A Heuristic Algorithm for Genome Rearrangements [1]

Qian-Ping Gu qian@u-aizu.ac.jp
Kazuyuki Iwata m5011202@u-aizu.ac.jp
Shietung Peng s-peng@u-aizu.ac.jp
Qi-Ming Chen qmchen@u-aizu.ac.jp
The University of Aizu, Aizu-Wakamatsu, Fukushima 965-80, Japan

1 Introduction

Recently, a new approach to analyze genomes evolving was proposed which is based on the global rearrangements (e.g., inversions and transpositions of fragments). Given the sequences of the identical genes of two species, if we express one sequence by $I = (12...n)$ then the other sequence can be expressed by a permutation $\pi = (\pi_1\pi_2...\pi_n)$ of $\{1, 2, ..., n\}$. Checking the similarity between genomes based on global rearrangements leads to a combinatorial problem of finding a shortest series of rearrangements that sorts the permutation π into the identity I. A signed permutation is a permutation π on $\{1, 2, ..., n\}$ with $+$ or $-$ sign associated with every element π_i of π. The identity of signed permutations is $I = (+1 + 2 + 3 + ... + n)$. Signed permutations are more relevant to genome rearrangements, since genes are usually considered oriented in DNA sequences. In this paper, we propose a heuristic algorithm for sorting a signed permutation by transpositions and reversals.

2 Algorithm

Three rearrangements, reversal, transposition, and reversal+transposition, are considered in this paper. Let $\pi = (\pi_1\pi_2...\pi_n)$ be a permutation of $\{1, 2, ..., n\}$. For $1 \le i < j \le n + 1$, a *reversal* $r(i, j)$ reverses the order of $\pi_i\pi_{i+1}...\pi_{j-1}$. For $1 \le i < j \le n + 1$ and $1 \le k \le n + 1$ with $k \notin [i, j]$, a *transposition* $t(i, j, k)$ moves $\pi_i\pi_{i+1}...\pi_{j-1}$ to a new location of π between π_{k-1} and π_k; and a *reversal+transposition* $rt(i, j, k)$ reverses $\pi_i\pi_{i+1}...\pi_{j-1}$ and then moves $\pi_{j-1}...\pi_i$ to a new location of π between π_{k-1} and π_k.

Our algorithm makes use of the notion of *breakpoint graph* introduced by Bafna and Pevzner [1]. Let τ be a signed permutation of n elements. We transform τ to an unsigned permutation π of $2n$ elements as follows: replace $+i$ with $(2i - 1, 2i)$ and replace $-i$ with $(2i, 2i - 1)$ for $1 \le i \le n$. Notice that the identity $I = (+1 + 2... + n)$ is transformed into the unsigned identity $(1234...(2n - 1)2n)$. Next, we extend $\pi = \pi_1\pi_2...\pi_{2n}$ by adding $\pi_0 = 0$ and $\pi_{2n+1} = 2n + 1$. Let $i \sim j$ if $|i - j| = 1$. We call a pair of consecutive elements π_i and π_{i+1} an *adjacency* if $\pi_i \sim \pi_{i+1}$, otherwise a *breakpoint*. Define a *breakpoint graph* $G(\pi)$ of π as follows: There are $2n + 2$ nodes $0, 1, 2, ..., 2n + 1$ in $G(\pi)$. There is a grey edge between i and j if $i \sim j$ and i, j are not consecutive in π. There is a black edge between i and j if (i, j) is a breakpoint. When we refer to the breakpoint graph of a signed permutation, it is implied that we refer to the breakpoint graph of the transformed unsigned permutation.

Given a signed permutation π, let $b(\pi)$ be the number of the breakpoints and $c(\pi)$ be the number of cycles with odd number of black edges (breakpoints) in $G(\pi)$. Then it has been proved that $(b(\pi) - c(\pi))/2$ is a lower bound on the number of rearrangements for sorting π into I [2].

Call a rearrangement ρ an i-move on π if $\pi \cdot \rho = \pi'$ and $(b(\pi) - c(\pi)) - (b(\pi') - c(\pi')) = i$. It is known that $i \le 2$ for any rearrangement [2]. Our algorithm follows a greedy strategy which always executes a rearrangement with the maximum i to sort the given π. In particular, we uses the heuristic given in Figure 1 to find a 2-move $\rho(i, j, k)$.

3 Computer simulation and results

The algorithm has been tested on the following data:

[1] This work was supported by the grant-in-aid for scientific research on the priority area "Genome Science" from the Ministry of Education, Science, Sports and Culture of Japan.

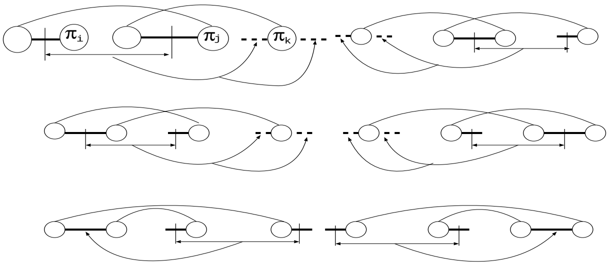

Figure 1: The heuristic for finding 2-moves.

a: Permutations $\pi = \pi_1\pi_2\cdots\pi_n$, each π_i is uniformly chosen from $\{1, 2, ..., n\}$ exclusively.

b: Permutations obtained by applying $f(n)$ random rearrangements to I, where a random rearrangement $\rho(i, j, k)$ is a rearrangement in which i, j, k are uniformly chosen from$[1, n]$.

The experiment results for the data of **a** and **b** are given in Figure 2 and 3, respectively. In the figures, 1 is the lower bound on the number of operations and 2 is the number of operations sorting π into I by the algorithm. For the data of **b** $f(n) = \sqrt{n}$.

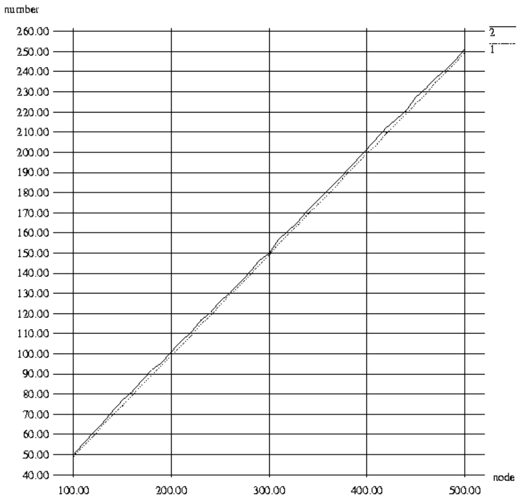

Figure 2:

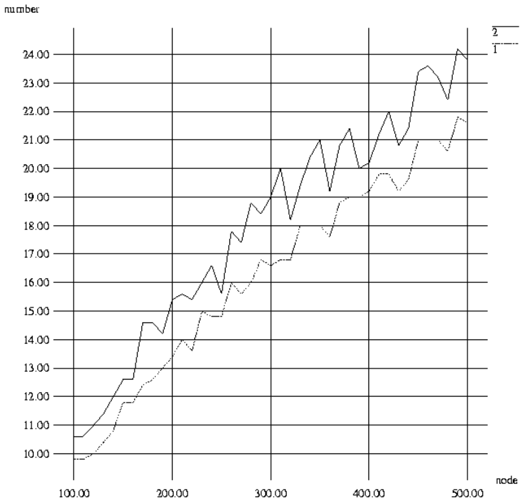

Figure 3:

References

[1] V. Bafna and P. Pevzner. Genome rearrangements and sorting by reversals. *SIAM J. on Computing*, 25(2):272–289, 1996.

[2] Q.P. Gu, S. Peng, and H. Sudborough. Approximation algorithms for genome rearrangements. In *Proc. of the 7th Workshop on Genome Informatics*, pages 13–22, 1996.