Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2007)

Operon Prediction in Microbial Genomes Using Decision Tree Approach

Dongsheng Che, Jizhen Zhao, Liming Cai and Ying Xu[1]
Department of Computer Science
[1]Department of Biochemistry and Molecular Biology, and Institute of Bioinformatics
University of Georgia, Athens, GA 30602
Email: {che, jizhen, cai}@cs.uga.edu, xyn@bmb.uga.edu

***Abstract* -- Identifying operons at the whole genome scale of microbial organisms can facilitate deciphering of transcriptional regulation, biological networks and pathways. A number of computational methods, such as naïve Bayesian and neural network approaches, have been employed for operon prediction to whole genome sequences of a number of prokaryotic organisms, based on features known to be associated with operons, such as intergenic distance, microarray expression data, phylogenetic profiles, clusters of orthologous groups (COG). In this paper, we introduce a decision tree approach to predict operon structures using three effective types of genomic data: intergenic distance, gene order conservation and COG. We calculated and analyzed frequency distributions of each attribute of known operons and non-operons of *Escherichia coli* (*E. coli*) K12 and *Bacillus subtilis* (*B. subtilis*) 168, and constructed decision trees based on training examples to predict operons. The overall prediction accuracy is 94.1% for *E. coli* K12 and 91.0% for *B. subtilis* 168. We also applied four other classifiers, logistic regression, naïve Bayesian, neural network and support vector machines on both organisms. The results indicate that the decision tree approach is the best classifier for operon prediction. The software package operonDT is freely available at http://www.cs.uga.edu/~che/OperonDT.**

I. INTRODUCTION

The operon structure is the one of special features unique to prokaryotic organisms, although a few eukaryotic organisms, such as *Caenorhabditis elegans*, do have operon-like structures [1]. An operon is defined as a set of genes that are arranged in a tandem and are co-transcribed as a unit, *i.e.*, all genes in an operon share a common promoter and terminator. These co-transcribed genes often act together in a specific biochemical pathway or in a common biological process. For example, genes of the lactose operon in *E. coli* participate in lactose metabolism. Therefore, identifying operons in microbial genomes is a fundamental step for biologists to further understand the gene regulation network and pathways.

Various features and methods have been used in operon prediction. The most direct approach is to identify the boundaries of transcriptional units based on known or computationally identified promoters and terminators [2-4]. For example, Yada *et al.* [4] trained an operon predictor based on known promoters and terminators of *E. coli*, using hidden Markov models (HMMs), to predict transcriptional units. 60% of 390 known transcriptional units were identified using their trained HMMs. In addition to transcriptional signal information, Bockhorst *et al.* [2, 5] also used gene lengths and intergenetic spacing, codon usage statistics and other features to construct Bayesian networks for operon prediction. This method could identify over 78% operons in *E. coli*, with only about 10% false positive rate. Based on the principle that the genes in the same operon usually have related functions and are involved in the successive reactions in metabolic pathways, Zheng *et al.* [6] developed a pipeline for operon prediction by using biochemical pathway knowledge. Sabatti *et al.* [7] used microarray expression data as a tool to predict operons.

Recent studies show that other features, such as intergenic distances [3, 8-13], cluster of orthologous groups (COG) [8, 10] are also very effective in operon prediction. Salgado *et al.* [11] found the intergentic distances of adjacent gene pairs within operons are usually shorter than those of at the borders. They used log-likelihood ratio of intergenic distance of adjacent gene pairs between within operons and at the borders to correctly predict 75% operons in *E. coli*. Chen *et al.* [8] developed a neural network using intergenic distance, COG function and phylogenetic profiles as inputs. They have achieved an overall accuracy of 83.8% in *E. coli* K12. More recently, Westover *et al.* [13] proposed a method that does not need extensive training data. They used naïve Bayesian approach with attributes such as intergenic distance, common annotation length and inclusion in a common cluster. This method achieves a true positive rate of 88% and 20% false positives in *E. coli*, and a true positive rate of 73% with 20% false positives in *Bacteroides theta*. Similar to Westover's work in terms of no prior training, Jacob *et al.* [9] proposed a fuzzy guided genetic algorithm-based approach using four scoring criteria: intergenic distance, participation in the same metabolic pathway, phylogenetic profiles and COG. The prediction accuracy of *E. coli* K12 and *B. subtitils* was evaluated by ROC (receiver operating characteristic) analysis, and the area under the ROC curve is around 0.9. Interestingly, without using any important features such as intergenic distance, Edward *et al.* used homologous gene pair information among multiple genomes based on BLAST search, and applied maximum bipartite matching algorithm to detect operons with prediction accuracy of 85% in *E. coli*

K12 [14]. More recently, Tran *et al.* used the predicted results from three popular operon predictors (JPOP, OFS and VIMSS) to train a neural network, and showed this approach could reach the prediction accuracy of around 90% in both *E. coli* K12 and *B. subtitils* 168 [submitted].

In this paper, we aim to seek a high accurate operon prediction method based on previous work. All previous prediction methods tried to classify whether two adjacent genes belong to the same operon or not based on the scores assigned to the gene pair. For instance, if gene *a* and gene *b* are an adjacent gene pair, and their COG code are the same, they will have a high score and thus have high probability of belonging to the same operon. Unlike the previous methods, we are trying to classify if a gene belongs to a member of operon structure based on the attribute values of the gene itself, which are in turn based on the information of its neighbor genes. The use of information of neighboring genes (usually more than two) instead of its adjacent gene has its statistical reasoning. Suppose two adjacent genes belong to the same operon if they have the same COG code, then the probability of a gene belongs to the same operon structure with its several consecutive neighbor genes should be much higher if we see all its neighbor genes have the same COG code as this gene. We use three effective types of information (*i.e.*, intergenic distance, gene order conservation and COG) to evaluate whether the gene belongs to a member of the operon or not. Since we now evaluate the attribute values on single genes instead of on gene pairs, calculations of these attributes are different. Detailed calculations of these feature values are described in the following section. Based on attribute values of the training examples, we construct decision trees for operon prediction on both *E. coli* K12 and *B. subtitils* 168, and then evaluate the prediction accuracies.

The remainder of the paper is organized as follows. Section 2 describes the methods used to implement the classifier. Section 3 shows the prediction results of decision trees for *E. coli* K12 and *B. subtitils* 168. The paper is concluded in Section 4 with a discussion of the possible direction in the future work.

II. METHODS

A. Data Sources

The annotated complete genome sequences were downloaded from NCBI GenBank database (http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi). The annotations of *E. coli* K12 and *B. subtitils* 168 are used for calculating intergenic distance and COG information. We picked 177 representative genomes as reference genomes and run BLAST to obtain homologous gene pairs, which were used for calculating gene order conservation. Experimentally confirmed operon dataset of *E. coli* K12 were downloaded from regulonDB database [15]. Operons of *B. subtitils* 168 were extracted from operon database (http://odb.kuicr.kyoto-u.ac.jp/) [16], in which operons were originally obtained from transcriptional maps stored in BSORF (http://bacillus.genome.jp/). We call operon genes if multiple genes are within a transcription unit (or operon), and a non-operon gene if there is only one gene contained in the corresponding transcription unit. This classification leads to 1140 operon genes and 230 non-operon genes for *E. coli* K12, 955 operon genes and 181 non-operon genes for *B. subtitils* 168.

B. Intergenic Distance Analysis

Intergenic distance has been proved to be one of the most effective attributes to discriminate operon genes from non-operon genes. Intergenic distances between an operon gene and its adjacent neighbor genes are usually very short, and it is very common that genes overlap, thus leading to negative intergenic distances. We define the intergenic distance of a gene and its adjacent gene as the number of base pairs between the two genes, or the number of base pairs overlapped. We denote $d(g_{i-1}, g_i)$ be the distance between a gene and its left adjacent gene, and similarly, $d(g_i, g_{i+1})$ be the distance between a gene and its right adjacent gene. The shorter distance (*S_Dist*) and longer distance (*L_Dist*) are defined as

$$S\_Dist = min\ (d(g_{i-1}, g_i), d(g_i, g_{i+1})) \quad (1)$$

$$L\_Dist = max\ (d(g_{i-1}, g_i), d(g_i, g_{i+1})) \quad (2)$$

Figure 1 shows frequency distributions of *S_Dist* for experimentally confirmed non-operon and operon genes for both *E. coli* K12 and *B. subtitils* 168. In both organisms, operon genes tend to have short *S_Dist*, with only a few having their *S_Dist* greater than 150. In contrast, *S_Dist* for non-operon genes are relatively more uniformly distributed. Frequency distributions of *L_Dist* for non-operon and operon genes are also shown in Figure 1. These frequency distributions indicate the feature of intergenic distance is an effective indicator for operon prediction.

C. Analysis of Gene Order Conservation

Although the whole gene order in many known operons is not conserved among different organisms, partial gene order conservation in operons does exist, such as the *Trp* operon [17]. The conservation of the gene order is possibly due to the physical interaction such as molecular complex, or at least some association in the same biological processes. Thus, developing a scoring scheme based on the conservation of gene order should be a good indicator for finding operon gene.

A simple but powerful scheme is to find out how many consecutive genes are gene-order conserved in which the gene is located between the query genome (G) and the reference genome (G') based on BLAST results. In practice, there could be several genes from G' that are homologous to gene g_i from G. We choose the maximum of all scores of gene order conservation (*goc*), denoted as *max*(*goc*). In particular, for every homologous gene pair (g_i, g_j) ($g_i \in G$ and $g_j \in G'$) for g_i, we first identify their corresponding directons (the set of consecutive genes having the same

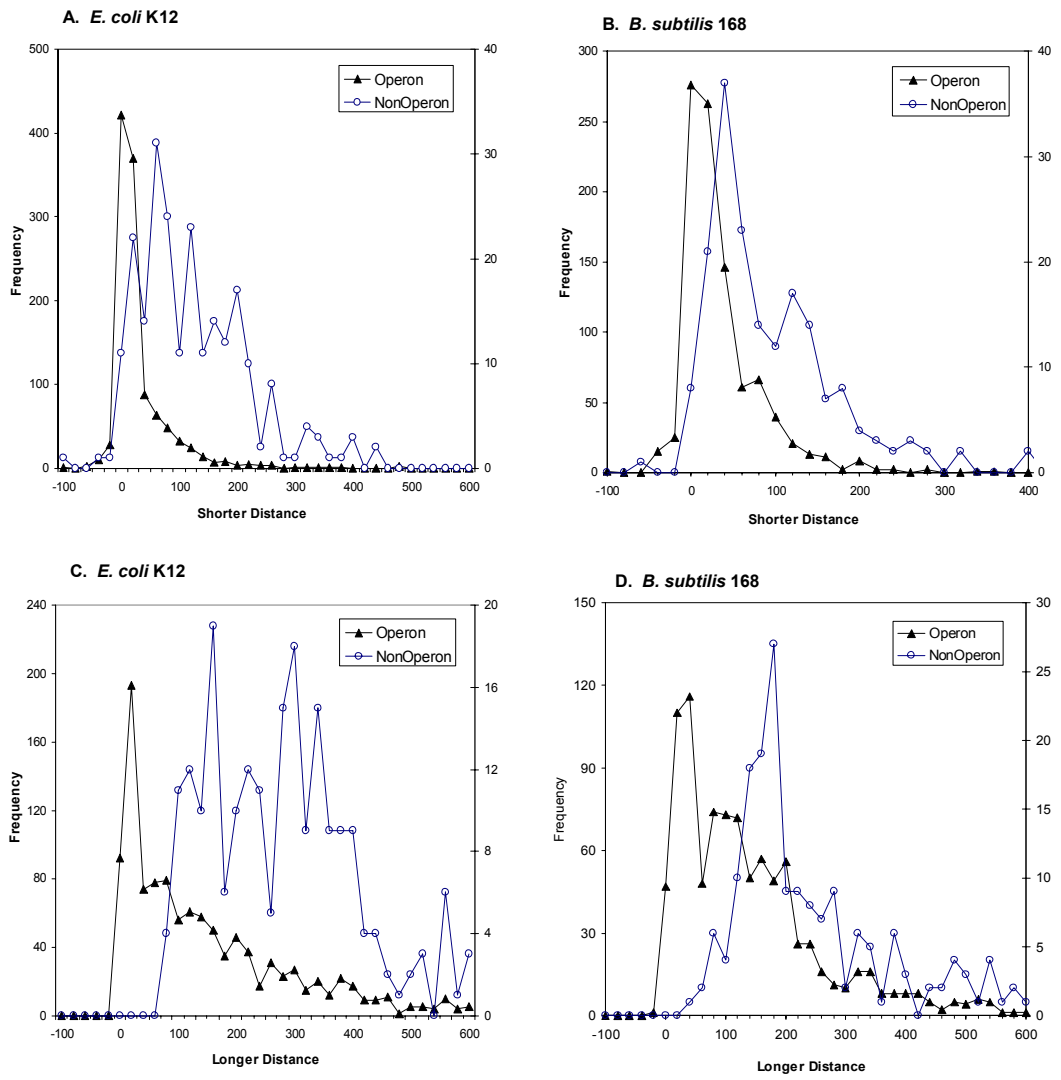

Fig. 1. Frequency distributions of shorter and longer intergenic distances for non-operon genes and operon genes of *E. coli* K12 (A and C) and *B. subtilis* 168 (B and D)

strand directions with gene g_i or g_j). We keep updating the gene order conservation score (s_k) until their corresponding neighbor gene pairs are not homologous, or the boundary of at least one directon has been reached. Thus, *max*(*goc*) is defined as

$$max(goc) = max\ (s_1, s_2, \ldots, s_k, \ldots, s_K) \quad (3)$$

where K is the total number of homologous gene pairs for gene g_i in G.

To make attribute values statistically meaningful, we have used multiple reference genomes to evaluate the gene order conservation for each gene in the genome of *E. coli* and *B. subtitils* 168. To avoid the redundant information of close related genomes, we picked one representative strain from each organism, and thus 177 reference genomes were chosen of all complete microbial genomes. Therefore, the overall gene order conservation score for gene g_i is the summation of all *max*(*goc*) against 177 reference genomes. Figure 2 describes the algorithm to calculate gene order conservation scores for all genes in the query genome G.

By applying this algorithm, we obtained the scores of gene order conservation for all genes in *E. coli* K12 and *B. subtitils* 168. Figures 3A and 3B show frequency distributions of calculated scores for experimentally confirmed non-operon and operon genes. For *E. coli*, only 17 out of 230 non-operon genes have gene order conservation scores higher than 40, which is equivalent to 7.4%. In contrast, 620 out of 1140 operon genes have the conservation score higher than 40, which accounts for 54.4%. For *B. subtitils* 168, 13 out of 181 (7.2%) non-operon genes have the conservation score higher

[Input] Query genome G with M genes ($g_1, g_2, \ldots, g_M$) and N directons ($D_1, D_2, \ldots D_N$)
R Reference genomes $G'_1, G'_2, \ldots G'_R$, with
genes ($g'_{11}, g'_{12}, \ldots, g'_{1M\_1}$) and directons ($D'_{11}, D'_{12}, \ldots D'_{1N\_1}$),
genes ($g'_{21}, g'_{22}, \ldots, g'_{2M\_2}$) and directons ($D'_{21}, D'_{22}, \ldots D_{2N\_2}$), …
genes ($g'_{R1}, g'_{R2}, \ldots, g'_{RM\_R}$) and directons ($D'_{R1}, D'_{R2}, \ldots D'_{RN\_R}$)
Homologous gene pairs between genomes of (G, G'_1), (G, G'_2), … (G, G'_R)

[Output] Gene order conservation score S_1, S_2, S_M for $g_1, g_2, \ldots, g_M$

Iterations

for $i = 1$ to M
  let $S_i = 0$;
  find $d_1 \in (D_1, D_2, \ldots D_N)$ s.t. $g_i \in d_1$
  for $j = 1$ to R
    let *max_s = 0;*
    for $k = 1$ to K homologous gene pairs ((g_i, g'_{j1}), (g_i, g'_{j2}), …, (g_i, g'_{jK}))
      $d_2 \in (D'_{j1}, D'_{j2}, \ldots D'_{jN\_j})$ s.t. $g_i \in d_1$
      let s_{ll}, s_{rr}, s_{lr} and s_{rl} be 0;
      let $g_1 = g_i$ and $g_2 = g'_{ik}$;
      while ($g_1 \in d_1$ && $g_2 \in d_2$ && g_1, g_2 are homologous)
        increment s_{ll} by 1;
        g_1 = the left neighbor gene of g_1;
        g_2 = the left neighbor gene of g_2;
      while ($g_1 \in d_1$ && $g_2 \in d_2$ && g_1, g_2 are homologous)
        increment s_{rr} by 1;
        g_1 = the right neighbor gene of g_1;
        g_2 = the right neighbor gene of g_2;
      while ($g_1 \in d_1$ && $g_2 \in d_2$ && g_1, g_2 are homologous)
        increment s_{lr} by 1;
        g_1 = the left neighbor gene of g_1;
        g_2 = the right neighbor gene of g_2;
      while ($g_1 \in d_1$ && $g_2 \in d_2$ && g_1, g_2 are homologous)
        increment s_{rl} by 1;
        g_1 = the right neighbor gene of g_1;
        g_2 = the left neighbor gene of g_2;
      $s_k = max(s_{ll} + s_{rr}, s_{lr} + s_{rl})$;
      if $s_k > max\_s$;
        $max\_s = s_k$;
    $S_i = S_i + max\_s$;
End for loop

Fig. 2. The algorithm for calculating gene order conservation scores

than 10, while 537 out of 955 (56.3%) operon genes have the conservation score higher than 10.

D. COG Analysis

COGs are created by identifying the best hit for each gene in complete pairwise comparisons of a set of genomes [18]. Each COG consists of individual proteins or groups of paralogs from at least three lineages. There are three levels in the COG function hierarchy, with the first level consisting of four categories (*i.e.*, information storage and processing, celllular processes, metabolism and poorly characterized). The second level is much more specific base on the first level. For example, Translation, ribosomal structure and biogenesis (J) belongs to the first category of the first level, and transcription (T) belongs to the second category of the first level. Totally, there are twenty-six categories of the second level.

Previous study showed that using the information of the first level can help to differentiate the operon and non-operon gene a little [8]. In this study, we use the category

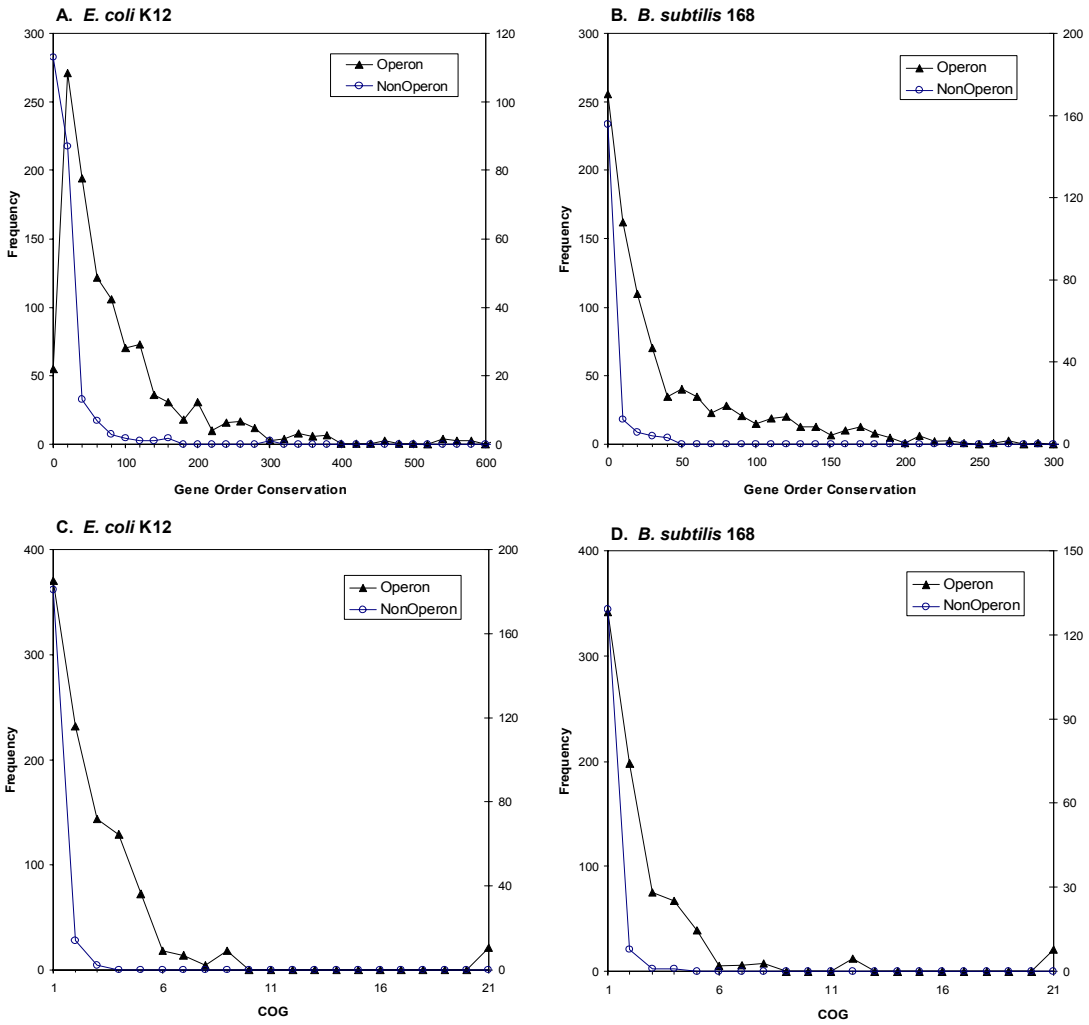

Fig. 3. Frequency distributions of gene order conservation and COG scores for non-operon genes and operon genes of *E. coli* K12 (A and C) and *B. subtilis* 168 (B and D)

information from the second level. In particular, for any gene g_i in genome G, we first identify the directon that contains g_i. We keep expanding to its neighbors until the COG code of its neighbor is not the same as that of g_i, or the boundary of the directon has been reached. The number of consecutive genes with the same COG code with that of g_i is the COG score for g_i. Since some genes do not have COG annotation, we could not evaluate their COG scores, and thus exclude them for COG analysis. Figures 3C and 3D show frequency distributions of COG scores for known non-operon and operon genes. In *E. coli*, only 16 out of 197 non-operon genes (8.1%) have their COG scores higher than one, while 652 out of 1022 operon genes (63.8%) have their COG scores higher than one. A similar COG distribution pattern exists in *B. subtitils* 168. 10 out of 139 non-operon genes (about 7.2%) have their COG scores higher than one, while 431 out 773 operon genes (about 55.8%) have their COG scores higher than one.

E. Decision Tree Classification

Decision tree classification is one of most widely used machine learning methods and has many biological applications. For example, Salzberg showed that the decision tree approach could accurately classify coding and noncoding DNA [19]. Selbig *et al.* predicted consensus protein secondary structure by decision tree trained based on known data [20]. Recently, Wong *et al.* combined multiple types of data to predict genetic interactions in *Saccharomyces cerevisiae* by using probabilistic decision trees [21].

In this study, five input attributes were used, including shorter distance, longer distance, score of COG, score of gene order conservation, strand direction information between the target gene and its adjacent genes. The predicted

attribute values are operon genes and non-operon genes. We use C4.5, one of the popular decision tree learning algorithms that employs a top-down, greedy search to construct decision trees [22]. In particular, we start with all genes of the training set in the root node, and pick the attribute that best classifies the training data based on the *information gain* (*IG*), which is defined as

$$IG(S,A) = E(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} E(S_v) \qquad (4)$$

where *Value*(*A*) is the set of all values for attribute *A* (*A* could be shorter distance, longer distance, COG, gene order conservation or strand direction information), and S_v is the subset of *S* for which attribute *A* has value *v* (*i.e.*, $S_v = \{s \in S \mid A(s) = v\}$). In our case, *v* is either an operon gene or a non-operon gene. $E(S)$ is the entropy of *S*, which is defined as

$$E(S) = -p_o \log_2 p_o - p_{no} \log_2 p_{no} \qquad (5)$$

p_o is the probability that the gene is an operon gene (*i.e.*, the percentage of positive examples in *S*), and p_{no} is the probability that the gene is a non-operon gene. We split the set based on the possible values of the selected feature. If the subset contains examples of only one class, then the process stops, and the node become a leaf node. On the other hand, if the subset does contain examples from two classes, we recursively split the node.

WEKA is open source software that contains many machine learning algorithms for data mining tasks [23]. It includes decision tree, logistic regression (LR), naïve Bayesian (NB), neural network (NN), support vector machines (SVMs) and many other classifiers. We applied C4.5 (J48) of WEKA as our decision tree classifier on our dataset.

F. Evaluation

A ten-fold cross-validation scheme was used to evaluate the prediction accuracy of decision tree approach. In particular, the known operon dataset is evenly separated into ten parts, and the first part is evaluated based on the model trainde from the remaining nine parts. This process continues until all ten parts are evaluated. The overall accuracy is the average of all ten separate evaluations. True positives (*TP*) were the number of operon genes predicted to be operon genes. False negatives (*FN*) were the number of operon genes predicted to be non-operon genes. True Negatives (*TN*) were the number of non-operon genes predicted to be non-operon genes. False positives (*FP*) were the number of non-operon genes predicted to be operon genes. Sensitivity (*Sen*) and specificity (*Spc*) were defined as equation (6) and (7). The overall accuracy was the average of sensitivity and specificity.

$$Sen = TP/(TP + FN) \qquad (6)$$

$$Spc = TN/(TN + FP) \qquad (7)$$

III. RESUITS

To evaluate the decision tree approach for predicting operons, we tested two well-studied organisms (*i.e.*, *E. coli* K12 and *B. subtilis* 168) because of the availability of known operon information. We applied J48 (with default parameter settings) on datasets generated by our program called 'operonFT', which can be obtained at http://www.cs.uga.edu/~che/operonDT. Table 1 lists results of sensitivity, specificity and overall accuracy. For *E. coli* K12, we have achieved sensitivity of 0.889 and specificity of 0.993. The overall accuracy is 0.941. For *B. subtilis* 168, the prediction sensitivity and specificity are 0.859 and 0.960, respectively. To compare the decision tree method with other classifiers in WEKA, we applied four classifiers to the same datasets, including LR, NB, voted percetron (VP), one of NN implementations, and Sequential Minimal Optimization (SMO), one of SVMs implementations. We also chose default parameter settings for these four classifiers. As shown in Table 1, prediction accuracies of the decision tree method on both organisms are higher than any of other four methods. For example, the prediction accuracy on *E. coli* is 94.1% using decision tree approach, while the prediction accuracies are 84.7% and 86.3% for SMO and LR methods respectively.

TABLE 1
Sensitivity, specificity and accuracy of operon prediction on *E. coli* K12 and *B. subtilis* 168

Organism	Method	Sensitivity	Specificity	Accuracy
E. coli k12	J48	0.889	0.993	0.941
	SMO	0.855	0.839	0.847
	LR	0.867	0.860	0.863
	VP	0.804	0.899	0.852
	NB	0.723	0.946	0.834
*B. subtilis*168	J48	0.859	0.960	0.910
	SMO	0.851	0.797	0.824
	LR	0.839	0.830	0.834
	VP	0.691	0.874	0.782
	NB	0.527	0.966	0.747
E. coli k12 +	J48	0.865	0.953	0.909
B. subtilis 168	SMO	0.858	0.833	0.846
	LR	0.864	0.845	0.854
	VP	0.745	0.900	0.822
	NB	0.628	0.959	0.793

The option of confidence factor in J48 can be adjusted to affect the size of the decision tree by pruning the tree. The default setting for the confidence value (*c*) is 0.25. Table 2 lists the tree sizes and prediction accuracies with different confidence values. In general, prediction accuracies and tree sizes increase with the increase of confidence values. For instance, the prediction accuracy for *E. coli* K12 is 0.917 with c = 0.05, and 0.941 with c = 0.5. The decision tree models with high confidence value might be suitable for predicting operons of genomes in which partial training sets available, such as *E. coli* K12 and *B. subtilis* 168. Thus,

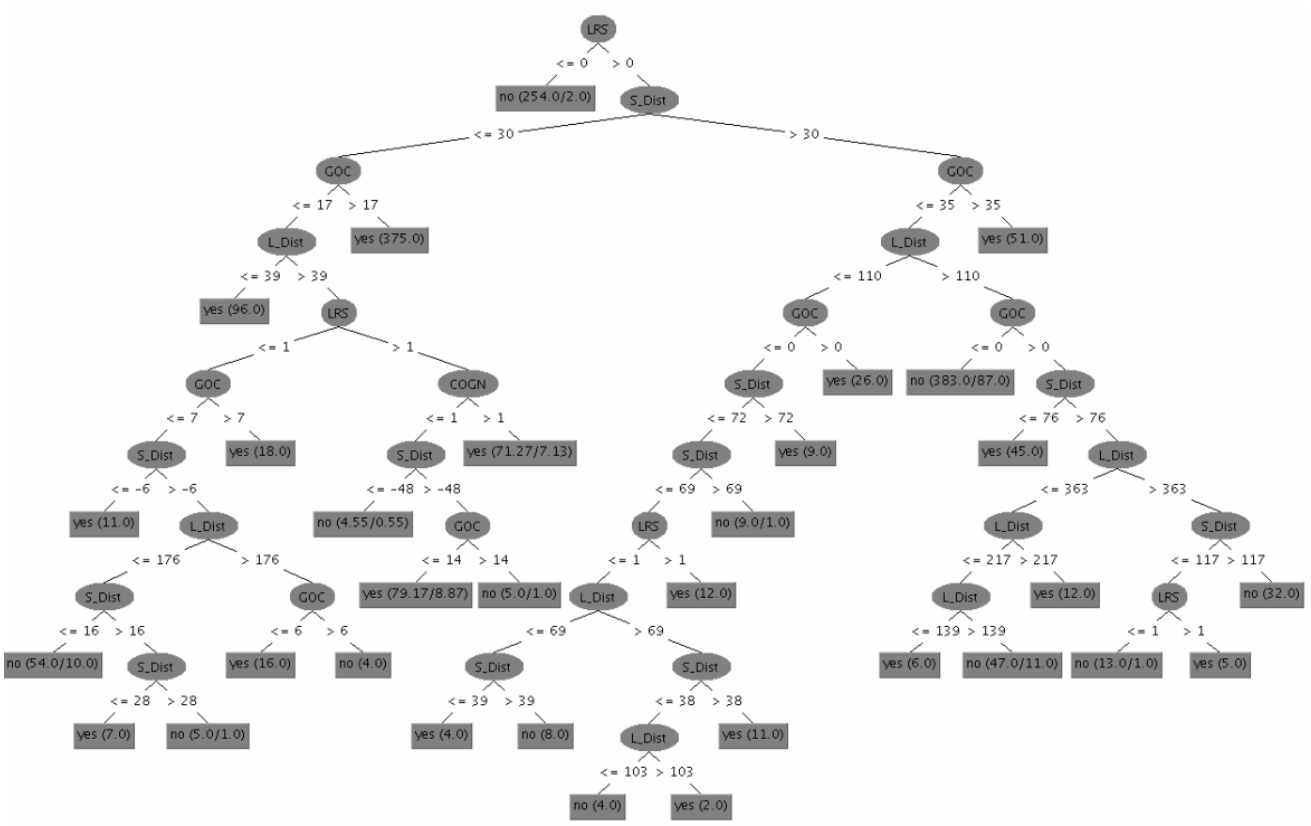

Fig. 4. An example of a decision tree model for *B. subtilis* 168 (confidence c= 0.1)

organism specific decision tree models with default confidence setting should work fine for predicting the remaining operons. The complete set of decision tree models and predicted operon structures can be viewed on our website at http://www.cs.uga.edu/~che/OperonDT. As an illustration, Fig. 4 shows the decision tree model of *B. subtilis* 168 with the confidence value of 0.1.

The real challenge in practice is to predict operon structures for genomes without any operon training sets. Thus, organism specific models with high accuracy might not be applicable to those genomes. One simple solution is to mix all available training sets from multiple organisms. In our experiment, we mixed the datasets of *E. coli* K12 and *B. subtilis* 168. In addition, we might use a low confidence value to make the model relatively universal. Table 2 shows the prediction accuracies and tree sizes of these models.

TABLE 2
Prediction accuracy, the number of leaves and tree sizes of the decision tree models with different confidence values

Organism	Confidence	Accuracy	Leaves	TreeSize
E. coli k12	0.050	0.917	51	101
	0.100	0.932	63	125
	0.250	0.941	63	125
B. subtilis 168	0.050	0.905	29	57
	0.100	0.907	32	63
	0.250	0.910	42	83
E. coli k12 +	0.050	0.893	73	145
B. subtilis 168	0.100	0.901	81	161
	0.250	0.909	101	201

IV. Discussions

Preliminary experimental results showed that decision tree approach is the best classifier for predicting operons in two genomes. Compared with other classifiers, the decision tree method can improve about 10% prediction accuracy. Further analysis of those attribute patterns might explain why the decision tree based approach might be better than other methods for this problem. We found that none of non-operon genes whose score of COG is higher than 3 for *E. coli*, while about 30% operon genes whose COG scores are higher than 3. Similarly, no non-operon gene in *B. subtilis* has COG score higher than 4, but 11.7% operon genes have their COG scores higher than 4. Distribution differences between operon and non-operon genes also exist in other attributes. These clear-cut differences make the decision tree-based

classification easily and accurately. On the other hand, prediction based on overall attribute values, such as NB approach, might misclassify some genes if these attribute values are conflicting. For example, if a gene has large intergenic distance with its adjacent genes on both sides, but the COG score is also high, say 10, then it is very hard for NB approach to correctly classify this gene since operon genes tend to have high COG scores and small intergenic distances. In contrast, decision tree approach simply classifies it as an operon gene since all higher COG scores (>4) are operon genes in the training set.

Although we have shown that the decision tree approach is a very powerful method in terms of prediction accuracy, we have also found that a few genes could not be correctly classified by using any machine learning method based on current features (results not shown). Therefore, a complete new feature might be included for the decision tree. Recently, Janga *et al.* found that the oligonucleotide signatures of promoter regions are different from the upstream regions in the middle of operons [24]. We believe that adding this new discovered feature information should make our decision tree approach more accurate.

The importance of the operon prediction problem is to predict those prokaryotic organisms without any operon information. Our prediction results on the dataset of two organism shows the overall prediction accuracy did not decrease compared with that of on two separate predictions. This indicates that the model trained from the mixed dataset maybe used for predicting other genomes. However, we are aware that this model was built based on training sets from two organisms. In our future work, we will build a general model by using all available operon data from ODB [16], including *B. subtilis*, *E. coli*, *Pseudomonas aeruginosa*, *Agro. tumefaciens*, *Synechocystis* sp. PCC6803, *Bradyrhizobium japonicum*, and *Pyrococcus furiosus*. In addition, a small confidence value will be chosen to make the model be general. We hope we can apply the universal model to predict operons of all other organisms with high prediction accuracy.

IV. Acknowledgement

This research was supported in part by National Science Foundation (#NSF/DBI-0354771, #NSF/ITR-IIS-0407204 and #NSF/DBI-0542119) and by a “distinguished scholar” grant from Georgia Cancer Coalition.

References

[1] T. Blumenthal, D. Evans, C. D. Link, A. Guffanti, D. Lawson, J. Thierry-Mieg, D. Thierry-Mieg, W. L. Chiu, K. Duke, M. Kiraly, and S. K. Kim, "A global analysis of Caenorhabditis elegans operons," *Nature,* vol. 417, pp. 851-4, Jun 20 2002.

[2] J. Bockhorst, M. Craven, D. Page, J. Shavlik, and J. Glasner, "A Bayesian network approach to operon prediction," *Bioinformatics,* vol. 19, pp. 1227-35, Jul 1 2003.

[3] 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 Res,* vol. 32, pp. 2147-57, 2004.

[4] 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. 987-93, Dec 1999.

[5] M. Craven, D. Page, J. Shavlik, J. Bockhorst, and J. Glasner, "A probabilistic learning approach to whole-genome operon prediction," *Proc Int Conf Intell Syst Mol Biol,* vol. 8, pp. 116-27, 2000.

[6] Y. Zheng, J. D. Szustakowski, L. Fortnow, R. J. Roberts, and S. Kasif, "Computational identification of operons in microbial genomes," *Genome Res,* vol. 12, pp. 1221-30, Aug 2002.

[7] 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 Res,* vol. 30, pp. 2886-93, Jul 1 2002.

[8] X. Chen, Z. Su, Y. Xu, and T. Jiang, "Computational Prediction of Operons in Synechococcus sp. WH8102," *Genome Inform Ser Workshop Genome Inform,* vol. 15, pp. 211-22, 2004.

[9] E. Jacob, R. Sasikumar, and K. N. Nair, "A fuzzy guided genetic algorithm for operon prediction," *Bioinformatics,* vol. 21, pp. 1403-7, Apr 15 2005.

[10] 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 Res,* vol. 33, pp. 880-92, 2005.

[11] H. Salgado, G. Moreno-Hagelsieb, T. F. Smith, and J. Collado-Vides, "Operons in Escherichia coli: genomic analyses and predictions," *Proc Natl Acad Sci U S A,* vol. 97, pp. 6652-7, Jun 6 2000.

[12] L. Wang, J. D. Trawick, R. Yamamoto, and C. Zamudio, "Genome-wide operon prediction in Staphylococcus aureus," *Nucleic Acids Res,* vol. 32, pp. 3689-702, 2004.

[13] B. P. Westover, J. D. Buhler, J. L. Sonnenburg, and J. I. Gordon, "Operon prediction without a training set," *Bioinformatics,* vol. 21, pp. 880-8, Apr 1 2005.

[14] 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 Res,* vol. 33, pp. 3253-62, 2005.

[15] H. Salgado, S. Gama-Castro, A. Martinez-Antonio, E. Diaz-Peredo, F. Sanchez-Solano, M. Peralta-Gil, D. Garcia-Alonso, V. Jimenez-Jacinto, A. Santos-Zavaleta, C. Bonavides-Martinez, and J. Collado-Vides, "RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in Escherichia coli K-12," *Nucleic Acids Res,* vol. 32, pp. D303-6, Jan 1 2004.

[16] S. Okuda, T. Katayama, S. Kawashima, S. Goto, and M. Kanehisa, "ODB: a database of operons accumulating known operons across multiple genomes," *Nucleic Acids Res,* vol. 34, pp. D358-62, Jan 1 2006.

[17] T. Dandekar, B. Snel, M. Huynen, and P. Bork, "Conservation of gene order: a fingerprint of proteins that physically interact," *Trends Biochem Sci,* vol. 23, pp. 324-8, Sep 1998.

[18] R. L. Tatusov, E. V. Koonin, and D. J. Lipman, "A genomic perspective on protein families," *Science,* vol. 278, pp. 631-7, Oct 24 1997.

[19] S. Salzberg, "Locating protein coding regions in human DNA using a decision tree algorithm," *J Comput Biol,* vol. 2, pp. 473-85, Fall 1995.

[20] J. Selbig, T. Mevissen, and T. Lengauer, "Decision tree-based formation of consensus protein secondary structure prediction," *Bioinformatics,* vol. 15, pp. 1039-46, Dec 1999.

[21] S. L. Wong, L. V. Zhang, A. H. Tong, Z. Li, D. S. Goldberg, O. D. King, G. Lesage, M. Vidal, B. Andrews, H. Bussey, C. Boone, and F. P. Roth, "Combining biological networks to predict genetic interactions," *Proc Natl Acad Sci U S A,* vol. 101, pp. 15682-7, Nov 2 2004.

[22] J. R. Quinlan, *C4.5 Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[23] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. San francisco: Morgan Kaufmann, 2005.

[24] S. C. Janga, W. F. Lamboy, A. M. Huerta, and G. Moreno-Hagelsieb, "The distinctive signatures of promoter regions and operon junctions across prokaryotes," *Nucleic Acids Res,* Aug 12 2006.