2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

An Improved Genetic Algorithm for Operon Prediction

Li-Yeh Chuang[1], Ruei-Hung Tsai[2] and Cheng-Hong Yang [2+], Member, IEEE

[1] Department of Chemical Engineering, I-Shou University, Kaohsiung, Taiwan
[2] Department of Computer Science and Information Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan

Abstract. An operon is a fundamental unit of transcription and contains specific functional genes for the construction and regulation of networks at the whole genome level. The prediction of operons is critical to understanding gene regulation and function in newly sequenced genomes. As experimental methods for operon detection tend to be non-trivial and time-consuming, various methods have been used for predicting operons in the literature. In this study, an improved genetic algorithm (IGA) with reward and penalty (RP) is used for operon prediction in bacterial genomes. The intergenic distance, participation in the same metabolic pathway and the cluster of orthologous groups (COG) are used to design a fitness function based on RP. The RP operation improves the fitness value of a chromosome in proportion to the accuracy. Experimental results show that the prediction accuracy of this method reached 89.5%, 94.7%, 96.4%, and 97.3% on the E. coli K12 genome, Bacillus subtilis genome, Pseudomonas aeruginosa PA01 genome and Staphylococcus aureus genome, respectively. This method has enabled us to predict operons with high accuracy from four genomes where limited data exists on the properties of the operon structure.

Keywords: Genetic algorithm, Operon prediction, COG function, Metabolic pathway, Prediction accuracy.

1. Introduction

Operons contain one or more consecutive genes on the same strand in a prokaryote. These genes are co-transcribed into a single-strand mRNA sequence. Co-transcribed genes have the same biological function and directly affect each other. Operons contain valuable information for studying drug design and protein functions [1]. However, knowledge of operons is very scarce, and using experimental methods to predict operons is very difficult [2]. To gain insight, the number and organization of operons in bacterial genomes has to be studied carefully. Understanding the rules of transcription is critical to operon prediction. As a result, operons can be predicted based on an organism's genomic sequence.

In recent years, a number of scientists have proposed properties were proposed that can accurately predict operons. These properties can be divided into the following five categories [3]: intergenic distance, conserved gene clusters, functional relations, genome sequence, and experimental evidence. In each of the aforementioned categories, it is pivotal to detect the promoter and terminator at the operon boundaries to identify the biologically most representative properties [2]. The simplest and most important prediction property is to observe whether the distance between gene pairs within the operon (WO pairs) is shorter than the distance between gene pairs at the borders of transcription units (TUB pairs) [4]. The distance property reveals not only the simplest prediction property, but also yields good results on the operon prediction.

Jacob et al. proposed a fuzzy guided algorithm for operon prediction [2]. This method does not rely on complicated mathematical formulas to calculate fitness values of chromosome. Genetic algorithms (GA) [1] use the four biological properties of intergenic distance, metabolic pathway, cluster of orthologous groups

[+] Corresponding author. Tel.: + 886 7 381 4526 ext 5639; fax: +886 7 383 6844. *E-mail address*: chyang@cc.kuas.edu.tw gene function (COG) and microarray expression data to predict operons. The above studies only use the intergenic distance to initialize the population, whereas the importance of a strand is ignored. This prevents the GA from obtaining better chromosomes for initialization. If a low crossover rate and mutation rate is set for a GA, it is also difficult to find an optimal solution chromosome.

In this paper, we propose an effective genetic algorithm for operon prediction. To validate the method, four bacterial genomes (Escherichia coli K12-MG 1655, Bacillus subtilis, Pseudomonas aeruginosa PA01 and Staphylococcus aureus) were selected as benchmark genomes of known operon structure. A restriction was first introduced in the strand form to initialize a basis for the intergenic distance property. After initialization, the population was updated by the processes of crossover and mutation at each generation. The intergenic distance, metabolic pathway and COG function were used to calculate the fitness value of each gene in the chromosome based on reward and penalty (RP) [5]. Experimental results indicate that the proposed method applied to the four test data sets obtained higher accuracy, sensitivity, and specificity compared to the other methods from the literature.

2. Method

2.1. Data sets preparation

The entire genome data were downloaded from the GenBank database (http://www.ncbi.nlm.-nih.gov/). The operon databases of the E. coli and the B. subtilis were obtained from RegulonDB (http://-regulondb.ccg.unam.mx/) [6] and DBTBS (http://dbtbs.hgc.jp/) [7], respectively. The operon databases of the P. aeruginosa PA01 genome and the S. aureus genome were obtained from ODB (http://odb.kuicr.kyoto-u.ac.jp/) [8]. The genomes' metabolic pathway data and COG data were obtained from KEGG (http://www.genome.ad.jp/kegg/pathway.html) and NCBI (http://www.ncbi.nlm.nih.gov/COG/), respectively.

2.2. Genetic algorithm

Genetic algorithms (GAs) are stochastic search algorithms modelled on the process of natural selection underlying biological evolution. GAs have been applied to many search, optimization, and machine learning problems. The basic concept of GAs is designed to simulate evolutionary processes in natural systems, specifically those that follow the principle of survival of the fittest first laid down by Charles Darwin. As such, they represent an intelligent exploitation of random search within a defined search space to solve a problem. GAs proceed in an iterative manner by generating new populations of strings from old ones. Every string is the encoded binary, real, etc., version of a candidate solution. An evaluation function associates a fitness measure with every string and indicates its fitness for the problem. Standard GAs apply genetic operators, such as selection, crossover, and mutation on an initially random population in order to compute an entire generation of new strings.

2.2.1. Initial population

The proposed method uses intergenic distance and strands to create P binary chromosomes. Each chromosome is initialized with a random threshold value of between 0 and 600bps. Adjacent genes on different strands must be non-operon pairs (NOP). In other words, if a threshold is 75bps, and two adjacent genes are on the same strand with the intergenic distance of 70bps, the gene pair is determined to be in the same putative operon. If a threshold is 65bps, and two adjacent genes are on the different strands, the adjacent gene pair is determined to be in a different putative operon. For example, if gene3, gene4 and gene6 are the last genes of the operon1, operon2 and operon3, respectively, the elements of the array will be 110010.

2.2.2. Fitness evaluation

In this study, we use the intergenic distance, metabolic pathway and the COG gene function to calculate a fitness value. The pair-score of an intergenic distance is calculated by using the Local-entropy-minimization method [1]. If the largest distance frequency of adjacent gene pairs is -4bps, the maximum pair-score appears in the intergenic distance from -6bps to 1bps. If the adjacent genes are in the same metabolic pathway, the pair-score of the adjacent gene is 1, otherwise it is 0.The pair-score of the COG function is calculated by using the log-likelihood method [1]. If the adjacent genes are in the same COG functional categories, for example “Information storage and processing”, the pair-score of the COG function is 1.0733. If the gene pair is an operon pair (OP), the pair-score is obtained by the calculation method above. However, if the gene pair is a non-operon pair (NOP), the obtained pair-score is multiplied by -1. Finally, the entire pair-score of adjacent genes is calculated as the sum of the obtained pair-scores from the three properties mentioned above.

As shown in the Fig. 1, the pair-score of adjacent genes is calculated based on the metabolic pathway and the COG function. The obtained pair-score for gene1 and gene2 is -3. The RP pair-score is obtained by multiplying the pair-score by -1. The fitness value of the operon is calculated by multiplying the average OP with the gene number in the same operon.

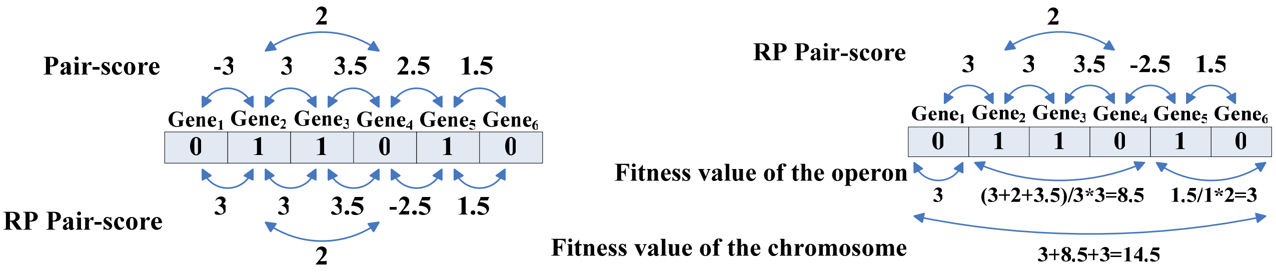

Fig. 1: Diagram of fitness evaluation

The fitness value of the cth putative operon is thus calculated by the following equation:

$$fitness_c = \sum_{i=1}^{m} d_i + \left(\frac{\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} S_{path}(gene_i, gene_j)}{n} + \frac{\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} S_{SOG}(gene_i, gene_j)}{n} \right) \times m \tag{2}$$

where di is the pair-score of intergenic distance of the ith gene in the cth operon, and m and n are the number of all genes and gene pairs in the cth operon, respectively.

Finally, the fitness value of a chromosome is calculated as the sum of fitness values from all putative operons in the chromosome and thus given by the following equation:

$$fitness = \sum_{i=1}^{c} fitness_i \tag{3}$$

where c is the number of operons in a chromosome.

2.2.3. Selection, crossover and mutation operations

In this study, the roulette wheel selection method is utilized for selecting two chromosomes with slots weighted in proportion to the fitness value of the chromosomes in the population. The chromosome with the higher fitness value is more likely to be selected into the next population.

The crossover operation is based on single-point crossover. Initially, a random position on the chromosome is chosen as a crossover point. Each of the two selected chromosomes is divided into two parts at this crossover point, and then the separated fragments are combined to form two different offsprings.

Two scenarios are possible for the mutation operator:

(1) If two adjacent genes belong to different operons in the same strand and the pair-score of the adjacent genes is greater than the average pair-score, the two operons are combined.

(2) If the putative operon and the next operon are in a different strand and the pair-score of the last gene pair of the putative operon is smaller than the average pair-score, the last gene is removed from the putative operon.

2.3. Parameter Settings

In the present study, the population number P was set to 20, the iteration number G was 100, the crossover rate (X-rate) was 1.0, and the mutation rate (M-rate) was 1/P [9]. The predictive performance [10] is evaluated based on the sensitivity and specificity shown in Table 1.

Table 1**:** Evaluation method for operon prediction

True data **Result of prediction**	+	-	**Evaluating the rate of positive and negative prediction**
+	True Positive (TP)	False Positive (FP)	**Positive Prediction Rate (PPR)** PPR=TP/(TP+FP)
-	False Negative (FN)	True Negative (TN)	**Negative Prediction Rate (NPR)** NPR=TN/(FN+TN)
Evaluating the accuracy of prediction	**Sensitivity** SN=TP/(TP+FN)	**Specificity** SP=TN/(FP+TN)	**Accuracy** ACC=(TP+TN)/(TP+FP+TN+FN)

Legend: The true positive and the true negative are the true predicted WO pair and the true predicted TUB pair, respectively.

3. Results and Discussions

Table 2: Accuracy, sensitivity, and specificity of operon prediction on four genomes

Genome	**Methodology**	**Accuracy**	**Sensitivity**	**Specificity**
E. coli	IGA	0.895	0.865	0.915
	IGA without RP	0.829	0.629	**0.967**
	Genetic Algorithm [1]	0.860	**0.894**	0.813
	Fuzzy Genetic Algorithm [2]	0.900	N/A	N/A
	Support vector machine [11]	0.856	0.888	0.802
	Using both genome-specific and general genomic information [10]	**0.937**	N/A	N/A
B. subtilis	IGA	**0.947**	**0.930**	**0.961**
	IGA without RP	0.850	0.833	0.871
	Genetic Algorithm [1]	0.883	0.873	0.897
	Fuzzy Genetic Algorithm [2]	0.882	N/A	N/A
	Support vector machine [11]	0.889	0.900	0.860
	Using both genome-specific and general genomic information [10]	0.902	N/A	N/A
P. aeruginosa PA01	IGA	**0.964**	**0.944**	**0.994**
	IGA without RP	0.793	0.670	0.988
	Genetic Algorithm [1]	0.813	0.870	0.763
S. aureus	IGA	**0.973**	**0.959**	**1.000**
	IGA without RP	0.845	0.767	**1.000**
	Genome-wide operon prediction in Staphylococcus aureus [4]	0.920	N/A	N/A

Legend: N/A represents non-available data.

An IGA was applied to search for the best putative operon at each generation. The best putative operon identified by this search was then compared to the experimental operon. In addition, IGA without RP was also applied for operon prediction. Table 2 shows the comparison of the results obtained by the proposed method and other methods reported in the literature. The various reported methods include a genetic algorithm [1], a fuzzy genetic algorithm [2], support vector machine [11], and using both genome-specific and general genomic information [10]. As Table 2 shows the prediction accuracy of the proposed method obtained the highest accuracy value on the B. subtilis (0.947), P. aeruginosa PA01 (0.964), and S. aureus (0.973) data sets. For the E. coli data set, IGA obtained the third highest accuracy (0.895). The proposed method also showed the best performance in terms of prediction sensitivity and specificity on most of the test bacterial genomes. Overall, the method proposed in this study obtained better results than the other methods tests for operon prediction.

Most prediction methods predict operons base on the properties of adjacent genes, which they try to identity as either OP or NOP. However, this procedure does not take the properties of near genes into account, and this result in generally lower accuracies for operon prediction. The IGA used in this study evaluates the properties of all genes, thereby increase the probability of finding an optimal solution. Gas, however, tend to get trapped in local optimal if a smaller mutation rate is used. In order to improve the performance of the GA, we set a higher crossover rate and a higher mutation rate to overcome the lack of chromosome updates at each generation. The probability of obtaining the best solution is thus increased.

Operon prediction accuracy can be increased if better chromosomes are selected in the initial step. Since the benefits of the initially superior chromosome are multiplied through the continued updating at each generation. In our study, the intergenic distance and the gene strand condition were evaluated for the initiation step. We obtained a very high specificity when the initiation threshold was set at 300bps. When the threshold was adjusted to 600bps, the sensitivity was raised even further, but the specificity was reduced. This indicates that in order to increase the sensitivity, specificity and accuracy of operon prediction, the initiation step is the critical point. By boosting the quality of chromosomes at the initiation, the best chromosome can be obtained through successive progression through the generations.

In the currently used methods the fitness value of a chromosome is not proportional to the prediction accuracy. Even if adjacent genes have relative properties, they might be in the different operons. This necessitates the implementation of the fitness function in the proposed method. In the literature methods, the GA only calculates the fitness value of an operon but not that of an individual gene. Since individual genes are not evaluated in the literature methods, the fitness value is higher when a chromosome has many operons which contain more genes. Therefore, we calculated the pair-score of each gene and also evaluated the fitness value of each gene. The fitness function was designed based on RP. Basically, if an adjacent gene has a higher score, then that score will be multiplied by -1 in order to change the gene's fitness value into a lower score. The experimental results prove that this fitness function enables us to identify better chromosomes.

Experimental data on the E. coli genome can be downloading from the RegulonDB database, but for other genomes extensive experimental data is not readily available. In order to apply the proposed method to other genomes which contain fewer attributes, only three common properties for operon prediction were used. As shown in Table 2, Wang's method, which uses four properties, led to better results than IGA without RP. Generally, methods using more properties for operon prediction achieve a higher accuracy. This could explain why the accuracy and sensitivity of the proposed method were not higher than two from the literature for the E. coli genome. For other bacterial genomes, however, our method achieved the highest accuracy of operon prediction.

4. Conclusions

In this paper, we propose a novel operon prediction method, called IGA. This study not only uses the restriction of strand, and the intergenic distance is commutated at the initiation step. IGA can obtain better chromosomes at the initialization of a population. We used the intergenic distance, metabolic pathway and COG gene functions based on RP to design a fitness function. The experimental results show that the proposed method not only increases the accuracy of operon prediction on four genome data sets, but also reduces the computation time for operon prediction. In the future, we intend to use different properties and other algorithms on the problems of operon prediction in order to further enhance the prediction performance.

5. References

[1] S. Wang, Y. Wang, W. Du, F. Sun, X. Wang, C. Zhou, and Y. Liang. A multi-approaches-guided genetic algorithm with application to operon prediction. *Artificial Intelligence in Medicine,* 2007, 41: 151-159.

[2] E. Jacob, R. Sasikumar, and K. N. R. Nair. A fuzzy guided genetic algorithm for operon prediction. *Bioinformatics,* 2004, 21 (8): 1403-1407.

[3] R. W.W. Brouwer, O. P. kuipers, and S. A. F. T. van Hijum. The relative value of operon predictions. *Bioinformatics,* 2008, 9 (5): 367-375.

[4] L. Wang, J. D. Trawick, R. Yamamoto, and C. Zamudio. Genome-wide operon prediction in Staphylococcus aureus. *Nucleic Acids Research,* 2004, 32 (12): 3689-3702.

[5] M. E. Harmon and Harmon, S.S. *Reinforcement learning: a tutorial,* 1996.

[6] M. Pertea, K. Ayanbule, M. Smedinghoff, and S. L. Salzberg. OperonDB: a comprehensive database of predicted operons in microbial genomes. *Nucleic Acids Research,* 2008, 1-4.

[7] N. Sierro, Y. Makita, M. D. Hoon, and K. Nakai. DBTBS: a database of transcriptional regulation in Bacillus subtilis containing upstream intergenic conservation information. *Nucleic Acids Research,* 2008, 36: D93-D96.

[8] S. Okuda, T. Katayama, S. Kawashima, S. Goto, and M. Kanehisa. ODB: a database of operons accumulating known operons across multiple genomes. *Nucleic Acids Research,* 2006, 34: D358-D362.

[9] O. K. Erol, and I. Eksin. A new optimization method: Big Bang-Big Crunch. *Advances in Engineering software*, 2006, 37: 106-111.

[10] P. Dam, V. Olman, K. Harris, Z. Su, and Y. Xu. Operon prediction using both genome-specific and general genomic information. *Nucleic Acids Research*, 2007, 35 (1): 288-298.

[11] G. Q. Zhang, Z. W. Cao, Q. M. Luo, Y. D. Cai, and Y. X. Li. Operon prediction based on SVM. *Comput Biol Chem*, 2006, 30: 233-240.