Multiple Sequence Alignment Using a Genetic Algorithm

Masamichi Isokawa
isokawa@cc.hirosaki-u.ac.jp

Masato Wayama
wayama@cc.hirosaki-u.ac.jp

Toshio Shimizu
slsimi@si.hirosaki-u.ac.jp

Department of Information Science, Faculty of Science,
Hirosaki University
Bunkyou-cho 3, Hirosaki 036 Japan

1 Introduction

The conventional multiple sequence alignment algorithms are classified into two categories: iterative improvement strategies (e.g., [1]) and simulated annealing methods (e.g., [6]). Recently, a genetic algorithm has been used in computational molecular biology as a powerful combinatorial optimizer. A genetic algorithm has been applied to the problem of multiple sequence alignment on a parallel computer (e.g., [7]).

In a simple genetic algorithm [3], a solution of a given problem is represented as "chromosomes" which consists of bit strings of 0's and 1's. The genetic operations, such as reproduction, crossover and mutation, are applied to a population of chromosomes to create a new population of chromosomes. This process is repeated many times so that we can obtain a nearly optimal alignment.

Here, we propose a improved method to apply a genetic algorithm to the problem of multiple sequence aligment. The processing was performed on a Fujitsu SPARCstation 20 and NEC UP4800.

2 Methods

We applied a genetic algorithm to the problem of multiple sequence alignment based on Goldberg's simple genetic algorithm. We define a chromosome as a N $\times$ M bit matrix [8] of which elements are strictly 0 or 1. A sequence, including gaps, in an alignment is represented as a bit string which consists of 0 and 1. In this bit string, '1' corresponds to a gap, with the total number of '0's being exactly the length of the sequence. The alignment is expressed with a matrix, which is a vertical arrangement of the bit strings.

Bit matrices as the first population are prepared in a random way: an element in each bit matrix is randomly determined to 0 or 1. The next population is generated by applying three genetic operations: reproduction, crossover and mutation.

The reproduction operation creates the next population from the matrices in the first population with use of tournament selection [4] and similarity score [2]. Next, "window-frame" crossover operation as shown in Figure 1 exchanges partly the information between two parent matrices selected randomly: the correspondence of each amino acid residue is strictly conserved in this operation. Then, "island-shift" mutation operation as shown in Figure 2 is applied to bit matrices in the next population: in this operation the resulted disorders in the amino acid residue correspondence are limited only in small regions of the bit matrix, i.e. "islands".

These processes are carried out repeatedly to obtain a nearly optimal alignment.

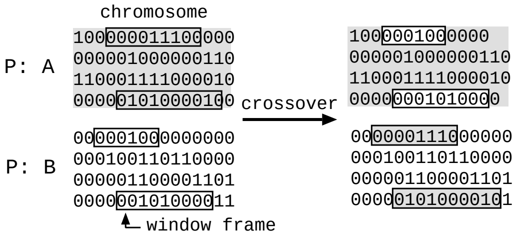

Figure 1: The "window-frame" crossover

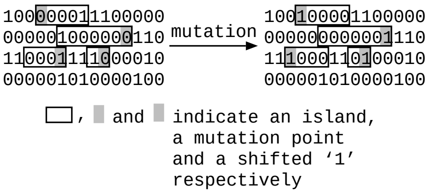

Figure 2: The "island-shift" mutation

3 Results and Discussion

We prepared two test sets which consist of 4 and 5 amino acid sequences in the database of SWISS-PROT release 30. The amino acid sequences were aligned with procedure described above. The alignment results are comparable to those obtained by CLUSTAL [5] which is the typical software for multiple sequence alignment based on the tree-based algorithm. In the case of alignment of short amino acid sequences, our method showed rather better quality results with high scores compared to CLUSTAL. It is also found that nearly optimal alignments could be obtained with this method.

The future objectives are to solve the problem of dependence of the genetic operations on random number sequence, to shorten the running time and to improve the quality of alignment further.

Acknowledgment

The computations have been performed with computers in the Hirosaki University Center for Computer and Communications, too.

References

[1] Berger, M. P. and Munson, P. J., *Comput. Applic. Biosci.*, Vol. 7, pp. 479-484, 1991.

[2] Dayhoff, M. O., Schwartz, R. M. and Orcutt, B. C., *National Biomedical Research Foundation, Washington, DC*, Vol. 5, Suppl. 3, pp. 345-352, 1978.

[3] Goldberg, D. E., *Addison-Wesley Publishing Company Inc.*, 1989.

[4] Goldberg, D. E., Korb, B. and Deb, K., *Complex Systems*, Vol. 3, pp. 493-530 1989.

[5] Higgins, D. G., Bleasby, A. J. and Fuchs, R., *Comput. Applic. Biosci.*, Vol. 8, pp. 189-191, 1992.

[6] Kim, J., Pramanik, S. and Chung, M. J., *Comput. Applic. Biosci.*, Vol. 10, pp. 419-426, 1994.

[7] Tajima, K., *Proc. Genome Informatics Workshop IV*, Universal Academy Press, pp. 183-187, 1993.

[8] Wayama, M., Takahashi, K. and Shimizu, T., *Proc. Genome Informatics Workshop 1995*, pp. Universal Academy Press, 122-123, 1995.