Approximating the number of Double Cut-and-Join scenarios

István Miklós[1,2] and Eric Tannier[3]

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

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

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

Abstract

The huge number of solutions in genome rearrangement problems calls for algorithms for counting and sampling in the space of solutions, and not only draw one arbitrary scenario. A closed formula exists for DCJ operations and co-tailed genomes, but no polynomial result has been published so far for arbitrary linear genomes and counting has been conjectured $\#P$-complete. We prove it admits a Fully Polynomial time Randomized Approximation Scheme. We use a MCMC almost uniform sampler and prove it converges to the uniform distribution in fully polynomial time.

Keywords: comparative genomics, genome rearrangement, FPRAS, DCJ, MCMC

1. Introduction

The genome rearrangement problem is a class of computational biology question which was introduced in Sturtevant and Novitski (1941). It consists of finding a minimum size scenario of rearrangements that 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 (Fertin et al., 2009), but often the number of minimum solutions is so high that finding one is almost meaningless. Some studies have focused on the enumeration, the structural classification, or computation of the size of the solution space for some rearrangements and small or restricted genomes (Braga et al., 2008;

Braga and Stoye, 2010; Ouangraoua and Bergeron, 2010), while statistical methods sample the space when the genomes are larger (Darling et al., 2008; Durrett et al., 2004; Larget et al., 2002, 2005; Miklós and Tannier, 2010). For the Double Cut-and-Join (DCJ) rearrangement, introduced by Yancopoulos et al. (2005), a linear algorithm gives one scenario (Bergeron et al., 2006), and it is possible to count their number in polynomial time if the genomes share the same telomeres (Braga and Stoye, 2010; Ouangraoua and Bergeron, 2010). Braga and Stoye (2010) 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 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 solution. 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, 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), and showing 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 incoming and outgoing degree at most 1, and each label is unique. Each edge is called a* marker*. The beginning of an edge is called* tail*, the end of an edge is called* head*, the joint name of heads and tails is* extremities*. The vertices with degree* 2 *are called* adjacencies*, the vertices with 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 genome is drawn on Figure 1. All adjacencies correspond to two marker extremities and telomeres to one.

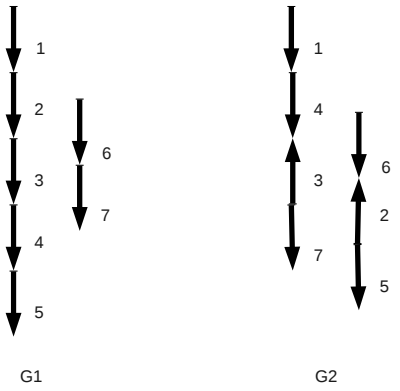

Figure 1: An example of two genomes with 7 markers.

For example, $(h1, t4)$ describes the vertex of genome 2 in Figure 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 label set are *co-tailed* if they have the same telomeres. It is the case for the two genomes of Figure 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 be also* (a, d) *and* (b, c)*, too.*
- *Take an adjacency* (a, b) *and a telomere* (c)*, and create a new adjacency and 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)*.*

Definition 3. *The* Most Parsimonious DCJ (MPDCJ) *problem is the following: given two genomes G_1 and G_2 with the same label set, what is the minimum number of DCJ operations necessary to transform G_1 into G_2. This number is called the DCJ distance and is denoted by $d_{DCJ}(G_1, G_2)$. Similarly, the* #MPDCJ *problem asks for the number of shortest DCJ scenarios transforming G_1 into G_2. The number of solutions will be denoted by $\#MPDCJ(G_1, G_2)$.*

For example, the DCJ distance between the two genomes of Figure 1 is three (exercise left to the reader).

$MPDCJ$ is an optimization problem, which has a natural corresponding decision problem asking if there is a solution 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 series of $d_{DCJ}(G_1, G_2)$ DCJ operations transforming G_1 into 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 $\#P$ is polynomial-time reducible to it. 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 true 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 a time complexity bounded by a polynomial of $|x|$, $1/\epsilon$ and $-\log(\delta)$

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 distribution p satisfying*

$$d_{TV}(p, U) \leq \epsilon \tag{2}$$

where U is the uniform distribution of the witnesses, $d_{TV}(\cdot, \cdot)$ is the total variational distance between two distributions, and the algorithm has a time complexity bounded by a polynomial of $|x|$, and $-\log(\epsilon)$. The variational

distance between two discrete distributions, p and π, over the set X is defined as

$$d_{TV}(p, \pi) = \frac{1}{2} \sum_{x \in X} |p(x) - \pi(x)| \tag{3}$$

2.2. Finding one solution is easy

Definition 6. *The* adjacency graph *$G(U \cup V, E)$ of two genomes, G_1 and G_2 with the same label set is a bipartite multigraph defined in the following way. U is the set of adjacencies and telomeres of G_1, V is the set of adjacencies and telomeres of G_2. The number of edges between $u \in U$ and $v \in V$ 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 be one of the 3 types:

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

Theorem 7. *(Yancopoulos et al., 2005; Bergeron et al., 2006)*

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

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 algorithm.

The DCJ operations on G_1 decreasing the DCJ distance from genome G_2 can be characterized using the adjacency graph. They act on the vertex set U of the adjacency graph, and have one of the following effects:

- 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 cycles,
- 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 all but the last type of DCJ operations act on a single component of the adjacency graph. However, the last type acts on two components, which must be handled separately.

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 (Braga and Stoye, 2010; Ouangraoua and Bergeron, 2010), or more generally if combining M and a W shaped paths is not allowed (co-tailed genomes imply 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. Note G_1^* be the genome which has adjacencies and telomeres of G_1 on all the M and W shaped paths of AG, and those of G_2 on all other components of AG. It is easy to see that $d_{DCJ}(G_1, G_2) = d_{DCJ}(G_1, G_1^*) + d_{DCJ}(G_1^*, G_2)$, and that G_1 and G_1^* are co-tailed, while the adjacency graph of G_1^* and G_2 has only M and W shaped paths.

Definition 8. *The* #MPDCJ$_{MW}$ problem *asks for the number of DCJ scenarios between two genomes when their adjacency graph contains only M and W shaped paths.*

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

Lemma 9.

$$\begin{aligned} \#MPDCJ(G_1, G_2) &= \frac{d_{DCJ}(G_1, G_2)!}{d_{DCJ}(G_1^*, G_2)! \prod_i (c_i - 1)! \prod_j (l_j - 1)!} \times \\ \times \prod_i c_i^{c_i - 2} \prod_j l_j^{l_j - 2} &\times \#MPDCJ_{MW}(G_1^*, G_2) \end{aligned} \tag{5}$$

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 U belonging to the ith cycle, j indexes the odd paths of the adjacency graph, l_j is the number of vertices in vertex set U belonging to the jth odd path.

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

$$\begin{gathered} \#MPDCJ(G_1, G_2) = \binom{d_{DCJ}(G_1, G_2)}{d_{DCJ}(G_1^*, G_2)} \times \\ \#MPDCJ(G_1, G_1^*) \times \#MPDCJ_{MW}(G_1^*, G_2) \end{gathered}$$

For the co-tailed genomes G_1 and G_1^*, we have from Braga and Stoye (2010) and Ouangraoua and Bergeron (2010) that

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

These two equations together with $d_{DCJ}(G_1, G_2) = d_{DCJ}(G_1, G_1^*) + d_{DCJ}(G_1^*, G_2)$ give the result. □

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

Theorem 10.

$$\#MPDCJ_{MW} \in FP \iff \#MPDCJ \in FP \tag{6}$$

$$\#MPDCJ_{MW} \in \#P\text{-}complete \iff \#MPDCJ \in \#P\text{-}complete \tag{7}$$

$$\#MPDCJ_{MW} \in FPRAS \iff \#MPDCJ \in FPRAS \tag{8}$$

$$\#MPDCJ_{MW} \in FPAUS \iff \#MPDCJ \in FPAUS \tag{9}$$

Proof. Both the multinomial factor and the two products in Eqn. 5. can be calculated in polynomial running 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 $FPRAS$ iff $\#MPDCJ$ is in $FPRAS$.

Concerning the last equivalence, the $\Longleftarrow$ part is trivial because $\#MPDCJ_{MW}$ is a particular case of $\#MPDCJ$. Now we prove that $\#MPDCJ_{MW} \in FPAUS \Longrightarrow \#MPDCJ \in FPAUS$. Suppose a 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

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

  where U is the uniform distribution over all possible DCJ sorting paths 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, since 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. As there is a polynomial number of sorting DCJ steps on each component, and after each sorting DCJ applied, the remaining components are still only cycles and odd paths, it is easy to draw such a scenario at random with a uniform distribution.

- Draw a sequence of 0s and 1s, containing $d_{DCJ}(G_1^*, G_2)$ 1s and $d_{DCJ}(G_1, G_1^*)$ 0s, uniformly from all $\binom{d_{DCJ}(G_1,G_2)}{d_{DCJ}(G_1^*,G_2)}$ such sequences.

- Merge the two paths constructed at the two first steps, according to the drawn sequence of 0s and 1s.

Note that the obtained DCJ scenario 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_π denote the set of all possible scenarios drawn by the above algorithm using a specific scenario π between G_1^* and G_2. Then

$$\sum_{\pi' \in X_\pi} |p'(\pi') - U'(\pi')| = |p(\pi) - U(\pi)| \tag{10}$$

Using Eqn. 10 we get that

$$d_{TV}(p', U') = \frac{1}{2} \sum_{\pi} \sum_{\pi' \in X_\pi} |p'(\pi') - U'(\pi')| = \frac{1}{2} \sum_{\pi} |p(\pi) - U(\pi)| = d_{TV}(p, U) \tag{11}$$

This proves that the above algorithm is an $FPAUS$ for $\#MPDCJ$, proving the left-to-right direction in Eqn. 9. □

We are going to show that $\#MPDCJ_{MW}$ is in $FPAUS$, thus $\#MPDCJ$ is in $FPAUS$. As $\#MPDCJ$ is a self-reducible counting problem (the continuations of any prefix of an optimal sorting path is exactly the sorting paths of another problem instance), the $FPAUS$ implies the existence of an $FPRAS$ (Jerrum et al., 1986). The $FPAUS$ algorithm for $\#MPDCJ_{MW}$ will be defined via a rapidly mixing Markov chain. First, we have to recall and 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 aim is to show that the number of DCJ scenarios in which an M and a W shaped path are sorted independently is not a negligible number compared to the number of all possible scenarios in which an M and a W shaped path are sorted, possibly merging them into two odd paths in one of the steps.

Theorem 11. *(Braga and Stoye, 2010) The number of DCJ scenarios of a W shaped even path in the adjacency graph with k nodes in the vertex set U is k^{k-2}.*

Theorem 12. *(Braga and Stoye, 2010) The number of DCJ scenarios of an M shaped even path in the adjacency graph with k adjacencies is $(k+1)^{k-1}$.*

Theorem 13. *The number of DCJ scenarios on a W and an M shaped even paths that sorts the two components independently 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 adjacencies of the W path and the M path, respectively.*

Proof. The W path can be sorted in $k_1^{k_1-2}$ different ways in k_1-1 steps. The M path can be sorted in $(k_2+1)^{k_2-1}$ steps in k_2 steps. The number of ways in which the two sortings can be merged is $\binom{k_1+k_2-1}{k_1-1}$. □

Theorem 14. *The number of DCJ scenarios that sorts a W and an M path at some steps combining the two components into two odd paths is less than $2(k_1+k_2+1)^{k_1+k_2-1}$, where k_1 and k_2 are the adjacencies of the W path and the M path, respectively.*

Proof. After combining the two components, in both so-emerging odd paths, there will be 1-1 telomeres in the start and the end genome. We can add two t_1s and two t_2s to them, and create two cycles from the two odd paths. It is easy to see that we could do this before starting sorting the M and W path, and hence any sorting of the so obtained cycles is also a sorting to the two components. There are two ways to connect the padded M and W paths into a cycle, and any joint sorting of the M and W shaped path is a sorting of this component. Hence there are at most $2(k_1+k_2+1)^{k_1+k_2-1}$ scenarios of the W and the M path in which the two paths are combined into two odd paths at some step. □

Theorem 15. *Let $T(k_1,k_2)$ denote the number of joint DCJ scenarios of a W path with k_1 adjacencies in the start genome (including the two telomeres) and an M path with k_2 adjacencies in the start genome. Let $I(k_1,k_2)$ denote the number of their independent scenarios.*

$$\frac{T(k_1,k_2)}{I(k_1,k_2)} = O\left(\frac{k_1^{1.5}k_2^{1.5}}{(k_1+k_2)^{0.5}}\right) \quad (12)$$

$$\frac{I(k_1,k_2)}{T(k_1,k_2)} = O\left(k_1k_2\right) \quad (13)$$

Proof. To prove Eqn. 12 is sufficient to show that

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

Using the Stirling formula, we get on the left hand side of Eqn. 14

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

After simplifications and algebraic rearrangement, we get

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

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

To prove the Eqn. 13 consider the subset of joint sortings of the W and M shaped paths which starts with passing a one long odd path from the M shaped path to the W shaped path. The result will be two odd paths with k_1 and k_2 vertices in the U set, which can be sorted in k_1-1 and k_2-1 steps in $k_1^{k_1-2}$ and $k_2^{k_2-2}$ different ways, respectively. Since we can combine any two particular solutions in $\binom{k_1+k_2-2}{k_1-1}$ ways, it is sufficient to show that

$$\frac{\binom{k_1+k_2-1}{k_1-1}k_1^{k_1-2}(k_2+1)^{k_2-1}}{\binom{k_1+k_2-2}{k_1-1}k_1^{k_1-2}k_2^{k_2-2}} = O(k_1k_2) \tag{17}$$

After minor algebraic simplification, the left hand side is

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

which is clearly $O(k_1k_2)$. □

5. The Markov chain on DCJ scenarios

We are going to sample DCJ scenarios in an indirect way. 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 paths are sorted jointly if and only if they are connected by an edge of $\mathcal{M}$. Since we can calculate the number of joint sortings of any M and W shaped paths in

polynomial time (Braga and Stoye, 2010), we can calculate the number of $\mathcal{M}$-compatible scenarios in polynomial time: it is the product of the number of possible sortings of each component and pair of components multiplied with an appropriate multinomial factor. Let $f(\mathcal{M})$ denote this number for a particular matching $\mathcal{M}$, and define a distribution π over the set of all matchings of the complete bipartite graph $K_{n,m}$ as

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

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

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

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

where $d_{TV}(\cdot, \cdot)$ denotes the total variation distance, π is the distribution defined in Eqn. 19, and U denotes the uniform distribution over all DCJ scenarios.

Proof.

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

We may decompose this sum into

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

$\sum_{(x \ \mathcal{M}-compatible \ DCJ \ scenario)} q(x)$ is $p(\mathcal{M})$ since x is drawn uniformly among the scenarios compatible with $\mathcal{M}$, and $\sum_{(x \ \mathcal{M}-compatible \ DCJ \ scenario)} U(x)$ is

$\pi(\mathcal{M})$. Furthermore, both $q(x)$ and $U(x)$ are constant for a particular matching $\mathcal{M}$, thus

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

yielding the result. □

So we are going to define an MCMC on matchings of $K_{n,m}$ converging to π. The rapid mixing 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: supposing the current state is a matching $\mathcal{M}$.

- 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 (Metropolis et al., 1953), 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

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

Then the so-emerging Markov chain is reversible and converges to π in Eqn. 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 Vempala (2005)).

The last important property of this Metropolis-Hastings Markov Chain is that the inverse of the transition probabilities are polynomially bounded. Indeed, the transition probabilities are the probabilities of the primer Markov chain multiplied by the Metropolis-Hastings ratio. The inverse of the transition probabilities in the primer Markov chain is polynomially bounded. And

as the ratio of the independent and the joint sorting of an M shaped and a W shaped component is polynomially bounded according to Theorem 15, and 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, the inverse of the Metropolis-Hastings ratio is also polynomially bounded.

We prove the rapid mixing 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 π. For this, we use the Multicommodity flow technique developed by Sinclair (1992). We note $T(\cdot|\cdot)$ the transition probabilities of the Markov chain.

The Markov graph $G(V, E)$ of our Markov chain on matchings is a directed graph which 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

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

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

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

Theorem 17. *Sinclair (1992) For any path system Γ,*

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

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 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 π its equilibrium distribution. Let the relaxation time be defined as

$$\tau_i(\epsilon) := \min\{n_0 \in \mathbb{N} : d_{TV}(p_i^n, \pi) \leq \epsilon \ \forall n \geq n_0\} \tag{27}$$

Theorem 18. *(Diaconis and Strock, 1991)*

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

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 fast mixing of the MCMC we defined, 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 of N, the number of markers in G_1 and G_2. Let U and V be the bipartition of the vertex set of $K_{n,m}$. Fix a total order on both vertex set U and V.

For two matchings $\mathcal{X}$ and $\mathcal{Y}$, we define a path from $\mathcal{X}$ to $\mathcal{Y}$ in the following way. Take the symmetric difference of $\mathcal{X}$ and $\mathcal{Y}$. It is a set of disjoint paths and cycles. Define an order on the components of $\mathcal{X}\Delta\mathcal{Y}$ in the following way: the smallest component is the one containing the smallest vertex, the second smallest component is the component having the smallest vertex in the remaining ones, etc. 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.

Note that in this construction, exactly one path for each pair (x, y), thus in the distribution $\Gamma_{x,y}$, its probability is 1. Furthermore, this path has length at most nm. So κ_Γ can be written:

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

As we already noticed that by Theorem 15, the inverse of the transition probabilities is bounded by a polynomial of N, we get

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

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

$$\hat{\mathcal{M}} = \mathcal{X}\Delta\mathcal{Y}\Delta\mathcal{Z} \tag{30}$$

Lemma 19. *The couple $\hat{\mathcal{M}}$ and $\mathcal{Z} \to \mathcal{Z}'$ unequivocally determines $\mathcal{X}$ and $\mathcal{Y}$.*

Proof. It is obvious that

$$\hat{\mathcal{M}}\Delta\mathcal{Z} = \mathcal{X}\Delta\mathcal{Y} \tag{31}$$

hence, $\mathcal{Z}$ and $\hat{\mathcal{M}}$ 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, from the $\mathcal{Z} \to \mathcal{Z}'$ we know the current component. We also know the begining 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 20. *A matching can be obtained from $\hat{\mathcal{M}}$ 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 $\hat{\mathcal{M}}$ 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 $\hat{\mathcal{M}}$ to get a matching $\tilde{\mathcal{M}}$. □

Call $\tilde{\mathcal{M}}$ this matching.

Lemma 21.

$$\frac{\pi(\mathcal{X})\pi(\mathcal{Y})}{\pi(\mathcal{Z})} = O(poly(N))\pi(\tilde{\mathcal{M}}) \tag{32}$$

Proof. We prove that

$$\frac{f(\mathcal{X})f(\mathcal{Y})}{f(\mathcal{Z})f(\tilde{\mathcal{M}})} = O(poly(N)) \tag{33}$$

It proves the lemma, as $\pi(\cdot)$ and $f(\cdot)$ differ only by a normalizing constant. $\tilde{\mathcal{M}}\Delta\mathcal{Z}$ differs at most in two edges from $\mathcal{X}\Delta\mathcal{Y}$. These edges appear in $\mathcal{X}\Delta\mathcal{Y}$, but not in $\tilde{\mathcal{M}}\Delta\mathcal{Z}$. The two vertices of any missing edges correspond to components which are independently sorted either in $\mathcal{Z}$ or $\tilde{\mathcal{M}}$, 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 of G_1 in $\mathcal{A}$, and k_2 the number of adjacencies of G_1 in $\mathcal{B}$. The ratio on the left-hand side of Eqn. 33 due to such difference is

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

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

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

and we can apply Theorem 15. □

These results together lead to the following theorem:

Theorem 22. *The Metropolis-Hastings Markov chain on the matchings defined above converges rapidly to π.*

Proof. From Lemma 21, Equation 29 may be written

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

By Lemmas 19 and 20, a matching $\tilde{\mathcal{M}}$ may appear only a polynomial number of times in this sum. So

$$\kappa_\Gamma \leq O(poly(N)) \sum_{\tilde{\mathcal{M}}} \pi(\tilde{\mathcal{M}}),$$

and as $\sum_{\tilde{\mathcal{M}}} \pi(\tilde{\mathcal{M}}) = 1$, κ_Γ is bounded by a polynomial of N, this proves the theorem. □

Using this result, we can prove the following theorem

Theorem 23. $\#MPDCJ_{MW} \in FPAUS$

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 polymonial with the size of the graph. For any start state i, $\log(1/\pi(i))$ 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 with both N and $\log(1/\epsilon)$, according to Theorem 18. 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

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

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 24. $\#MPDCJ \in FPRAS$

Proof. $\#MPDCJ_{MW} \in FPAUS$ according to Theorem 23. Then $\#MPDCJ \in FPAUS$ according to Theorem 10. Since $\#MPDCJ$ is a self-reducible counting problem, it is in $FPRAS$ (Jerrum et al., 1986). □

7. Conclusion

Sampling from reversal scenarios has been conjectured to be $\#P$-complete Miklós and Tannier (2010), 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, and is given a solution in Grinstead and J.L. (2006)). 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 (2010) prove that altering 3 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, as the one in Miklós and Tannier (2010), and the rigorous study of the modes of evolution of the eukaryote chromosomes.

References

Bergeron, A., Mixtacki, J., Stoye, J., 2006. A unifying view of genome rearrangements. LNCS 4175, 163–173.

Braga, M., Sagot, M.-F., Scornavacca, C., Tannier, E., 2008. Exploring the solution space of sorting by reversals with experiments and an application to evolution. IEEE-ACM Transactions on Computational Biology and Bioinformatics 5, 348–356.

Braga, M., Stoye, J., 2010. The solution space of sorting by DCJ. Journal of Computational Biology 17 (9), 1145–1165.

Darling, A., Miklós, I., Ragan, M., 2008. Dynamics of genome rearrangement in bacterial populations. PLoS Genetics 4 (7), e1000128.

Diaconis, P., Strock, D., 1991. Geometric bounds for eigenvalues of Markov chains. The Annals of Applied Probability 1 (1), 36–61.

Durrett, R., Nielsen, R., York, T., 2004. Bayesian estimation of genomic distance. Genetics 166, 621–629.

Fertin, G., Labarre, A., Rusu, I., Tannier, E., Vialette, S., 2009. Combinatorics of genome rearrangements. MIT press.

Grinstead, C., J.L., S., 2006. Introduction to Probability. American Mathematical Society.

Jerrum, M., Valiant, L., Vazirani, V., 1986. Random generation of combinatorial structures from a uniform distribution. Theoretical Computer Science 43, 169–188.

Larget, B., Simon, D., Kadane, B., 2002. Bayesian phylogenetic inference from animal mitochondrial genome arrangements. J. Roy. Stat. Soc. B. 64 (4), 681–695.

Larget, B., Simon, D., Kadane, J., Sweet, D., 2005. A Bayesian analysis of Metazoan mitochondrial genome arrangements. Mol. Biol. Evol. 22 (3), 485–495.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., 1953. Equations of state calculations by fast computing machines. J. Chem. Phys. 21 (6), 1087–1091.

Miklós, I., Tannier, E., 2010. Bayesian sampling of genome rearrangement scenarios via dcj. Bioinformatics 26, 3012–3019.

Ouangraoua, A., Bergeron, A., 2010. Combinatorial structure of genome rearrangements scenarios. Journal of Computational Biology 17 (9), 1129–1144.

Sinclair, A., 1992. Improved bounds for mixing rates of Markov chains and multicommodity flow. Combinatorics, Probability and Computing 1, 351–370.

Sturtevant, A., Novitski, E., 1941. The homologies of chromosome elements in the genus Drosophila. Genetics 26, 517–541.

Vempala, S., 2005. Geometric random walks: A survey. Combinatorial and Computational Geometry 52, 573–612.

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