.M.M., A.J., and D.S. conceived and carried out the research, and wrote the article.

AUTHOR DISCLOSURE STATEMENT

The authors declare they have no conflicting financial interests.

FUNDING INFORMATION

Research and publication supported, in part, by Discovery grant RGPIN-2022-05212 to D.S. from the Natural Sciences and Engineering Research Council of Canada. D.S. holds the Canada Research Chair in Mathematical Genomics.

REFERENCES

Alexeev, N., and Alekseyev, M.A. 2017. Estimation of the true evolutionary distance under the fragile breakage model. *BMC Genomics* 18, 356.

Berestycki, N., and Durrett, R. 2006. A phase transition in the random transposition random walk. *Probab Theory Relat Fields* 136, 203–233.

Bergeron, A., Mixtacki, J., and Stoye, J. 2006. A unifying view of genome rearrangements, 163–173. *Workshop on Algorithms in Bioinformatics.*Springer.

Biller, P., Guéguen, L., and Tannier, E. 2015. Moments of genome evolution by double cut-and-join. *BMC Bioinformatics* 16, 1–10.

Bollobás, B. 2001. *Random Graphs*. Cambridge University Press.

Braga, M.D.V., Willing, E., and Stoye, J. 2011. Double cut and join with insertions and deletions. *J Comput. Biol.* 18, 1167–1184.

Fertin, G., Labarre, A., Rusu, I., et al. 2009. *Combinatorics of Genome Rearrangements*. MIT Press.

Hannenhalli, S., and Pevzner, P.A. 1995a. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proceedings of IEEE 36th Annual Foundation of Computer Science*. IEEE Computer Society Press, 581–592.

Hannenhalli, S., and Pevzner, P.A. 1995b. Towards a computational theory of genome rearrangements. *Computer Science Today: Recent Trends and Developments*; Jan van Leeuwen, ed. Vol. 1000 of *Lecture Notes in Computer Science*; Springer-Verlag, 184–202.

Hilker, R., Sickinger, C., Pedersen, C.N., et al. 2012. UniMoG—a unifying framework for genomic distance calculation and sorting based on DCJ. *Bioinformatics* 28, 2509–2511.

Jamshidpey, A., and Sankoff, D. 2013. Phase change for the accuracy of the median value in estimating divergence time. *BMC Bioinformatics* 14, 1–6.

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

Address correspondence to:
Professor David Sankoff
Department of Mathematics and Statistics
University of Ottawa
Ottawa
Ontario K1N 6N5
Canada

E-mail: sankoff@uottawa.ca