Published online 12 April 2010

Nucleic Acids Research, 2010, Vol. 38, No. 12 ***e128***
doi:10.1093/nar/gkq204

Binary particle swarm optimization for operon prediction

Li-Yeh Chuang[1], Jui-Hung Tsai[2] and Cheng-Hong Yang[3,4,*]

[1]Department of Chemical Engineering, I-Shou University, [2]Department of Computer Science and Information Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan, [3]Department of Network Systems, Toko University, Chiayi and [4]Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan

Received October 3, 2009; Revised February 26, 2010; Accepted March 9, 2010

ABSTRACT

An operon is a fundamental unit of transcription and contains specific functional genes for the construction and regulation of networks at the entire genome level. The correct prediction of operons is vital for understanding gene regulations and functions in newly sequenced genomes. As experimental methods for operon detection tend to be nontrivial and time consuming, various methods for operon prediction have been proposed in the literature. In this study, a binary particle swarm optimization is used for operon prediction in bacterial genomes. The intergenic distance, participation in the same metabolic pathway, the cluster of orthologous groups, the gene length ratio and the operon length are used to design a fitness function. We trained the proper values on the *Escherichia coli* genome, and used the above five properties to implement feature selection. Finally, our study used the intergenic distance, metabolic pathway and the gene length ratio property to predict operons. Experimental results show that the prediction accuracy of this method reached 92.1%, 93.3% and 95.9% on the *Bacillus subtilis* genome, the *Pseudomonas aeruginosa PA01* genome and the *Staphylococcus aureus* genome, respectively. This method has enabled us to predict operons with high accuracy for these three genomes, for which only limited data on the properties of the operon structure exists.

INTRODUCTION

Operons in prokaryote organisms contain one or more consecutive genes on the same strand, although a few eukaryotic organisms also have operon-like structures, e.g. *Caenorhabditis elegans* (1). These genes are co-transcribed into a single-strand mRNA sequence. Co-transcribed genes likely have the same biological functions and directly affect each other. Operon prediction can therefore be used to infer the function of putative proteins if the functions of other genes in the same operon are known. A well-known example is the lactose operon in *Escherichia coli*. This operon contains the three consecutive structural genes, *lac*Z, *lac*Y and *lac*A, which all share the same promoter and terminator.

Operons of bacterial genomes contain information valuable for drug design and determining protein functions (2). The Gram-positive *Staphylococcus* bacterium, for example, is a human pathogen that is responsible for community-acquired and nosocomial infections (3). Operon prediction on this bacterium can facilitate drug target identification and the development of antibiotic drugs. However, knowledge of operons is scarce, and experimental methods for predicting operons are generally difficult to implement (4). To gain better insight, the number and organization of operons in bacterial genomes have to be studied in greater detail. A detailed understanding of the transcription rules is critical, as it would allow scientists to accurately predict operons based on an organism's genomic sequence.

A number of scientists have proposed properties that can accurately predict operons. These properties can be divided into the following five categories (5): 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 the terminator at the operon boundaries to identify the biologically most representative properties (4). The simplest and most important prediction property is to observe whether the distance between gene pairs within an operon (WO pairs) is shorter than the distance between gene pairs at the borders of the transcription units (TUB

*To whom correspondence should be addressed. Tel: +886 7 3814526; Fax: +886 7 3836844; Email: chyang@cc.kuas.edu.tw

pairs) (3). The distance property yields very good operon prediction results.

Many computational algorithms have been proposed to properly balance the sensitivity and specificity of operon prediction. Jacob *et al*. (4) proposed an algorithm guided by fuzzy logic. Fuzzy logic does not rely on complex mathematical formulas to calculate fitness values of a chromosome. Genetic algorithms (GA) (2) use the intergenic distance, metabolic pathways, cluster of orthologous groups (COG) and microarray expression data to predict operons. Zhang *et al*. (6) presented a support vector machine algorithm (SVM) to predict operons. This method uses the four biological properties as SVM input vectors and divides gene pairs into operon pairs (OPs) and non-operon pairs (NOPs). The experimental accuracy of prediction was 0.9. In our study, we compare additional predictors [genome-specific (7), DVDA (8), FGENESB, ODB (9), OFS (10), OPERON (11), JPOP (12), VIMSS (13), UNIPOP (1) and genome-wide operon prediction in *Staphylococcus aureus* (3)], in addition to the above-mentioned methods.

In this paper, we propose an effective binary particle swarm optimization (BPSO) for operon prediction. To validate the feasibility of the method, we calculated the logarithmic likelihood of each property in the *E. coli* (NC_000913) genome as a fitness value of each gene in the particle. Three bacterial genomes [*Bacillus subtilis* (NC_000964), *Pseudomonas aeruginosa PA01* (NC_002516) and *S. aureus* (NC_002952)] were selected as benchmark genomes of known operon structure. In a first step, a restriction was introduced in the strand form to initialize a basis for the intergenic distance property. In order to select the best possible combination of properties, we employed the concept of feature selection to implement operon prediction. The five features investigated were the intergenic distance, metabolic pathways, COG, gene length ratio and operon length. Based on the experimental results and our analysis thereof, the intergenic distance, metabolic pathways and gene length ratio were selected after the feature selection process to calculate the fitness value of each gene in a particle. The particle was subsequently updated by an update formula at each generation. The detailed updating process is described in the next section. The experimental results indicate that the proposed method obtained a higher accuracy, sensitivity and specificity on the test data sets when compared to other methods from the literature.

MATERIALS AND METHODS

Data set preparation

The complete microbial genome data were downloaded from the GenBank database (http://www.ncbi.nlm.nih.gov/). The data contain a total of 4225, 5651 and 2845 genes in the *B. subtilis* genome, *P. aeruginosa PA01* genome and *S. aureus* genome, respectively. The related genomic information consists of the gene name, gene ID, position, strand and product. The operon databases of *E. coli* and *B. subtilis* were obtained from RegulonDB (http://regulondb.ccg.unam.mx/) (14) and DBTBS (http://dbtbs.hgc.jp/) (15), 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/) (9). 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.

Definition of a potential operon pair

In order to gain valuable information pertaining to drug and protein functions, operons have to be predicted based on an organism's genomic sequence. The entire genome is scanned for adjacent gene pairs on the same string, and each gene pair is then classified into one of three types: (i) adjacent; (ii) WO pair; or (iii) TUB pair. The latter two are defined as positive and negative, respectively, before the accuracy for a putative operon map is calculated. The WO pairs of adjacent genes shown in Supplementary Figure S1 are in the same operon. 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 gene is of uncertain status at the end of the border of the transcription unit (16), the gene pair cannot be labeled a TUB pair. In addition, the first gene of an operon and the upstream gene are TUB pairs.

Binary particle swarm optimization

Overview. The particle swarm optimization (PSO) technique is a population-based evolutionary algorithm developed by Kenney and Eberhart in 1995 (17). PSO has been developed through simulation of the social behavior of organisms, e.g. fish in a school or birds in a flock. The method is similar to a genetic algorithm, in which particles are initialized within a random population and search for global optimal solutions at each generation. However, PSO is not suitable for optimization problems in a discrete feature space. Hence, Kenney and Eberhart developed binary PSO (BPSO) to overcome this problem (18). The basic elements of BPSO are briefly introduced below:

(i) *Population*: A swarm (population) consists of N particles.
(ii) *Particle position, x_i*: Each candidate solution can be represented by a D-dimensional vector; the ith particle can be described as $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, where x_{iD} is the position of the ith particle with respect to the Dth dimension.
(iii) *Particle velocity, v_i*: The velocity of the ith particle is represented by $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, where v_{iD} is the velocity of the ith particle with respect to the Dth dimension. In addition, the velocity of a particle is limited within $[V_{\min}, V_{\max}]^D$.
(iv) *Inertia weight, w*: The inertia weight is used to control the impact of the previous velocity of a particle on the current velocity. This control parameter affects the trade-off between the exploration and exploitation abilities of the particles.

(v) *Individual best*, $pbest_i$: $pbest_i$ is the position of the *i*th particle with the highest fitness value at a given iteration.
(vi) *Global best, gbest*: The best position of all *pbest* particles is called global best.
(vii) *Stopping criteria*: The process is stopped after the maximum allowed number of iterations is reached.

In the BPSO algorithm, each particle represents a candidate solution to the problem, and a swarm consists of N particles moving around a D-dimension search space until the computational limitations are reached. A flowchart of BPSO is shown in Figure 1. An inertia weight with a value of 1 is used at each generation (18). The *gbest* value is reached after the maximum number of 100 iterations has been executed. Detailed steps are shown below and in the flowchart of Figure 1.

Step (i): Each particle is initialized based on the gene strand and a random threshold value of between 0 and 600 bp.
Step (ii): The pair-score of each gene is calculated based on its properties.
Step (iii): The fitness value of the putative operon is calculated by Equation (11).
Step (iv): The fitness value of each particle is calculated by Equation (12).
Step (v): Each particle is updated based on the PSO update formula, and a search for pbest_i and gbest of the population is conducted.
Step (vi): Steps 3 and 4 are repeated until the stopping criteria are satisfied.

In BPSO, each particle is updated based on the following equations:

$$v_{id}^{\text{new}} = w \times v_{id}^{\text{old}} + c_1 \times r_1 \times \left(pbest_{id} - x_{id}^{\text{old}}\right) + c_2 \times r_2 \times \left(gbest_{id} - x_{id}^{\text{old}}\right) \quad (1)$$

$$\text{if } v_{id}^{\text{new}} \notin (V_{\min}, V_{\max}) \text{ then } v_{id}^{\text{new}} = \max\left(\min\left(V_{\max}, v_{id}^{\text{new}}\right), V_{\min}\right) \quad (2)$$

$$S\left(v_{id}^{\text{new}}\right) = \frac{1}{1+e^{-v_{id}^{\text{new}}}} \quad (3)$$

$$\text{if } \left(r_3 < S\left(v_{id}^{\text{new}}\right)\right) \text{ then } x_{id}^{\text{new}} = 1 \; x_{id}^{\text{new}} = 0 \quad (4)$$

where w is the inertia weight that controls the impact of the previous velocity of a particle. c_1 and c_2 are acceleration constants that control the distance a particle moves at each generation; r_1, r_2 and r_3 are random numbers between [0, 1]. v_{id}^{new} and v_{id}^{old} represent the velocity of the new and old particles, respectively. Particles x_{id}^{old} and x_{id}^{new} denote the position of the current particle and the updated particle, respectively. The velocity of a dimension in Equation (2) is limited within $[V_{\min}, V_{\max}]^D$. The positions of the updated particles are calculated by Equation (3) (19). If the function $S\left(v_{id}^{\text{new}}\right)$ is greater than r_3, the position of the particle is updated to {1} (meaning this gene is part of the operon). If $S\left(v_{id}^{\text{new}}\right)$ is smaller than r_3, the position is updated to {0} (i.e. this gene is the final gene of the operon).

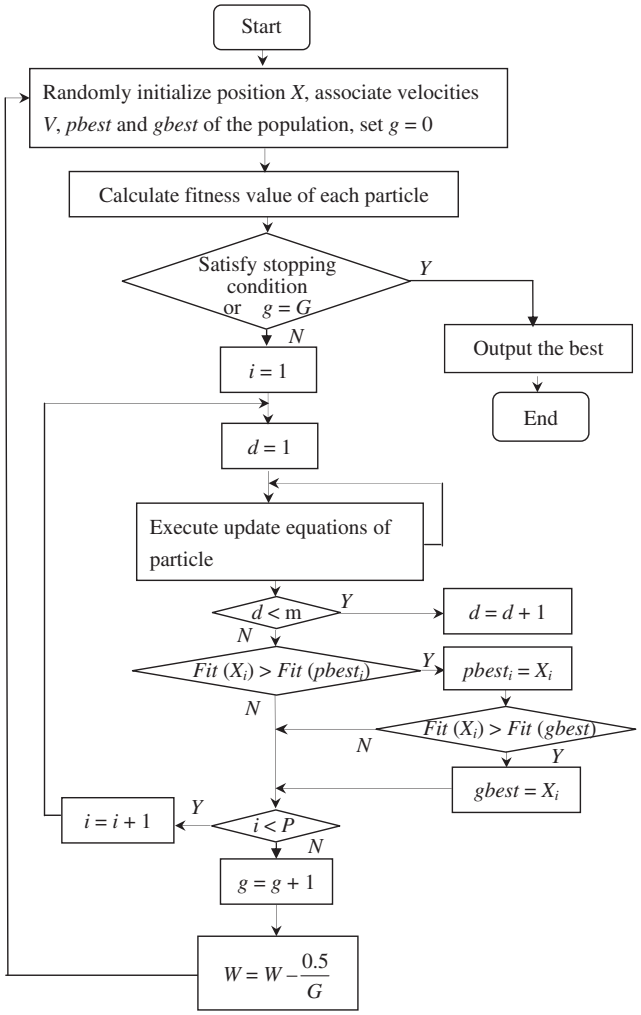

Figure 1. BPSO flowchart. The algorithm starts out by initializing a population of random particles. Then the fitness values of particles are calculated and searches for *pbest* and *gbest* are executed at each generation. Afterwards, the position and velocity of the *i*-th particle are updated by $pbest_i$ and *gbest* in the swarm, and the search for the best solution is continued by updating the generations until the stopping criteria are satisfied. Each particle makes use of its own memory and knowledge gained by the swarm as a whole to find the best solution. The updated position and velocity of the particles confined within $[X_{\min}, X_{\max}]^D$ and $[V_{\min}, V_{\max}]^D$ are obtained.

Initial population. The proposed method uses the intergenic distance and strands to create P binary particles. Each particle is initialized with a random threshold value of between 0 and 600 bp (4). For adjacent genes to be considered in the same operon, they must conform to the following two conditions: the distance of adjacent genes must be smaller than the random threshold value, and adjacent genes must be on the same strand. If the distance between adjacent genes is greater than the random threshold value, we assume that the two adjacent genes are within a different operon. Adjacent genes on different strands are considered NOPs). Supplementary Figure S2 illustrates these criteria. The encoding used is shown in Supplementary Figure S3.

Fitness function. As stated previously, many properties can be used to predict operons. The five properties we used in this study are individually described in the following section. Supplementary Table S1 shows the pair-score of the intergenic distance, metabolic pathway, COG gene function and the gene length ratio calculated by the logarithmic likelihood ratio test. The pair-score of the operon length is calculated by the Bernoulli process.

Intergenic distance. This property allows operon prediction in genomes for which the genomic sequence is completely mapped. In order to prevent mRNA degradation, the distance of adjacent genes in the same operon is kept short (20). The Supplementary Figure S4 shows the operon diagram. The intergenic distance is calculated using base pairs of adjacent genes. However, adjacent genes sometimes overlap. The intergenic distance distribution of WO and TUB pairs is shown in Supplementary Figure S5. Genes with a smaller intergenic distance are more likely located within the same operon (2). The maximum frequency of the distance of WO pairs is –4 (21). However, the distance distribution frequency of TUB pairs increases with the distance and gradually becomes higher than the frequency of WO pairs. Hence, this property can be used to identify operons. As shown in Supplementary Table S2, we calculate the score of each separated interval in 10-bp bins (22) based on an intergenic distance from –100 bp to 300 bp using the following equation:

$$\mathrm{LL}_{dist}(gene_i, gene_j) = \ln\left(\frac{N_{\mathrm{WO}}(dist)/TN_{\mathrm{WO}}}{N_{\mathrm{TUB}}(dist)/TN_{\mathrm{TUB}}}\right) \quad (5)$$

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

Metabolic pathways. Gene ontology contains three levels of biological functions, namely a biological process, a molecular function and a cellular component (23). However, genes within an operon often participate in the same biological process (6). Therefore, adjacent genes have the same metabolic pathway, and we can reasonably assume that the gene pair is located in the same operon. The pathway pair-score is only taken into account when two adjacent genes have the same pathway. Otherwise, the pathway pair-score is 0 (2). Equation (6) is used to calculate the pathway pair-score.

$$\mathrm{LL}_{path}(gene_i, gene_j) = \ln\left(\frac{N_{\mathrm{WO}}(path)/TN_{\mathrm{WO}}}{N_{\mathrm{TUB}}(path)/TN_{\mathrm{TUB}}}\right) \quad (6)$$

where $N_{\mathrm{WO}}(path)$ and $N_{\mathrm{TUB}}(path)$ correspond to the total number of WO and TUB pairs in the same metabolic pathway.

COG gene function. The COGs consist of three main levels. The first level contains the following four classes: information storage and processing, cellular processing and signaling, metabolism and different COG categories. Each class is subdivided into multiple functional categories. Adjacent genes are often of the same class, so we assume that the gene pair is located in the same operon. The pair-score of the COG gene function is calculated based on the first level. The following equations are used (12):

$$\mathrm{LL}_{\mathrm{COG}}(gene_i, gene_j) = \ln\left(\frac{N_{\mathrm{WO}}(\mathrm{COG})/TN_{\mathrm{WO}}}{N_{\mathrm{TUB}}(\mathrm{COG})/TN_{\mathrm{TUB}}}\right) \quad (7)$$

$$\mathrm{LL}_{\mathrm{COG}d}(gene_i, gene_j) = \ln\left(\frac{1 - N_{\mathrm{WO}}(\mathrm{COG})/TN_{\mathrm{WO}}}{1 - N_{\mathrm{TUB}}(\mathrm{COG})/TN_{\mathrm{TUB}}}\right) \quad (8)$$

where $N_{\mathrm{WO}}(\mathrm{COG})$ and $N_{\mathrm{TUB}}(\mathrm{COG})$ are the total number of WO and TUB pairs in the same COG gene function. $\mathrm{LL}_{\mathrm{COG}d}$ ($gene_i$, $gene_j$) in Equation (8) represents the pair-score of adjacent genes with a different COG gene function.

Gene length ratio. TUB pairs are often associated with small values of the natural logarithm of the length ratio when the $\log_n$ of the length ratio is examined. The length ratio influences the probability of the gene pair being located within an operon (7). Dam *et al*. (7) used their experimental results to verify that the gene length ratio is a powerful tool for discerning operons. The pair-score of the gene length ratio is calculated as the natural logarithm of the length ratio of upstream genes and downstream genes (7). It is defined by the following equation:

$$\mathrm{LL}_{glr}(gene_i, gene_j) = \ln\left(\frac{\mathrm{length}_i}{\mathrm{length}_j}\right) \quad (9)$$

where length_i and length_j are the length of the upstream and downstream gene, respectively.

Operon length. The operon length is given by the number of genes in an operon (24). De Hoon *et al*. (25) evaluated the distribution of the operon length based on a list of 635 experimentally verified operons and calculated a prior probability of adjacent gene pairs within the same operon. If an operon contains just a single gene, the structure is called a singleton operon; if an operon consists of multiple genes, the operon appearance probability decreases. The Bernoulli process is the discrete equivalent of a Poisson process (25). The probability P, i.e. the pair-score of the operon length, is calculated by the following equation:

$$P_i = \frac{\overline{n} - 1}{\overline{n}} \quad (10)$$

$\overline{n}$ is the average operon length that is given by the total number of genes in all operons divided by the total number of operons in the genome. P_i represents the probability of the next gene being located in the same operon. If a random number between [0, 1] is smaller than P_i, we infer that the gene pair is in the same operon.

While the individual pair-scores are obtained by the above calculations, the overall pair-score of adjacent genes is calculated as the sum of the individual pair-scores

from the five properties mentioned above. Supplementary Figure S6 shows the fitness evaluation.

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

$$fitness_c = \sum_{i=1}^{m-1}(d_i + p_i + \mathrm{LL}_{glr}(gene_i, gene_j)) + \left(\frac{\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}(\mathrm{LL}_{path}(gene_i, gene_j) + \mathrm{LL}_{\mathrm{COG}}(gene_i, gene_j))}{n} \right) \times m,\ m = n+1 \quad (11)$$

where d_i is the pair-score of the intergenic distance of the ith gene in the cth operon, and m and n are the total number of genes and gene pairs in the cth operon, respectively. In equation (11), the pathway and COG fitness values divided by n are used to calculate the average value of the two gene pair properties, and then these averages are multiplied by m to obtain the fitness value of the two properties for the operon. An example is given in Supplementary Table S3.

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

$$fitness = \sum_{i=1}^{c} fitness_i \quad (12)$$

where c is the number of operons in the particle.

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 (26) and $V_{\max}$ and $V_{\min}$ were 6 and –6, respectively (18).

Example

An example of the performed calculations is given in Supplementary Data.

RESULTS AND DISCUSSION

Performance measurement

In this study, we used the *E. coli* genome to estimate the fitness value, and then conducted accuracy tests on other genomes. To do this, the training data set was further divided in order to be able to estimate the prediction accuracy during the search. For a large data set like *E. coli*, it is easy to build a predictor that clearly identifies WO and TUB pairs. Most previous efforts to predict operons have focused on the *E. coli* genome, which has lead to an extensive database of experimentally identified transcripts for this genome. For these reasons, *E. coli* was chosen as the training data set. We used the entire data set to estimate the fitness values, since dividing the data set into subgroups does not provide a clear advantage over using the entire data set (7). In order to verify the generalization ability of our method, test data sets do not contain the *E. coli* genome, which have the genome-specific properties. The predictive performance (7) was evaluated based on the sensitivity and specificity (Supplementary Table S4). True positive (TP) and false negative (FN) are the numbers of correctly and incorrectly predicted operon gene pairs among the WO gene pairs, respectively, whereas true negative (TN) and false positive (FP) are the numbers of correctly and incorrectly predicted operon gene pairs among the TUB gene pairs. We calculated the sensitivity, specificity and accuracy based on TP, FN, TN and FP; results are shown in Supplementary Table S4. We present an instance in which the experimental operon encoding of the genome is 111010, and the predicted operon encoding is 110110. The third and fourth genes are FN and FP, respectively. The first, second and fifth gene are TP, and the sixth gene is TN (Supplementary Figure S7). We compare the accuracy obtained to other methods and note that a good balance between sensitivity and specificity was achieved.

Receiver operating characteristic curve analysis

The sensitivity and specificity express the accuracy of the two operon prediction factors. The sensitivity is the ability to predict the WO pairs, and the specificity is the ability to predict the TUB pairs. Sensitivity and specificity show a reciprocal relationship; if one of the two factors increases, the other is decreased. Receiver operating characteristic (ROC) curves are used to express the relationship between sensitivity and specificity (27). By convention, the false-positive rate is plotted on the abscissa and the true-positive rate on the ordinate (6). The point on the ROC curve where the tangent has a slope of one is the point of maximum sensitivity and specificity. The area under the curve (AUC) represents the prediction accuracy of the method (6).

The operon prediction ROC curves are shown in Figure 2A–C. The figures show that the intergenic distance property is the most effective property for operon prediction. Figure 2A and B show a smaller AUC when the metabolic pathway or COG gene function properties are left out. It can thus be assumed that metabolic pathway and COG gene function properties are very important for operon prediction. Nevertheless, these two important properties have flaws. The false positive of the COG gene function property is very high. Metabolic pathway data are not sufficiently available in all databases, so results show a different performance on different genomes (2). In Figure 2C, the ROC curve of *S. aureus* is not smooth, since the experimentally verified operon data set is too small (1).

Contribution of selected features to operon prediction

The implemented feature selection was based on the intergenic distance, metabolic pathway, COG gene function, gene length ratio and operon length since these properties have powerful identification capabilities for operon prediction. The intergenic distance property not

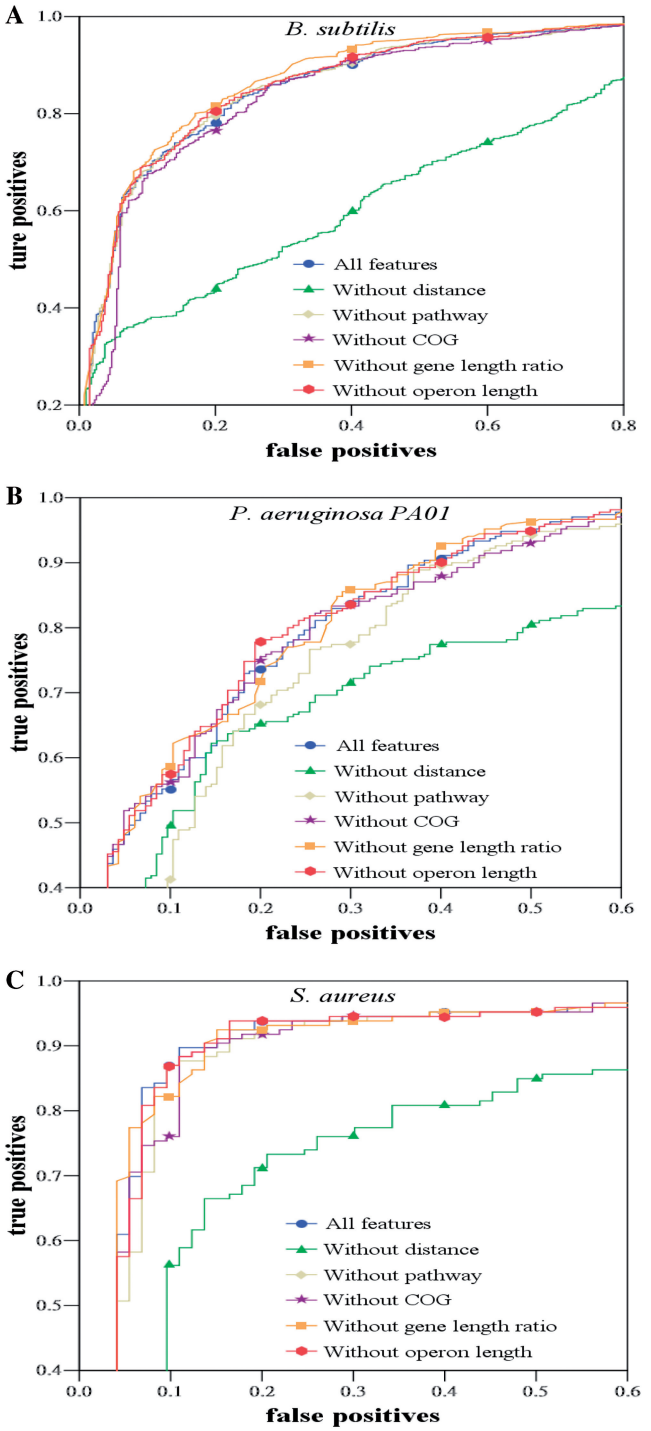

Figure 2. ROC curves of operon prediction. This study estimates the predictive ability under the circumstances of leaving a single property out on the *B. subtilis*, *P. aeruginosa PA01* and *S. aureus* data, respectively.

only plays an important role in the initial step, but also yields good results for operon prediction (3,10,20,24,28). This property can be used to universally predict 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 by the literature (4). 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% of being within the same operon on the *E. coli* genome (22). However, since the metabolic pathway and the COG gene function belong to the function relation category, the method only searches regions where these properties overlap (4). 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. Since the metabolic pathway property only determines whether adjacent genes have the same pathway or not, the COG must be used to estimate if a gene is within a functional category. In addition, the ratio of the length of gene pairs is a powerful discerning property for operon prediction (7). The operon length is a severely biased method of prediction, since the probability is directly dependent on the number of WO pairs and TUB pairs (25). Both, the gene length ratio and operon length, fall into the genome sequence category. In order to avoid searching overlapping regions of the two properties, we selected either property for operon prediction. As shown in the experimental results, the combination of the gene length ratio and the metabolic pathway property yielded superior prediction results compared to other methods; hence, the gene length ratio property was finally selected in this study.

Comparison to other methods

BPSO was applied to search for the best putative operon at each generation. The best putative operon identified by the search was then compared to experimentally verified operons. We have compared our method with various reported methods, including genetic algorithm (2), a fuzzy genetic algorithm (4), support vector machine (6) using both genome-specific and general genomic information (7), genome-wide operon prediction in *S. aureus* (3) and prediction results taken from the literature (1,8–13). In Table 1, all prediction properties of the *B. subtilis* organism available in the literature are listed and the prediction performance is compared to our method. As Table 2 shows, the prediction accuracy of the proposed method obtained the highest value on the *B. subtilis* (0.921), *P. aeruginosa PA01* (0.933) and *S. aureus* (0.959) data sets. The proposed method also showed the best performance in terms of prediction sensitivity and specificity on most of the tested bacterial genomes. For *B. subtilis*, our method had the highest sensitivity (0.930). ODB had the highest specificity (0.992), but had a low sensitivity (0.499). ODB does not achieve a good balance between sensitivity and specificity. For *P. aeruginosa PA01* and *S. aureus*, our predictor obtained a higher accuracy, sensitivity, and specificity compared to the other methods from the literature. Overall, the proposed method

Table 1. Prediction features used by each computational method on the data set of *B. subtilis*

Methodology	Features used																	
	ID	PA	GLR	HG	COG	PD	MI	MO	GO	GOC	PP	CGA	CF	PF	CAI	GCC	PR	TE
BPSO	√	√	√															
UNIPOP (1)				√														
GA (2)	√	√			√		√											
Using both genome-specific and general genomic information (7)	√		√	√		√		√	√									
SVM (6)	√	√		√						√	√							
ODB (9)	√	√					√			√								
DVDA (8)				√														
OFS (10)	√									√		√						
VIMSS (13)	√				√								√		√			
FGA (40	√	√		√										√				
JPOP (12)	√									√	√							
OPERON (11)																√		
FGENESB (http://www.softberry.com)	√									√							√	√

ID, intergenic distance; PA, pathway; GLR, gene length ratio; HG, homologous genes; COG, cluster of orthologous groups; PD, phylogenetic distance; MI, microarray; MO, motif; GO, gene ontology; GOC, gene order conservation; PP, phylogenetic profile; CGA, common gene annotation; CF, comparative features; PF, protein functions; CAI, codon adaptation index; GCC, gene cluster conservation; PR, promoter; TE, terminator.

Table 2. Accuracy, sensitivity, and specificity of operon prediction on three genomes

Genome	Methodology	Accuracy	Sensitivity	Specificity
B. subtilis (NC_000964)	BPSO	**0.921**	**0.930**	0.899
	BPSO (initiation threshold = 300 bp)	0.905	0.887	0.945
	UNIPOP (1)	0.792	0.782	0.821
	GA (2)	0.883	0.873	0.897
	Using both genome-specific and general genomic information (7)	0.902	N/A	N/A
	SVM (6)	0.889	0.900	0.860
	ODB (9)	0.632	0.499	**0.992**
	DVDA (8)	0.485	0.319	0.932
	OFS (10)	0.683	0.765	0.439
	VIMSS (13)	0.780	0.764	0.871
	FGA (4)	0.882	N/A	N/A
	JPOP (12)	0.746	0.720	0.900
	OPERON (11)	0.629	0.531	0.892
	FGENESB (http://www.softberry.com)	0.771	0.721	0.904
P. aeruginosa PA01 (NC_002516)	BPSO	**0.933**	**0.930**	0.939
	BPSO (initiation threshold = 300 bp)	0.910	0.885	**0.951**
	GA (2)	0.813	0.870	0.763
S. aureus (NC_002952)	BPSO	**0.959**	**0.959**	**0.958**
	BPSO (initiation threshold = 300 bp)	0.936	0.924	**0.958**
	Genome-wide operon prediction in *Staphylococcus aureus* (3)	0.920	N/A	N/A

N/A: Data not available.
Highest values in bold type.

obtained better results than the other methods tested for operon prediction.

DISCUSSION

GA and BPSO are both optimization algorithms. In the literature, the metabolic pathway and COG properties are often used to predict operons. Based on ROC curves published in the literature, we know that the identification ability of the microarray property is inferior to the ability of the gene length ratio property (2). If the initiation threshold is set at 300 bp, a rather small value, the prediction results do not attain a good balance between sensitivity and specificity. The reasons for the improved performance of our method compared to other methods can be found in the following factors: (i) the superiority of the BPSO algorithm; (ii) an improved initialization procedure; (iii) a fitness function designed based on statistics; and (iv) the selection of relevant properties. Each of these factors is discussed below.

(i) Most methods predict operons based 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 thus generally results in lower accuracies for operon prediction. The BPSO used in this study evaluates the properties of all genes, and thereby increases the probability of finding an optimal solution. In order to raise the BPSO prediction performance, we set the inertia weight to 1, and limit the velocity of BPSO to between $V_{\min}$ and $V_{\max}$. If the velocity is close to 0, the probability of a state changing is increased, and vice versa. Hence, BPSO has global and local search capabilities. The probability of obtaining the best solution is thus increased.

(ii) Operon prediction accuracy can be increased if better particles are selected in the initial step since the benefits of the initially superior particle are multiplied through the repeated updating process at each generation. In our study, the intergenic distance and the gene strand condition were evaluated in the initiation step. As shown in Table 2, we obtained a higher specificity and lower sensitivity when the initiation threshold was set to 300 bp. When the threshold was adjusted to 600 bp, the sensitivity was raised, but the specificity was reduced. A sensitivity and specificity value of higher than 0.8 represents a good balance between the two parameters (6). In order to obtain a good balance between sensitivity and specificity and increase the accuracy of operon prediction, proper settings at the initiation step are of critical importance. By boosting the quality of particles at the initiation, the best particles can be obtained by successive progression through the generations.

(iii) Generally, the fitness value of a particle is proportional to the prediction accuracy. Although adjacent genes have related properties, they still have a different probability of being in different operons. This necessitates the implementation of a fitness function in the proposed method. We calculate the fitness value of each particle based on the logarithmic likelihood ratio test since this method is designed on the basis of statistics. Therefore, 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.

(iv) Experimental data on the *E. coli* genome can be downloaded from the RegulonDB database, but for other genomes extensive experimental data are not readily available. In order to apply the proposed method to other genomes with fewer attributes, only five common properties for operon prediction were used. Theoretically, methods using more properties for operon prediction achieve a higher accuracy. Some of the methods in Table 1 use numerous properties, yet our BPSO method only uses three such properties and still achieves better results. The simplicity of our method can thus be considered a great attribute for operon prediction. When we used the five original properties to predict operons, the prediction accuracy did not improve, but the prediction time was increased (data not shown). Table 1 shows that the intergenic distance, homologous genes and pathway property are frequently used. ODB uses four properties for operon prediction, but the method suffers from a low prediction sensitivity (1). In addition, the WO pair and TUB pair performance of DVDA was <0.5 in the gene pair analyses performed, and the operon prediction performance based on the literature (5) was <0.2 based on the complete operons of *E. coli* and *B. subtilis*. We thus omitted the homologous gene property, and used two properties more suitable for identification of the WO and TUB pairs. The gene length ratio is used somewhat less frequently than other properties, but the literature (7) hints at the powerful identification ability of this property. Our method achieved the highest accuracy for operon prediction even though it only uses three properties on all bacterial genomes. The contributions to operon prediction are thus self-evident.

CONCLUSION

We propose a novel operon prediction method called BPSO for operon prediction in bacterial genomes. The intergenic distance and strand are applied at the initiation step, and BPSO thus superior particles are used at the initialization of a population. We used the intergenic distance, metabolic pathway, COG gene functions, gene length ratio and the operon length of the *E. coli* genome for feature selection and designed a fitness function. Finally, BPSO was used to predict operons based on the intergenic distance, metabolic pathway and gene length ratio properties. The experimental results show that the proposed method not only increases the accuracy of operon prediction on the three genome data sets tested, but also reduces the computation time needed for the prediction. In the future, we intend to investigate different properties and other algorithms on the problems of operon prediction in order to increase the prediction performance further.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

FUNDING

The National Science Council in Taiwan under grant (NSC96-2221-E-214-050-MY3, NSC96-2622-E-214-004-CC3, and NSC97-2622-E-151-008-CC2). Funding for open access charge: National Science Council in Taiwan (NSC96-2221-E-214-050-MY3).

Conflict of interest statement. None declared.

REFERENCES

1. Li,G., Che,D. and Xu,Y. (2009) A universal operon predictor for prokaryotic genomes. *J Bioinform Comput Biol.*, **7**, 19–38.
2. Wang,S., Wang,Y., Du,W., Sun,F., Wang,X., Zhou,C. and Liang,Y. (2007) A multi-approaches-guided genetic algorithm with application to operon prediction. *Artif. Intell. Med.*, **41**, 151–159.
3. Wang,L., Trawick,J.D., Yamamoto,R. and Zamudio,C. (2004) Genome-wide operon prediction in *Staphylococcus aureus*. *Nucleic Acids Res.*, **32**, 3689–3702.

4. Jacob,E., Sasikumar,R. and Nair,K.N. (2005) A fuzzy guided genetic algorithm for operon prediction. *Bioinformatics*, **21**, 1403–1407.
5. Brouwer,R.W., Kuipers,O.P. and van Hijum,S.A. (2008) The relative value of operon predictions. *Brief Bioinform.*, **9**, 367–375.
6. Zhang,G.Q., Cao,Z.W., Luo,Q.M., Cai,Y.D. and Li,Y.X. (2006) Operon prediction based on SVM. *Comput. Biol. Chem.*, **30**, 233–240.
7. Dam,P., Olman,V., Harris,K., Su,Z. and Xu,Y. (2007) Operon prediction using both genome-specific and general genomic information. *Nucleic Acids Res.*, **35**, 288–298.
8. Edwards,M.T., Rison,S.C., Stoker,N.G. and Wernisch,L. (2005) A universally applicable method of operon map prediction on minimally annotated genomes using conserved genomic context. *Nucleic Acids Res.*, **33**, 3253–3262.
9. Okuda,S., Katayama,T., Kawashima,S., Goto,S. and Kanehisa,M. (2006) ODB: a database of operons accumulating known operons across multiple genomes. *Nucleic Acids Res.*, **34**, D358–D362.
10. Westover,B.P., Buhler,J.D., Sonnenburg,J.L. and Gordon,J.I. (2005) Operon prediction without a training set. *Bioinformatics*, **21**, 880–888.
11. Ermolaeva,M.D., White,O. and Salzberg,S.L. (2001) Prediction of operons in microbial genomes. *Nucleic Acids Res.*, **29**, 1216–1221.
12. Chen,X., Su,Z., Dam,P., Palenik,B., Xu,Y. and Jiang,T. (2004) Operon prediction by comparative genomics: an application to the *Synechococcus* sp. WH8102 genome. *Nucleic Acids Res.*, **32**, 2147–2157.
13. Price,M.N., Huang,K.H., Alm,E.J. and Arkin,A.P. (2005) A novel method for accurate operon predictions in all sequenced prokaryotes. *Nucleic Acids Res.*, **33**, 880–892.
14. Gama-Castro,S., Jimenez-Jacinto,V., Peralta-Gil,M., Santos-Zavaleta,A., Penaloza-Spinola,M., Contreras-Moreira,B., Segura-Salazar,J., Muniz-Rascado,L., Martinez-Flores,I. and Salgado,H. (2007) RegulonDB (version 6.0): gene regulation model of *Escherichia coli* K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. *Nucleic Acids Res.*, **36**, D120–D124.
15. Sierro,N., Makita,Y., de Hoon,M. and Nakai,K. (2008) DBTBS: a database of transcriptional regulation in *Bacillus subtilis* containing upstream intergenic conservation information. *Nucleic Acids Res.*, **36**, D93–D96.
16. Sabatti,C., Rohlin,L., Oh,M.K. and Liao,J.C. (2002) Co-expression pattern from DNA microarray experiments as a tool for operon prediction. *Nucleic Acids Res.*, **30**, 2886–2893.
17. Kennedy,J. and Eberhart,R. (1995) Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 4, pp. 1942–1948.
18. Kennedy,J. and Eberhart,R. (1997) A discrete binary version of the particle swarm algorithm. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 5, pp. 4104–4108.
19. Crammer,K. and Singer,Y. (2002) On the learnability and design of output codes for multiclass problems. *Machine Learn.*, **47**, 201–233.
20. Salgado,H., Moreno-Hagelsieb,G., Smith,T.F. and Collado-Vides,J. (2000) Operons in *Escherichia coli*: genomic analyses and predictions. *Proc. Natl Acad. Sci. USA*, **97**, 6652–6657.
21. Yan,Y. and Moult,J. (2006) Detection of operons. *Proteins*, **64**, 615–628.
22. Romero,P.R. and Karp,P.D. (2004) Using functional and organizational information to improve genome-wide computational prediction of transcription units on pathway-genome databases. *Bioinformatics*, **20**, 709–717.
23. Tran,T.T., Dam,P., Su,Z., Poole,F.L. II, Adams,M.W., Zhou,G.T. and Xu,Y. (2007) Operon prediction in *Pyrococcus furiosus*. *Nucleic Acids Res.*, **35**, 11–20.
24. Bockhorst,J., Craven,M., Page,D., Shavlik,J. and Glasner,J. (2003) A Bayesian network approach to operon prediction. *Bioinformatics*, **19**, 1227–1235.
25. De Hoon,M.J., Imoto,S., Kobayashi,K., Ogasawara,N. and Miyano,S. (2004) Predicting the operon structure of *Bacillus subtilis* using operon length, intergene distance, and gene expression information. *Pac. Symp. Biocomput.*, 276–287.
26. Kennedy,J., Eberhart,R. and Shi,Y. (2001) *Swarm Intelligence*. Springer, New York.
27. Roback,P., Beard,J., Baumann,D., Gille,C., Henry,K., Krohn,S., Wiste,H., Voskuil,M.I., Rainville,C. and Rutherford,R. (2007) A predicted operon map for *Mycobacterium tuberculosis*. *Nucleic Acids Res.*, **35**, 5085–5095.
28. Moreno-Hagelsieb,G. and Collado-Vides,J. (2002) A powerful non-homology method for the prediction of operons in prokaryotes. *Bioinformatics*, **18**, S329–S336.