Operon Prediction Using Chaos Embedded Particle Swarm Optimization

Li-Yeh Chuang, Cheng-Huei Yang, Jui-Hung Tsai, and Cheng-Hong Yang

Abstract—Operons contain valuable information for drug design and determining protein functions. Genes within an operon are co-transcribed to a single-strand mRNA and must be coregulated. The identification of operons is, thus, critical for a detailed understanding of the gene regulations. However, currently used experimental methods for operon detection are generally difficult to implement and time consuming. In this paper, we propose a chaotic binary particle swarm optimization (CBPSO) to predict operons in bacterial genomes. The intergenic distance, participation in the same metabolic pathway and the cluster of orthologous groups (COG) properties of the *Escherichia coli* genome are used to design a fitness function. Furthermore, the *Bacillus subtilis*, *Pseudomonas aeruginosa* PA01, *Staphylococcus aureus* and *Mycobacterium tuberculosis* genomes are tested and evaluated for accuracy, sensitivity, and specificity. The computational results indicate that the proposed method works effectively in terms of enhancing the performance of the operon prediction. The proposed method also achieved a good balance between sensitivity and specificity when compared to methods from the literature.

Index Terms—Operon, particle swarm optimization, chaos

1 Introduction

Valuable information for drug design and protein functions can be acquired from the operons of bacterial genomes [1]. Operons in prokaryote organisms contain one or more consecutive genes on the same strand. The genes are cotranscribed into a single-strand mRNA sequence and are, thus, likely to have the same biological functions, as well as affect each other directly. Understanding the gene regulations is, thus, critical for improving the operon prediction process. However, information regarding operons is scarce, and experimental methods for predicting operons are generally difficult to implement. To gain a deeper insight, operon-related research projects have to be investigated in further detail. In recent years, many operon prediction features have been proposed in the literature. Features commonly used to determine the existence of an operon are the intergenic distance, metabolic pathway, homologous genes, terminator, gene order conservation, clusters of orthologous groups, and the gene length ratio [2]. Out of the above features, the promoter and the terminator property in the genome sequence feature are the most representative properties [3]. The intergenic distance is the simplest and most widely used prediction property. It is used to observe whether the distance between gene pairs within an operon (WO pairs) is shorter than the distance between gene pairs at the transcription unit borders (TUB pairs) [4].

Scientists have proposed many methods to predict operons, including the Bayesian method [5], [6], [7], [8], machine learning [9], clustering approaches [10], logistic regression method [11], and graphical theoretic approaches [12], [13]. Some advanced techniques use artificial intelligence to predict operons; genetic algorithms [1], [3], [14], particle swarm optimization (PSO) [2], and neural networks [15], [16] fall into this category. These artificial intelligence methods have shown a high operon prediction accuracy. Some databases that fall into neither of the above two categories have been constructed and made available, for example, RegulonDB [17], DBTBS [18], DOOR (Database of prOkaryotic OpeRons) [19], MicrobesOnline [20], and the ODB (Operon Database) [21].

In this study, we propose a chaotic binary particle swarm optimization (CBPSO) to predict operons. PSO constitutes a randomized search and optimization technique that derives its working principles from the social behavior of organisms. Chaos is a nonlinear system with deterministic dynamic behavior. It has stochastic and regularity properties, as well as ergodicity, and is very sensitive to the initial conditions and parameters. Small differences in the initial conditions result in great differences after many iterations [22]. These characteristics of a chaotic system can be used to enhance the search ability of PSO. The *Escherichia coli (E. coli)* genome was selected for training the genome based on the intergenic distance, the participation in the same metabolic pathway, and the clusters of orthologous groups (COG). The *Bacillus subtilis* (*B. subtilis*), *Pseudomonas*

- *L.-Y. Chuang is with the Department of Chemical Engineering, Institute of Biotechnology and Chemical Engineering, I-Shou University, No. 1, Sec. 1, Syuecheng Road, Dashu District, Kaohsiung 84001, Taiwan, R.O.C. E-mail: chuang@isu.edu.tw.*
- *C.-H. Yang is with the Department of Electronic Communication Engineering, National Kaohsiung Institute of Marine Technology, No. 142, Haijhuan, Road, Nandh District, Kaohsiung 81157, Taiwan, R.O.C. E-mail: chyang@mail.nkmu.edu.tw.*
- *J.-H. Tsai is with the Department of Electronic and Communication Engineering, National Kaohsiung Marine University, Kaohsiung, Taiwan, R.O.C. E-mail: bigblack918@hotmail.com.*
- *C.-H. Yang is with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan, R.O.C. E-mail: chyang@cc.kuas.edu.tw.*

Manuscript received 15 June 2012; revised 10 May 2013; accepted 13 May 2013; published online 20 May 2013.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-2012-06-0147.
Digital Object Identifier no. 10.1109/TCBB.2013.63.

1545-5963/13/$31.00 © 2013 IEEE Published by the IEEE CS, CI, and EMB Societies & the ACM

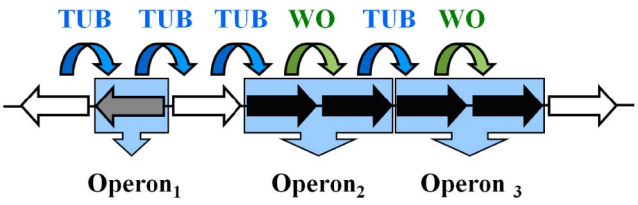


Fig. 1. WO and TUB pairs. The white arrows represent genes that are experimentally unclassified, and the gray arrow represents a singleton operon. In addition, the black arrows represent operons that consist of several genes.

aeruginosa PA01 (*P. aeruginosa* PA01), *Staphylococcus aureus* (*S. aureus*), and *Mycobacterium tuberculosis* (*M. tuberculosis*) genome were selected as target genomes. The CBPSO computational results demonstrate that the prediction ability of CBPSO is superior to the other methods from the literature it was compared to [8], [9], [10], [11], [12], [13], [14], [15], [16].

2 Methods

2.1 Data Set Preparation

The entire genome data of *E. coli*, *B. subtilis*, *P. aeruginosa* PA01, *S. aureus*, and *M. tuberculosis* were downloaded from the GenBank database (http://www.ncbi.nlm.nih.gov/). The related genomic information contains the gene name, the gene ID, the position, the strand, and the product. The experimental operon data set of the *E. coli* and *B. subtilis* genomes were obtained from RegulonDB (http://regulondb.ccg.unam.mx/) [17] and DBTBS (http://dbtbs.hgc.jp/) [18], respectively, which contains highly reliable data of validated experimental operons of the *E. coli* and *B. subtilis* genomes [19]. The experimental operon data sets of the *P. aeruginosa* PA01, *S. aureus*, and *M. tuberculosis* genomes were obtained from ODB (http://www.genome.sk. ritsumei.ac.jp/odb/) [21]. The metabolic pathway and COG data of the genomes were obtained from KEGG (http://www.genome.ad.jp/kegg/pathway.html) and NCBI (http://www.ncbi.nlm.nih.gov/COG/), respectively.

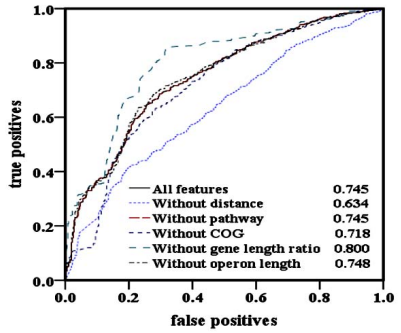
2.2 Operon Pairs

An operon is defined as a sequence of one or more genes that, under certain conditions, are transcribed as a unit. Adjacent genes in the same operon are called a WO pair. If the operon contains a single gene and the downstream gene is of unknown status, the gene pair is called a TUB pair. However, if the upstream gene is the last gene of an operon, then the downstream gene is of uncertain status, and thus the gene pair cannot be labeled a TUB pair [5]. Fig. 1 shows a simple illustration of WO and TUB pairs.

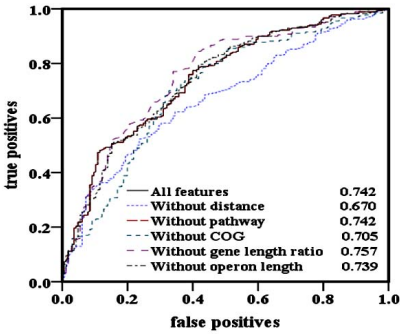
2.3 Operon Properties

2.3.1 Features Selected for Operon Prediction

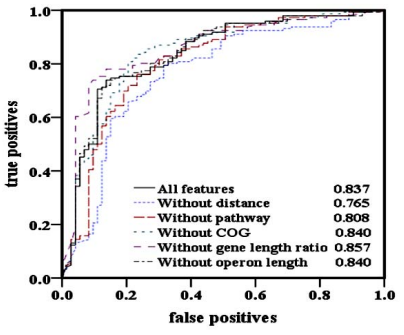
Five properties were originally considered for the prediction of operons, i.e., the intergenic distance, the metabolic pathway, the COG gene function, the gene length ratio, and the operon length. However, Fig. 2 indicates that the gene length ratio and the operon length are not as suitable for operon prediction as the other three features. Thus, we selected the intergenic distance, the metabolic pathway, and the COG gene function to predict operons. The intergenic



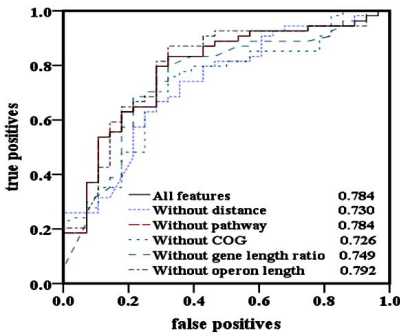
(a) *B. subtilis* genome



(b) *P. aeruginosa* genome



(c) *S. aureus* genome



(d) *M. tuberculosis* genome

Fig. 2. ROC curves of operon prediction. The false-positive rate is plotted as the abscissa and the true-positive rate as the ordinate. (a) *B. subtilis* genome. (b) *P. aeruginosa* genome. (c) *S. aureus* genome. (d) *M. tuberculosis* genome.

distance property not only plays an important role in the initial step but also yields good prediction results [3], [16], [17], [23], [24]. This property can be used to universally predict operons in bacterial genomes with a completed chromosomal sequence. In the functional relations category, we used the metabolic pathway and the COG gene function

to predict operons. The metabolic pathway property has a high prediction accuracy on the *E. coli* data set, as indicated in the literature [3]. When adjacent genes have the same pathway, the probability of a pair being within the same operon is very high. The reason we selected the COG gene function is that genes which belong to the same first-level functional category or fall into the fourth category have a probability of 83.5 percent of being within the same operon on the *E. coli* genome [25]. However, since the metabolic pathway and the COG gene function belong to the functional relations category, the method only searches regions where these properties overlap [3]. Since the same prediction results were obtained when either one of these properties was used, is should be noted that the metabolic pathway property is more efficient for operon prediction. However, the metabolic pathway property only determines whether adjacent genes have the same pathway or not, and thus COG must be used to estimate if a gene is within a functional category. A detailed description of the above mentioned three properties is given below.

2.3.2 Intergenic Distance

As shown in (1), the intergenic distance is calculated based on the base pairs of adjacent genes. In general, the distance of WO pairs is shorter than the distance of TUB pairs [23]. The maximum frequency of the WO pairs distance is -4 bps [26]. The distribution frequency of the TUB pairs increases with the distance. The intergenic distance distribution of WO and TUB pairs of the *E. coli*, *B. subtilis*, *P. aeruginosa* PA01, and *S. aureus* genomes are shown in Figs. 3a, 3b, 3c, and 3d. The figures indicate that property can be effectively used for operon prediction

$$\text{Distance} = \text{Gene}_2\_\text{start} - (\text{Gene}_1\_\text{end} + 1). \quad (1)$$

2.3.3 Metabolic Pathway

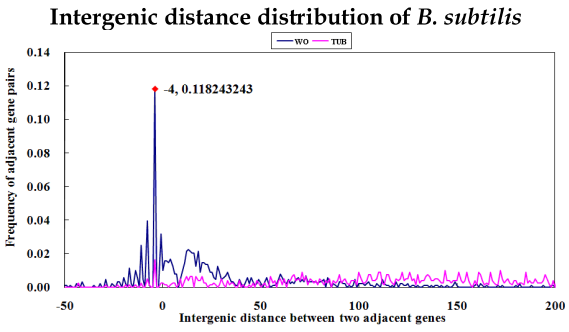
Three levels of biological functions commonly used in gene ontology are the biological process, the molecular function, and the cellular component [27]. Genes within an operon often have the same biological function [1]. Therefore, if adjacent genes are annotated with the same metabolic pathway, we can infer that the gene pair is from the same operon.

2.3.4 COG Gene Function

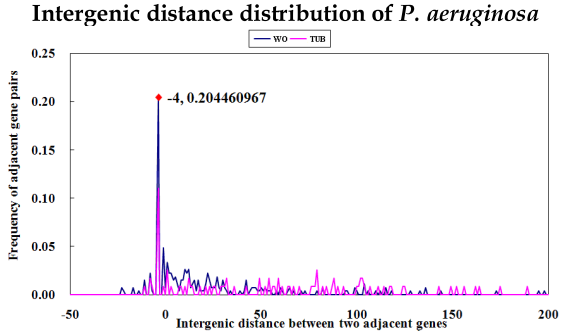
COG consists of three main levels. The first level contains four classes, namely, information storage processing, cellular processing, signaling, and metabolism. Each of the classes is divided into multiple functional categories, and adjacent genes often remain in the same class [15]. Hence, we consider a gene pair in a same operon when the adjacent genes are within the same class.

2.4 Binary Particle Swarm Optimization

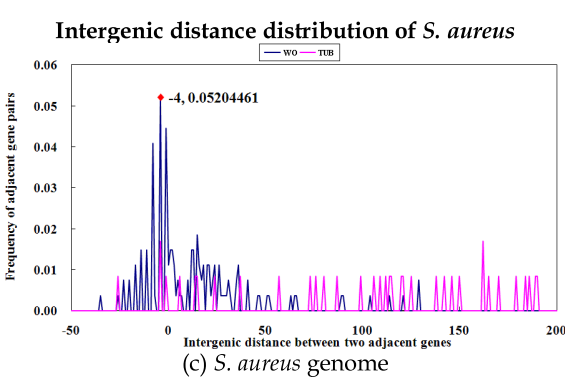
Particle swarm optimization is a population-based evolutionary computation technique developed by Kennedy and Eberhart [28]. The concept of PSO was developed through the observation of the social behavior of birds in a flock or fish in a school. Each individual is affected by its past experience and the swarm behavior. In PSO, each solution can be considered an individual particle in a given search space, which has its own position and velocity. During movement, each particle adjusts its position by changing its velocity based on its own experience, as well as the experience of its companions, until an optimum position is reached by itself and its companions [29]. All of the particles have fitness values based on the calculations of a fitness function. Particles are updated by following two parameters called *pbest* and *gbest* at each iteration. Each



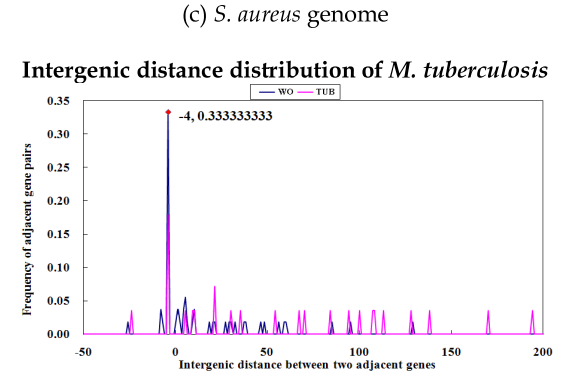
(a) *B. subtilis* genome



(b) *P. aeruginosa* PA01 genome



(c) *S. aureus* genome



(d) *M. tuberculosis* genome

Fig. 3. Intergenic distance distribution diagram. The diagram shows the intergenic distance distribution of WO and TUB pairs of the (a) *B. subtilis* genome, (b) *P. aeruginosa* PA01, genome, (c) *S. aureus* genome, and (d) *M. tuberculosis* genome.

particle is associated with the best solution (fitness) the particle has achieved so far in the search space. This fitness value is stored, and represents the position called *pbest*. The value *gbest* is the global optimum value for the entire population.

PSO was originally developed to solve real-value optimization problems. Many optimization problems occur in a space featuring discrete, qualitative distinctions between variables and levels of variables. To extend the real-value version of PSO to a binary/discrete space, Kennedy and Eberhart proposed a binary version of the PSO method (BPSO).

The position of each particle is represented in binary string form by $X_p = \{X_{p1}, X_{p2}, \ldots, X_{pd}\}$ and is randomly generated. The bit values {0} and {1} represent a nonselected and a selected feature, respectively. The velocity of each particle is represented by $V_p = \{V_{p1}, V_{p2}, \ldots, V_{pd}\}$ (p is the number of particles, and d is the number of dimensions (features) of a given data set). The initial velocities in particles are probabilities limited to a range of $\{0.0 \sim 1.0\}$. In BPSO, once the adaptive values *pbest* and *gbest* are obtained, the features of the *pbest* and *gbest* particles can be tracked with regard to their position and velocity. Each particle is updated based on the following equations:

$$v_{pd}^{new} = w \times v_{pd}^{old} + c_1 \times r_1 \times \left(pbest_{pd} - x_{pd}^{old}\right) + c_2 \times r_2 \times \left(gbest_d - x_{pd}^{old}\right), \tag{2}$$

if $v_{pd}^{new} \notin (V_{\min}, V_{\max})$, then

$$v_{pd}^{new} = \max\left(\min\left(V_{\max}, v_{pd}^{new}\right), V_{\min}\right), \tag{3}$$

$$S\left(v_{pd}^{new}\right) = \frac{1}{1 + e^{-v_{pd}^{new}}}, \tag{4}$$

$$\text{If}\left(rand < S\left(v_{pd}^{new}\right)\right), \text{ then } x_{pd}^{new} = 1; \text{else } x_{pd}^{new} = 0. \tag{5}$$

In (2), w is the inertia weight, c_1 and c_2 are acceleration parameters, and *rand*, r_1 and r_2 are three independent random numbers between [0, 1]. Velocities v_{pd}^{new} and v_{pd}^{old} are those of the updated particle and the particle before being updated, respectively; x_{pd}^{old} is the original particle position (solution), and x_{pd}^{new} is the updated particle position (solution).

In (3), particle velocities of each dimension are tried to a maximum velocity $V_{\max}$. If the sum of accelerations causes the velocity of that dimension to exceed $V_{\max}$, then the velocity of that dimension is limited to $V_{\max}$. Both $V_{\max}$ and $V_{\min}$ are user-specified parameters (in our case $V_{\max} = 6$, $V_{\min} = -6$).

In (4) and (5), the updated features are calculated by the function $S(v_{pd}^{new})$, in which v_{pd}^{new} is the updated velocity value. $S(v)$ is a sigmoid limiting transformation and rand() is a quasi-random number selected from a uniform distribution in [0.0, 10]. If $S(v_{pd}^{new})$ is larger than a randomly produced disorder number that is within $0.0 \sim 1.0$, then its position value $S_n, n = 1, 2, \ldots, m$ is represented by {1}, meaning this feature is selected as a required feature for the next update. If $S(v_{pd}^{new})$ is smaller than a randomly produced disorder number that is within $0.0 \sim 1.0$, then its position value $F_n, n = 1, 2, \ldots, m$ is represented by {0}, meaning this feature is not selected as a required feature for the next update.

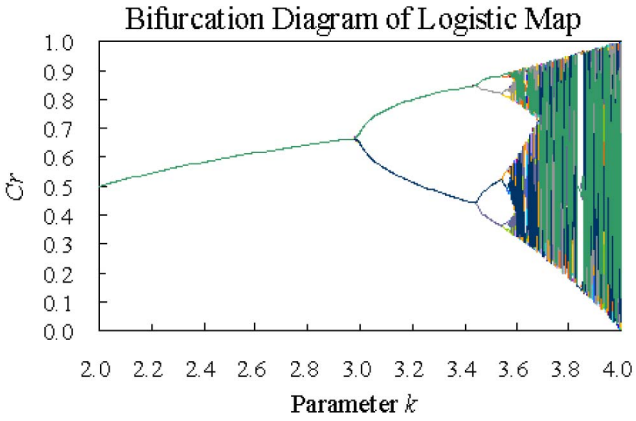


Fig. 4. Bifurcation diagram of logistic map. The diagram shows no overlap when adapting the *r1* and *r2* values from CBPSO.

2.5 Chaotic Sequences for Updating Inertia Weight

The inertia weight controls the balance between the global exploration and the local search ability. A large inertia weight facilitates the global search, while a small inertia weight facilitates the local search. Proper adjustment of the inertia weight value is important. The inertia weight w is the key factor influencing the convergence, and thus will greatly affect the BPSO search process and, through it, the resulting classification accuracy. The BPSO process often suffers from entrapment of particles in a local optimum. This causes the premature convergence mentioned above. We employed chaotic binary particle swarm optimization to prevent this early convergence and thus achieve better classification results:

$$Cr_{(t+1)} = k \times Cr_{(t)} \times (1 - Cr_{(t)}). \tag{6}$$

In (6), $Cr_{(0)}$ is generated randomly for each independent run, with $Cr_{(0)}$ not being equal to {0, 0.25, 0.5, 0.75, 1} and k equal to 4. The driving parameter k of the logistic map, controls the behavior of $Cr_{(t)}$ (as t goes to infinity). Fig. 4 shows the behavior of the logistic map for various values of the parameter k. For low values of $k (k < 3)$, *Cr* eventually converges to a single number. When $k = 3$, *Cr* oscillates between two values. This characteristic change in behavior is called a bifurcation. For $k > 3$, *Cr* goes through further bifurcations, eventually resulting in chaotic behavior. In fact, the bifurcation diagram itself is a fractal [30]. Fig. 4 shows that when k equals 4, the chaotic sequence value *Cr* is bounded within [0, 1]. The velocity update equation for CBPSO can be formulated as

$$v_{id}^{new} = w \times v_{id}^{old} + c_1 \times Cr \times \left(pbest_{id} - x_{id}^{old}\right) + c_2 \times (1 - Cr) \times \left(gbest_d - x_{id}^{old}\right). \tag{7}$$

In (7), *Cr* is a function based on the results of the logistic map with values between 0.0 and 1.0.

2.5.1 Encoding and Initialization

When an adjacent gene is inferred to be a WO, the upstream gene is encoded as 1. If an adjacent gene is inferred to be a TUB, the upstream gene is encoded as 0. In addition, the

TABLE 1
Intervals of Intergenic Distance Using the Log-Likelihood Method for *E. coli*

Interval	Score	Interval	Score	Interval	Score
[- ∞ , -99]	-0.82457	[30, 39]	0.568643	[170, 179]	-1.83357
[-100, -91]	0	[40, 49]	-0.67375	[180, 189]	-1.98772
[-90, -81]	1.478014	[50, 59]	-0.52852	[190, 199]	-1.51772
[-80, -71]	0	[60, 69]	-0.43437	[200, 209]	-2.35497
[-70, -61]	-0.31375	[70, 79]	-0.6435	[210, 219]	-1.98772
[-60, -51]	0	[80, 89]	-0.6322	[220, 229]	-3.4918
[-50, -41]	0.533552	[90, 99]	-0.55887	[230, 239]	-2.23556
[-40, -31]	-0.22673	[100, 109]	-1.48787	[240, 249]	-2.25966
[-30, -21]	0.379401	[110, 119]	-1.15683	[250, 259]	-2.79865
[-20, -11]	2.019145	[120, 129]	-1.43768	[260, 269]	0
[-10, -1]	2.22656	[130, 139]	-1.84221	[270, 279]	-3.33417
[0, 9]	2.2105	[140, 149]	-2.66512	[280, 289]	-2.1329
[10, 19]	2.340637	[150, 159]	-1.80384	[290, 299]	-2.83947
[20, 29]	1.564274	[160, 169]	-1.78965	[300, ∞]	-2.96611

The score of each separated interval in 10 bp bins is calculated based on an intergenic distance in the range of -100 bps to 300 bps.

initial position $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ and velocity $v_i = (v_{i1}, v_{i2}, \ldots, v_{id})$ of p particles are given. The position of each particle is initialized with its strand and a random threshold value between 0 and 600 bps [3]; the WO pair must be on the same strand. The initial velocity of each particle is 0, and the velocity is limited to between -6 and 6 [31].

2.5.2 Particle Update

Each particle is updated through an individual best ($pbest_i$), a global best (*gbest*) value, as well as other parameters. The $pbest_i$ value represents the position of the ith particle with the highest fitness value at a given iteration, and *gbest* represents the best position of all *pbest* particles. In CBPSO, each particle is updated according to (2)-(5). The velocity update equation in BPSO is (2), which is replaced by (7) in CBPSO.

2.5.3 Fitness Function

1. *Calculation of pair-score.* The three properties used in this study to predict operons are the intergenic distance, the metabolic pathway, and the COG gene function. The fitness values of the three properties are calculated based on the log-likelihood method as shown below.

   a. *Intergenic distance.* As shown in Table 1, (8) is used to calculate the pair-score of intergenic distance [25]

   $$\mathrm{LL}_{\mathrm{Proporty}}(gene_i, gene_j) = \ln\left(\frac{N_{WO}(\mathrm{Pr}\,operty)/TN_{WO}}{N_{TUB}(\mathrm{Pr}\,operty)/TN_{TUB}}\right), \quad (8)$$

   where N_{WO}*(property)* and N_{TUB}*(property)* correspond to the number of WO and TUB pairs in the interval distance (10, 20, 30...). TN_{WO} and TN_{TUB} are the total pair numbers within WO and TUB, respectively.

   b. *Metabolic pathways.* The pair-score of the metabolic pathway is also calculated by the log-likelihood method. The pathway pair-score is only taken into account when the two adjacent genes have the same pathway. Equation (8) is used to calculate the pathway pair-score. If the adjacent genes are in the same metabolic pathway, the pair-score of the adjacent gene is calculated to be 0.223. Otherwise, the pathway pair-score is 0 [1].

   c. *COG gene function.* We used the log-likelihood method to calculate the pair-score of the COG gene function based on three main levels. Equations (8) and (9) are used to calculate the COG pair-score [32]. The pair-scores of the COG function of adjacent genes are 0.936, 1.4996, and 1.1543 in "Information storage and processing," "Cellular processing and signaling," and "Metabolism," respectively. If a gene pair has a different COG function, the pair-score of the adjacent gene is calculated to be -0.4112. Otherwise, the pair-score of the adjacent gene is 0.

   $$\mathrm{LL}_{COGd}(gene_i, gene_j) = \ln\left(\frac{1 - N_{WO}(COG)/TN_{WO}}{1 - N_{TUB}(COG)/TN_{TUB}}\right), \quad (9)$$

   where $LL_{COGd}(gene_i, gene_j)$ represents the pair-score of adjacent genes with a different COG gene function.

2. *Calculation of operon fitness value.* While the pair-scores of each particle are calculated based on the metabolic pathway and the COG function, the fitness value of the operon in CBPSO is calculated by multiplying the pair-score average with the gene number in the same operon. The fitness of the cth putative operon is calculated by (10).

$$fitness_{cth} = \sum_{i=1}^{m-1} d_i + \left(\left\{\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}(\mathrm{LL}_{path}(gene_i, gene_j) + \mathrm{LL}_{COG}(gene_i, gene_j))\right\} \Big/ \{n\}\right) \times m, m = n + 1, \quad (10)$$

TABLE 2
Evaluation Method for Operon Prediction

Predicted result \ True data	Positive	Negative
Positive	TP	FP
Negative	FN	TN
Sensitivity (SN)	SN=TP/(TP+FN)	
Specificity (SP)	SP=TN/(FP+TN)	
Accuracy (ACC)	ACC=(TP+TN)/(TP+FP+TN+FN)	

True positive and false negative is the number of correctly and incorrectly predicted operon gene pairs among the WO gene pairs, respectively. True negative and false positive is the number of correctly and incorrectly predicted operon gene pairs among the TUB gene pairs. The sensitivity, specificity and accuracy are calculated based on TP, FP, TN, and FP.

where cth is the number of putative operons in a particle, m represents the number of genes within an operon, and n is the total number of gene pairs within an operon.

3. *Calculation of particle fitness value.* Finally, the fitness value of a particle is calculated as the sum of the fitness values from all putative operons in the particle

$$fitness = \sum_{i=1}^{c} fitness_i. \tag{11}$$

2.6 Performance Measurement

In Table 2, true positive (TP) and false negative (FN) are numbers that represent correct and incorrect predictions of gene pairs among the WO gene pairs, respectively, whereas false positive (FP) and true negative (TN) are the numbers of incorrect and correct predictions of gene pairs among the TUB gene pairs. We used TP, TN, FP, and FN to calculate the sensitivity (SN), the specificity (SP), and the accuracy (ACC). SN, SP, and ACC were then used to evaluate the prediction results [14]. The operon prediction flowchart of CBPSO is shown in Fig. 5. An illustrative example is shown in the supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2013.63.

2.7 Parameter Settings

The population number p was set to 20, the iteration number G was 100, the initial inertia weight w was 1, c_1 and c_2 were 2 [33], Cr_0 was 0.1, and $V_{\max}$ and $V_{\min}$ were 6 and -6, respectively [31].

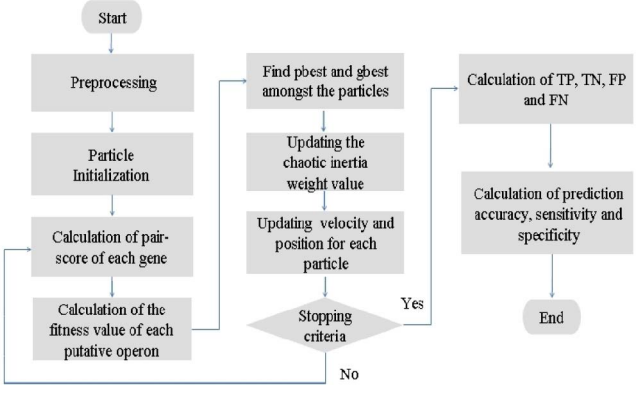


Fig. 5. Operon prediction flowchart of CBPSO.

3 Results and Discussion

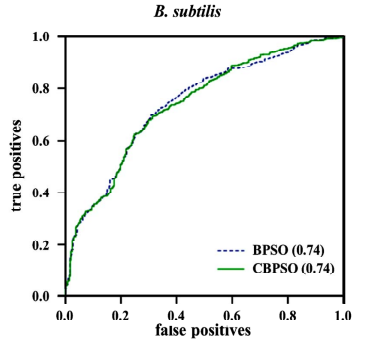
3.1 Receiver Operating Characteristic Curve Analysis

In this study, we initially used the five operon features intergenic distance, metabolic pathway, COG gene function, gene length ratio, and operon length to explore the performance of the proposed method. In a next step, we removed a single feature to observe how the receiver operating characteristic (ROC) curves changed. The operon prediction ROC curves are shown in Figs. 2a, 2b, 2c, and 2d for the *B. subtilis*, *P. aeruginosa* PA01, *S. aureus*, and *M. tuberculosis* genomes, respectively.

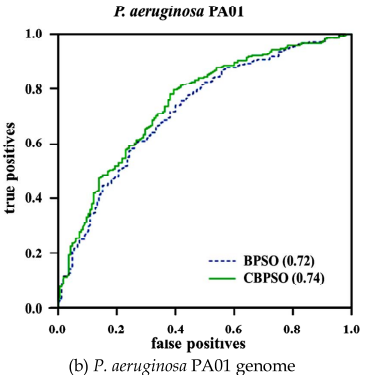
Considering that ROC curves help to better understand the prediction values of our algorithm. ROC curves express the relationship between the sensitivity and the specificity [23] as we vary a threshold on the confidence of our predictions. Figs. 2, 3, 4, 5, and 6 show ROC curves, in which the false-positive rate is plotted on the abscissa and the true-positive rate on the ordinate. The points on the ROC curves where the tangents have a slope of one is the point of maximum sensitivity and specificity. The area under a curve (AUC) is proportional to the prediction accuracy of the method, and thus a larger area represents better results.

The ROC curves indicate that the metabolic pathways and COG information can be expected to be highly correlated. We show the ROC area in Figs. 2a, 2b, 2c, and 2d for the results of all features. As stated above, a larger area implies better results. The numbers in the legends of the figures indicate the area size relative to the entire area shown (1.0). These figures attest to the quality of the prediction results. Figs. 2a, 2b, and 2d show that the prediction results without the COG information feature are worse than the results obtained when all features are used, with the sole exception of *S. aureus* (see Fig. 2c), for which results without COG information are actually slightly improved (area enlargement of 0.002). Overall, the COG information is an important feature for operon prediction. Since the ROC area without the metabolic pathway is the same as the results with all features in Figs. 2a, 2b, and 2d, the quality cannot be improved without the metabolic pathway feature. However, the *S. aureus* genome (see Fig. 2c) shows obvious improvement, which means that both, the metabolic pathway and the COG information, are integral for operon prediction.

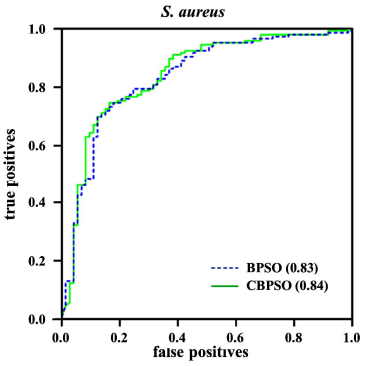
The figures further indicate that the intergenic distance feature is the single most important feature and that it has



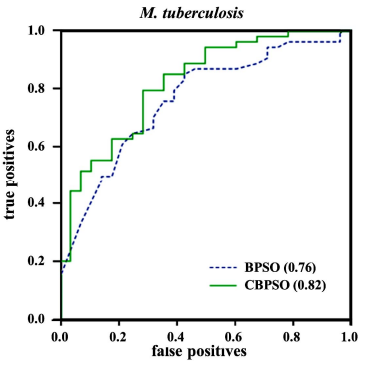
(a) *B. subtilis* genome



(b) *P. aeruginosa* PA01 genome



(c) *S. aureus* genome



(d) *M. tuberculosis* genome

Fig. 6. CBPSO ROC curves of operon prediction compared to the BPSO method (a) *B. subtilis* genome (b) *P. aeruginosa* PA01 genome (c) *S. aureus* genome (d) *M. tuberculosis* genome.

the best prediction ability. Through measurements of the areas under the curves (AUC) without COG, it was determined that the metabolic pathway is the second most important feature. We, thus, used the three most important features (intergenic distance, metabolic pathway, and COG gene function), respectively, to determine putative operons.

3.2 Comparison with Other Methods

In Table 3, we compare the achieved accuracy, sensitivity, and specificity to BPSO [2], GA [1], SVM [10], FGENESB (available at http://www.sofberry.com) and several other methods which have genome-wide operon prediction in *Staphylococcus aureus* [4], OFS [8], VIMSS [11], DVDA [12], UNIPOP [13], JPOP [15], ODB [21], and a predicted operon. map for *Mycobacterium tuberculosis* [34]. The experimental results show that the prediction accuracy of CBPSO had the highest values for the four target genomes *B. subtilis* (92.6 percent), *P. aeruginosa* PA01 (94.7 percent), *S. aureus* (95.9 percent), and *M. tuberculosis* (96.3 percent).

In this paper, the *E. coli* genome was selected to train the fitness function since abundant experimental data about this genome is available and can easily be downloaded from RegulonDB database. In contrast, only scarce data is available for other genomes. Even though related genes have related properties, there still exists a possibility of them being located in different operons. However, the fitness function of each particle is not the accuracy estimated. Hence, improvement of the fitness function is critical for solving the operon prediction problem. In this paper, we used the log-likelihood method of the statistical results to improve the reliability of the fitness function. The experimental results prove that the fitness function of CBPSO can effectively search for more optimal particles (solutions).

The intergenic distance, metabolic pathways, and COG gene function were used to predict operons on four target genomes. The properties currently most frequently used are the intergenic distance, metabolic pathway, and homologous genes. In DVDA (directon versus directon analysis), the homologous gene property is used to predict operons. Both the sensitivity and specificity are below 50 percent on the *E. coli* and *B. subtilis* genomes [35]. In addition, genes in the *E. coli* genome belong to the same first level of functional categories in the COG gene function, which have a probability of 83.5 percent of being within the same operon [25]. Hence, this property can be replaced by the superior COG gene function property.

Some methods only use the properties of adjacent genes to identify WO or TUB pairs and ignore the properties of nearby genes. GA, FGA, and BPSO are methods that try to overcome this problem. The GA algorithm often entraps the chromosomes in a local optimum. However, BPSO has global and local search capabilities since the algorithm is capable of adjusting the velocity of each particle through the inertia weight. The chaos embedded in the BPSO algorithm of this study further enhances the multiplicity of particles in the swarm. Figs. 6a, 6b, 6c, and 6d show that CBPSO effectively improves the probability of a higher accuracy, specificity, and sensitivity for the *B. subtilis*, *P. aeruginosa* PA01, *S. aureus*, and *M. tuberculosis* genomes, respectively, when compared to the BPSO method. In the results, the ROC curves of *M. tuberculosis* are significantly different due to the very small number of known genes in it. Table 4, which contains the SN, TP, SP, and FP for the four genomes, shows that the number of total TPs is very small on the *M. tuberculosis* genome compared to the other genomes.

In CBPSO, the initiation step is very important for the operon prediction procedure. In this study, we used two

TABLE 3
Accuracy, Sensitivity, and Specificity of Operon Prediction on Four Genomes

Genome	Methodology	ACC(%)	SN(%)	SP(%)
Bacillus subtilis (NC_000964)	CBPSO	**92.6**	88.6	**95.9**
	BPSO (Chuang and others 2010)	92.1	88.7	94.5
	UNIPOP (Li and others 2009)	79.2	78.2	82.1
	GA (Wang and others 2007)	88.3	87.3	89.7
	Decision tree-based classification (Dam and others 2007)	90.2	**90.8**	90.5
	SVM (Zhang and others 2006)	88.9	90.0	86.0
	ODB (Okuda and others 2006)	63.2	49.9	99.2
	DVDA (Edwards and others 2005)	48.5	31.9	93.2
	OFS (Westover and others 2005)	68.3	76.5	43.9
	VIMSS (Price and others 2005)	78.0	76.4	87.1
	JPOP (Chen and others 2004)	74.6	72.0	90.0
	OPERON (Ermolaeva and others 2001)	62.9	53.1	89.2
	FGENESB (http://www.softberry.com)	77.1	72.1	90.4
***Pseudomonas aeruginosa* PA01** (NC_002516)	CBPSO	**94.7**	**94.4**	**95.1**
	BPSO (Chuang and others 2010)	93.3	93.0	93.9
	GA (Wang and others 2007)	81.3	87.0	76.3
Staphylococcus aureus (NC_002952)	CBPSO	**95.9**	**95.9**	**95.9**
	BPSO (Chuang and others 2010)	**95.9**	**95.9**	**95.9**
	Genome-wide operon prediction in *Staphylococcus aureus* (Wang and others 2004)	92.0	N/A	N/A
Mycobacterium tuberculosis (NC_000962)	CBPSO	**96.3**	**96.3**	**96.3**
	BPSO	95.1	94.4	**96.3**
	A Predicted Operon map for *Mycobacterium tuberculosis* (Roback and others 2007)	90.8	90.8	90.9

CBPSO obtained a higher accuracy, sensitivity, and specificity compared to the other methods from the literature. All performance measures are taken from the literature and are based on different data sets. UPN: Used property number. N/A: Data not available. Highest values are given in bold type.

TABLE 4
SN, TP, SP, and FP for Operons Predicted on Four Different Genomes

Genome	Method	TP predicted/total	SN (%)	FP predicted/total	SP (%)
B. subtilis	BPSO	788/888	88.7	22/399	94.5
	CBPSO	787/888	88.6	16/399	95.9
P. aeruginosa PA01	BPSO	251/270	93.0	10/165	93.9
	CBPSO	255/270	94.4	8/165	95.1
S. aureus	BPSO	140/146	95.9	3/73	95.9
	CBPSO	140/146	95.9	3/73	95.9
M. tuberculosis	BPSO	51/54	94.4	1/28	96.3
	CBPSO	52/54	96.3	1/28	96.3

important properties (the intergenic distance and strands) to initialize each particle. The initially superior particles are further improved through a repetition of the updating process in each generation, thus further improving the prediction performance. A sensitivity and specificity value higher than 80 percent represents a good balance between the two parameters [10]. Table 3 shows that the proposed method achieved a good balance between sensitivity and specificity and at the same time also improved the prediction accuracy. The prediction accuracy of CBPSO is superior to the other methods it was compared to, but the specificity and sensitivity of the *B. subtilis* genome is lower than the ones for ODB and SVM, respectively. However, ODB does not yield a good balance between sensitivity and specificity. Its sensitivity only reached 49.9 percent, and the accuracy was only 63.2 percent. The respective sensitivity

TABLE 5
Prediction Features Used by Each Computational Method on the *Bacillus subtilis* Genome

Year of publication		2010		2009	2008	2007		2006		2005				2004		2003	2002	2001	2000	1999	Others
Methodology		NN	BPSO	UNIPOP	PSWM	GA	Decision tree Classification	SVM	ODB	DVDA	OFS	VIMSS	FGA	JPOP	CCG	BN	BC	OPERON	NB	HMM	FGENESB
Reference		[2]	[2]	[13]	[37]	[1]	[14]	[10]	[21]	[12]	[8]	[[11]	[3]	[15]	[4]	[6]	[5]	[38]	[9]	[39]	*
Features	Intergenic distance	√	√			√	√	√	√		√	√	√	√		√					√
	Metabolic pathway		√			√		√	√				√								
	Homologous genes			√			√	√		√			√								
	Terminator														√	√			√	√	√
	Gene order conservation								√		√			√							√
	Promoter															√			√	√	√
	Microarray					√			√							√	√				
	Cluster of orthologous groups (COG)	√				√						√									
	Gene length ratio		√				√														
	Phylogenetic profile							√						√							
	Operon length															√			√		
	Phylogenetic distance						√														
	Motif						√														
	Gene ontology						√														
	Common gene annotation										√										
	Comparative features											√									
	Protein functions												√								
	Codon adaptation index											√									
	Transcription factor binding site				√																
	Gene cluster conservation																	√			
	Remaining Orthologous Groups (ROG)	√																			
	Spacing within operon															√					
	Ribosome binding																			√	

*http://www.softberry.com.

and accuracy values of CBPSO were both higher than in ODB. The sensitivity of SVM (90.0 percent) was slightly higher than the one of the proposed method (88.6 percent). Nevertheless, the specificity of CBPSO (95.9 percent) was considerably higher than that of SVM (86.0 percent).

A prediction method that uses a higher number of properties does not necessary translate into an improved prediction accuracy. Table 5 shows the properties used in methods taken from the literature. As shown, the number of properties used in GA, decision tree classification, SVM [10], ODB [21], VIMSS [11], and FCENESB (available at http://www.sofberry.com) are all higher than the number of properties used by CBPSO, yet the prediction accuracy obtained by these methods is lower than the prediction accuracy obtained by CBPSO.

The better performance of our method compared to the other methods it was compared to is substantiated by the following factors: 1) the superiority of the CBPSO algorithm and the design of the fitness function based on statistics; and 2) the selection of relevant properties. These factors are discussed below:

1. Some methods try to predict operons based on the properties of adjacent genes that are either identified as a WO pair or TUB pair, but these methods do not take the properties of near genes into account. Hence, the results are generally of a lower accuracy for operon prediction. CBPSO can evaluate the properties of all genes, and thereby increases the probability of finding an optimal solution. To improve the traditional BPSO method's performance, we use the chaos theory to improve the inertia weight of the BPSO update function. This effectively controls the balance between the global exploration and the local search ability, and thus the probability of obtaining the best solution is increased. Another factor is the designed fitness value. Although adjacent genes have related properties (e.g., pathways and COG), they still have a probability of being in different operons. Consequently, a fitness function needs to be implemented in the proposed method. We calculated the fitness value of each particle based on the logarithmic likelihood ratio test, since this method is designed on the basis of statistics, i.e., the intergenic distance, the metabolic pathway, and the COG properties. This means that the fitness value of a putative operon is directly proportional to the prediction accuracy. The experimental results prove that this fitness function identifies better particles.
2. The *E. coli* genome has been widely investigated and valuable information pertaining to it can be readily downloaded from the RegulonDB database, but for other genomes extensive experimental data is not readily available. To apply the proposed method to other prokaryote genomes with fewer attributes, only five common operon prediction properties need to be used. In general, a higher number of properties selected for operon prediction results in a better accuracy. However, the proposed CBPSO method uses only three properties to achieve a high accuracy and better results. The advantage here is that the necessary prediction time is not increased. Table 5 shows the features used in the different methods. Decision tree-based classification uses six properties for operon prediction and achieves the highest sensitivity, but the method suffers from a lower prediction specificity and lower accuracy than our method. ODB uses four properties for operon

prediction, but the method suffers from a low prediction sensitivity and accuracy. In other words, these methods only achieve either a high sensitivity or a high specificity, but not both. We used the intergenic distance, the metabolic pathway, and the COG properties to identify the WO and TUB pairs. The metabolic pathway is used with a higher frequency than other the properties; it often carries out highly specific activities in a biochemical metabolic pathway [1], [3], [36]. The COG is used somewhat less frequently than the other properties, but literature reports [11], [15] have proved the powerful identification ability of this property due to one operon often having the same or similar COG functions. The accuracy, sensitivity, and specificity results that our method achieved are the highest for operon prediction even though the method only uses three properties on all bacterial genomes.

4 Conclusions

This study proposes CBPSO for the prediction of operons in bacterial genomes. The embedded chaos enhances the random diversity and thus improves the probability of finding optimal results. In addition, the log-likelihood method was employed to design a fitness function. The evaluation accuracy of the fitness function was further increased through the use of statistical theory. The experimental results show that the use of only three properties in CBPSO was sufficient to obtain the highest accuracy on the four target genomes. The proposed method also achieved a good balance between sensitivity and specificity. In the future, we intend to construct an operon prediction system for bioinformatics research to obtain further valuable information about operons.

Acknowledgments

This work was partly supported by the National Science Council in Taiwan under grants 102-2221-E-151-024-MY3, 102-2622-E-151-003-CC3, 101-2622-E-151-027-CC3, and 102-2221-E-214-039.

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,* vol. 41, pp. 151-159, Oct. 2007.

[2] L.Y. Chuang, J.H. Tsai, and C.H. Yang, "Binary Particle Swarm Optimization for Operon Prediction," *Nucleic Acids Research,* vol. 38, article e128, 2010.

[3] E. Jacob, R. Sasikumar, and K.N.R. Nair, "A Fuzzy Guided Genetic Algorithm for Operon Prediction," *Bioinformatics,* vol. 21, pp. 1403-1407, Apr. 2005.

[4] L. Wang, J.D. Trawick, R. Yamamoto, and C. Zamudio, "Genome-Wide Operon Prediction in *Staphylococcus aureus*," *Nucleic Acids Research,* vol. 32, pp. 3689-3702, 2004.

[5] C. Sabatti, L. Rohlin, M.K. Oh, and J.C. Liao, "Co-Expression Pattern from DNA Microarray Experiments as a Tool for Operon Prediction," *Nucleic Acids Research,* vol. 30, pp. 2886-2893, July 2002.

[6] J. Bockhorst, M. Craven, D. Page, J. Shavlik, and J. Glasner, "A Bayesian Network Approach to Operon Prediction," *Bioinformatics,* vol. 19, pp. 1227-35, July 2003.

[7] M.J. De Hoon, S. Imoto, K. Kobayashi, N. Ogasawara, and S. Miyano, "Predicting the Operon Structure of *Bacillus subtilis* Using Operon Length, Intergene Distance, and Gene Expression Information," *Proc. Pacific Symp. Biocomputing,* pp. 276-87, 2004.

[8] B.P. Westover, J.D. Buhler, J.L. Sonnenburg, and J.I. Gordon, "Operon Prediction without a Training Set," *Bioinformatics.* vol. 21, pp. 880-888, Apr. 2005.

[9] M. Craven, D. Page, J. Shavlik, J. Bockhorst, and J. Glasner, "A Probabilistic Learning Approach to Whole-Genome Operon Prediction," *Proc. Int'l Conf. Intelligent Systems for Molecular Biology,* vol. 8, pp. 116-27, 2000.

[10] G.Q. Zhang, Z.W. Cao, Q.M. Luo, Y.D. Cai, and Y.X. Li, "Operon Prediction Based on SVM," *Computational Biology and Chemistry,* vol. 30, pp. 233-240, June 2006.

[11] M.N. Price, K.H. Huang, E.J. Alm, and A.P. Arkin, "A Novel Method for Accurate Operon Predictions in All Sequenced Prokaryotes," *Nucleic Acids Research,* vol. 33, pp. 880-892, 2005.

[12] M.T. Edwards, S.C. Rison, N.G. Stoker, and L. Wernisch, "A Universally Applicable Method of Operon Map Prediction on Minimally Annotated Genomes Using Conserved Genomic Context," *Nucleic Acids Research,* vol. 33, pp. 3253-3262, 2005.

[13] G. Li, D. Che, and Y. Xu, "A Universal Operon Predictor for Prokaryotic Genomes," *Bioinformatics and Computational Biology,* vol. 7, pp. 19-38, Feb. 2009.

[14] P. Dam, V. Olman, K. Harris, Z. Su, and Y. Xu, "Operon Prediction Using Both Genome-Specific and General Genomic Information," *Nucleic Acids Research,* vol. 35, pp. 288-298, 2007.

[15] X. Chen, Z. Su, Y. Xu, and T. Jiang, "Computational Prediction of Operons in *Synechococcus* sp. WH8102," *Genome Informatics,* vol. 15, pp. 211-222, 2004.

[16] B. Taboada, C. Verde, and E. Merino, "High Accuracy Operon Prediction Method Based on STRING Database Scores," *Nucleic Acids Research,* vol. 38, article e130, 2010.

[17] S. Gama-Castro, V. Jimenez-Jacinto, M. Peralta-Gil, A. Santos-Zavaleta, M. Penaloza-Spinola, B. Contreras-Moreira, J. Segura-Salazar, L. Muniz-Rascado, I. Martinez-Flores, and H. Salgado, "RegulonDB (Version 6.0): Gene Regulation Model of *Escherichia coli* K-12 Beyond Transcription, Active (Experimental) Annotated Promoters and Textpresso Navigation," *Nucleic Acids Research,* vol. 36, pp. D120-D124, 2007.

[18] N. Sierro, Y. Makita, M. De Hoon, and K. Nakai, "DBTBS: A Database of Transcriptional Regulation in *Bacillus subtilis* Containing Upstream Intergenic Conservation Information," *Nucleic Acids Research,* vol. 36, p. D93, 2008.

[19] F. Mao, P. Dam, J. Chou, V. Olman, and Y. Xu, "DOOR: A Database for Prokaryotic Operons," *Nucleic Acids Research,* vol. 37, p. D459, 2009.

[20] P.S. Dehal, M.P. Joachimiak, M.N. Price, J.T. Bates, J.K. Baumohl, D. Chivian, G.D. Friedland, K.H. Huang, K. Keller, and P.S. Novichkov, "MicrobesOnline: An Integrated Portal for Comparative and Functional Genomics," *Nucleic Acids Research,* vol. 38, p. D396, 2010.

[21] 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,* vol. 34, p. D358, 2006.

[22] H.G. Schuster and W. Just, *Deterministic Chaos.* Wiley, 1988.

[23] H. Salgado, G. Moreno-Hagelsieb, T.F. Smith, and J. Collado-Vides, "Operons in *Escherichia coli*: Genomic Analyses and Predictions," *Proc. Nat'l Academy of Sciences USA,* vol. 97, pp. 6652-6657, June 2000.

[24] G. Moreno-Hagelsieb and J. Collado-Vides, "A Powerful Non-Homology Method for the Prediction of Operons in Prokaryotes," *Bioinformatics,* vol. 18, pp. S329-S336, 2002.

[25] P.R. Romero and P.D. Karp, "Using Functional and Organizational Information to Improve Genome-Wide Computational Prediction of Transcription Units on Pathway-Genome Databases," *Bioinformatics,* vol. 20, pp. 709-717, Mar. 2004.

[26] Y. Yan and J. Moult, "Detection of Operons," *Proteins,* vol. 64, pp. 615-28, Aug. 2006.

[27] T.T. Tran, P. Dam, Z. Su, F.L. Poole, M.W.W. Adams, G.T. Zhou, and Y. Xu, "Operon Prediction in *Pyrococcus furiosus*," *Nucleic Acids Research,* vol. 35, p. 11, 2006.

[28] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Proc. IEEE Int'l Joint Conf. Neural Network,* vol. 4, pp. 1942-1948, 1995.

[29] J. Kennedy, "The Particle Swarm: Social Adaptation of Knowledge," *Proc. IEEE Int'l Conf. on Evolutionary Computation,* pp. 303-308, 1997.
[30] D. Kuo, "Chaos and Its Computing Paradigm," *IEEE Potentials,* vol. 24, no. 2, pp. 13-15, Apr./May 2005.
[31] J. Kennedy and R. Eberhart, "A Discrete Binary Version of the Particle Swarm Algorithm," *Proc. IEEE Int'l Conf. System, Man, and Cybernetics,* pp. 4104-4108, 1997.
[32] X. Chen, Z. Su, P. Dam, B. Palenik, Y. Xu, and T. Jiang, "Operon Prediction by Comparative Genomics: An Application to the *Synechococcus* sp. WH8102 Genome," *Nucleic Acids Research,* vol. 32, pp. 2147-2157, 2004.
[33] J. Kennedy, "Swarm Intelligence," *Handbook of Nature-Inspired and Innovative Computing,* pp. 187-219, Springer, 2006.
[34] P. Roback, J. Beard, D. Baumann, C. Gille, K. Henry, S. Krohn, H. Wiste, M.I. Voskuil, C. Rainville, and R. Rutherford, "A Predicted Operon Map for *Mycobacterium tuberculosis*," *Nucleic Acids Research,* vol. 35, pp. 5085-5095, 2007.
[35] R.W. Brouwer, O.P. Kuipers, and S.A. van Hijum, "The Relative Value of Operon Predictions," *Briefings in Bioinformatics,* vol. 9, pp. 367-75, Sept. 2008.
[36] Y. Zheng, J.D. Szustakowski, L. Fortnow, R.J. Roberts, and S. Kasif, "Computational Identification of Operons in Microbial Genomes," *Genome Research,* vol. 12, pp. 1221-1230, Aug. 2002.
[37] E. Laing, K. Sidhu, and S.J. Hubbard, "Predicted Transcription Factor Binding Sites as Predictors of Operons in *Escherichia coli* and *Streptomyces coelicolor*," *BMC Genomics,* vol. 9, article 79, Feb. 2008.
[38] M.D. Ermolaeva, O. White, and S.L. Salzberg, "Prediction of Operons in Microbial Genomes," *Nucleic Acids Research,* vol. 29, pp. 1216-1221, Mar. 2001.
[39] T. Yada, M. Nakao, Y. Totoki, and K. Nakai, "Modeling and Predicting Transcriptional Units of *Escherichia coli* Genes Using Hidden Markov Models," *Bioinformatics,* vol. 15, pp. 987-993, Dec. 1999.

Li-Yeh Chuang received the MS degree from the Department of Chemistry, University of North Carolina, in 1989 and the PhD degree from the Department of Biochemistry, North Dakota State University, in 1994. She is a professor and director of the Department of Chemical Engineering and the Institute of Biotechnology and Chemical Engineering at I-Shou University, Kaohsiung, Taiwan. Her main areas of research include bioinformatics, biochemistry, and genetic engineering.

Cheng-Huei Yang received the BS degree from the National Taipei Institute of Technology, Taiwan, in 1978, the MS degree from Northeastern University, Boston, Massachusetts, in 1987, and the PhD degree in electrical engineering from the National Chen Kung University, Tainan, Taiwan, in 2001. Currently, he is a professor in the Department of Telecommunication and Computer Engineering, National Kaohsiung Institute of Marine Technology, Taiwan. His research interests include network communication, electronic instrument systems, and image processing.

Jui-Hung Tsai received the MS degrees from the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Taiwan, in 2008 and 2010, respectively. He has rich experience in computer programming, database design and management, and systems programming and design. His main areas of research include bioinformatics and computational biology.

Cheng-Hong Yang received the MS and PhD degrees in computer engineering from North Dakota State University in 1988 and 1992, respectively. He is a professor in the Department of Electronic Engineering at the National Kaohsiung University of Applied Sciences and serves as president of the university. His main areas of research include evolutionary computation, bioinformatics, and assistive tool implementation.