Genome Rearrangement in Mitochondria and Its Computational Biology

István Miklós[1] and Jotun Hein[2]

[1] Hungarian Academy of Science and Eötvös Loránd University of Science,
Theoretical Biology and Ecology Group
1117 Budapest, Pázmány Péter sétány 1/c, Hungary
miklosi@ramet.elte.hu
[2] Oxford Centre for Gene Function
University of Oxford
South Parks Road, Oxford, OX1 3QB, UK
hein@stats.ox.ac.uk

Abstract. In the first part of this paper, we investigate gene orders of closely related mitochondrial genomes for studying the properties of mutations rearranging genes in mitochondria. Our conclusions are that the evolution of mitochondrial genomes is more complicated than it is considered in recent methods, and stochastic modelling is necessary for its deeper understanding and more accurate inferring. The second part is a review on the Markov chain Monte Carlo approaches for the stochastic modelling of genome rearrangement, which seem to be the only computationally tractable way to this problem. We introduce the concept of partial importance sampling, which yields a class of Markov chains being efficient both in terms of mixing and computational time. We also give a list of open algorithmic problems whose solution might help improve the efficiency of partial importance samplers.

1 Introduction

The idea that differences between the gene orders of two genomes can be used as a measurement of evolutionary distance was proposed more than six decades ago [1]. It was rediscovered in the eighties [2], and since then a large set of papers on optimisation methods for genome rearrangement problems has been published. However, except the case of sorting signed permutations by inversions [3–7] or by translocations [8], only approximations [9–13] and heuristics [14] exist. Most of the methods concerning with more types of mutations either penalise all the mutations with the same weight [12], or exclude a whole set of possible mutations due to a special choice of weights [11].

The principle of choosing solutions by minimising the amount of evolution is also called parsimony and has been widespread in phylogenetic analysis. Over the last two decades the parsimony method of phylogenetic reconstruction has been severely criticised and lost terrain to methods based on stochastic modelling of evolution [15, 16]. Statistical methods give not only more consistent estimations, but also the possibility of hypotheses testing and goodness-of-fit testing

J. Lagergren (Ed.): RECOMB 2004 Ws on Comparative Genomics, LNBI 3388, pp. 85–96, 2005.

of the underlying model. Therefore it is a natural attempt to develop statistical methods also for genome rearrangement.

Recently a few papers were published on probability models of genome rearrangement. Two of them considered only inversions [17, 18], and a third one worked on multi-chromosomal genomes rearranged by both inversions and translocations [19]. We also started a Bayesian approach to the genome rearrangement problem considering insertions, transpositions and inverted transpositions in unichromosomal genomes [20, 21]. As we will explain it in detail in Section 4, all of these methods have the same computational concept: instead of analytical solutions, they apply a Markov chain Monte Carlo (MCMC) converging to the distribution defined by the underlying stochastic model. Samples from this Markov chain estimate statistics of interest like the posterior distribution of number of mutations happened, evolutionary parameters, etc.

All the underlying models used so far in genome rearrangement problems are relatively simple, and an obvious way to achieve further improvements on this methodological line is to improve models describing the evolution of gene orders. We will focus on Metazoa mitochondrial genomes for several reasons. Mitochondrial genomes are unichromosomal circular genomes consisting of usually 13 protein coding genes and 24 RNA genes. This gene content as well as the genes themselves are very conservative. Each gene is represented in one copy except a very few cases that helps to identify homologous gene pairs. Mitochondrial genomes are very compact lacking introns and transposons, which are known to influence the rate of mutations rearranging genomes. Previous results showed that there were no selection on any particular gene order in mitochondria.

The large amount of available mitochondrial genome data allows the investigation of closely related genomes. Our aim is to reveal elementary steps of the evolution of gene orders. In Section 3, we report our findings based on the investigation of the NCBI database on gene orders in mitochondria, and we review previous hypotheses, too. In Section 4, we describe a general framework for modelling the dynamics behind the evolution of gene orders in mitochondria. Efficient computation in biologically more realistic models needs a core logic being significantly different from that used in optimisation methods. We give a list of open algorithmic questions related to this framework to demonstrate that Markov chain Monte Carlo is not a boring technique but a source of interesting computational problems. In Section 5, we discuss how similar research could be conducted on bacterial or nuclear genomes.

2 Mathematical Description of Genome Rearrangement

We consider that two genomes have the same gene content, each gene is represented in one copy in both genomes. We describe their gene orders as signed permutations, numbers correspond to genes, signs represent the reading direction of genes. Based on elementary group theory, a series of mutations transforming a genome π_1 to genome π_2 also transforms a permutation $\pi_2^{-1}\pi_1$ to the identical permutation. Therefore we will always consider *sorting* permutations, namely

transforming a permutation into the identical one. For circular genomes, we choose an anchor gene being in position 1 and having a positive sign. On the NCBI database, this is the NADH dehydrogenase subunit 1. It is easy to show that the rest of the genome mimics the evolution of a linear genome, hence the genome rearrangement problem of circular genomes having n genes is equivalent with the genome rearrangement problem of linear genomes with $n-1$ genes. We will talk about linear signed permutations in the rest of the paper. We follow the convention representing a signed permutation of length n as an unsigned permutation of length $2n$, we replace $+i$ with $2i-1, 2i$, and $-i$ with $2i, 2i-1$. This unsigned permutation is then framed to 0 and $2i+1$. Only segments $[2i+1, 2j]$ are allowed to mutate in the unsigned representation.

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. We consider the permutation as a graph, called *graph of reality and desire*, whose vertexes 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 or reality edges, and arcs are named grey edges or desire edges. A black edge is a breakpoint if its cycle is longer than a black edge and a grey arc. The breakpoint distance between genomes π_1 and π_2 is the number of breakpoints in $\pi_2^{-1}\pi_1$.

3 Inferring Closely Related Mitochondrial Genomes

To date, 466 fully sequenced Metazoa mitochondrial genomes have been published in the NCBI database yielding 140 different gene orders. For these 140 genomes, all pairs of genomes were compared. Compared pairs were sorted based on their breakpoint distance. Results can be downloaded form the World Wide Web, http://www.stats.ox.ac.uk/~miklos/metazoa.tar.gz.

Among the $\binom{140}{2} = 9730$ comparisons, 36 have 0 breakpoint distance. Although these pairs have different gene content, the intersection of genes have the same order. All of the genomes in these comparisons belong to the Vertebrata phylum.

A breakpoint distance 1 is mathematically impossible. We found 72 genome pairs having breakpoint distance 2. Remarkably, 70 of them are caused by inversions of a single gene. In all of these cases, the inverted gene was a tRNA gene. One of the long insertions is two long (*Pollicipes polymerus* vs. *Tetraclita japonica*), and the second one is eighteen long (*Pisaster ochraceus* vs. several urchins having the same gene order).

Breakpoint distance 3 is always caused by a transposition, an inverted transposition or two adjacent inversions. We found 146 comparisons with breakpoint distance 3. Several of these pairs differ in a swap of two adjacent genes. A frequent pattern is a transposition of a single gene that is moved far from its original place. There are several cases for inverted transpositions, even a two long fragment can be inverted, for example *Caelorinchus kishinouyei* vs. *Gonostoma gracile*. Remarkably again, we did not found any transposed fragment longer than two. Two

adjacent inversions acting on neighbour tRNA genes coding cystein and tyrosine might also cause a breakpoint distance 3 in several vertebrates.

The first really interesting breakpoint distance is 4, because it cannot be caused by a single inversion, transposition or inverted transposition. In terms of such mutations it might be the result of two non-adjacent inversions or two mutations having a common black edge, of which at least one of them is a transposition or inverted transposition. Surprisingly, behind almost half of the breakpoint distances 4 we can see the later configuration (40 from 93). In most of these surprising cases, the pattern can be described with a transposition followed by the inversion of a single gene at one of the ends of the inverted fragment. If mutations happened independently we would see about one tenth of the inversion-transposition pairs connected (recall that the vast majority of inversions acts on a single gene). We can say that connected inversion-transposition pairs are more frequent than we would expect considering independence, 35 connected pairs versus 121 independent pairs having breakpoint distance 5. However, it is hard to elaborate statistical significance, since genomes were not sequenced randomly but based on biological interest, hence we have a biased sample. In all cases, the common black edge was in the vicinity of a control region. It has already been reported that genome rearrangements are more frequent around the control region [22].

We show two examples on Fig. 1. A transposition and an inversion separates the *O. bicirrhosum* mitochondrial genome from several bird genomes. Next to the common point of the two mutations lies the control region in both genomes. Fig. 1b shows genome rearrangements between ticks *I. hexagonus* and *R. sanguineus*. In *R. sanguineus*, a new control region has emerged. This new CR clearly attracts mutations.

We must mention that there might be mutations increasing significantly the breakpoint distance. Boore proposed a duplication-loss model a few years ago [22]. In this model, a part of the genome is duplicated, and after the duplication, one copy of all gene pairs are eliminated. Boore found an excellent example strongly supporting this model. In primitive Holothuroidea, a set of 15 tRNA genes can be found in one part of the mitochondrial genome. In *Cucumaria*, 9 of them stay in the same place, while six of them moved to another part of the genome. Only a couple of these six were adjacent in the original rearrangement, however, all of them kept the reading direction and their relative order. Moreover, several nonassignable nucleotides can be found in *Cucumaria* exactly at those places where unused genes are being eliminated.

We might conclude that all the three classical mutation types (inversions, transpositions and inverted transpositions) occur during the evolution of mitochondrial gene orders, and these mutations usually act on short segments and they are frequently correlated. The correlation is suspiciously caused by control regions. Other type of mutations should be also considered. A duplication-loss model was already published, a type of mutation that can cause significant increase of breakpoint distance in one step. Since there is a selection pressure keeping mitochondrial genomes very compact, the speed of gene loss should be

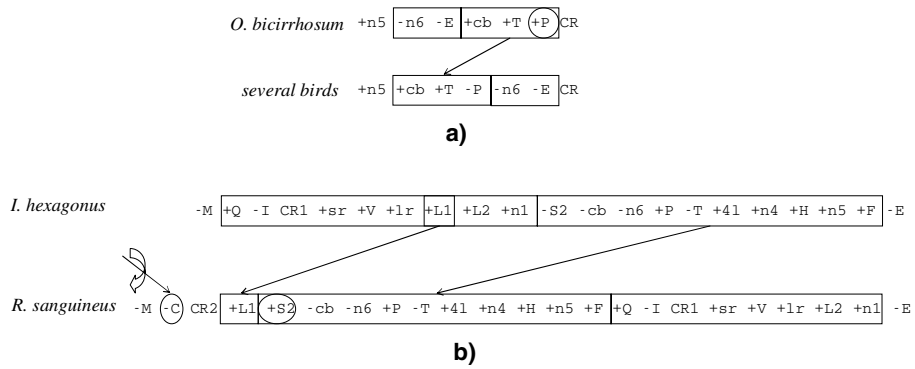

Fig. 1. Genome rearrangement between **a)** several birds and *Osteoglossum bicirrhosum* and **b)** ticks *Ixodes hexagonus* and *Rhipicephalus sanguineus*. Arrows show transpositions, circles show inversions of single genes, arrow with a twisted arrow on it indicates inverted transposition. This transposed and inverted tRNA-Cys gene originated from a region not shown in the figure. Abbreviations: `n1, n4, n5`: NADH dehydrogenase subunit 1, 4 and 5; `sr`: 12S rRNA; `lr`: 16S rRNA; `CR, CR1, CR2`: control regions; tRNA genes are denoted by the one-letter amino acid abbreviation.

very fast, and the entire scenario can be modelled as a single mutation happening in infinitesimally small time.

All these findings might question the usefulness of methods considering only inversions. In the next section, we give a review on more complicated models for genome rearrangement and their computational properties. We do not have definite solutions on how to treat duplication-loss events, but we describe an approach which might be fruitful for correlated mutations.

4 Stochastic Modelling of Genome Rearrangements

4.1 Time-Continuous Markov Model

Time-continuous Markov models have been the standard approach for stochastic modelling of molecular evolution. In case of genome rearrangements, points in the state space of the Markov model are the possible gene orders. The number of these points grows with the factorial of the number of genes, therefore the brute force calculation working fine for example for substitution models of nucleic acids is computationally intractable here. What we can calculate is the likelihood of a trajectory, which is the probability that a given sequence of mutations happened in a time span conditional on a set of parameters describing the model. It is easy to find a closed form for this likelihood when the sum of the rates of mutations does not depend on the actual state [20, 23], it is

$$e^{-tr\times t}\frac{\prod_{i=1}^{l} r_i t}{l!} \tag{1}$$

where tr is the sum of rates of all possible mutations at each state (total mutation rate or *exit rate*, see [23]) t is the spanning time of the trajectory, r_i is the rate of the ith mutation, and l is the length of the trajectory. The exit rate always keeps constant when we work on unichromosomal genome, insertions and deletions are excluded and mutations happen independently. Durrett *et al.* recognised that the sum of rates changes along the trajectory for multi-chromosomal genomes, and they introduced pseudo-events to maintain the constant total rate along the trajectory [19]. When the rate of mutations depends on the positions of previous mutations, the sums of rates are changing along the trajectory, since different mutations act on different number of positions, and hence in biologically more realistic problems, we will face with the same problem, even for unichromosomal genomes. Duplications and deletions or emergence of a new control region also cause the exit rate changing. The approximation introducing pseudo-events is not necessary, since exact likelihood values can be calculated even when the sum of the rates are varying with states of the Markov model. However, exact calculations need more computational time, the state-of-the-art algorithm [23] calculating trajectory likelihoods runs in $O(l^2)$ time, where l is the length of the trajectory, namely, the number of states in it.

4.2 Metropolised Partial Importance Sampler: A New MCMC Strategy

The aim is to sample trajectories with probabilities proportional to their likelihood, and this can be done by using Markov chain Monte Carlo (MCMC) [24, 25]. In all of the published methods, an update in the Markov chain replaces a part of the trajectory. The proposal for the new sub-trajectory is drawn from a distribution that mimics the target distribution we would like to sample from, and the new proposal is independent from the old sub-trajectory. We call this strategy *Metropolised Partial Importance Sampler* [21]. The success of such a sampler is based on how well we can approximate a target distribution in small dimensions. Here the target distribution is the posterior distribution of trajectories given two genomes that the trajectories should connect. If genomes are close to each other, trajectories are short (low dimension). When we resample short sub-trajectories, the two rearrangements at the ends of the sub-trajectory are similar.

The discrepancy between the proposal and the target distribution is corrected by accepting the proposal with probability

$$\min\left\{1\ ,\ \frac{P(X|Y)\pi(Y)}{P(Y|X)\pi(X)}\right\} \tag{2}$$

where P is the proposal distribution, π is the target one, X is the actual state of the chain, and Y is the proposal, and the chain remains in state X with the complement probability. The probability in Equation 2 is known as the Metropolis-Hastings ratio [24, 26]. The beauty of MCMC is that whenever the Markov chain based on the proposals is ergodic on the state space of the target distribution, and

$$P(X|Y) > 0 \quad \text{if and only if} \quad P(Y|X) > 0 \tag{3}$$

the Metropolis-Hastings ratio transforms it into a chain converging to the target distribution. However, the convergence might be very slow, and a slow convergence also implies high correlation between neighbour points of the chain. From a computational point of view it is also important question how much time it takes to get a proposal and to calculate the Metropolis-Hastings ratio. We are going to discuss both mixing and computational properties of Markov chain Monte Carlo for genome rearrangement.

Mixing of Markov Chains. For fast convergence and good mixing, the proposal distribution should be selected carefully. An ideal proposal distribution has low dependence on the actual state and yields Metropolis-Hastings ratios always close to 1. Indeed, if the proposal did not depend on the actual state of the chain and the Metropolis-Hastings ratio was always 1, namely, if

$$P(Y|X) = P(Y) = \pi(Y) \tag{4}$$

then the Markov chain would consist of independent samples of the target distribution. (Although a sampler might be super-efficient having smaller sampling variance than that of independent sampling [25], we will not discuss it here.) The proposed sub-trajectory does not depend on the removed one, therefore the dependency of the new trajectory correlates only with the length of the sub-trajectory being resampled. From this point of view, resampling long sub-trajectories seems to be a good idea. On the other hand, we can sample from a distribution only *approximating* the target one. The overlapping of the proposal and target distributions will obviously decrease with the length of the resampled sub-trajectory causing small Metropolis-Hastings ratio, and hence, small acceptance ratio. An additional observation is that the proposed and original sub-trajectories might have different length, therefore all possible length for sub-trajectories should be proposed to satisfy Condition 3. We conjecture that it is very hard to prove that a particular type of distribution is better than another for the purpose. However, several types of distributions were used successfully in the literature [18, 20]. Due to the trade-off between dependency in the Markov chain and acceptance ratio, there should be an optimal expected length of the resampled sub-trajectory for which the acceptance ratio is still sufficiently high, while on the other hand, the autocorrelation of the samples got small. The distribution can be optimised in preliminary performance studies [18].

The mixing of the Markov chain also depends on how well the proposal distribution can mimic the target distribution. Published methods propose new paths step by step. They measure the departure of the actual rearrangement from the rearrangement which the sub-trajectory must arrive to, and propose a mutation which decrease the measurement of the departure ('good' mutations) with high probability and propose other ones ('bad' mutations) with low probability. This philosophy seems to be essential, since random mutations would reach the target rearrangement with a very small probability. Based on the measurement of

goodness of mutations, there are several possibilities to propose good and bad mutations differing also in the amount of computational time needed. We are going to discuss them below.

Efficiency of Sampling Sub-trajectories. The inversion distance problem is one of the few problems in genome rearrangement which has polynomial time solution and has been most intensively investigated [3, 4]. It is also possible to enumerate all sorting inversions, namely, inversions decreasing the inversion distance [7]. Similarly we can enumerate inversions that do not change or increase the inversion distance. An obvious strategy uses these enumerations for sampling sub-trajectories [17]: it samples a sorting inversion with high probability (uniformly among them), and non-sorting inversion with small probability. However, the running time for sampling a mutation grows with $\Omega(n^2)$, where n is the number of genes, hence it gets inefficient for large genomes.

An alternative strategy distinguishes inversions based on how they change the number of cycles in the graph of desire and reality [18, 20]. We distinguish three groups of inversions, $+1$-, 0- and -1-inversions, based on whether they change the number of cycles by 1, 0 or -1, respectively, see Fig. 2. A sampling strategy samples uniformly inside the three group, but the group of $+1$-inversions with high probability, and with decreasing probability the groups of 0- and -1-inversions. Although we know that an inversion increasing the number of cycles is not necessarily a sorting inversion, it is still a good measurement, since the identical permutation has $n+1$ cycles, where n is the number of genes, and all other rearrangements have less cycles. The advantage of this strategy is that we can calculate the number of inversions in each group in linear time:

$$\#1\text{-inversion} = \sum_{i=0}^{k} \left(l(c_i) - p(c_i)\right) p(c_i) \tag{5}$$

$$\#0\text{-inversion} = \sum_{i=0}^{k} \binom{l(c_i) - p(c_i)}{2} + \binom{p(c_i)}{2} \tag{6}$$

$$\begin{aligned} \#-1\text{-inversion} &= \binom{n+1}{2} - \\ &-(\#1\text{-inversion}) - (\#0\text{-inversion}) \end{aligned} \tag{7}$$

where k is the number of cycles, $l(c_i)$ is the length of cycle c_i, and $p(c_i)$ is the number of positive black edges of cycle c_i, that is the number of black edges one passes left to right on a tour of the cycle [20].

In an earlier work [20], we sampled inversions with the rejection method [27, 25], namely, we sampled an inversion uniformly, and if it was an inversion of the prescribed type, we accepted it, otherwise we rejected and drew a new sample till success. It is also possible to sample inversions of a prescribed type in a running time growing linearly with the number of genes [21], which turns out to be more efficient in practise. For example, we sample $+1$-inversions in the following way. We create a list of positive and negative edges for each cycle. We first sample

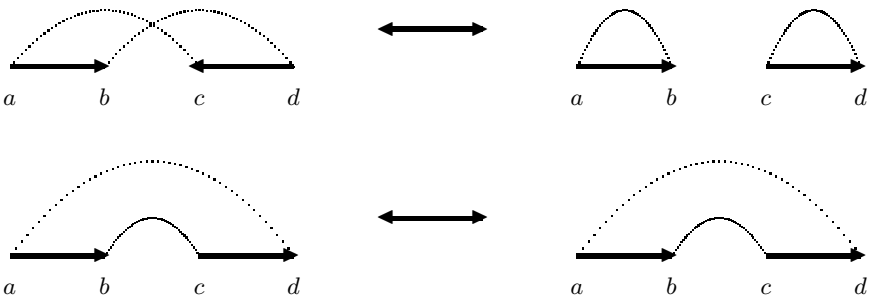

Fig. 2. How an inversion changes the number of cycles in a permutation. Note that each dashed line represents a path connecting two ends of black lines, which is not necessarily a single arc. A +1-inversion acts on two black edges of a cycle having different orientation. A −1-inversion acts on two cycles. A 0-inversion acts on two edges of a cycle having the same orientation.

a random cycle with a probability proportional to the number of +1-inversions acting on it, and then we choose a random positive and a random negative edge of this cycle uniformly. These two edges define the sampled +1-inversion. Due to the weighted sampling of cycles, the procedure guarantees uniform sampling on all +1-inversions. Similar strategies exist for uniform sampling of 0- and −1-inversions.

Sampling transpositions and inverted transpositions is not so well elaborated. Only one strategy is known, it distinguishes +2, +1 transpositions and inverted transpositions, and treats all the rest in the same way [20]. +2- and +1-mutations are listed, and sampled using this list, while other mutations are sampled with the rejection method. +2- and +1-transpositions and inverted transpositions act always on one cycle, and the enumeration investigates all the possible triplets of edges in all cycles. Although the running time for this sampler grows with the number of genes cubed in the worst case, in practise it is still fast, since long cycles are rare, especially when we resample short sub-trajectories. However, a better sampling strategy would be a great advantage.

It is not enough to somehow sample mutations from a reasonable distribution, but proposal and back-proposal probabilities ($P(Y|X)$ and $P(X|Y)$) should also be calculated. The rejection method is a good example that a sampling strategy is not necessarily able to calculate sampling probabilities. Sampling with rejection is always coupled with an algorithm enumerating the size of the set we are sampling from to get sampling probabilities. Therefore we need a thinking that might be unusual in optimisation strategies to improve present techniques in stochastic modelling: we need reasonable measurements for the goodness of a mutation which can be sampled fast, as well as we should be able to calculate proposal and back-proposal probabilities. In the next section we are going to list some open algorithmic problems.

4.3 Open Problems

Propose Transpositions in Sub-cubic Time. The state-of-the-art algorithm needs $O(n^3)$ time to propose a transposition from a reasonable distribution, where n is the number of genes in the genome. Can it be improved? Is there other distributions for which a faster strategy exist? Distinguishing transpositions based on the number of breakpoints they remove might be a promising way.

Polynomial Proposal for the Duplication-Loss Model. There are exponentially many possible duplication-loss events. How can we characterise good mutations and how can we efficiently sample from them?

Sub-squared Time Calculation for the General Trajectory Likelihood. Can we calculate trajectory likelihoods faster than $\Theta(n^2)$ time?

Reusing Information During Sub-trajectory Proposal. Can we update efficiently auxiliary variables storing information about cycle decomposition, number of different type of mutations, etc. during sampling trajectories using informations about the previous mutation sampled?

5 Discussion

Most of the methods in the scientific literature consider inversions, transpositions and inverted transpositions as elementary mutations rearranging genomes [28]. In this paper, we investigated mitochondrial genomes to get a better picture what elementary rearrangment events really are. We found all the tree types of mutations mentioned above, and we also showed that short rearrangements are frequent and mutations are not independent. The dependency is very likely caused due to the increased mutation rate around the control region.

In the second part of this paper, we gave an overview on recent progress in stochastic modelling genome rearrangement. Available techniques provide a partial solution how to incorporate more prior knowledge into the models to improve them. We can introduce different rates for different length of mutations as well as different rates for mutations acting around the control region without any problem, since they do not change the exit rate. Dependency on the acting points of the previous mutation does change the exit rate, since transpositions act on three black edges of the breakpoint graph, while inversions act only on two. An interesting problem would be to change samplings such that dependent mutations are proposed more frequently, but changing the likelihood calculations on its own enough to get a Markov chain converging to the desired new distribution. Modified likelihood calculations need increased running time, and this might be a drawback when trajectories are long.

Introducing dependency between mutations would be a simple model of more complicated mutations like duplication-loss events. Indeed, in such a model,

a sequence of mutations mimicing a duplication-loss event would have higher likelihood than a sequence of similar mutations not having a common edge in the breakpoint graph. However, a reassuring solution would be the explicit modelling of these events.

The number of sequenced genomes grows quickly, and recent sequencing projects meet the requirement of comparative genetics to sequence closely related genomes. It would be worth investigating bacterial and Eukaryota nuclear genomes, as well. However, such a work will definitely be more complicated: there might be several copies of genes, hence we cannot describe a genome as a signed permutation. Additionally, pseudogenes, transposons, repetitive elements should be modelled in a reasonable way. In spite of the difficulties, the authors believe that stochastic modelling and MCMC will be the main key in modern comparative genetics.

Acknowledgements

This work was funded by EPSRC grant HAMJW and MRC grant HAMKA. I.M. was further funded by a Békésy György postdoctoral fellowship.

References

1. Sturtevant, A.H., Novitski, E.: The homologies of chromosome elements in the genus Drosophila. Genetics **26** (1941) 517–541
2. Palmer, J.D., Herbon, L.A.: Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence. J. Mol. Evol. **28** (1988) 87–97
3. Bader, D.A., Moret, B.M.E., Yan, M.: A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. J. Comp. Biol. **8(5)** (2001) 483–491
4. Bergeron, A.: A very elementary presentation of the Hannenhalli-Pevzner theory. In: Proceedings of CPM2001 (2001) 106–117
5. Hannenhalli, S., Pevzner, P.A.: Transforming Cabbage into Turnip: Polynomial Algorithm for Sorting Signed Permutations by Reversals. Journal of ACM **46(1)** (1999) 1–27
6. Kaplan, H., Shamir, R., Tarjan, R.: A faster and simpler algorithm for sorting signed permutations by reversals. SIAM J. Comput. **29(3)** (1999) 880–892
7. Siepel, A.: An algorithm to find all sorting reversals. In: Proceedings of RECOMB2002 (2002) 281–290
8. Hannenhalli, S.: Polynomial algorithm for computing translocation distance between genomes. In: Proceedings of CPM1996 (1996) 168–185
9. Bafna, V., Pevzner, A.: Sorting by transpositions. SIAM J. Disc. Math. **11(2)** (1998) 224–240
10. Berman, P., Hannenhalli, S., Karpinski, M.: 1.375-Approximation Algorithm for Sorting by Reversals. In: Proceedings of ESA2002 (2002) 200–210
11. Eriksen, N.: $(1+\varepsilon)$-approximation of sorting by reversals and transpositions. In: Proceedings of WABI2001, LNCS **2149** (2001) 227–237
12. Gu, Q-P., Peng, S., Sudborough, H.I.: A 2-Approximation Algorithm for Genome Rearrangements by Reversals and Transpositions. Theor. Comp. Sci. **210(2)** (1999) 327–339

13. Kececioglu, J.D., Sankoff, D.: Exact and Approximation Algorithms for Sorting by Reversals, with Application to Genome Rearrangement. Algorithmica **13(1/2)** (1995) 180–210
14. Blanchette, M., Kunisawa, T., Sankoff, D: Parametric genome rearrangement. Gene **172** (1996) GC11–GC17
15. Felsenstein, J.: Inferring phylogenies. Sinauer Associates (2003)
16. Hein, J., Wiuf, C., Knudsen, B., Moller, M.B., Wibling, G.: Statistical alignment: Computational properties, homology testing and goodness-of-fit. J. Mol. Biol. **203** (2000) 265–279
17. Larget, B., Simon, D.L., Kadane, B.J.: Bayesian phylogenetic inference from animal mitochondrial genome arrangements. J. Roy. Stat. Soc. B. **64(4)** 681–695
18. York, T.L., Durrett, R., Nielsen, R.: Bayesian estimation of inversions in the history of two chromosomes. J. Comp. Biol. **9** (2002) 808–818
19. Durrett, R., Nielsen, R., York, T.L.: Bayesian estimation of genomic distance. Genetics **166** (2004) 621–629
20. Miklós, I.: MCMC Genome Rearrangement. Bioinformatics **19** (2003) ii130–ii137
21. Miklós, I., Ittzés, P., Hein, J.: ParIS genome rearrangement server. Bioinformatics (2004) advance published, doi:10.1093/bioinformatics/bti060
22. Boore, J.L.: The duplication/random loss model for genome rearrangement exemplified by mitochondrial genomes of deuterostome animals. In: Sankoff, D., Nadau, J.H. (eds.): Comparative Genomics. Computational Biology Series **1** (2000) Kluwer Academic Publishers
23. Miklós, I., Lunter, G.A., Holmes, I.: A 'long indel' model for evolutionary sequence alignment. Mol. Biol. Evol. **21(3)** (2004) 529–540
24. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equations of state calculations by fast computing machines. J. Chem. Phys. **21(6)** (1953) 1087–1091
25. Liu, J.S.: Monte Carlo strategies in scientific computing. Springer Series in Statistics, New-York. (2001)
26. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57(1)** (1970) 97–109
27. von Neumann, J.: Various techniques used in connection with random digits. National Bureau of Standards Applied Mathematics Series **12** (1951) 36–38.
28. Nadau, J.H., Taylor, B.A.: Lengths of chromosome segments conserved since divergence of man and mouse. PNAS **81** (1984) 814–818