

Computing Genomic Midpoints

Yannet Interian* and Richard Durrett*,†

†Department of Mathematics and *Center for Applied Mathematics,
Cornell University, Ithaca, New York 14853

December 7, 2005

Abstract

This paper proposes a new algorithm for the genomic median problem that combines greedy and stochastic search. Our computational experiments suggest that for more complex problems our algorithm finds better solutions than previous approaches. In particular we find an improved midpoint for a human-mouse-rat comparison with 424 markers. In order to understand why such problems are hard, we explore a phase transition in the complexity of the median problem for random data, associated with the emergence of a giant component in the breakpoint graph.

1 Introduction

Genomes evolve not only by nucleotide substitutions but also by inversions that rearrange the order of genes on a chromosome, by translocations that exchange material between chromosomes, and by fissions and fusions that change chromosome number. Hannenhalli and Pevzner (1995a) developed a polynomial time algorithm for computing the inversion distance between two chromosomes, and later extended this to compute the distance between two genomes (Hannenhalli and Pevzner, 1995b). Their procedures are called parsimony methods by biologists since they use the shortest path to estimate the true evolutionary path.

In order to determine what rearrangement events took place and when they occurred in evolution, we need to consider multiple species. Here we are thinking of examples where the phylogeny is known. Simon and Larget (2001) and Larget et al. (2002) have used genome rearrangements to estimate phylogenetic relationships.

The *Median Problem* can be stated as follows: given three genomes $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ and a measure of distance d , find an ancestral genome \mathcal{G} that minimizes the total

distance $d(\mathcal{G}, \mathcal{G}_1) + d(\mathcal{G}, \mathcal{G}_2) + d(\mathcal{G}, \mathcal{G}_3)$. A more general problem, the *Multiple Genome Rearrangement Problem* (MGR), asks the following: given n genomes $\mathcal{G}_1, \dots, \mathcal{G}_n$ find a tree $T = (V, E)$ and genomes for the vertices, so that the leaves of T are the genomes \mathcal{G}_i $1 \leq i \leq n$, and the total distance $D(T) = \sum_{(G,H) \in E} d(G, H)$ is minimized.

We consider here the Median Problem for multichromosomal genomes with the *genomic distance*. The genomic distance is the minimum number of *basic rearrangements* needed to transform one genome into another. There are four kinds of *basic rearrangements*: *reversals*, *translocations*, *fusions* and *fissions* (see the appendix for precise definitions). For unichromosomal genomes we consider the *reversal distance*, which is the minimum number of reversals to transform one genome into the other.

Sankoff and Blanchette (1997,1998) have considered the Median Problem and the MGR Problem for unichromosomal genomes and the “breakpoint” distance $b(\mathcal{G}, \mathcal{G}')$, which is 1/2 the number of markers adjacent in one genome that fail to be adjacent in the other, rounded up to the next integer. Blanchette et al. (1999) used BPAAnalysis, an implementation of the breakpoint analysis (Blanchette et al., 1997) on a problem with 11 genomes and 35 markers. More recently, an improvement of the BPAAnalysis, called GRAPPA has been developed (Moret et al., 2001).

The Median Problem for the reversal distance was first considered in (Hannenhalli et al., 1995) for three herpes viruses. However, there were only 7 total changes in the minimum solution, so they could find the midpoint by examining all of the arrangements within a fixed distance of the three genomes. Bourque and Pevzner (2002) have recently proposed a new approach, the Multiple Genome Rearrangement algorithm (MGR-MEDIAN) which uses the genomic distance and applies to n species. Given three (multichromosomal) genomes $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$, a *good rearrangement* in genome \mathcal{G}_1 is a rearrangement that reduces the genomic distance between \mathcal{G}_1 and \mathcal{G}_2 and between \mathcal{G}_1 and \mathcal{G}_3 . Good rearrangements in genomes \mathcal{G}_2 and \mathcal{G}_3 , are defined similarly. For the case of three genomes, the MGR-MEDIAN algorithm iteratively looks for good rearrangements (if there are any) in $\mathcal{G}_1, \mathcal{G}_2$ and \mathcal{G}_3 , and performs them until each of the genomes is turned into the same genome. If it runs out of good rearrangements before the three genomes converge it relaxes the definition of good rearrangement and uses another heuristic to find the next rearrangement.

MGR-MEDIAN and GRAPPA are the most commonly programs for the Genomic Median Problem. There are three problems with GRAPPA. First, it only works for unichromosomal genomes. Second, it is based on the breakpoint analysis, which in general does not give accurate picture of the evolutionary distance. And third, when the complexity of the problem increases, the running time of the algorithm blows up. On the other hand, the MGR-MEDIAN algorithm is a fast algorithm

that performs very well on a variety of problems in which the original genomes are close to each other but, as we are going to discuss below, it fails to find the best median in more complicated scenarios.

In this paper, we describe a new approach to the median problem. Our algorithm combines *greedy search* (rearrangements that improve the objective function) with *non-improving* moves that allow the algorithm to escape from local minima. The algorithm starts at a (possibly random) genome and iteratively moves from one genome to a *neighbor* genome, which is obtained from the previous one by some *elementary rearrangement*.

Our algorithm has some advantages over previous approaches. It can be initialized at any point in the space and is a randomized algorithm. So it produces not only one solution but many good quality midpoint solutions. Comparing these solutions help us understand the confidence we have in various features of our predicted midpoints (Durrett and Interian, 2005). The non-greedy aspect of our algorithm allows us to search a larger space of solutions, and for some difficult data sets we have been able to find better solutions than the MGR-MEDIAN algorithm (Bourque and Pevzner, 2002). In particular, our analysis suggests new improved midpoint solutions for a comparison of Human-Mouse-Rat genomes (comparison constructed by C. Denwey and L. Pachter that appeared in the issue of Nature (Rat Genome Sequencing Consortium, 2004) announcing the sequencing of the brown Norway rat). We ran our algorithm starting at 100 random midpoints and found 19 solutions with value 346, 81 with value 347 in contrast with one solution found by MGR-MEDIAN with value 350 (personal communication from G. Bourque, 2005).

2 Algorithm

Our algorithm (MedRbyLS) starts at an arbitrary genome and iteratively performs a sequence of rearrangements to examine possible ancestral genomes. After a pre-defined number of steps the algorithm outputs the best midpoint seen so far. At each step t , the algorithm has a “current” genome \mathcal{G}_t and a set of elementary rearrangements \mathcal{M}_t to get from \mathcal{G}_t to \mathcal{G}_{t+1} . The algorithm chooses a rearrangement uniformly at random (u.a.r) from the set \mathcal{M}_t of elementary rearrangements. If applying the rearrangement gives a genome with a better total distance than \mathcal{G}_t the move is accepted, otherwise, the move is accepted with probability p . That is, any move that leads to a better genome is accepted (greedy move), otherwise “non-improving” moves are accepted with probability p . Usually we initialize our algorithm either with one of the three original genomes, or with a randomly chosen genome.

Given three genomes $\mathcal{G}^1, \mathcal{G}^2, \mathcal{G}^3$ and an initial midpoint genome \mathcal{G} , denote the total distance by $d_{\mathcal{G};\mathcal{G}^1,\mathcal{G}^2,\mathcal{G}^3} = d(\mathcal{G}, \mathcal{G}^1) + d(\mathcal{G}, \mathcal{G}^2) + d(\mathcal{G}, \mathcal{G}^3)$.

```

MedRbyLS(input  $\mathcal{G}^1, \mathcal{G}^2, \mathcal{G}^3$  and  $\mathcal{G}$ )
 $d = d_{\mathcal{G};\mathcal{G}^1,\mathcal{G}^2,\mathcal{G}^3}$ 
 $d_{min} = d, \mathcal{G}_{min} = \mathcal{G}$ 
For  $t=1$  to STEPS
    take  $\rho \in \mathcal{M}_t$  u.a.r
     $\mathcal{G}' = \rho \mathcal{G}$     let  $d' = d_{\mathcal{G}';\mathcal{G}^1,\mathcal{G}^2,\mathcal{G}^3}$ 
    if  $d > d'$  or else with probability  $p$ 
         $\mathcal{G} = \mathcal{G}', d = d'$ 
    if  $d_{min} > d'$ ,
         $d_{min} = d', \mathcal{G}_{min} = \mathcal{G}'$ 
output  $\mathcal{G}_{min}$ 

```

A key element in the success of the algorithm comes from the choice of the set \mathcal{M}_t of possible moves or rearrangements at time t . By results of (Hannenhalli and Pevzner, 1995b) given two genomes \mathcal{G} and \mathcal{G}' we can identify a sequence of rearrangements to get from \mathcal{G} to \mathcal{G}' in the minimum number of moves. We define $\mathcal{M}(\mathcal{G}, \mathcal{G}')$ to be a special subset of these moves (see the appendix for the exact definition) and define $\mathcal{M}_t = \cup_{i=1}^3 \mathcal{M}\{\mathcal{G}_t, \mathcal{G}^i\}$. Thus at each step, we are taking a step in an optimal path toward one (or more) of the original genomes.

Another important feature of the algorithm is its ability to make either strictly improving, “sideways” or “uphill” moves. The later moves are mechanisms to escape from local minima that allow us to search a larger space of solutions.

3 Results

We compare our algorithm with GRAPPA, an implementation of the breakpoint analysis (Moret et al., 2001) and with MGR-MEDIAN (Bourque and Pevzner, 2002). We use both simulated and real data for the comparison. The simulated data is generated as follows. For unichromosomal data we start with the identity permutation with n genes/markers, and we perform k random reversals to get each genome \mathcal{G}^i . For the case of multichromosomal genomes we start with identity permutation with n markers, we break it in five identical pieces, and then k rearrangements are applied at random (with probability 0.2 the rearrangements are translocations and with probability 0.8 are inversions).

3.1 Simulated data

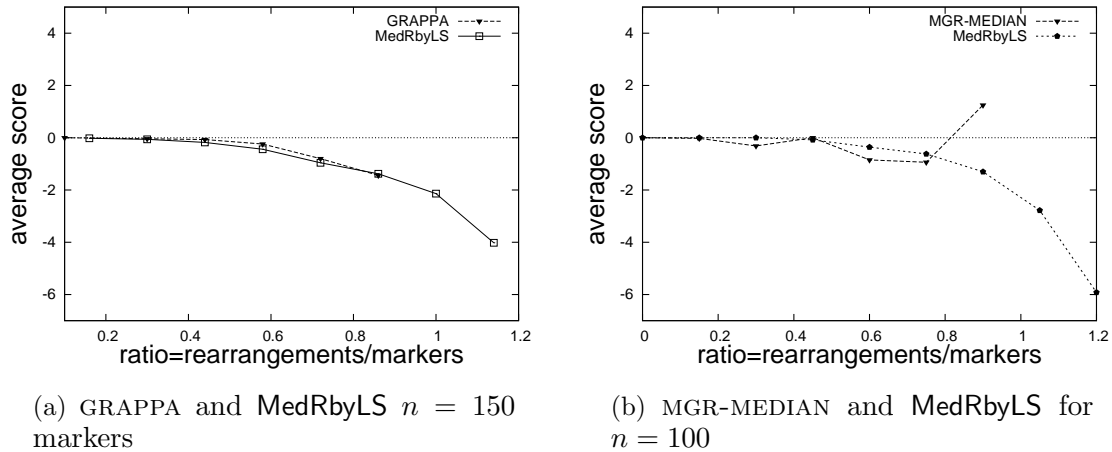


Figure 1: (a) Comparison between GRAPPA and MedRbyLS for one chromosome genomes, $n = 150$. The score is the sum of the distances between the midpoint found by the algorithm and the original genomes minus $3k$. Each point is the average value of the score, for 30 simulations with different data point, as a function of the ratio $r = 3k/n$. (b) Comparison between MGR-MEDIAN and MedRbyLS for $n = 100$.

Figure 1a gives a comparison of GRAPPA and MedRbyLS. The score is $\sum_{i=1}^3 d(\mathcal{G}, \mathcal{G}^i) - 3k$, the sum of the distance $\sum_{i=1}^3 d(\mathcal{G}, \mathcal{G}^i)$ between the genomes \mathcal{G}^i and the midpoint \mathcal{G} found by the algorithm, minus $3k$, the sum of the distances between the \mathcal{G}^i and the identity permutation. Figure 1a shows that the quality of the solutions found by GRAPPA and MedRbyLS are very similar until $r = 0.86$. But for the values $r = 1$ and $r = 1.14$ GRAPPA did not finish some of the instances in 24 hours.

Figure 1b shows the comparison between MGR-MEDIAN and MedRbyLS. Note that below the ratio $r = 0.75$ the results of MGR-MEDIAN and MedRbyLS are similar but for $r = 0.9$ the midpoints that MGR-MEDIAN finds are far from optimal. The values of the MGR-MEDIAN experiment were taken from the paper (Bourque and Pevzner, 2002). The solver web interface, the only publicly available version of this program, only allows data sets with at most 30 markers. [We were not able to get the executable from the authors].

As we can see from Figure 1 there are some values of the ratio r of rearrangements to markers for which GRAPPA and MGR-MEDIAN stop working while MedRbyLS still finds good solutions. In the case of GRAPPA the time for finding

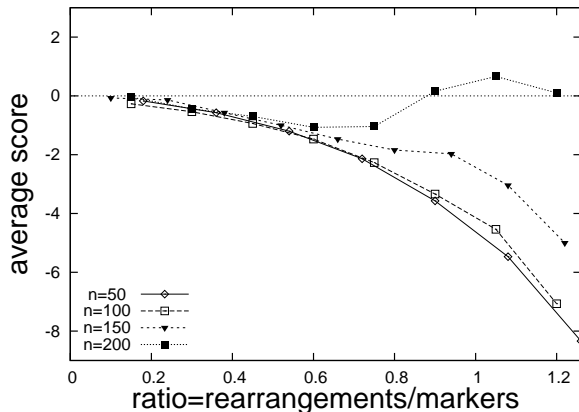


Figure 2: Results from MedRbyLS on generated data for multichromosomal genomes and $n = 50, 100, 150, 200$. Each point is the average value of the score, for 30 simulations with different data point, as a function of the ratio $r = 3k/n$.

solution blows up for $r = 1$ and $r = 1.14$ and for $r = 0.9$ MGR-MEDIAN does not find good solutions.

Looking at Figure 1, the reader might be surprised to see negative values. However, these are to be expected. If $k \geq n/4$, which corresponds to $r = 0.75$ the results in (Berestycki and Durrett, 2005) imply that one can get from \mathcal{G}^i to \mathcal{G}^j in fewer than $2k$ steps. Thus, it is not surprising that the median we found is closer than the ancestral genome.

Figure 2 shows the results on simulated data for multichromosomal genomes for $n = 50, 100, 150, 200$. For $n = 50$ and $n = 100$ we believe MedRbyLS finds midpoints very close to the best values, since we see curves similar to the ones from unichromosomal genomes. For $n = 150$ and 200 and for r greater than 0.6 MedRbyLS starts having difficulty finding the best midpoints. In particular for $n = 200$ and large values of r the score is positive.

The Median problem is NP-hard in general (Caprara, 1999a). However, for many biological and random data seems relatively easy to find optimal or rear-optimal solutions. We would like to understand why algorithms are not finding optimal solutions in some situations.

One possible explanation comes from changes in the structure of the breakpoint graph. Consider our randomly generated data. Let $r = 3k/n$ where n is the number of markers and k the number of rearrangements along each lineage from the identity genome. The breakpoint graph for random data is a random graph. In most random graph models, as a certain parameter increases the size of the largest component abruptly changes from having size of order $\log n$ to a giant

component with size of order n . We think that this phenomenon is responsible for the increase in difficulty in this random data. The plots in Figure 3 show the fraction of the number of markers in the largest component of the three-genome breakpoint graph as a function of r . The appearance of the giant component for this model seems to take place around $r = 0.6$.

When r is small, all the components are small. Intuitively, one can then attack the problem by considering the components separately. This cannot always be done, since this might lead to a midpoint with a circular chromosome. However, in practice one can use this approach on many problems and it breaks even very large problems into a number of simple small ones.

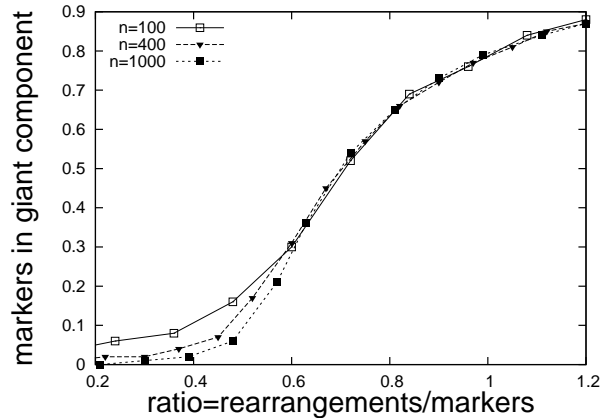


Figure 3: Fraction of the number of markers in the giant component of the three-genome breakpoint graph

3.2 Real data sets

In the companion paper (Durrett and Interian, 2005), we considered three examples: a human-lemur-tree shrew comparison derived from chromosome painting experiments, several versions with different resolutions of a human-cow-cat comparison of Murphy et al (2003), and a human-mouse-rat comparison constructed by C. Denwey and L. Pachter that appeared in the issue of Nature (Rat Genome Sequencing Consortium, 2004) announcing the sequencing of the brown Norway rat. We refer the reader to the companion paper for more details. In the human-mouse-rat comparison, 30 percent of the markers belong to the same component making the problem very hard to solve. We ran our algorithm starting at 100 random midpoints and found 19 solutions with value 346, 81 with value 347

in contrast with one solution found by MGR-MEDIAN with value 350 (personal communication G. Bourque, 2005).

References

- Bafna, V. and Pevzner, P. A. (1996). Genome rearrangements and sorting by reversals. *SIAM J. Comput.*, 25(2):272–289.
- Berestycki, N. and Durrett, R. (2005). A phase transition in the random transposition random walk. *Prob. Theor. Rel. Fields.*, to appear.
- Blanchette, M., Bourque, G., and Sankoff, D. (1997). Breakpoint phylogenies. *Miyano, S., and Takagi, T., eds. Genome Informatics*, pages 25–34.
- Blanchette, M., Kunisawa, T., and Sankoff, D. (1999). Gene order breakpoint evidence in animal mitochondrial phylogeny. *J. Mol. Evol.*, 49:193–203.
- Bourque, G. and Pevzner, P. A. (2002). Genome-scale evolution: reconstructing gene orders in the ancestral species. *Genome Research*, 12(1):26–36.
- Caprara, A. (1999a). Formulations and hardness of multiple sorting by reversals. In *Proceedings 3rd Conf. Computational Molecular Biology RECOMB99*, ACM Press, New York, pages 84–93.
- Caprara, A. (1999b). On the tightness of the alternating-cycle lower bound for sorting by reversals. *J. Comb. Optim.*, 3(2-3):149–182.
- Caprara, A. (2003). The reversal median problem. *INFORMS Journal on Computing*, 15(1):93–113.
- Consortium, R. G. S. (2004). Genome sequence of the brown norway rat yields insights into mammalian evolution. *Nature*, 428:493–521.
- Durrett, R. and Interian, Y. (2005). Genomic midpoints: Computation and evolutionary implications. *In preparation*.
- Hannenhalli, S., Chappey, C., Koonin, E. V., and Pevzner, P. A. (1995). Genome sequence comparison and scenarios for gene rearrangements: a test case. *Genomics*, 30(2):299–311.
- Hannenhalli, S. and Pevzner, P. A. (1995a). Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. In *Proceedings of the 27th Annual ACM Symposium on the Theory of Computing*, pages 178–189.

- Hannenhalli, S. and Pevzner, P. A. (1995b). Transforming men into mice (polynomial algorithm for genomic distance problem). In *FOCS '95: Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95)*, page 581, Washington, DC, USA. IEEE Computer Society.
- 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., and Kadane, J. (2002). Bayesian phylogenetic inference from animal mitochondrial genome rearrangements. *J. Roy. Stat. Soc.*, 64:681–693.
- Moret, B., Wyman, S., Bader, D., Warnow, T., and Yan, M. (2001). A new implementation and detailed study of breakpoint analysis. In *Proceedings of the 6th Pacific Symposium on Biocomputing (PSB2001), Hawaii*, pages 583–594.
- Simon, D. L. and Larget, B. (2001). Phylogenetic inference from mitochondrial genome arrangement data. In *International Conference on Computational Science (2)*, pages 1022–1030.

A Notations and Background

Genomes with one chromosome are represented as *signed permutations* where each integer corresponds to a gene or marker and the sign gives its orientation. A multichromosomal genome can be written as signed permutation divided into pieces called chromosomes. More precisely, given a set of markers $N = \{1, 2, \dots, n\}$, a *chromosome* τ is an ordering of a subset of the markers in which each marker has a sign, i.e., $\tau = (\tau_1 \dots \tau_m)$ where $|\tau_i| \in N$, $m \leq n$ and we identify $(\tau_1 \dots \tau_m)$ and $(-\tau_m \dots -\tau_1)$. A *genome* \mathcal{G} on a set of markers N is a collection of chromosomes in which each marker appears exactly once.

We consider four elementary kinds of rearrangements in a genome: *reversals*, *translocations*, *fusions* and *fissions*. A *reversal* $\rho = \rho_{i,j}$ of the interval (i, j) , $1 \leq i \leq j \leq m$, applied to a chromosome $\pi = (\pi_1 \dots \pi_m)$ takes π to

$$\pi \rho_{i,j} = (\pi_1 \dots \pi_{i-1} \quad -\pi_j \quad -\pi_{j+1} \dots \quad -\pi_{i+1} \quad -\pi_i \quad \pi_{j+1} \dots \pi_m)$$

A *translocation* $\rho = \rho_{i,j}$, $1 \leq i \leq m+1$, $1 \leq j \leq l+1$, applied to the two chromosomes $\pi = (\pi_1 \dots \pi_m)$ and $\tau = (\tau_1 \dots \tau_l)$ results in the two new chromosomes.

There are two possibilities. The simplest is

$$\{\pi; \tau\} \rho_{i,j} = \{(\pi_1 \dots \pi_{i-1} \tau_j \dots \tau_l), (\tau_1 \dots \tau_{j-1} \pi_i \dots \pi_m)\}$$

but we could also flip the first chromosome before translocating ending up with:

$$\{-\pi; \tau\} \rho_{i,j} = \{(-\pi_m \dots -\pi_i \tau_j \dots \tau_l), (\tau_1 \dots \tau_{j-1} -\pi_{i-1} \dots -\pi_1)\}$$

A *fusion* is a particular kind of translocation $\rho = \rho_{m+1,1}$ that concatenates two chromosomes π and τ resulting in a chromosome $(\pi_1 \dots \pi_m \tau_1, \dots, \tau_l)$ and an empty chromosome (we could also flip one of the chromosomes before fusing). A *fission* is the translocation $\rho = \rho_{i,1}$ that takes π and the empty chromosome resulting in two chromosomes $(\pi_1 \dots \pi_i)$ and $(\pi_{i+1} \dots \pi_m)$.

A.1 The breakpoint graph and the genomic distance

In the study of the genomic distance for unichromosomal genomes Kececioglu and Sankoff (1995) and Bafna and Pevzner (1996) introduced the *breakpoint graph* for signed permutations. Caprara (1999a) generalized this notion to study the median problem for unichromosomal genomes. In this paper we further extend the notion of the breakpoint graph to the case of multichromosomal genomes and to more than two species.

The first step is to double the markers. Consider the function u from the set of signed chromosomes to the unsigned chromosome such that $u(\tau) = (u(\tau_1), \dots, u(\tau_m))$, where for a signed marker $+x$, $u(+x) = 2x - 1, 2x$ and $u(-x) = 2x, 2x - 1$. Let $u(\tau_i) = x_{2i-1}x_{2i}$ for $1 \leq i \leq m$, that is, $u(\tau) = (x_1x_2, \dots, x_{2m-1}x_{2m})$. In this case the adjacency graph for the chromosomes will have edges

$$\{(x_{2i}, x_{2i+1}) : i \in 1, \dots, n\} \cup \{(H, x_1), (H, x_{2m})\}.$$

For a genome we apply this procedure to each chromosome and take the union to form the adjacency graph $\Gamma(\mathcal{G})$. The H 's in this graph denote the adjacencies to the chromosome "ends". They are all different, but we denote them with the same symbol to simplify the notation.

Given two genomes $\mathcal{G} = \{\tau^1, \dots, \tau^k\}$ and $\mathcal{G}' = \{\pi^1, \dots, \pi^l\}$, the breakpoint graph $\Gamma(\mathcal{G}, \mathcal{G}')$ is defined by combining the adjacency graphs $\Gamma(\mathcal{G})$ and $\Gamma(\mathcal{G}')$ using different labels H and H' for chromosome ends corresponding to different genomes, and different colors for edges, say *black* for \mathcal{G} and *gray* for \mathcal{G}' . See Figure 4 for an example. The dashed lines are gray edges.

Note that, except for the special nodes H and H' , all the other nodes in $\Gamma(\mathcal{G}, \mathcal{G}')$ have degree 2, and are incident with one gray and one black edge. There

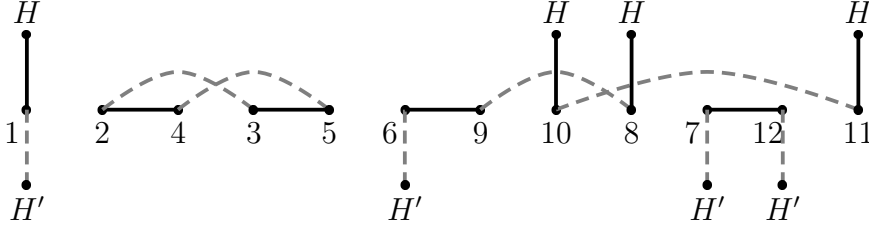


Figure 4: Figure 4: Breakpoint graph $\Gamma(\mathcal{G}, \mathcal{G}')$ for $\mathcal{G} = \{1 - 2 3 5; -4 - 6\}$ and $\mathcal{G}' = \{1 2 3; 4 5 6\}$

are $2k$ copies of node H and $2l$ copies of node H' , each of them with degree 1. Without loss of generality $k \geq l$. If $k > l$ we add $k - l$ empty chromosomes to \mathcal{G}' . We can write $\Gamma(\mathcal{G}, \mathcal{G}')$ in a unique way as a union of paths that start and end at the special nodes (with no special node in the middle of each path), and cycles of non-special nodes. Let $c(\mathcal{G}, \mathcal{G}')$ be the number of paths and cycles, including empty chromosomes. We call these the *components* of the graph. Let $\#(H, H')$ be the number of cycles that start in H and end with H' .

For example, consider in Figure 4 the breakpoint graph $\Gamma(\mathcal{G}, \mathcal{G}')$, for $\mathcal{G} = \{1-2 3 5; -4-6\}$ and $\mathcal{G}' = \{1 2 3; 4 5 6\}$. In this example $k = l = 2$, we have five components $\{H 1 H'\}$, $\{2 4 5 3 2\}$, $\{H' 6 9 8 H\}$, $\{H 10 11 H\}$ and $\{H' 7 12 H'\}$ so $c(\mathcal{G}, \mathcal{G}') = 5$, and the number of same genome cycles $\#(H, H) = \#(H', H') = 1$.

The *graph distance* between two genomes $\mathcal{G} = \{\tau^1, \dots, \tau^k\}$, $\mathcal{G}' = \{\pi^1, \dots, \pi^l\}$ is defined as

$$d(\mathcal{G}, \mathcal{G}') = n + k - c(\mathcal{G}, \mathcal{G}') + \#(H', H').$$

The graph distance for the example in Figure 4 is $6 + 2 - 5 + 1 = 4$. It is easy to check that we just need four rearrangements to transform \mathcal{G} into \mathcal{G}' .

In this paper we will work with the graph distance d instead of the harder to compute genomic distance, which we denote by d_0 . Computational experiments, and biological examples have shown that the graph distance is equal to the genomic distance in most cases. Furthermore, Caprara (1999b) showed that for random unichromosomal genomes both distances agree with high probability.

The next lemma, whose proof is a simple extension of Lemma 1 in (Caprara, 2003), shows that the graph distance d satisfies the triangle inequality.

Lemma 1. *Given three genomes $\mathcal{G}^1, \mathcal{G}^2, \mathcal{G}^3$ $d(\mathcal{G}^1, \mathcal{G}^2) + d(\mathcal{G}^1, \mathcal{G}^3) \geq d(\mathcal{G}^2, \mathcal{G}^3)$*

With Lemma 1 established, using an argument of Hannenhalli et al. (1995) shows that for any other genome M we have:

$$d(\mathcal{G}^1, M) + d(\mathcal{G}^2, M) + d(\mathcal{G}^3, M) \geq \frac{d(\mathcal{G}^1, \mathcal{G}^3) + d(\mathcal{G}^2, \mathcal{G}^3) + d(\mathcal{G}^2, \mathcal{G}^1)}{2} \quad (1)$$

A.2 Elementary rearrangements: definition of \mathcal{M}_t

We have two types of connected components in the breakpoint graph $\Gamma(\mathcal{G}, \mathcal{G}')$, one without special vertices $\mathcal{C} = \{x_1x_2 \dots x_{2r}\}$ that we call cycles, and a second type that begins and ends with special vertices: $\mathcal{C} = \{H'x_1x_2 \dots x_{2r+1}H\}$, $\mathcal{C} = \{Hx_1x_2 \dots x_{2r}H\}$, $\mathcal{C} = \{H'x_1x_2 \dots x_{2r}H'\}$ that we call paths.

We say that a component is *elementary* if it has the form $\{H x H'\}$ or $\{x y\}$. The first kind corresponds to a common end, the second one corresponds to x and y being adjacent in both genomes.

We say a rearrangement ρ acting on a cycle (path) of the breakpoint graph $\Gamma(\mathcal{G}, \mathcal{G}')$ is *elementary* if $\Gamma(\mathcal{G} \rho, \mathcal{G}')$ is obtained from $\Gamma(\mathcal{G}, \mathcal{G}')$ by one of the following operations:

- splitting one of the cycles, (with or without special vertices), \mathcal{C} into a 2-cycle and a smaller cycle. This applies to all paths and cycles
- splitting one of the cycles (paths) \mathcal{C} with special vertices into a 3-vertex path and a smaller path.

We define $\mathcal{M}\{\mathcal{G}, \mathcal{G}'\}$ as the set of elementary rearrangements acting on cycles (paths) of the breakpoint graph $\Gamma(\mathcal{G}, \mathcal{G}')$.