A Greedy Genetic Algorithm for Minimal Cost Convergence of Gene Sequences

Anupam Bhattacharjee, Kazi Zakia Sultana, Zalia Shams,

Tanjil Ahmed, A.K.M. Saifun Nabi

Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology,
BUET, Dhaka-1000, Bangladesh.
abrbuet@yahoo.com, z_sultana00@yahoo.com, setu_18@yhaoo.com,
tanim_buet62@yahoo.com, shabuj103@yahoo.com

Abstract

Researchers of bioinformatics, nowadays, focus on various applications of well-known algorithms in different fields of bioinformatics. One of the most famous algorithms in this regard is genetic algorithm. In this paper, we enhance the application of genetic algorithm on gene sequence related problems. Given two sets of gene sequences A and B, we are to find a minimal cost path to convert the set A to the set B through mutations and crossover operations. As the problem of minimum cost convergence has no polynomial time solution, researchers continue to find out experimental solutions such that the convergence cost becomes as small as possible. In this paper, we present a greedy modified genetic experimental algorithm to solve the problem of the minimal cost convergence of gene sequences. Experimental result shows that our algorithm performs better than the earlier algorithms. The solution to the problem has high significance on various bioinformatics methods in data mining, data cleaning, duplicate data detection, data clustering, mining frequent pattern, micro array data analysis and so on.

Keywords: Bioinformatics, Crossover, Data mining, Mutation.

I. INTRODUCTION

Researches in *Bioinformatics* have been growing rapidly with the growth of biological data everywhere in the world. Recent development in biotechnology has produced a massive amount of raw biological data which accumulates at an exponential rate (Fig. 1). Most of the data are erroneous or duplicate. That is why, *data mining* has ramified in various problems of bioinformatics.

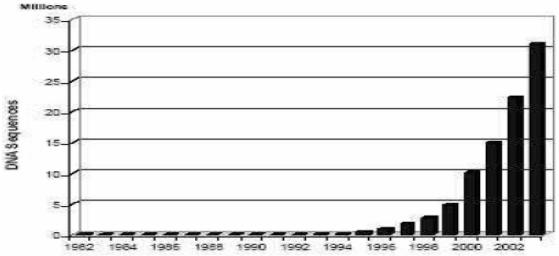

Figure 1: Growth of GeneBank

With the advent of different applications of various problems in bioinformatics, researchers are interested nowadays in the result of applying the earlier well-known general algorithms on those problems. One of such problems is the problem of *minimum cost convergence of gene sequences*. The problem can be defined as follows.

Given a set of *n* gene sequences *A*, the problem is to find the minimum cost to convert the set to another given set of *n* sequences *B* through *mutation* and *crossover* operations. The costs of a mutation and a crossover operation are given a priori.

As the problem is NP-Hard in general [1], to find an optimal solution to the problem, scientists are accentuating on the establishment of different experimental algorithms such that the convergence cost becomes as small as possible. The problem *minimal cost convergence of gene sequences* can be defined as follows.

We are given two sets *A* and *B*, each set contains *n* genes each of which has length *L*. The task is to find a set of crossover and mutation operations by which the genes of set *A* can be converted to the other set minimizing the total cost of the mutation and crossover operations. In other words, we have to find a one-to-one mapping from $A \rightarrow B$ after applying two types of operations:

i. A gene can be mutated to form another gene.
ii. Two genes can be crossed over at a position l $[1 \le l \le L]$

Parent	A C C T A G A
Offspring	A C G T A G T

Figure 2: Two mutations at 3rd and 7th positions

Parent1	A C C T | A G A
Parent2	T G A C | A A G
Offspring1	A C C T A A G
Offspring2	T G A C A G A

Figure 3: An example of crossover at 5th position.

Fig-2 and Fig-3 show two examples for each of the above operations.

The assumptions in the paper are:

1. The cost of a single mutation is 2.

2. The cost of a single crossover for a gene is 1, i.e., if two genes take part in a crossover operation then cost occurred for each gene is 1.

For a mutation operation two bonds need to be broken and two new bonds need to be built. But for a crossover operation, only one bond needs to be degenerated and a new bond needs to be created. If we take the cost of blotting negligible, the cost of mutation is higher than that of crossover which corroborates the assumptions.

Let us describe the problem by suitable examples. In Fig. 4 and Fig. 5, we are given the set *A* to be converted to set *B*. Here, $e = mutate(b)$ denotes that mutations occur at the proper positions of *b* to change it to *e*. The operation $(f, g) = cross\ (a, b, 2)$ means genes *a* and *b* are crossed over at the 2nd position resulting two new genes *f* and *g*. From the set of genes *a*, *b*, *c*, *d*, *e*, *f*, *g*, *h*, *i*, and *j*, in the first intermediate step in Fig. 4, "*choose e, g, h, j*" operation discards genes *a*, *b*, *c*, *d*, *f*, and *i*. Fig. 4 shows two intermediate steps with a high convergence cost. On the other hand, Fig. 5 shows another way with two intermediate steps and lower cost. So the objective of the solution is to find a set of steps optimizing total cost.

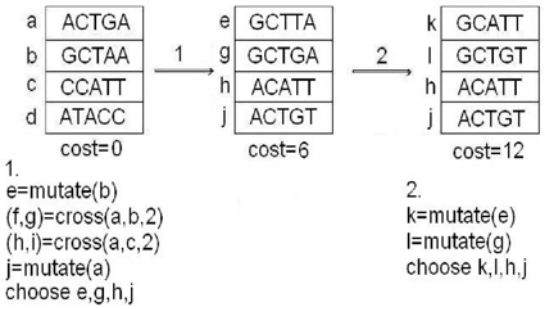

Figure 4: Gene sequence convergence through two intermediate steps with cost=12.

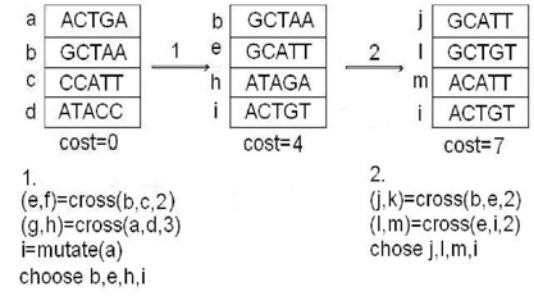

Figure 5: Gene sequence convergence through two intermediate steps with cost=7.

Gene sequence convergence problem has real world applications in clinical diagnosis, data cleaning, duplicate biological data detection, clustering of expressed gene data, microarray data analysis and so on. The goal of clinical diagnosis is to classify whether a sample is normal or not. While using the method of prognosis, the process of gene expression matching can be used to predict whether a patient will relapse or not [5], [8].

Moreover, with the increase of data, different types of errors and redundancies are introduced. The process of detecting and removing defective and duplicate data is referred to as data cleaning and usually carried out in proprietary or ad-hoc manner, sometimes manually [2], [5], [6]. For example, in the paper [2], [4], [8] stringent selection criteria is used to select 310 complete and unique records of *Homo sapiens* splice sites from 4300 raw records in the EMBL database. Although rigorous elimination of data is effective in removing redundancy, it may result in loss of critical information. Hence the approach of gene sequence convergence is much more preferable in this regard.

In the work of Koh, Lee, Khan, Tan and Brusic [2], a similar problem is introduced in the context of text mining. Again, the work of Li *et al*. [7] introduces the idea of using general algorithms in the context of gene sequence mapping problems. The work in [3] presents an experimental algorithm to the mapping problem with 82% accuracy.

In this paper, we present an experimental algorithm to solve the minimal cost convergence problem of gene sequences. The algorithm is a modification of the general genetic algorithm. Earlier works show that application of genetic algorithms in many problems of bioinformatics results better accuracy. We also applied a modified version of the general genetic algorithm and tested it against different known datasets e.g. EMBL dataset. The result corroborates that our algorithm is better than the previous algorithms along with dramatic speedup and accuracy (90%).

II. PRELIMINARIES

Gene is the segment of a DNA molecule and the fundamental biological unit of heredity. It carries information for the biosynthesis of a specific product in a cell. The information content in a DNA molecule comes from the specific sequence of its nucleotides. Only four different bases are used in DNA molecules: *Guanine*, *Adenine*, *Thymine*, and *Cytosine* (G, A, T, and C). The nitrogenous base guanine with its two-ringed structure is simply too large to pair with a two-ringed adenine or another guanine in the space that usually exists between two DNA strands. By the same token, the N-base thymine with its single-ringed structure is too small to interact with another single-ringed cytosine or thymine. Hence only the pairing between the nitrogenous bases G-C and A-T are stable. Each base is attached to a phosphate group and a deoxyribose sugar to form a nucleotide. Phosphate attaches to 5' carbon of one base sugar and the 3' carbon of the next. Thus, each nucleic acid strand has a 5' → 3' directionality.

Now, Changes to DNA occur due to mutation (Fig. 2) and crossover (Fig. 3). Mutation operates on a DNA sequence from one organism by randomly changing a single point on the strand in order to make a new DNA strand, whereas, crossover operate on two different DNA sequences by replacing part of a DNA sequence with that of another sequence in order to create a new DNA strand. Large scale gene expression mapping is motivated by the premise that the information on the functional state of an organism is largely determined by the information on gene expression.

III. PSEUDOCODE OF THE ALGORITHM

Definition:

leastCostMatch Algorithm: Given a set of m genes, with cost associated with each gene, the algorithm chooses n genes from them such that total cost of the chosen genes is minimum. There are many known algorithms of matching in this regard with $O(n^3)$ time complexity.

Input:

A: A set of n genes each of length L.
B: Another set of n genes each of length L.
threshold: given *threshold* value.

Output:

Minimal cost of convergence and intermediate steps.

Algorithm MinimalCostConvergence:

BEGIN
1. *max_cost* := *mutation_cost*(A, B);
2. *cost* := 0;
3. while (*cost* < (*max_cost* × *threshold*)) and (A != B)
   3.1. *temp_list* := A;
   3.2. for any two genes p and q in A
      3.2.1. for *position* := 1 *to* L
         3.2.1.1. (r, s) := *crossover*(p, q, *positon*);
         3.2.1.2. Choose two random positions pos_1 and pos_2 to mutate for genes r and s respectively;
         3.2.1.3. Apply mutation operation on the genes in the selected positions with probability $e^{\text{cost-max\_cost}}$;
   3.3. A := a list of chosen n genes from *temp_list* with *leastCostMatch* algorithm;
   3.4. *cost* = *cost* + cost of crossover + mutation for the genes in A;
4. if ((*cost* ≥ (*max_cost* × *threshold*)) and (A != B))
   cost := *cost* + *mutation-cost*(A, B);
5. return *cost*;
END

IV. DESCRIPTION

Our algorithm iteratively transforms the gene sequences of set A using crossovers and mutations until the cost of mapping(*cost*) from $A \rightarrow B$ exceeds a certain percentage(*threshold*) of the maximum cost(*max_cost*). Here, maximum cost is the cost of mapping the original gene sequences of set A to the gene sequences of set B using minimum number of mutations (*step 1*). In each iteration all possible pairs of genes of set A are crossed over at a common point between 1 and L engendering two new genes r and s. To avoid local extrima problem two positions pos_1 and pos_2 are chosen randomly from the genes r and s respectively. Then mutation operation is applied on the two genes in the selected positions with probability $e^{\text{cost-max\_cost}}$. From the set of original and newly created genes, we choose n genes that are best matched to n genes of set B (*step* 3.2 - 3.3). All these aforementioned operations are continued and the total cost increases iteratively until it exceeds a certain percentage of the maximum cost (*step*-3). The total cost of the process is the cumulative cost of the crossover and mutation operations at each iteration (*step*-3.4). If cost exceeds a certain percentage of the maximum cost before A is fully matched to B, the final mutation cost to convert A into B is added to current cost(*step*-4).

V. OBSERVATION

If A_i and A_{i+1} are two sets of genes obtained from two consecutive iterations, then the similarity between A_{i+1} and B is more than that of A_i and B, i.e., the convergence cost of A_{i+1} to B is less than that of A_i to B.

Illustration:

Let d and e be two genes in A_i and A_{i+1} respectively. Without loss of generality, we assume that all other genes of A_i and A_{i+1} are same. The algorithm chooses e in place of d if and only if the cost of matching of e in B is equal or less than that of d. So, the set A_{i+1} has lower convergence cost than A_i. Otherwise, if the cost is higher A_{i+1} would be equal to A_i. Thus, the observation is corroborated.

VI. RESPONSE TIME

Step 3 of the algorithm needs $O(n^2)$ time. Step 3.2 and 3.2.1 take $O(n^2)$ and $O(L)$ time respectively. Step 3.3 has $O(n^3)$ time complexity. So, total runtime of the algorithm is $O(n^3L)$.

VII. SIMULATION RESULT

Experiment shows that the algorithm differs very little as compared with the optimum cost for the number of sequences < 3000 (Fig. 6). Moreover, it can be shown from experiment results that the accuracy (Cost found by applying the algorithm divided by minimum cost found by Brute-Force method) of the method is almost 90% (Fig. 7). The accuracy is in the allowed limit used in detection of duplicate data in EMBL database. If it is possible to store all the intermediate results of crossover and mutation operations, it would be possible to find out the optimum result for larger set of gene sequences.

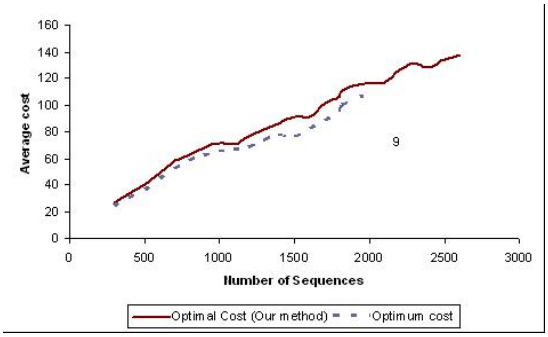

Figure 6: Minimal cost compared to Minimum cost

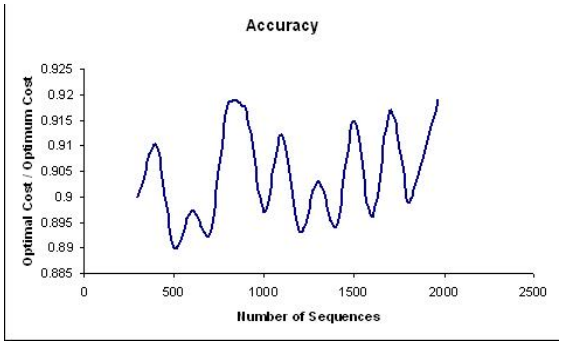

Figure 7: Accuracy Graph

VIII. CONCLUSION

In this paper, we presented a greedy algorithm for optimal gene sequence convergence problem. Experiment showed that our approach gave well performance comparing to optimum result. This problem can be extended to various gene related problems and has high significance on data mining and various bioinformatics approaches.

REFERENCES

[1] Rastogi, Mendiratta, and Rastogi, "*BIOINFORMATICS Methods and Applications. Genomics, Proteomics and Drug Discovery*", New Delhi, Prentics-Hall, 2004.

[2] Koh, J. L. Y., Lee, M. L., Khan, A. M., Tan P. T. J., and Brusic V., "Duplicate Data Detection in Biological Data using Association Rule Mining", *Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics*, 1992.

[3] Sese J., Kurokawa Y.,Monden M., Kato k., and Morishita S., "Constrained clusters of gene expressions profiles with pathological features", *Bioinformatics*, 20(17): 3137-3145, 2004.

[4] Blockeel H., De Raedt L., and Ramon J., "Top-down induction of clustering trees", *Proceedings of the Fifteenth International Conference on Machine Learning(ICML 1998)*, 55-63, Morgan kaufmann, 1998.

[5] Blaak J., and Kuba P., "Mining frequent patterns in complex structured data", *Proceedings of DATAKON*, 193-203, Brno, October, 2003.

[6] kuba P., "Mining frequent patterns in object-relational databases", *Proceedings of the international Conference on Knowledge Based Computer Systems(KBCS)*, 59-68, Mumbai, December 2002.

[7] Li J., Liu H., and Wong L., "Meanentropy Discretized Features are Effective for Classifying High-dimensional Biomedical Data", *Proceedings of 3rd ACM SIGKDD Workshop on Data Mining in Bioinformatics(BIOKDD03)*, 17-24, 2003.

[8] Li H., Marsolo K., Parthasarathy S., and Polshakov D., "A New Approach to Protein Structure Mining and Alignment", *Proceedings of 4th Workshop on Data Mining in Bioinformatics(BIOKDD04)*, 1-10, 2004.