AUTHOR QUERY FORM

ELSEVIER	**Journal:** Theoretical Computer Science **Article Number:** 8788	**Please e-mail or fax your responses and any corrections to:** **E-mail: corrections.esch@elsevier.river-valley.com** **Fax: +44 1392 285879**

Dear Author,

Please check your proof carefully and mark all corrections at the appropriate place in the proof (e.g., by using on-screen annotation in the PDF file) or compile them in a separate list. Note: if you opt to annotate the file with software other than Adobe Reader then please also highlight the appropriate place in the PDF file. To ensure fast publication of your paper please return your corrections within 48 hours.

For correction or revision of any artwork, please consult http://www.elsevier.com/artworkinstructions.

Location in article	**Query / Remark click on the Q link to go** **Please insert your reply or correction at the corresponding line in the proof**
Q1	Please confirm that given names and surnames have been identified correctly. confirmed
Q2	Property 17 mentioned here has been changed to Observation 17. Please check, and correct if necessary.

checked, no correction is necessary

Thank you for your assistance.

Theoretical Computer Science xx (xxxx) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Approximating the number of Double Cut-and-Join scenarios

Q1 István Miklós[a,b,*], Eric Tannier[c]

[a] Department of Stochastics, Rényi Institute, 1053 Budapest, Reáltanoda u. 13-15, Hungary

[b] Data Mining and Search Research Group, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, Hungary

[c] INRIA Rhône-Alpes; Université de Lyon; Université Lyon 1; CNRS, UMR5558, Laboratoire de Biométrie et Biologie Évolutive, F-69622, Villeurbanne, France

A R T I C L E I N F O

Article history:
Received 25 March 2011
Received in revised form 2 February 2012
Accepted 7 March 2012
Communicated by R. Giancarlo

Keywords:
Comparative genomics
Genome rearrangement
FPRAS
DCJ
MCMC

A B S T R A C T

The huge number of solutions in genome rearrangement problems calls for algorithms for counting and sampling in the space of solutions, rather than drawing one arbitrary scenario. A closed formula exists for counting the number of DCJ scenarios between co-tailed genomes, but no polynomial result has been published so far for arbitrary genomes. We prove here that it admits a Fully Polynomial time Randomized Approximation Scheme. We use an MCMC almost uniform sampler and prove that it converges to the uniform distribution in fully polynomial time. The MCMC can be used to quickly draw a sample of DCJ scenarios from a prescribed distribution and test some hypotheses on genome evolution.

1. Introduction

The genome rearrangement problem is a class of computational biology question which was first formulated by Sturtevant and Novitski [21]. It consists of finding a minimum size scenario of rearrangements which can explain the structural differences between two genomes. According to what a genome and what a rearrangement is, a number of variants have been studied [11], but often the number of minimum solutions is so high that finding one is almost meaningless. Evolutionary hypotheses can be tested by drawing scenarios of rearrangements like the Random Breakpoint Model [2,4] or the sizes and positions of inversions [1,8]. Drawing conclusions from one scenario only can be misleading as argued by Bergeron et al. [4], while the absence of a uniform sampler makes it impossible to state unbiased results.

Some studies have focused on the enumeration, the structure, or computation of the size of the solution space for some rearrangements and small or restricted genomes [19,6,7,18], while statistical methods sample the space when the genomes are larger [8,10,14,15,17]. For the Double Cut-and-Join (DCJ) rearrangement, a simple variant of the "reversal/translocation" model introduced by Yancopoulos et al. [23], a linear time algorithm gives one scenario [3], and it is possible to count their number in polynomial time if the genomes share the same telomeres [7,18]. Braga and Stoye [7] give an algorithm for the general case which runs in exponential time, and the complexity of the counting problem in the general case is not known.

In this paper we prove that this problem admits a Fully Polynomial time Randomized Approximation Scheme (FPRAS). This means that there is an algorithm which is polynomial in the size of the data and in $1/\epsilon$, which gives an ϵ-approximation of the number of solutions. We use for this a Markov chain Monte Carlo (MCMC) sampler, which samples the DCJ scenarios with a distribution converging to the uniform distribution in polynomial time.

The paper is organized as follows. The first section gives the definition of genomes, DCJ, scenarios, and the basic objects we will work on. Then in Section 3, we reduce the problem to pairs of genomes without common telomeres (the hard part),

* Corresponding author at: Department of Stochastics, Rényi Institute, 1053 Budapest, Reáltanoda u. 13-15, Hungary. Tel.: +36 14838304.
E-mail address: miklos.istvan.74@gmail.com (I. Miklós).

0304-3975/$ – see front matter © 2012 Published by Elsevier B.V.
doi:10.1016/j.tcs.2012.03.006

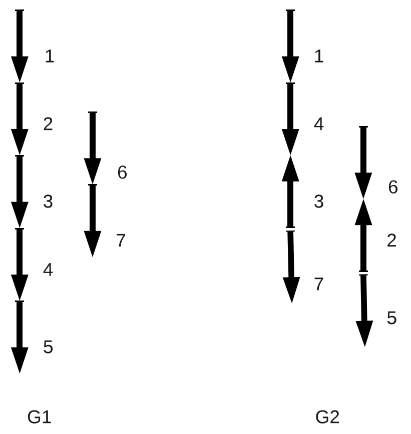

Fig. 1. An example of two genomes with 7 markers.

and show that it is equivalent in complexity to the general case. In Section 4, we prove some partial results on counting the number of DCJ scenarios. Then in Section 5, we describe the MCMC sampler and eventually prove its fast convergence with multicommodity flow techniques in Section 6.

2. Preliminaries

2.1. Genomes and rearrangements

Definition 1. A *genome* is a directed, edge-labelled graph, in which each vertex has total degree (indegree plus outdegree) 1 or 2, and each label is unique. Each edge is called a *marker*. The beginning of an edge is called its *tail* and the end of an edge is called its *head*; the joint name of heads and tails is *extremities*. Vertices with total degree 2 are called *adjacencies*, and vertices with total degree 1 are called *telomeres*.

By definition a genome is a set of disjoint paths and cycles, and neither the paths nor the cycles are necessarily directed. The components of the genome are the *chromosomes*. An example of a genome is drawn in Fig. 1. All adjacencies correspond to an unordered set of two marker extremities and telomeres to one marker extremity. For example, the $(h1, t4)$ describes the vertex of genome 2 in Fig. 1 in which the head of marker 1 and the tail of marker 4 meet, and similarly, $(h7)$ is the telomere where marker 7 ends. A genome is fully described by a list of such descriptions of adjacencies and telomeres. Two genomes with the same edge label set are *co-tailed* if they have the same telomeres. It is the case for the two genomes of Fig. 1.

Definition 2. A DCJ *or Double Cut and Join* operation transforms one genome into another by modifying the adjacencies and telomeres in one of the following 4 ways:

- Take two adjacencies (a, b) and (c, d) and create two new adjacencies (a, c) and (b, d). The adjacency descriptors are not ordered: namely, the two new adjacencies might instead be (a, d) and (b, c).
- Take an adjacency (a, b) and a telomere (c), and create a new adjacency and a new telomere from the 3 extremities: either (a, c) and (b) or (b, c) and (a).
- Take two telomeres (a) and (b), and create a new adjacency (a, b).
- Take an adjacency (a, b) and create two new telomeres (a) and (b).

Given two genomes G_1 and G_2 with the same label set, it is always possible to transform one into the other by a sequence of DCJ operations [23]. Such a sequence is called a DCJ *scenario* for G_1 and G_2. The minimum length of a scenario is called the DCJ *distance* and is denoted by $d_{\mathrm{DCJ}}(G_1, G_2)$.

Definition 3. The *Most Parsimonious* DCJ *(*MPDCJ*)* scenario problem for two genomes G_1 and G_2 is to compute $d_{\mathrm{DCJ}}(G_1, G_2)$. The #MPDCJ problem asks for the number of scenarios of length $d_{\mathrm{DCJ}}(G_1, G_2)$, denoted by #MPDCJ$(G_1, G_2)$.

For example, the DCJ distance between the two genomes of Fig. 1 is three and there are nine different most parsimonious scenarios.

MPDCJ is an optimization problem, which has a natural corresponding decision problem asking if there is a scenario with a given number of DCJ operations. So we may write that #MPDCJ $\in$ #P, which means that #MPDCJ asks for the number of witnesses of the decision problem “Is there a scenario for G_1 and G_2 of size $d_{\mathrm{DCJ}}(G_1, G_2)$?”.

We will use complexity classes in #P. FP is the class of problems in #P which have a polynomial solution. A problem in #P is #P-complete if it is in #P and any problem in #P can be reduced to it by a polynomial-time counting reduction. Two other classes, FPRAS and FPAUS, concern the approximability of the solutions.

Definition 4. A counting problem in #P is in FPRAS if there exists a randomized algorithm such that for any problem instance x, and $\epsilon, \delta > 0$, it generates an approximation $\hat{f}$ for the number of solutions f, satisfying

$$P\left(\frac{f}{1+\epsilon} \leq \hat{f} \leq f(1+\epsilon)\right) \geq 1-\delta \tag{1}$$

and the algorithm has time complexity bounded by a polynomial in $|x|$, $1/\epsilon$ and $-\log(\delta)$.

The variational distance between two discrete distributions p and q over a set X is defined as

$$d_{TV}(p,q) := \frac{1}{2}\sum_{x\in X}|p(x)-q(x)|. \tag{2}$$

Definition 5. A counting problem in #P is in FPAUS if there exists a randomized algorithm such that for any problem instance x, and $\epsilon > 0$, it generates a random witness of the corresponding decision question following a distribution p satisfying

$$d_{TV}(p,U) \leq \epsilon \tag{3}$$

where U is the uniform distribution over the space of witnesses, and the algorithm has a time complexity bounded by a polynomial in $|x|$, and $-\log(\epsilon)$.

2.2. Finding one solution is easy

Definition 6. The *adjacency graph* $G(V_1 \cup V_2, E)$ of two genomes G_1 and G_2 with the same edge label set is a bipartite multigraph with V_1 the set of adjacencies and telomeres of G_1, V_2 the set of adjacencies and telomeres of G_2. The number of edges between $u \in V_1$ and $v \in V_2$ is the number of extremities they share.

Each vertex of the adjacency graph has either degree 1 or 2, and thus, the adjacency graph falls into disjoint cycles and paths. The paths might belong to one of three types:

- odd path, containing an odd number of edges and an even number of vertices,
- even path with two endpoints in V_1: we will call them W-shaped paths,
- even path with two endpoints in V_2: we will call them M-shaped paths.

In addition, cycles with two edges and paths with one edge are called *trivial* components.

Theorem 7 (*Yancopoulos et al. [23]; Bergeron et al. [3]*).

$$d_{\mathrm{DCJ}}(G_1,G_2) = N - \left(C + \frac{I}{2}\right) \tag{4}$$

where N is the number of markers, C is the number of cycles in the adjacency graph of G_1 and G_2, and I is the number of odd paths in the adjacency graph of G_1 and G_2.

Since calculating C and I is easy, MPDCJ is clearly in P and has a linear running time algorithm. Bergeron et al. [3] also give a linear time algorithm to find one scenario of length $d_{\mathrm{DCJ}}(G_1, G_2)$.

A DCJ operation on a genome G_1 which decreases the DCJ distance to a genome G_2 is called a *sorting* DCJ for G_1 and G_2. It is possible to characterize the effect of a sorting DCJ on the adjacency graph of genomes G_1 and G_2. It acts on the vertex set V_1 and has one of the following effects [7]:

- splitting a cycle into two cycles,
- splitting an odd path into a cycle and an odd path,
- splitting an M-shaped path into a cycle and an M-shaped path,
- splitting an M-shaped path into two odd paths,
- splitting a W-shaped path into a cycle and a W-shaped path,
- merging the two ends of a W-shaped path, thus transforming it into a cycle,
- combining an M-shaped and a W-shaped path into two odd paths.

Note that trivial components are never affected by these operations, and all but the last type of DCJ operations act on a single component of the adjacency graph. The last type of DCJ acts on two components, which are M and W-shaped paths.

In this context, *sorting* a component involving adjacencies A on G_2 means applying a sequence of sorting DCJ operations to vertices of this component so that the resulting adjacency graph has only trivial components involving the adjacencies of A. In a minimum length DCJ scenario, every component is sorted independently, except M and W-shaped paths, which can be sorted together. If in a DCJ scenario one operation acts on both an M and a W-shaped path, we say that they are sorted *jointly*; otherwise we say that they are sorted *independently*.

3. Decomposing the #MPDCJ problem

The complexity status of #MPDCJ is not known. It is solvable in polynomial time when the genomes are co-tailed [7,18], or more generally in the absence of M and W-shaped paths. So the hard part is dealing with M and W-shaped paths. We show here that for the general case, we may restrict ourselves to this hard part, and suppose that there are only M and W-shaped paths in the adjacency graph.

Given two genomes G_1 and G_2 with the same label set, let AG be the adjacency graph of G_1 and G_2. Denote by G_1^* the genome which has the adjacencies and telomeres of G_1 whenever they are implied in an M or W-shaped paths of AG, and those of G_2 when they are implied in another component of AG. By definition the adjacency graph between G_1 and G_1^* has no M and W-shaped paths, while the adjacency graph between G_1^* and G_2 has only trivial components and M and W-shaped paths.

Lemma 8. $d_{\text{DCJ}}(G_1, G_2) = d_{\text{DCJ}}(G_1, G_1^*) + d_{\text{DCJ}}(G_1^*, G_2)$.

Proof. Recall the characterization of the effect of DCJ operations on the adjacency graph implies that in a minimum length DCJ scenario between G_1 and G_2, a DCJ operation never acts on two vertices involved in different components of AG, except if these two components are M and W-shaped paths. This implies that the DCJ operations of a minimum length scenario are of two kinds: those which act on the vertices involved in M and W-shaped paths, and the others.

The subsequence of DCJ operations of the first kind transforms G_1 into G_1^*, and the complementary subsequence transforms G_1^* into G_2. This proves the lemma. □

Definition 9. The #MPDCJ$_{MW}$ *problem* asks for the number of DCJ scenarios between two genomes when their adjacency graph contains only trivial components and M and W-shaped paths.

The correspondence between solutions for #MPDCJ$_{MW}$ and #MPDCJ is stated by the following lemma.

Lemma 10.

$$\#\text{MPDCJ}(G_1, G_2) = \frac{d_{\text{DCJ}}(G_1, G_2)!}{d_{\text{DCJ}}(G_1^*, G_2)! \prod_i (c_i - 1)! \prod_j (l_j - 1)!} \times \prod_i c_i^{c_i - 2} \prod_j l_j^{l_j - 2} \times \#\text{MPDCJ}_{MW}(G_1^*, G_2) \qquad (5)$$

where i indexes the cycles of the adjacency graph of G_1 and G_2, c_i denotes the number of vertices in vertex set V_1 belonging to the ith cycle, j indexes the odd paths of the adjacency graph, l_j is the number of vertices in vertex set V_1 belonging to the jth odd path.

Proof. As M and W-shaped paths and other components are always treated independently, we have

$$\#\text{MPDCJ}(G_1, G_2) = \binom{d_{\text{DCJ}}(G_1, G_2)}{d_{\text{DCJ}}(G_1^*, G_2)} \times \#\text{MPDCJ}(G_1, G_1^*) \times \#\text{MPDCJ}_{MW}(G_1^*, G_2).$$

For the genomes G_1 and G_1^*, whose adjacency graph do not contain M and W-shaped paths, we have from Braga and Stoye [7] and Ouangraoua and Bergeron [18] that

$$\#\text{MPDCJ}(G_1, G_1^*) = \prod_i c_i^{c_i - 2} \prod_j l_j^{l_j - 2} \times \frac{d_{DCJ}(G_1, G_1^*)!}{\prod_i (c_i - 1)! \prod_j (l_j - 1)!}.$$

These two equations together with Lemma 8 give the result. □

The following theorem says that the hardness of the #MPDCJ problem is the same as the #MPDCJ$_{MW}$ problem.

Theorem 11.

$$\#\text{MPDCJ}_{MW} \in \text{FP} \iff \#\text{MPDCJ} \in \text{FP} \qquad (6)$$
$$\#\text{MPDCJ}_{MW} \in \#\text{P-}\textit{complete} \iff \#\text{MPDCJ} \in \#\text{P-}\textit{complete} \qquad (7)$$
$$\#\text{MPDCJ}_{MW} \in \text{FPRAS} \iff \#\text{MPDCJ} \in \text{FPRAS} \qquad (8)$$
$$\#\text{MPDCJ}_{MW} \in \text{FPAUS} \iff \#\text{MPDCJ}. \in \text{FPAUS} \qquad (9)$$

Proof. Both the multinomial factor and the two products in Eq. (5) can be calculated in polynomial time. Thus the transformation between the solutions to the two different counting problems is a single multiplication or division by an exactly calculated number. This proves that #MPDCJ$_{MW}$ is in FP if and only if #MPDCJ is in FP, as well as #MPDCJ$_{MW}$ is in #P-complete if and only if #MPDCJ is in #P-complete.

Such a multiplication and division keeps the relative error when the solution of one of the problems is approximated. This proves that #MPDCJ$_{MW}$ is in FPRAS if and only if #MPDCJ is in FPRAS.

Concerning the last equivalence, the $\Leftarrow$ part is trivial because #MPDCJ$_{MW}$ is a particular case of #MPDCJ. Now we prove that #MPDCJ$_{MW} \in$ FPAUS $\Rightarrow$ #MPDCJ $\in$ FPAUS. Suppose an FPAUS exists for #MPDCJ$_{MW}$, and let G_1 and G_2 be two arbitrary genomes. The following algorithm gives a FPAUS for #MPDCJ.

- Draw a DCJ scenario between G_1^* and G_2 following a distribution p satisfying

  $$d_{TV}(p, U) \le \epsilon$$

  where U is the uniform distribution over all possible most parsimonious DCJ scenarios between G_1^* and G_2.
- Generate a DCJ scenario between G_1 and G_1^*, following the uniform distribution. This scenario can be sampled sharply uniformly in polynomial time: (1) there are only cycles and odd paths in the adjacency graph of G_1 and G_1^*, so the number of scenarios can be calculated in polynomial time; (2) there is a polynomial number of sorting DCJ steps on each component, and a sorting DCJ operation results in an adjacency graph that also only has cycles and odd paths.
- Draw a sequence of 0s and 1s, containing $d_{\mathrm{DCJ}}(G_1^*, G_2)$ 1s and $d_{\mathrm{DCJ}}(G_1, G_1^*)$ 0s, uniformly from all $\binom{d_{\mathrm{DCJ}}(G_1,G_2)}{d_{\mathrm{DCJ}}(G_1^*,G_2)}$ such sequences.
- Merge the two paths constructed at the two first steps, according to the drawn sequence of 0s and 1s.

Note that the DCJ scenario obtained transforms G_1 into G_2. Let us denote the distribution of paths generated by this algorithm by p', and the uniform distribution over all possible DCJ scenarios between G_1 and G_2 by U'. Let X_s denote the set of all possible scenarios drawn by the above algorithm using a specific scenario s between G_1^* and G_2. Then

$$\sum_{s' \in X_s} |p'(s') - U'(s')| = |p(s) - U(s)|. \tag{10}$$

Using Eq. (10) we get that

$$d_{TV}(p', U') = \frac{1}{2} \sum_s \sum_{s' \in X_s} |p'(s') - U'(s')| = \frac{1}{2} \sum_s |p(s) - U(s)| = d_{TV}(p, U). \tag{11}$$

This proves that the above algorithm is an FPAUS for #MPDCJ, proving the left-to-right direction in Eq. (9). □

We will show that #MPDCJ$_{MW}$ is in FPAUS, thus #MPDCJ is in FPAUS. As MPDCJ is a self-reducible problem (there is a polynomial time reduction of the decision problem to the search problem), the FPAUS implies the existence of an FPRAS [13]. The FPAUS algorithm for #MPDCJ$_{MW}$ will be defined via a rapidly mixing Markov chain. And first, we have to recall or prove some properties on the number of independent and joint sortings of M and W-shaped paths.

4. Independent and joint sorting of *M* and *W*-shaped paths

Our goal is to show that the number of DCJ scenarios in which an M and a W-shaped path are sorted independently is a significant fraction of the total number of scenarios sorting these M and W-shaped paths (independently or jointly). We build on the following results by Braga and Stoye [7].

Theorem 12 (*Braga and Stoye [7]*). • *The number of minimum length* DCJ *scenarios sorting a cycle with $k > 1$ vertices in G_1 is k^{k-2}.*
- *The number of minimum length* DCJ *scenarios sorting an odd path with $k > 1$ vertices in G_1 is k^{k-2}.*
- *The number of minimum length* DCJ *scenarios sorting a W-shaped path with $k > 1$ vertices in G_1 is k^{k-2}.*
- *The number of minimum length* DCJ *scenarios sorting an M-shaped path with $k > 0$ vertices in G_1 is $(k+1)^{k-1}$.*

Theorem 13. *The number of* DCJ *scenarios that independently sort a W and an M-shaped path is $\binom{k_1+k_2-1}{k_1-1} k_1^{k_1-2}(k_2+1)^{k_2-1}$ where k_1 and k_2 are the number of vertices of G_1 in the W and M-shaped paths, respectively.*

Proof. It is a consequence of the previous theorem. The W-shaped path is sorted in $k_1 - 1$ operations, and the M-shaped path is sorted in k_2 operations. Thus there are $\binom{k_1+k_2-1}{k_1-1}$ ways to merge two scenarios. □

Theorem 14. *The number of* DCJ *scenarios that jointly sort a W and an M-shaped path is less than $2(k_1+k_2)^{k_1+k_2-2}$, where k_1 and k_2 are the number of vertices of G_1 in the W and M-shaped paths, respectively.*

Proof. Let t_1^W and t_2^W be the two telomeres of the W-shaped path, and t_1^M and t_2^M be the two telomeres of the M-shaped path. Let G_1' and G_2' be constructed from genomes G_1 and G_2 by adding a gene g, with extremities g^h and g^t, and replacing the telomeres of G_1 by adjacencies (t_1^W, g^t), (t_2^W, g^h) and the telomeres of G_2 by adjacencies (t_1^M, g^t), (t_2^M, g^h). In addition let G_1'' and G_2'' be constructed from G_1 and G_2 also by adding a gene g, and replacing the telomeres of G_1 by adjacencies (t_1^W, g^t), (t_2^W, g^h) and the telomeres of G_2 by adjacencies (t_1^M, g^h), (t_2^M, g^t). In both cases the M and W-shaped paths in G_1 and G_2 are transformed into a cycle with $k_1 + k_2$ adjacencies in both genomes. Call the cycles C' for the first case, and C'' for the second.

We prove that any scenario that jointly sort the W and M-shaped paths has a corresponding distinct scenario either sorting the cycle C' or sorting the cycle C''. This proves the theorem, because there are $(k_1+k_2)^{k_1+k_2-2}$ scenarios sorting each cycle.

A scenario jointly sorting the M and W-shaped paths can be cut into two parts: the first contains DCJ operations which act only on the M or only on the W-shaped path; the second part starts with a DCJ operation transforming an M and a W-shaped path into two odd paths, and continues with operations independently sorting the two odd paths.

In the first part, either a DCJ operation acts on two adjacencies of the M or W-shaped path, and the corresponding operation acts on the same two adjacencies on C' or C'', or it acts on an adjacency and a telomere of the W-shaped path, and the corresponding operation acts on two adjacencies of C' or C'', one of them containing an extremity of g. So there is a correspondence between being a telomere in the W-shaped path, and being adjacent to an extremity of g in C' or C''.

Now the corresponding operation of the DCJ transforming the two paths into two odd paths has to create two cycles from C' or C''. Choose C' or C'' so that it is the case. Now sorting an odd path exactly corresponds to sorting a cycle, by replacing being a telomere in the path by being adjacent to an extremity of g in the cycle.

So two different scenarios jointly sorting the M and W-shaped paths correspond to two different scenarios sorting either C' or C''. Then the number of scenarios jointly sorting the M and W-shaped paths is less than $2(k_1+k_2)^{k_1+k_2-2}$. □

Theorem 15. *Let $T(k_1,k_2)$ denote the number of* DCJ *scenarios jointly sorting a W and an M-shaped path with respectively k_1 and k_2 vertices G_1. Let $I(k_1,k_2)$ denote the number scenarios independently sorting the same paths. We have that*

$$\frac{T(k_1,k_2)}{I(k_1,k_2)} = O\left(\frac{k_1^{1.5}k_2^{1.5}}{(k_1+k_2)^{1.5}}\right) \tag{12}$$

$$\frac{I(k_1,k_2)}{T(k_1,k_2)} = O\left(k_1+k_2\right). \tag{13}$$

Proof. To prove Eq. (12) it is sufficient to show that

$$\frac{2(k_1+k_2)^{k_1+k_2-2}}{\binom{k_1+k_2-1}{k_1-1}k_1^{k_1-2}(k_2+1)^{k_2-1}} = O\left(\frac{k_1^{1.5}k_2^{1.5}}{(k_1+k_2)^{0.5}}\right). \tag{14}$$

Using Stirling's formula, we get on the left hand side of Eq. (14)

$$\frac{2\sqrt{2\pi(k_1-1)}\left(\frac{k_1-1}{e}\right)^{k_1-1}\sqrt{2\pi(k_2)}\left(\frac{k_2}{e}\right)^{k_2}(k_1+k_2)^{k_1+k_2-2}}{\sqrt{2\pi(k_1+k_2-1)}\left(\frac{k_1+k_2-1}{e}\right)^{k_1+k_2-1}k_1^{k_1-2}(k_2+1)^{k_2-1}} \tag{15}$$

After simplifications and algebraic rearrangement, we get

$$2\sqrt{\frac{2\pi(k_1-1)k_2}{k_1+k_2-1}}\left(\frac{k_1+k_2}{k_1+k_2-1}\right)^{k_1+k_2-1}\left(\frac{k_1-1}{k_1}\right)^{k_1-1}\left(\frac{k_2}{k_2+1}\right)^{k_2}\left(\frac{k_1(k_2+1)}{k_1+k_2}\right). \tag{16}$$

from which Eq. (14) follows with applying $(1+1/n)^n$ tends to e, and $(1-1/n)^n$ tends to $1/e$.

To prove Eq. (13) consider the subset of DCJ scenarios jointly sorting the W and M-shaped paths, and starting with a DCJ operation which acts on a telomere of the W-shaped path, and on an adjacency which is link with a telomere of the M-shaped path. The result is two odd paths with respectively k_1 and k_2 adjacencies and telomeres in G_1. They can be sorted in respectively k_1-1 and k_2-1 steps, in $k_1^{k_1-2}$ and $k_2^{k_2-2}$ different ways. Since we can combine any two particular solutions in $\binom{k_1+k_2-2}{k_1-1}$ ways, $\frac{I(k_1,k_2)}{T(k_1,k_2)}$ is bounded by

$$\frac{\binom{k_1+k_2-1}{k_1-1}k_1^{k_1-2}(k_2+1)^{k_2-1}}{\binom{k_1+k_2-2}{k_1-1}k_1^{k_1-2}k_2^{k_2-2}}. \tag{17}$$

After minor algebraic simplification, this expression is equal to

$$\frac{k_1+k_2+1}{k_2}\left(1+\frac{1}{k_2}\right)^{k_2-1}k_2, \tag{18}$$

which is clearly $O(k_1+k_2)$. □

5. The Markov chain on DCJ scenarios

Assume that there are n W-shaped paths and m M-shaped paths, and consider the complete bipartite graph $K_{n,m}$. Let $\mathcal{M}$ be a matching of $K_{n,m}$, which might range from the empty graph up to any maximum matching. A DCJ scenario is said to be $\mathcal{M}$-compatible when an M-shaped and a W-shaped path are sorted jointly if and only if they are connected by an edge of $\mathcal{M}$.

We denote by $\{P_i\}_i$ set of degree 0 vertices in $\mathcal{M}$, and by $\{M_iW_i\}_i$ the set of edges in $\mathcal{M}$. Let $l(P_i)$ be the minimum length of a DCJ scenario independently sorting P_i, and $l(M_iW_i)$ be the minimum length of a DCJ scenario jointly sorting M_i and W_i. We can calculate $N(M_i, W_i)$, the number of joint sortings of M_i and W_i, in polynomial time [7]. Denote by $N(P_i)$ the number of independent sortings of a path P_i. The number of $\mathcal{M}$-compatible scenarios is

$$f(\mathcal{M}) = \left(\frac{(\sum_i l(M_iW_i) + \sum_i l(P_i))!}{l(M_i, W_i)!, \ldots, l(P_i)!} \right) \Pi_i N(M_i, W_i) \Pi_i N(P_i),$$

and we can compute it in polynomial time. Define a distribution θ over the set of all matchings of the complete bipartite graph $K_{n,m}$ as

$$\theta(\mathcal{M}) \propto f(\mathcal{M}) \tag{19}$$

We first show that sampling DCJ scenarios from the uniform distribution is equivalent to sampling matchings of $K_{n,m}$ from the distribution θ.

Theorem 16. *Let a distribution q over the scenarios of n W-shaped paths and m M-shaped paths be defined by the following algorithm.*

- *Draw a random matching $\mathcal{M}$ of $K_{n,m}$ following a distribution p.*
- *Draw a random $\mathcal{M}$-compatible* DCJ *scenario from the uniform distribution of all $\mathcal{M}$-compatible ones.*

Then

$$d_{TV}(p, \theta) = d_{TV}(q, U) \tag{20}$$

where θ is the distribution defined in Eq. (19), and U denotes the uniform distribution over all DCJ *scenarios.*

Proof.

$$d_{TV}(q, U) = \frac{1}{2} \sum_{x \text{ scenario}} |q(x) - U(x)|.$$

We may decompose this sum into

$$\frac{1}{2} \sum_{(\mathcal{M} \text{ matching of } K_{n,m})} \sum_{(x\ \mathcal{M}-\text{compatible scenario})} |q(x) - U(x)|$$

$\sum_{(x\ \mathcal{M}-\text{compatible scenario})} q(x)$ is $p(\mathcal{M})$ since x is drawn uniformly among the scenarios compatible with $\mathcal{M}$, and $\sum_{(x\ \mathcal{M}-\text{compatible scenario})} U(x)$ is $\theta(\mathcal{M})$. Furthermore, both $q(x)$ and $U(x)$ are constant for a particular matching $\mathcal{M}$, thus

$$\frac{1}{2} \sum_{(\mathcal{M} \text{ matching of } K_{n,m})} \sum_{(x\ \mathcal{M}-\text{compatible scenario})} |q(x) - U(x)| = \frac{1}{2} \sum_{(\mathcal{M} \text{ matching of } K_{n,m})} |p(\mathcal{M}) - \theta(\mathcal{M})| = d_{TV}(p, \theta) \tag{21}$$

yielding the result. $\square$

So we are going to define an MCMC on matchings of $K_{n,m}$ converging to θ. The rapid convergence of this MCMC will imply that #MPDCJ$_{WM}$ admits an FPAUS, and hence #MPDCJ $\in$ FPAUS, and then #MPDCJ $\in$ FPRAS. The primer Markov chain walks on the matchings of $K_{n,m}$ and is defined by the following steps: suppose the current state is a matching $\mathcal{M}$, and

- with probability 1/2, the next state of the Markov chain is the current state $\mathcal{M}$;
- with probability 1/2, draw a random $i \sim U[1, n]$ and $j \sim U[1, m]$; if $ij \in \mathcal{M}$, then remove ij from $\mathcal{M}$; else if $\deg_{\mathcal{M}}(i) = 0$ and $\deg_{\mathcal{M}}(j) = 0$, then add ij to $\mathcal{M}$.

It is easy to see that this Markov chain is irreducible and aperiodic. We apply the standard Metropolis–Hastings algorithm on this chain [16], namely, when we are in state $\mathcal{M}$, we propose the next state $\mathcal{M}_{new}$ according to the primer Markov chain, and accept the proposal with probability

$$\min \left\{ 1, \frac{f(\mathcal{M}_{new})}{f(\mathcal{M})} \right\}. \tag{22}$$

The obtained Markov chain is reversible and converges to the distribution θ defined in Eq. (19). Furthermore, this is a lazy Markov chain (due to remaining in the current state with at least probability 1/2 in each step), providing that all its eigenvalues are positive real numbers (see for example [22]).

An important property of this Markov chain is that

Observation 17. *The non zero transition probabilities as well as their inverses are polynomially bounded.*

Indeed, the transition probability from $\mathcal{M}$ to $\mathcal{M}_{new}$, if non zero, is at least

$$\frac{1}{2 \times n \times m} \frac{f(\mathcal{M}_{new})}{f(\mathcal{M})}.$$

$\mathcal{M}$ and $\mathcal{M}_{new}$ vary by at most one edge M_iW_i, and on this edge, according to Theorem 15, the ratio of number of scenarios jointly and independently sorting M_i and W_i is polynomial. Furthermore, the combinatorial factors appearing in $f(\mathcal{M})$ and $f(\mathcal{M}_{new})$ due to merging the sorting steps on different components are the same. So $\frac{f(\mathcal{M}_{new})}{f(\mathcal{M})}$ as well as its inverse are polynomially bounded.

We now prove the rapid convergence of this Markov chain using a Multicommodity flow technique.

6. Fast convergence of the MCMC

In this section, we prove that the constructed Markov chain rapidly converges to its stationary distribution. From its construction, this distribution is θ as defined in Eq. (19). For this, we use the Multicommodity flow technique developed by Sinclair [20]. We denote by $T(\cdot|\cdot)$ the transition probabilities of the Markov chain.

The Markov graph $G(V, E)$ of our Markov chain on matchings is a directed graph whose vertices are the states of the Markov chain, and there is an arc between two states u and v if there is a transition from u to v. We define the load of an arc $e = (u, v)$ as

$$Q(e) := T(u|v)\theta(u). \tag{23}$$

A path system Γ in a Markov graph is a set of distributions of paths for each ordered pair (x, y), $x, y \in V$. We will denote the distribution of paths defined for (x, y) by $\Gamma_{x,y}$, and then

$$\Gamma := \cup_{(x,y)\in V\times V} \Gamma_{x,y}. \tag{24}$$

Let $p_{(x,y)}(\gamma)$ denote the probability of a path γ in the distribution $\Gamma_{x,y}$ of a path system Γ.
Let

$$\kappa_\Gamma := \max_{e=(u,v)\in E} \sum_{(x,y)\in V\times V} \sum_{\gamma\in\Gamma_{(x,y)}:e\in\gamma} \theta(x)\theta(y)p_{x,y}(\gamma)\frac{|\gamma|}{Q(e)} \tag{25}$$

Theorem 18 (*Sinclair [20]*)**.** *For any path system Γ,*

$$\frac{1}{1-\lambda_2} \le \kappa_\Gamma, \tag{26}$$

where λ_2 is the second eigenvalue of the transition matrix of the Markov chain.

This yields that if κ_Γ is bounded by a polynomial in the size of the data, then the Markov chain can be used for an FPAUS, based on the following theorem.

Let p_i^n denote the distribution of an irreducible, aperiodic, reversible Markov chain after n steps starting at a particular state i, and θ its equilibrium distribution. Let the relaxation time be defined as

$$\tau_i(\epsilon) := \min\left\{n_0 \in \mathbb{N} : d_{TV}(p_i^n, \theta) \le \epsilon \ \forall n \ge n_0\right\}. \tag{27}$$

Theorem 19. *[9]*

$$\tau_i(\epsilon) \le \frac{1}{1-\rho}(\log(1/\theta(i)) + \log(1/\epsilon)) \tag{28}$$

where ρ is the second largest eigenvalue modulus, i.e.*, the maximum of the second largest eigenvalue and the absolute value of the smallest eigenvalue (a reversible Markov chain has only real eigenvalues).*

So to prove that the Markov chain we defined on bipartite matchings has a polynomial relaxation time, we need to construct a path system Γ on the set of matchings of $K_{n,m}$, such that κ_Γ is bounded by a polynomial in N, the number of markers in G_1 and G_2.

In our case the path system between two matchings $\mathcal{X}$ and $\mathcal{Y}$ is a unique path with probability 1. Here is how we construct it.

Fix a total order on the vertex set of $K_{n,m}$. Take the symmetric difference of $\mathcal{X}$ and $\mathcal{Y}$, denoted by $\mathcal{X}\Delta\mathcal{Y}$. It is a set of disjoint paths and cycles. Define an order on the components of $\mathcal{X}\Delta\mathcal{Y}$, such that a component C is smaller than a component D if the smallest vertex in C is smaller than the smallest vertex in D. Now we orient each component in the following way: the beginning of each path is its extremity with the smaller vertex. The starting vertex of a cycle is its smallest vertex, and the direction is going towards its smaller neighbour.

We transform $\mathcal{X}$ to $\mathcal{Y}$ by visiting the components of $\mathcal{X}\Delta\mathcal{Y}$ in increasing order. Let the current component be C, and the current matching is $\mathcal{Z}$ (at first $\mathcal{Z} = \mathcal{X}$). If C is a path or cycle starting with an edge in $\mathcal{X}$, then the transformation steps are the following: delete the first edge of C from $\mathcal{Z}$, delete the third edge of C from $\mathcal{Z}$, add the second edge of C to $\mathcal{Z}$, delete the 5th edge of C from $\mathcal{Z}$, add the 4th edge of C to $\mathcal{Z}$, etc.

If C is a path or cycle starting with an edge in $\mathcal{Y}$, then the transformation steps are the following: delete the second edge of C from $\mathcal{Z}$, add the first edge of C to $\mathcal{Z}$, delete the 4th edge of C from $\mathcal{Z}$, add the third edge of C to $\mathcal{Z}$, etc.

This path has length at most nm, and κ_Γ can be written:

$$\kappa_\Gamma \le nm \max_{e=(u,v)\in E} \sum_{(x,y)\in V\times V: e\in\Gamma_{x,y}} \frac{\theta(x)\theta(y)}{Q(e)}.$$

By Observation 17, the inverse of the transition probabilities is bounded by a polynomial in N, so we get Q2

$$\kappa_\Gamma \le O(\text{poly}(N)) \max_{e=(u,v)\in E} \sum_{(x,y)\in V\times V: e\in\Gamma_{x,y}} \frac{\theta(x)\theta(y)}{\theta(u)}. \quad (29)$$

We then have to show that $\sum \frac{\theta(x)\theta(y)}{\theta(u)}$ can be bounded by a polynomial in N. Let $\mathcal{Z} \to \mathcal{Z}'$ be an edge on the path from $\mathcal{X}$ to $\mathcal{Y}$. We define

$$\widehat{\mathcal{M}} := \mathcal{X}\Delta\mathcal{Y}\Delta\mathcal{Z}. \quad (30)$$

Lemma 20. *The couple $\widehat{\mathcal{M}}$ and $\mathcal{Z} \to \mathcal{Z}'$ determines $\mathcal{X}$ and $\mathcal{Y}$.*

Proof. It is obvious that

$$\widehat{\mathcal{M}}\Delta\mathcal{Z} = \mathcal{X}\Delta\mathcal{Y} \quad (31)$$

hence, $\mathcal{Z}$ and $\widehat{\mathcal{M}}$ determine the symmetric difference of $\mathcal{X}$ and $\mathcal{Y}$. From the transition $\mathcal{Z} \to \mathcal{Z}'$, we can trace back which transition steps have been already made in the following way. The order of the components of $\mathcal{X}\Delta\mathcal{Y}$ is determined, and from the transition $\mathcal{Z} \to \mathcal{Z}'$ we know the current component. We also know the beginning and the direction of the component, be it either a path or a cycle, hence, we know which edges have been changed in the component so far, and which ones not yet. From these, we can reconstruct $\mathcal{X}$ and $\mathcal{Y}$. □

Lemma 21. *A matching can be obtained from $\widehat{\mathcal{M}}$ by deleting at most two edges.*

Proof. On each component in $\mathcal{X}\Delta\mathcal{Y}$, we delete at most two edges before putting back one. Hence $\widehat{\mathcal{M}}$ contains at most either 4 consecutive edges along a path or 2 pair of edges, and all remaining edges are independent. Therefore it is sufficient to delete at most two edges from $\widehat{\mathcal{M}}$ to get a matching. □

Denote this matching by $\widetilde{\mathcal{M}}$.

Lemma 22.

$$\frac{\theta(\mathcal{X})\theta(\mathcal{Y})}{\theta(\mathcal{Z})} = O(\text{poly}(N))\theta(\widetilde{\mathcal{M}}). \quad (32)$$

Proof. We prove that

$$\frac{f(\mathcal{X})f(\mathcal{Y})}{f(\mathcal{Z})f(\widetilde{\mathcal{M}})} = O(\text{poly}(N)). \quad (33)$$

It proves the lemma, as $\theta(\cdot)$ and $f(\cdot)$ differ only by a normalizing constant. $\widetilde{\mathcal{M}}\Delta\mathcal{Z}$ differs at most in two edges from $\mathcal{X}\Delta\mathcal{Y}$. These edges appear in $\mathcal{X}\Delta\mathcal{Y}$, but not in $\widetilde{\mathcal{M}}\Delta\mathcal{Z}$. The two vertices of any missing edges correspond to components which are independently sorted either in $\mathcal{Z}$ or $\widetilde{\mathcal{M}}$, but jointly in either $\mathcal{X}$ or $\mathcal{Y}$. Amongst these two vertices, one of them correspond to a W-shaped component $\mathcal{A}$, the other to an M-shaped component $\mathcal{B}$. Let k_1 be the number of adjacencies and telomeres of G_1 in $\mathcal{A}$, and k_2 the number of adjacencies and telomeres of G_1 in $\mathcal{B}$. The ratio on the left-hand side of Eq. (33) due to such difference is

$$\frac{T(k_1,k_2)}{(k_1+k_2+1)!} \Big/ \frac{I(k_1)I'(k_2)}{k_1!(k_2+1)!} \quad (34)$$

where $I(x)$ denotes the independent sorting of a W-shaped component of size x, and $I'(x)$ denotes the independent sorting of an M-shaped component of size x. However, it is polynomially bounded, since

$$\frac{I(k_1)I'(k_2)}{\binom{k_1+k_2+1}{k_1}} = I(k_1,k_2) \quad (35)$$

and we can apply Theorem 15. □

These results together lead to the following theorem:

Theorem 23. *The Metropolis–Hastings Markov chain on the matchings defined above converges rapidly to θ.*

Proof. From Lemma 22, Eq. (29) may be written

$$\kappa_\Gamma \leq O(\text{poly}(N)) \max_{e=(u,v)\in E} \sum_{(x,y)\in V\times V: e\in\Gamma_{x,y}} \theta(\widetilde{\mathcal{M}}).$$

By Lemmas 20 and 21, a matching $\widetilde{\mathcal{M}}$ may appear only a polynomial number of times in this sum. So

$$\kappa_\Gamma \leq O(\text{poly}(N)) \sum_{\widetilde{\mathcal{M}}} \theta(\widetilde{\mathcal{M}}),$$

and as $\sum_{\widetilde{\mathcal{M}}} \theta(\widetilde{\mathcal{M}}) = 1$, κ_Γ is bounded by a polynomial in N. This proves the theorem. □

Using this result, we can prove the following theorem

Theorem 24. #MPDCJ$_{MW}$ $\in$ FPAUS

Proof. The above defined Markov chain on partial matchings is an aperiodic, irreducible and reversible Markov chain, with only positive eigenvalues. Furthermore, a step can be performed in running time polynomial with the size of the graph. For any start state i, $\log(1/\theta(i))$ is polynomially bounded with the size of the corresponding genomes G_1^* and G_2, since there are $O(N^2)$ DCJ operations, the length of the DCJ paths is less than N, thus the number of sorting DCJ paths are $O(N^{2N})$, and the inverse of the probability of any partial matching is less than this. Thus, the relaxation time is polynomial in both N and $\log(1/\epsilon)$, according to Theorem 19. This means that in fully polynomial running time (polynomial both in N and $-\log(\epsilon)$) a random partial matching can be generated from a distribution p satisfying

$$d_{TV}(p,\theta) \leq \epsilon. \tag{36}$$

But then a random DCJ path can be generated in fully polynomial running time following a distribution q satisfying

$$d_{TV}(q,U) \leq \epsilon \tag{37}$$

according to Theorem 16. This is what we wanted to prove. □

Now we are ready to conclude by our main theorem:

Theorem 25. #MPDCJ $\in$ FPRAS

Proof. #MPDCJ$_{MW}$ $\in$ FPAUS according to Theorem 24. Then #MPDCJ $\in$ FPAUS according to Theorem 11. Since #MPDCJ is a self-reducible counting problem, it is in FPRAS [13]. □

7. Conclusion

Sampling from reversal scenarios has been conjectured to be #P-complete [17], but almost all counting problems on genome rearrangement scenarios have an open complexity status (the only exception we are aware of is counting tandem duplication and random loss scenarios, which is equivalent to counting the number of riffle shuffles of a deck of card [5], and is given a solution in [12]). We conjecture that sampling from DCJ scenarios is also #P-complete, and we proved in this paper that it admits an FPRAS. Braga and Stoye [7] prove that altering three consecutive steps in a DCJ sorting path is sufficient to get an irreducible Markov chain. Such a Markov chain can be also used in a Metropolis–Hastings algorithm to converge to the uniform distribution of all DCJ sorting scenarios. We conjecture that it is also a rapidly mixing Markov chain, which would give a more direct proof of the results in this paper.

This complexity result allows the device of theoretically grounded samplers in the space of genome rearrangements, such as the one used in [17]. Its efficiency makes it practical to use in rigorous studies of modes of evolution in eukaryotes. Miklós and Tannier [17] use the Braga and Stoye [7] sampler altering three consecutive DCJs in a scenario to sample among the DCJ scenario space. By a parallel tempering technique, Miklós and Tannier [17] also sample in reversal and translocation scenarios, which are a subset of DCJ scenarios, and apply the method on mammalian and yeast data. It is thus a bioinformatics application using the sampling of genome rearrangement scenarios, in which we compute the rates of different kinds of rearrangements. This result, as well as a rigorous breakpoint re-use computation [4], is inaccessible without methods to sample in the space of scenarios. So the next open problems for the combinatorics of genome rearrangements, which take a very small place in the first exhaustive review of the field [11], will probably be related to dealing with the entire solution space instead of only one representative.

Acknowledgements

We are grateful to the reviewers for their numerous and careful suggestions in improving the presentation of the article. E.T. was funded by the "Investissement d'Avenir" programme, through the Agence Nationale pour la Recherche (ANCESTROME ANR-10-BINF-01-01). I.M. was supported by OTKA PD 84297.

References

[1] Y. Ajana, J. Lefebvre, E. Tillier, N. El-Mabrouk, Exploring the set of all minimal sequences of reversals — an application to test the replication-directed reversal hypothesis, in: Algorithms in Bioinformatics, WABI'02, in: LNCS, vol. 2452, 2002, pp. 300–315.
[2] M.A. Alekseyev, P.A. Pevzner, Comparative genomics reveals birth and death of fragile regions in mammalian evolution, Genome Biol 11 (11) (2010) R117.
[3] A. Bergeron, J. Mixtacki, J. Stoye, A unifying view of genome rearrangements, in: LNCS, vol. 4175, 2006, pp. 163–173.
[4] A. Bergeron, J. Mixtacki, J. Stoye, On computing the breakpoint reuse rate in rearrangement scenarios (preview), in: Proceedings of RECOMB-CG 2008, in: LNBI, vol. 5267, 2008, pp. 226–240.
[5] M. Bernt, K.-Y. Chen, M.-C. Chen, A.-C. Chu, D. Merkle, H.-L. Wang, K.-M. Chao, M. Middendorf, Finding all sorting tandem duplication random loss operations, Journal of Discrete Algorithms 9 (1) (2011) 32–48.
[6] M. Braga, M.-F. Sagot, C. Scornavacca, E. Tannier, Exploring the solution space of sorting by reversals with experiments and an application to evolution, IEEE–ACM Transactions on Computational Biology and Bioinformatics 5 (2008) 348–356.
[7] M. Braga, J. Stoye, The solution space of sorting by dcj, Journal of Computational Biology 17 (9) (2010) 1145–1165.
[8] A. Darling, I. Miklós, M. Ragan, Dynamics of genome rearrangement in bacterial populations, PLoS Genetics 4 (7) (2008) e1000128.
[9] P. Diaconis, D. Strock, Geometric bounds for eigenvalues of markov chains, The Annals of Applied Probability 1 (1) (1991) 36–61.
[10] R. Durrett, R. Nielsen, T. York, Bayesian estimation of genomic distance, Genetics 166 (2004) 621–629.
[11] G. Fertin, A. Labarre, I. Rusu, E. Tannier, S. Vialette, Combinatorics of Genome Rearrangements, MIT Press, 2009.
[12] C. Grinstead, S. J L, Introduction to Probability, American Mathematical Society, 2006.
[13] M. Jerrum, L. Valiant, V. Vazirani, Random generation of combinatorial structures from a uniform distribution, Theoretical Computer Science 43 (1986) 169–188.
[14] B. Larget, D. Simon, B. Kadane, Bayesian phylogenetic inference from animal mitochondrial genome arrangements, Journal of the Royal Statistical Society. Series B 64 (4) (2002) 681–695.
[15] B. Larget, D. Simon, J. Kadane, D. Sweet, A bayesian analysis of metazoan mitochondrial genome arrangements, Molecular Biology Evolution 22 (3) (2005) 485–495.
[16] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equations of state calculations by fast computing machines, Journal of Chemical Physics 21 (6) (1953) 1087–1091.
[17] I. Miklós, E. Tannier, Bayesian sampling of genome rearrangement scenarios via dcj, Bioinformatics 26 (2010) 3012–3019.
[18] A. Ouangraoua, A. Bergeron, Combinatorial structure of genome rearrangements scenarios, Journal of Computational Biology 17 (9) (2010) 1129–1144.
[19] A. Siepel, An algorithm to enumerate sorting reversals for signed permutations, Journal of Computational Biology 10 (2003) 575–597.
[20] A. Sinclair, Improved bounds for mixing rates of markov chains and multicommodity flow, Combinatorics, Probability and Computing 1 (1992) 351–370.
[21] A. Sturtevant, E. Novitski, The homologies of chromosome elements in the genus drosophila, Genetics 26 (1941) 517–541.
[22] S. Vempala, Geometric random walks: a survey, Combinatorial and Computational Geometry 52 (2005) 573–612.
[23] S. Yancopoulos, O. Attie, R. Friedberg, Efficient sorting of genomic permutations by translocation, inversion and block interchange, Bioinformatics 21 (16) (2005) 3340–3346.