



MCMC genome rearrangement

István Miklós

Department of Statistics, University of Oxford, Oxford, OX1 3TG, UK

Received on March 17, 2003; accepted on June 9, 2003

ABSTRACT

Motivation: As more and more genomes have been sequenced, genomic data is rapidly accumulating. Genome-wide mutations are believed more neutral than local mutations such as substitutions, insertions and deletions, therefore phylogenetic investigations based on inversions, transpositions and inverted transpositions are less biased by the hypothesis on neutral evolution. Although efficient algorithms exist for obtaining the inversion distance of two signed permutations, there is no reliable algorithm when both inversions and transpositions are considered. Moreover, different type of mutations happen with different rates, and it is not clear how to weight them in a distance based approach.

Results: We introduce a Markov Chain Monte Carlo method to genome rearrangement based on a stochastic model of evolution, which can estimate the number of different evolutionary events needed to sort a signed permutation. The performance of the method was tested on simulated data, and the estimated numbers of different types of mutations were reliable. Human and Drosophila mitochondrial data were also analysed with the new method. The mixing time of the Markov Chain is short both in terms of CPU times and number of proposals.

Availability: The source code in C is available on request from the author.

Contact: miklos@stats.ox.ac.uk

INTRODUCTION

In classical methods of string comparison, strings may only mutate by operations that act on individual characters. New applications in computational biology have motivated the study of large scale mutations such as inversions, transpositions and inverted transpositions.

The aim is to find parsimonious series of mutations that explain the difference in the gene order between two genomes. The number or summed weights of mutations can be used as a measure of the evolutionary distance between two species (Palmer and Herbon, 1988). A large set of papers on optimisation methods of genome rearrangement was published in the last decade, however, except the case of sorting signed permutations by inversions (Bader *et al.*, 2001; Bergeron, 2001; Hannenhalli and

Pevzner, 1999; Kaplan *et al.*, 1999; Siepel, 2002) or by translocations (Hannenhalli, 1996), only approximations (Bafna and Pevzner, 1998; Berman *et al.*, 2002; Eriksen, 2001; Gu *et al.*, 1999; Kececioğlu and Sankoff, 1995) and heuristics (Blanchette *et al.*, 1996) exist. Most of the papers concerning with more types of mutations either penalise all the mutations with the same weight (Gu *et al.*, 1999), or exclude a whole set of possible mutations due to a special choice of weights (Eriksen, 2001). An exception is the work of Blanchette *et al.* (1996), which we will describe in details later in this paper.

On the other hand, statistically well-based methods on genome rearrangement are rare. To our best of knowledge, only two papers were published on probability models of genome rearrangement, and both of them discussed a phylogenetic inference based on only inversions (Larget *et al.*, 2002; Sankoff and Blanchette, 1999). In this paper we introduce a Markov Chain Monte Carlo method based on a stochastic model of inversions, transpositions and inverted transpositions and test it on simulated and biological data. Our method does not need specific weights for the different types of mutations, and still can estimate the number of mutations happened.

METHODS

Stochastic modelling of inversions, transpositions and inverted transpositions

We consider a stochastic time-continuous evolutionary dynamics in which each insertion has a rate α , and each transposition and inverted transposition has a rate β . We have $\binom{n+1}{2}$ different inversions, $\binom{n+1}{3}$ different transpositions and $2\binom{n+1}{3}$ different inverted transpositions, where n is the length of the permutation. The probability that a given type of mutation at a given position happens k times in a time span t is

$$e^{-\alpha t} \frac{(\alpha t)^k}{k!} \quad (1)$$

for inversions, and

$$e^{-\beta t} \frac{(\beta t)^k}{k!} \quad (2)$$

for transpositions and inverted transpositions. We suppose that mutations happen independently.

Our aim is to calculate the probability of a trajectory, that is, the probability that a given sequence of mutations happened in a time span t . Let l denote the total number of inversions and let m denote the total number of transpositions and inverted transpositions in the sequence. We can decompose l and m into

$$l = \sum_{i=0}^{\infty} i l_i \quad (3)$$

$$m = \sum_{j=0}^{\infty} j m_j \quad (4)$$

l_i is the number of positions where inversions happened i times, m_j is the number of positions where transpositions or inverted transpositions happened j times. Obviously,

$$\sum_{i=0}^{\infty} l_i = \binom{n+1}{2} \quad (5)$$

$$\sum_{j=0}^{\infty} m_j = 3 \binom{n+1}{3} \quad (6)$$

The probability for a set of mutations is

$$\prod_{i=0}^{\infty} \left(\frac{e^{-\alpha t} \alpha^i}{i!} \right)^{l_i} \prod_{j=0}^{\infty} \left(\frac{e^{-\beta t} \beta^j}{j!} \right)^{m_j} \quad (7)$$

Using equations 3, 4, 5 and 6, equation 7 becomes

$$\frac{e^{-(\frac{n+1}{2})\alpha t} (\alpha t)^l e^{-3(\frac{n+1}{3})\beta t} (\beta t)^m}{\prod_{i=0}^{\infty} (i!)^{l_i} \prod_{j=0}^{\infty} (j!)^{m_j}} \quad (8)$$

However, permutations are not commutative, therefore different orderings of mutations lead to different permutations. Due to symmetry, the above probability distributed identically on

$$\frac{(l+m)!}{\prod_{i=0}^{\infty} (i!)^{l_i} \prod_{j=0}^{\infty} (j!)^{m_j}} \quad (9)$$

different combinations, therefore the probability of a given trajectory is

$$\frac{e^{-(\frac{n+1}{2})\alpha t} (\alpha t)^l e^{-3(\frac{n+1}{3})\beta t} (\beta t)^m}{(l+m)!} \quad (10)$$

It is a general phenomenon in the stochastic evolutionary modelling that probabilities depend on the product of rate and time, therefore, unless we have a serial sample of data from which the absolute time can be estimated, the product of rate and time is not separable. For short, we will write α and β instead of αt and βt .

Inferring the evolutionary parameters

Let G_1 and G_2 be two genomes containing n common genes evolved from a common ancestor according to the above model. It is easy to show that the evolutionary process described above is reversible, and the equilibrium distribution is the uniform one on the signed permutations. Therefore the likelihood can be obtained as:

$$\begin{aligned} P(G_1, G_2 | \alpha, \beta) &= P_{\infty}(G_1) P(G_2 | G_1, \alpha, \beta) \\ &= \frac{P(G_2 | G_1, \alpha, \beta)}{2^n n!} \end{aligned} \quad (11)$$

where $P_{\infty}(G_1)$ is the probability of G_1 in equilibrium, and $P(G_2 | G_1, \alpha, \beta)$ is the probability that G_2 evolved from G_1 under the described model with parameters α and β . This later is

$$P(G_2 | G_1, \alpha, \beta) = \sum_{t \in \text{Traj}(G_1, G_2)} P(t | \alpha, \beta) \quad (12)$$

where $\text{Traj}(G_1, G_2)$ is the set of trajectories transforming G_1 into G_2 . It is not clear how to sum the probabilities of all possible trajectories; indeed, even it is not clear how to find the most probable trajectory.

However, we can obtain the posterior distribution of α and β in a Markov Chain Monte Carlo (MCMC) framework. By definition, the posterior distribution is

$$\begin{aligned} P(\alpha, \beta | G_1, G_2) &= \frac{P_{\infty}(G_1)}{P(G_1, G_2)} P(G_2 | G_1, \alpha, \beta) P(\alpha) P(\beta) \\ &= \frac{\sum_{t \in \text{Traj}(G_1, G_2)} P(t | \alpha, \beta) P(\alpha) P(\beta)}{2^n n! P(G_1, G_2)} \\ &= \frac{\sum_{t \in \text{Traj}(G_1, G_2)} P(t, \alpha, \beta)}{2^n n! P(G_1, G_2)} \end{aligned} \quad (13)$$

where

$$\begin{aligned} P(G_1, G_2) &= \int_{\alpha, \beta} P(G_1, G_2 | \alpha, \beta) P(\alpha) P(\beta) d\alpha d\beta \end{aligned} \quad (14)$$

and $P(\alpha)$ and $P(\beta)$ are the prior probabilities of the parameters. We obtain $P(\alpha, \beta | G_1, G_2)$ by sampling from $P(t, \alpha, \beta) = P(t | \alpha, \beta) P(\alpha) P(\beta)$, namely, we joint sample the trajectories and parameters. We do not have an a priori information about the parameters, so if we do not want the prior to influence our estimation, we should choose a flat prior with a cut-off at an arbitrary high value, namely, we should sample from $P(t | \alpha, \beta)$ instead.

The MCMC strategy

We are now going to describe our method of joint sampling trajectories and parameters. Given a fixed trajectory,

we Gibbs-sample α and β (Geman and Geman, 1984). According to equation 10, and due to the flat prior:

$$P(\alpha, \beta | t) \propto e^{-(\frac{n+1}{2})\alpha} \alpha^l e^{-3(\frac{n+1}{3})\beta} \beta^m \\ = f(\alpha)g(\beta) \quad (15)$$

The cumulative density functions of f and g have a closed form, therefore we can sample from f and g easily, (see Liu, 2001, p. 24 for details).

Given fixed parameters, we do a Metropolised Independent Sampling (MIS) for drawing a new trajectory (Hastings, 1970 and Liu, 2001, p. 115). We draw a new t^* from an auxiliary distribution g . If the actual trajectory is t , then we accept t^* as a new sample with probability

$$\min \left\{ 1, \frac{P(t^* | \alpha, \beta)g(t)}{P(t | \alpha, \beta)g(t^*)} \right\} \quad (16)$$

otherwise the new sample is t . Equation 16 is called Metropolis-Hastings ratio. The success of the MIS sampling depends on the goodness of the auxiliary distribution: the auxiliary distribution must be reasonably close to the target distribution, namely, to $P(t | \alpha, \beta)$. If we propose very unlikely trajectories, then we will accept it with very low probability. On the other hand, if a likely trajectory t has a low probability in the auxiliary distribution, then $g(t)/P(t | \alpha, \beta)$ will be very low. What follows, once the Markov chain reaches t , it gets stuck, since the Metropolis-Hastings ratio will be very low for most of the proposals. In both cases the mixing time of the MCMC is very long and thus, the whole performance is poor. Therefore, we need a careful design of g , for which we use the theory of sorting signed permutations invented in distance-based research.

Based on basic group theory, we sort the permutation $\pi_2^{-1}\pi_1$ instead of transforming π_1 into π_2 . We follow the convention representing a signed permutation of length n as an unsigned permutation of length $2n$, we replace $i > 0$ with $2i - 1, 2i$, and $i < 0$ with $2i, 2i - 1$. The permutation is then framed to 0 and $2i + 1$. Only segments $[2i + 1, 2j]$ are allowed to mutate in the unsigned representation. Prokaryotes and cellular organelles have circular genomes, which can be represented with a circular permutation. For circular permutations, we connect the first and last element of the permutation instead of framing the permutation into 0 and $2n + 1$. Though the representation, and thus, the detailed computations differ for circular permutations, all theorems presented here hold for circular permutations, as well.

Starting with 0, we connect every other position in the permutation with a straight line, and starting also with 0, we connect every other number of the permutation with an arc. If we consider the permutation as a graph, whose vertices are the numbers from 0 to $2n + 1$, and

edges are the straight lines and arcs, the permutation can be unequivocally decomposed into cycles. Following the convention, we call the straight lines black edges, and arcs are named grey edges.

The basic idea for the distribution g is that we propose a mutation increasing the number of cycles with high probability, mutations leaving the number of cycles unchanged get lower probability and mutations decreasing the number of cycles are proposed very rarely. The reasoning behind the idea is that only the identical permutation has $n + 1$ cycles, other permutations have less cycles. We call a mutation changing the number of cycles with k a k -mutation (k -inversion, k -transposition and k -inverted transposition).

Inversions can increase the number of cycles at most by 1 (Pevzner, 2000). Moreover, it is well known that a minimal sequence of inversions sorting permutation π consists of $n + 1 - c_\pi$ 1-inversions and $h_\pi + f_\pi$ 0-inversions, where n is the length of π , c_π is the number of cycles in π , h_π is the number of hurdles in π , and f_π is 1 if π is a fortress, otherwise 0. (For definition of hurdles and fortresses see Pevzner (2000)) Inversions are characterised with the black edges they act on (Setubal and Meidanis, 1997; Siepel, 2002) in the following way (see Fig. 1).

- An inversion is a 1-inversion iff it acts on black edges belonging to the same cycle, and on traversing (Setubal and Meidanis, 1997; Siepel, 2002) this cycle, the two edges have different orientations.
- An inversion is a 0-inversion iff it acts on black edges belonging to the same cycle, and on traversing this cycle, the two edges have the same orientation.
- An inversion is a -1 -inversion iff it acts on black edges belonging to different cycles.

Transpositions and inverted transpositions act on three black edges. The six vertices of the three black edges can be connected with grey edges in 15 different ways. The characterisations of transpositions and inverted transpositions are more complicated. We consider three groups of them: 2-mutations, 1-mutations and the rest ($\{0, -1, -2\}$ -mutations). Figure 2. shows the possible 2- and 1-transpositions. The possible 2- and 1-inverted transpositions can be obtained from 2- and 1-transpositions easily with inverting b and c or d and e in permutations on the left part of Figure 2. If this inversion does not increase the number of cycles, then there is an inverted transposition which acts on the three black edges of the modified permutation and is a 1- or 2-inverted transposition. As can be seen, all these mutations act on a single cycle. We call the 1-mutations and 2-mutations together benign mutations. We are now going to introduce a few lemmas on which our sampling strategy is based.

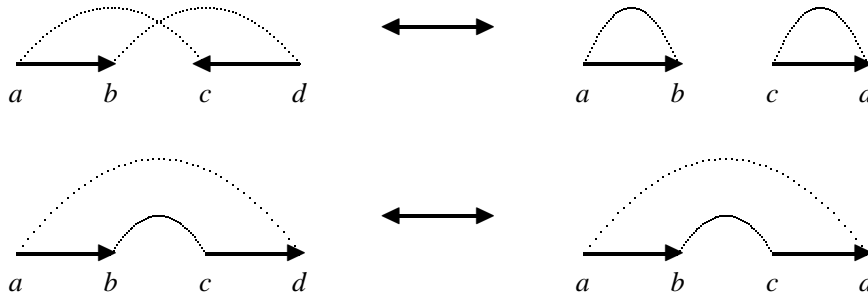


Fig. 1. How an inversion changes the number of cycles in a permutation. Note that each dashed line represents a path connecting two ends of black lines, which is not necessarily a single arc.

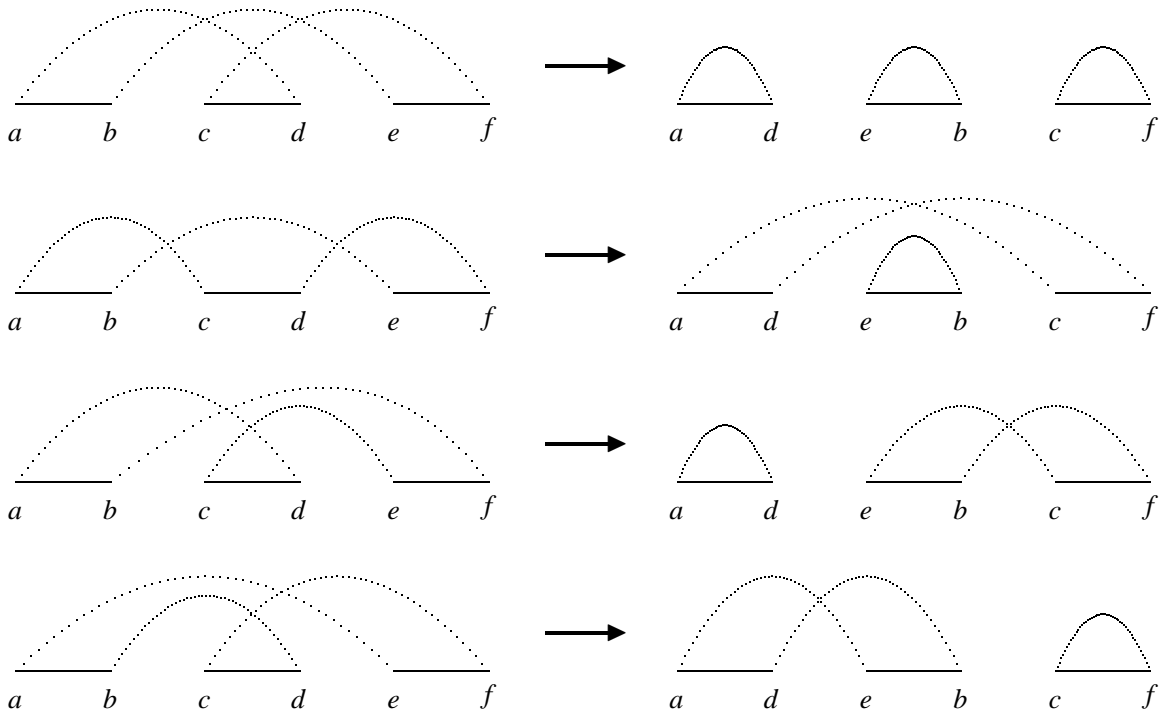


Fig. 2. How a transposition increases the number of cycles in a permutation. Note that each dashed line represents a path connecting two ends of black lines, which is not necessarily a single arc.

LEMMA 1. *For any permutation with length bigger than 1, either we have a non-benign transposition/inverted transposition or the length of the permutation is odd, and we have the permutation $n/2, 1, n/2 + 1, 2 \dots n, n/2 - 1$.*

PROOF. If the permutation contains more than 1 cycle, a transposition or an inverted transposition act on them will be non-benign. If the permutation has only one cycle, and each possible triplet of black edges has the

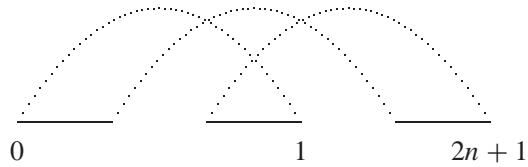
configuration that can be seen in the first row of Figure 2 then it is easy to show that we have the permutation $n/2, 1, n/2 + 1, 2 \dots n, n/2 - 1$. Otherwise, there exists a different triplet for which at least one of the possible transpositions/inverted transpositions will be non-benign.

LEMMA 2. *For any permutation with length bigger than 1, we have at least one 0-inversion or one -1 -inversion.*

PROOF. If we do not have any 0-inversion, then every cycle has a length less than 3, therefore we have at least two cycles, and an inversion acts on two cycles is a -1 -inversion. If we do not have -1 -inversion, then the permutation contains only one cycle, whose length is at least 3. This cycle has two black edges with the same orientation, and an inversion acts on them is a 0-inversion.

LEMMA 3. *For any permutation, we have either a -1 -inversion or a benign mutation.*

PROOF. It is enough to prove that at least one benign mutation exists for any permutation containing only one cycle. If this cycle's length is two, then an inversion acts on it is a 1-inversion. If the length of the cycle is at least three, then either it contains black edges with different orientation, and then an inversion acts on them is a 1-inversion, or all the orientations are the same. In the later case, we have the following situation:



The transposition acts on these three black edges is a 2-transposition.

The new trajectory is proposed in the following way. In each step, we travel the cycles, and this gives a fast decomposition of the cycles. On travelling a cycle, we mark each black edge with $+1$ or -1 depending on whether we travel the edge from left to right or not. For each long cycle containing at least three black edges, we extend the list of possible 2- and 1-transpositions with these kinds of transpositions acting on this cycle, and we do the same for 2- and 1-inverted transpositions, too. Knowing the number of cycles and the number of $+1$ and -1 sign in each cycle, it is easy to calculate the number of 1-, 0- and -1 -inversions:

$$\#1\text{-inversion} = \sum_{i=0}^k (l(c_i) - p(c_i)) p(c_i) \quad (17)$$

$$\#0\text{-inversion} = \sum_{i=0}^k (l(c_i) - p(c_i))^2 + p(c_i)^2 \quad (18)$$

$$\begin{aligned} \# - 1\text{-inversion} &= \binom{n+1}{2} - \\ &- (\#1\text{-inversion}) - (\#0\text{-inversion}) \end{aligned} \quad (19)$$

where (c_i) is the length of the i th cycle, and (c_i) is the number of $+1$ signs in this cycle. We decompose

the mutations into four subsets: benign mutations, 0-inversions, -1 -inversions and non-benign transpositions and inverted transpositions. According to Lemmas 1, 2 and 3, six possible cases might arise. In each case, we choose one of the subsets with probability given in Table 1. After choosing a subset, we choose one of its elements uniformly, and we apply this mutation for the permutation. If the result is not the identity permutation, then the outcome will be the input of the next step. If we get the identity permutation, we stop with probability 0.99, and with probability 0.01 we propose the identity permutation as the input of the next step. Since we know the cardinality of all the four subsets of mutations, we can calculate $g(t)$ easily.

The given probabilities were obtained empirically. Although the Markov chain mixes quite fast, we cannot say that these probabilities are optimal in any sense. However, the only thing must be provided for a statistically correct investigation in an MCMC framework is the ergodicity of the Markov chain, namely, non-optimal parameters do not destroy the correctness of the method. Since all possible trajectories are proposed and accepted with a non-zero probability, and all parameter value has a non-zero probability density in the Gibbs sampling part, the Markov chain is ergodic undoubtedly.

RESULTS

Testing the method on simulated data

Transpositions and inverted transpositions are not distinguished in our stochastic model. Therefore, we treat these two types of mutations in the same way, and for short, transposition means either transposition or inverted transposition below.

We tested our method on random permutations with prescribed numbers of inversions and transpositions. For each couple of numbers investigated, three independent 50-long random permutations were generated as the input of the method. We completed a run of 2.5 million cycles for each permutation; a cycle consisted of a Gibbs sampler and MIS step. Model parameters, the numbers of the different types of mutations of the actual trajectory and the acceptance ratio were reported after every thousand cycles. Trace plots of log-likelihoods indicated that the burn-in was very rapid, and the Markov chain mixed well (for an example, see Fig. 3). We discarded the initial 10% of each run, and the average numbers of different types of mutations in a trajectory were calculated on the rest of the run (see Table 2).

Each run took approximately 3 hours of CPU time on a 1.5 GHz Pentium4 PC. The average acceptance ratio was around 10%, but rarely it dropped to 1% (and then back

Table 1. Probabilities for the sampling strategy. See text for details

Benign mutation	We have		Benign mutations	Probability of choosing the subset of		Non-benign transpositions and inverted transpositions
	0-inversion	-1inversion		0-inversions	-1-inversions	
Yes	Yes	Yes	0.99	0.005	0.004	0.001
Yes	Yes	No	0.99	0.009	-	0.001
Yes	No	Yes	0.99	-	0.009	0.001
No	Yes	Yes	-	0.95	0.04	0.01
No	No	Yes	-	-	0.99	0.01
Permutation $n/2, 1, n/2 + 1, 2 \dots n, n/2 - 1$			0.99	0.01	-	-

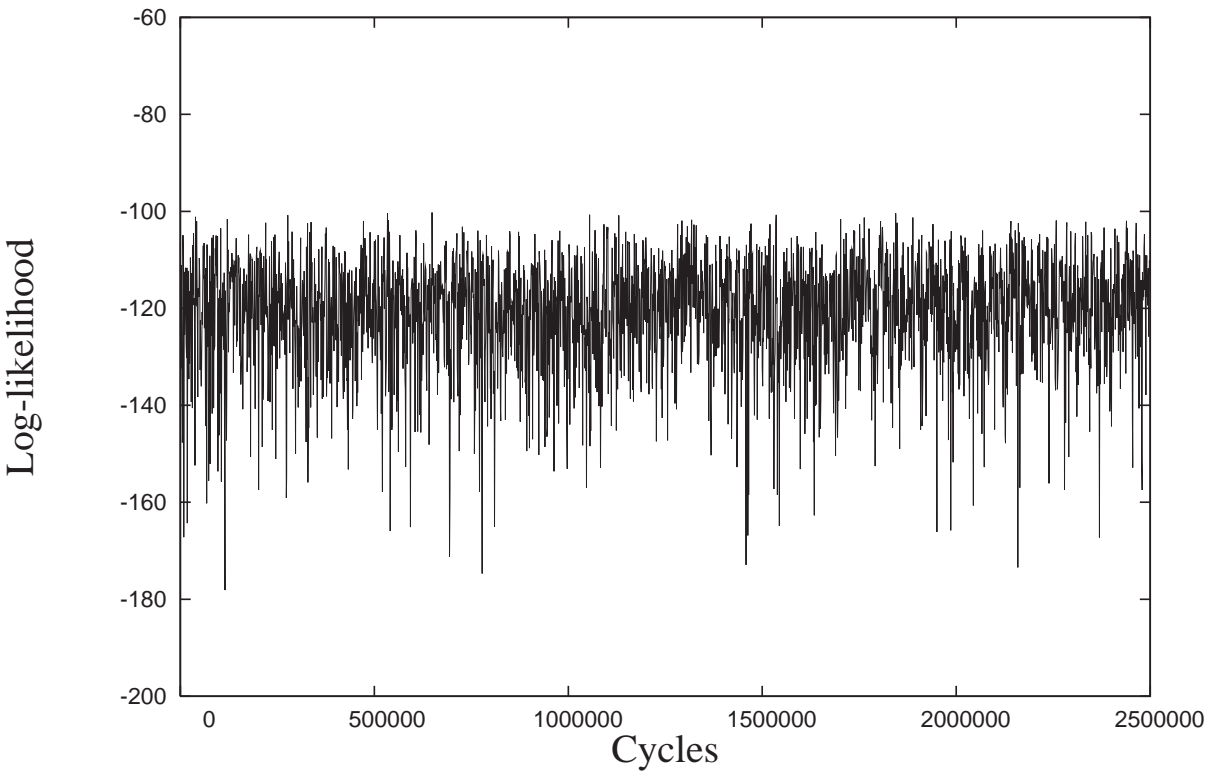


Fig. 3. The log-likelihood trace of the presented MCMC approach. The length of the input permutation was 50, and this permutation was generated using 5 random inversions and 5 random transpositions.

to the average ratio) indicating possible big differences between the auxiliary and target distribution. Indeed, some of the 0-mutations might be a part of likely trajectories, for example, an inversion merging or cutting hurdles (Pevzner, 2000, pp. 205–208). Trajectories containing 0-mutations are proposed more rarely than their likelihood would indicate, and the Metropolis-Hastings ratio corrects this bias providing that each trajectory observed in the MCMC with a frequency proportional to the likelihood of the trajectory.

When the number of inversions and the number of transpositions were the same in the initial random permutation, the estimations are close to the true. When one type of the mutations was 0, the estimations were a little bit biased. In all cases the sum of the estimated numbers of mutations were bigger than in the reality, since in many cases it is impossible to sort the proposed permutation in fewer steps than it was generated, while unlikely trajectories with many mutations have a small, but not negligible probability in our stochastic model.

Table 2. Estimated average numbers of different types of mutations and standard deviations from three independent runs. See text for details

Number of inversions		Number of transpositions	
Generated	Estimated	Generated	Estimated
5	4.96 ± 2.59	5	5.17 ± 0.73
0	2.64 ± 2.32	10	8.85 ± 1.10
10	6.46 ± 0.83	0	1.80 ± 0.21
10	8.58 ± 2.51	10	10.99 ± 1.43

Comparing human and *Drosophila* mitochondrial genomes

Blanchette *et al.* (1996) developed DERANGE II, a parametric genome rearrangement algorithm. DERANGE II is a greedy algorithm as it decomposes the possible steps into two subsets, ‘good’ and ‘bad’ mutations, and it never chooses a ‘bad’ mutation, namely, some of the trajectories do not have a chance to be chosen. However, there is no guarantee that the most likely trajectory consists of only ‘good’ mutations. The method generates a set of trajectories sorting a given permutation using different weights for inversions and transpositions, and the composition (namely, the number of different mutations) of the best trajectory is compared to a null-statistics. They found significant deviations from the null-statistics when the weight of the transpositions and inverted transpositions was slightly more than twice of the weight of inversions.

They applied DERANGE II for gene orders of human versus *Drosophila* mitochondrial genomes. For transposition weight less than twice of the inversion weight, DERANGE II obtained an optimal scenario with 12 mutations, 3 of them being transpositions. When the ratio of the two weights was between 2.0 and 2.5, the optimal scenario contained 13 transpositions and 3 inversions. Since the deviation from the randomness was the greatest in this case, Blanchette *et al.* concluded this scenario could be an appropriate result of the analysis.

We analysed the same data with different results. Our algorithm is theoretically sensitive on the representation of the data: while in an optimisation method, segments conserving the same gene orders can be treated as a single gene in the signed permutation without influencing the result, this representation leads to loss of information (namely information about the number of positions where we do not observe changes) in our approach, and thus, might influence the result. To check the robustness of our method, we analysed two representations. In one representation, each gene was counted individually, except the couple of genes of ATP-synthase F0 subunit 6 and 8, since these genes are overlapping, and this overlap seems conserved. In the second representation, each conserved segment got a single number. In the first case, the estimated average number of inversions was 6.76 ± 3.32 ,

the average number of transpositions was 6.80 ± 1.12 . For the second representation, these numbers were 6.95 ± 3.52 and 6.16 ± 1.88 , respectively. Therefore we can conclude that our method is robust to the representation of data. It is worth mentioning that we found a sorting scenario involving only 2 inversions and 9 transpositions, so 11 mutations together, which is better than that found by the greedy optimisation of DERANGE II.

CONCLUSIONS

We introduced a Markov Chain Monte Carlo method for a statistical investigation of the genome rearrangement problem. Tests on simulated data revealed that the new approach could give a reliable estimation to the number of mutations happened. The exact numbers were not expected to be exactly recovered, however, except the extreme case when the input data was generated using only inversions, the difference between the estimated and real number of mutations was less than the standard deviation.

The new method estimates slightly more transpositions and fewer inversions than the actual number of mutations generated the input random permutation. This is in accordance with the previous observations that using same weights for inversions and transpositions, an optimal sorting scenario contains many more transpositions than inversions. However, the bias is not so huge in the introduced method than the bias in the distance-based approach, and the bias could be reduced using prior distributions on the evolutionary parameters. For doing this, we should understand deeply the dynamics of the proposed stochastic evolutionary process.

The proposed auxiliary distribution in the MIS step is far from the optimal. Some of the 1-inversions are not safe inversions (Pevzner, 2000, p. 200), and therefore they might be less frequent in likely trajectories than they are proposed. On the other hand, some of the 0-inversions are sorting inversions in an optimisation scenario, and so, they are more frequent in likely trajectories than they are proposed. Better proposals could increase the acceptance ratio, and thus, they could improve the mixing properties of the Markov chain.

The present implementation of the method does an exhaustive search for the 2- and 1-transpositions. Theoretically it leads to a third order algorithm, though in practice, long cycles are rare, and hence, the running time is acceptable. A better implementation of the approach might improve both the theoretical time complexity and the running time in practice.

ACKNOWLEDGEMENTS

This work was funded by EPSRC grant HAMJW and MRC grant HAMKA. Gerton Lunter, Roald Forsberg and Rune Lyngsø are thanked for useful suggestions and

discussions. The author thanks Jotun Hein for support and encouragement.

REFERENCES

- Bader,D.A., Moret,B.M.E. and Yan,M. (2001) A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *J. Comp. Biol.*, **8**(5), 483–491.
- Bafna,V. and Pevzner,P.A. (1998) Sorting by transpositions. *SIAM J. Disc. Math.*, **11**(2), 224–240.
- Berman,P., Hannenhalli,S. and Karpinski,M. (2002) 1.375-Approximation algorithm for sorting by reversals. In *Proceedings of ESA2002*. pp. 200–210.
- Bergeron,A. (2001) A very elementary presentation of the Hannenhalli–Pevzner theory. In *Proceedings of CPM2001*. pp. 106–117.
- Blanchette,M., Kunisawa,T. and Sankoff,D. (1996) Parametric genome rearrangement. *Gene*, **172**, GC11–GC17.
- Eriksen,N. (2001) $(1+\epsilon)$ -approximation of sorting by reversals and transpositions. In *Proceedings of WABI2001, LNCS. 2149*, pp. 227–237.
- Geman,S. and Geman,D. (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 721–741.
- Gu,Q-P., Peng,S. and Sudborough,I.H. (1999) A 2-approximation algorithm for genome rearrangements by reversals and transpositions. *Theor. Comp. Sci.*, **210**(2), 327–339.
- Hannenhalli,S. (1996) Polynomial algorithm for computing translocation distance between genomes. 168–185.
- Hannenhalli,S. and Pevzner,P.A. (1999) Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *J. ACM*, **46**(1), 1–27.
- Hastings,W.K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**(1), 97–109.
- Kaplan,H., Shamir,R. and Tarjan,R. (1999) A faster and simpler algorithm for sorting signed permutations by reversals. *SIAM J. Comput.*, **29**(3), 880–892.
- Kececioglu,J.D. and Sankoff,D. (1995) Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. *Algorithmica*, **13**(1/2), 180–210.
- Larget,B., Simon,D.L. and Kadane,J.B. (2002) Bayesian phylogenetic inference from animal mitochondrial genome arrangements. *J. Roy. Stat. Soc. B*, **64**(4), 681–695.
- Liu,J.S. (2001) *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics, New York.
- Palmer,J.D. and Herbon,L.A. (1988) Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence. *J. Mol. Evol.*, **28**, 87–97.
- Pevzner,P. (2000) *Computational Molecular Biology*. MIT Press, Cambridge.
- Sankoff,D. and Blanchette,M. (1999) Probability models for genome rearrangements and linear invariants for phylogenetic inference. In *Proceedings of the 3rd International Conference on Computational Molecular Biology (RECOMB99)*. pp. 302–309.
- Setubal,J. and Meidanis,J. (1997) *Introduction to Computational Molecular Biology*. PWS Publishing, Boston.
- Siepel,A. (2002) An algorithm to find all sorting reversals. In *Proceedings of RECOMB2002*. pp. 281–290.