Operon prediction in *Pyrococcus furiosus*

Thao T. Tran, Phuongan Dam¹, Zhengchang Su¹, Farris L. Poole II¹, Michael W. W. Adams¹, G. Tong Zhou and Ying Xu¹,*

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA and ¹Department of Biochemistry and Molecular Biology, Institute of Bioinformatics, University of Georgia, Athens, GA 30602, USA

Received July 18, 2006; Revised October 25, 2006; Accepted October 26, 2006

**ABSTRACT**

Identification of operons in the hyperthermophilic archaeon *Pyrococcus furiosus* represents an important step to understanding the regulatory mechanisms that enable the organism to adapt and thrive in extreme environments. We have predicted operons in *P.furiosus* by combining the results from three existing algorithms using a neural network (NN). These algorithms use intergenic distances, phylogenetic profiles, functional categories and gene-order conservation in their operon prediction. Our method takes as inputs the confidence scores of the three programs, and outputs a prediction of whether adjacent genes on the same strand belong to the same operon. In addition, we have applied Gene Ontology (GO) and KEGG pathway information to improve the accuracy of our algorithm. The parameters of this NN predictor are trained on a subset of all experimentally verified operon gene pairs of *Bacillus subtilis*. It subsequently achieved 86.5% prediction accuracy when applied to a subset of gene pairs for *Escherichia coli*, which is substantially better than any of the three prediction programs. Using this new algorithm, we predicted 470 operons in the *P.furiosus* genome. Of these, 349 were validated using DNA microarray data.

**INTRODUCTION**

*Pyrococcus furiosus* is a hyperthermophilic anaerobic archaeon that grows optimally near 100°C using carbohydrates and peptides as carbon and energy sources (1). This organism is commonly found in hydrothermal vents on the seafloor near volcanos. Its ability to grow to high cell densities under laboratory conditions without the need of elemental sulfur, and thus production of toxic hydrogen sulfide, has made it a useful model organism with which to study thermostable enzymes and adaptations to high-temperature environments (2). The genome sequence of *P.furiosus* has been determined (3,4) and is available at http://www.ncbi.nlm.nih.gov/entrez/viewer.fcgi?db=nucleotide&val=18976372. The latest annotation contains 2125 genes, which is used for the study herein. Similar to most genome annotations that of *P.furiosus* provides a unique source of information for molecular and biochemical studies, but it is mostly gene-centric and does not provide much structural and functional information about higher-level organizations. In this paper, we present our recent work on the identification of operons in *P.furiosus*.

An operon is defined as a basic transcriptional unit in prokaryotes. Characterization of operons represents an important step in understanding many cellular processes and deciphering transcriptional regulatory networks. Insights into the function and regulation of genes in the context of pathways and networks can be gained if we can annotate operons accurately. Due to the arduous nature of experimentally determining operons on an individual basis, several computational approaches have been proposed for predicting them (5–10). Generally, these approaches use information derived from comparative genomics, transcriptional signals upstream and downstream of operons, features such as intergenic distances, functional annotation of genes and experimentally derived DNA microarray data.

We have recently developed a novel method for operon prediction by integrating three existing operon prediction methods and have applied it to the bacteria *Escherichia coli* and *Bacillus subtilis* and *P.furiosus*. The three methods were chosen because they are considered as better prediction methods among all publicly available operon prediction programs. All three prediction programs assume that genes within an operon are in the same direction, which is defined as consecutive open reading frames (ORFs) transcribed in the same direction with no intervening ORF on the opposite strand (6,8,10). The first program is the JPOP (Joint Prediction of Operons) program (10), which classifies each pair of consecutive genes as an ‘operonic’ or an ‘non-operonic’ boundary based on their intergenic distance, similarity between their phylogenetic profiles, and relatedness of their annotated functions from COGs (11). Each of these sets of supporting data is integrated using a neural network (NN) to generate operon
predictions. The second program, Operon Finding Software (OFS) (8), combines conserved gene-order information across multiple genomes with intergenic distance and similarity information of annotated gene functions to make operon predictions. This work generalized the gene-order conservation approach used in (7) by relaxing the adjacency and orthology criteria. The authors of OFS (8) claimed to be able to predict operons without extensive training. The third program, developed by the Virtual Institute for Microbial Stress and Survival (VIMSS) (6), is similar to the first two programs because it uses intergenic distance and COG information. VIMSS differs, however, in also employing the codon adaptation index (CAI) and applying a different approach for comparative genome analysis to make operon predictions. The comparative genome analysis examines how often orthologous genes are close to each other within 5 kb across multiple genomes, while the CAI measures synonymous codon usage.

Another operon prediction method developed by The Institute for Genomic Research (TIGR) (7) was considered but not integrated into our method due to the high number of missing confidence values between adjacent gene pairs in the same direction. A summary of the default operon prediction results for the three programs for *E.coli* and *B.subtilis* are shown in Supplementary Figure S1. The Venn diagram displays the number of gene pairs predicted to be within operons by each program and the overlap in gene pairs predicted among the three programs. Out of a total of 2985 adjacent gene pairs in the same direction in *E.coli*, 1885 gene pairs are predicted to be in operons by at least one program and only 55% (=1037/1885) of gene pairs are predicted to be in operons by all three programs. Likewise in *B.subtilis* with a total of 3005 gene pairs, 2122 gene pairs are predicted to be in operons, but only 28% (=599/2122) of gene pairs are predicted by all three programs. With low consensus among individual operon prediction programs, there is a need to incorporate the additional information provided by each program into a general operon predictor.

Our initial prediction stems from training a NN-based classifier (to classify a pair of adjacent genes as either operonic boundary or not), based on the outputs of the three aforementioned programs. Furthermore, we use additional computational data from (i) Gene Ontology (GO) information, (ii) known pathway information and (iii) log-likelihood intergenic distance to improve the operon prediction accuracy. The GO classification is used to compute a functional similarity score between pairs of adjacent genes in the same direction. Additionally, we have computed KEGG pathway scores based on whether or not gene pairs belong to common KEGG pathways. The intergenic distance feature as used in previous studies is also inputted directly into the NN to aid prediction since it has been found to be a strong discriminatory feature (10,12,13). Three-fold cross-validation and train/test set validation was analyzed for *E.coli* and *B.subtilis* to examine the performance of these features within and across species, respectively. Using the optimal training set, our method is applied to make operon predictions in *P.furiosus*. We have used experimental data obtained from time-course microarray gene expression data to verify these operon predictions. The idea is that genes in the same operon should in general exhibit similar expression patterns under any experimental conditions. All predicted operons are available at http://csbl.bmb.uga.edu/~tran/operators along with the prediction program.

**MATERIALS AND METHODS**

We present our method by first introducing how the positive and negative sets for training and testing were generated. Then, we discuss the various features used to train the operon predictor: confidence scores from JPOP, OFS and VIMSS, GO similarity scores, KEGG pathway scores and intergenic distances. Next, we explore the design of our NN-based predictor. Finally, we examine how we use the available microarray data to validate our predictions in *P.furiosus*.

**Generation of the positive and negative datasets**

Since no genome-wide operons in *P.furiosus* have been experimentally determined, we have benchmarked our program using operons from *E.coli* and *B.subtilis*, which have been experimentally validated. The true positive (TP) set in *E.coli* are the transcriptional unit gene pairs extracted from the RegulonDB database (14). The generation of the negative set represents a challenge in this study since current operon prediction programs typically only output the confidence score of adjacent gene pairs in the same direction. Defining a negative set using gene pairs from opposite strands as used in (10) is not applicable to our approach because the confidence scores are not defined. These gene pairs are considered trivial by current prediction methods since opposite strand information alone is enough to classify them. Furthermore, using an intergenic distance cutoff to generate a negative set may impose certain biases and prevent the identification of operonic gene pairs that are located far away. We generate our negative dataset as follows: two adjacent genes in the same direction are considered as a true negative (TN) gene pair if they are not transcriptionally co-expressed, i.e. not in the same transcriptional unit. These TN gene pairs also include adjacent genes not in a transcriptional unit with one that is present in a transcription unit. This approach has been used by other authors (7,12,15–17). Operons of *B.subtilis* obtained from (18) were extracted similarly. In addition, we consider only gene pairs with confidence measures from all three prediction programs when generating the TP and TN sets. The number of TP gene pairs in *E.coli* and *B.subtilis* are 711 and 628, respectively. The number of TN gene pairs in *E.coli* and *B.subtilis* are 374 and 556, respectively.

In addition to the above two sets, we also consider the whole operonic gene pair set (TP set plus those without confidence scores defined by all three prediction programs) and a non-operon set defined as pairs of adjacent genes, with one gene on one strand and the other gene on the opposite strand. We refer to these sets as operon and non-operon gene pair sets, not to be confused with the TP and TN sets as defined earlier. The operon set contains 821 gene pairs in *E.coli* and 806 gene pairs in *B.subtilis*, while the non-operon set contains 1256 gene pairs in *E.coli* and 1099 gene pairs in *B.subtilis*. By having the non-operon set, we know for sure that gene pairs from the opposite strands do not belong to the same operon. This allows us to detect any biases induced by our definition of the TN set when we examine the different features.
Operon prediction by existing software

Executable codes for JPOP were obtained from http://csbl.bmb.uga.edu/downloads/ (10). The software for OFS was downloaded from http://www.cse.wustl.edu/~jbuhler/research/operons. We have applied OFS using the default values for all parameters and $\beta = 0.35$ as used in (8). VIMSS had precompiled operon predictions for some organisms, available at http://www.microbesonline.org/operons (6). Perl scripts were written to extract the confidence values for each gene pair in the TP and TN set. For all organisms studied in this paper, the focus is on operons with two or more genes.

Gene Ontology (GO) similarity analysis

One additional source of evidence used to improve our NN predictor is the GO classification (19), which encompasses three levels of biological functions: biological process, molecular function and cellular component. It is known that genes in the same operon are involved in the same or similar biological processes; hence, GO ontology information should in principle be helpful for operon prediction. A similarity score is used as defined in (20), which examines the GO-based functional assignments for each gene pair and uses an acyclic graph (21) in the form of a tree structure to compute the depth of the common terms. The higher the score, the more similar the two genes are since they share more common GO terms.

The distribution of GO similarity scores for operon and non-operon gene pairs of *E.coli* and *B.subtilis* are shown in Supplementary Figure S2. We also include the distribution for the (TP,TN) set in order to compare with the {operon, non-operon} set. As shown in Supplementary Figure S2, the non-operon and TN gene pairs tend to have lower GO similarity scores compared with operon and TP gene pairs. This capability to discriminate between operon and non-operon gene pairs makes this feature useful in our NN predictor. The coverage of GO similarity scores with annotation in *E.coli*, *B.subtilis* and *P.furiosus* is 50%, 45% and 29%, respectively.

Pathway assignment

Genes within the same operon generally encode proteins that function in the same metabolic pathway or biological process. As such, we have generated scores based on the pathway information collected from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database (22). The KEGG (prokaryotic) pathways are organized into four general categories (level 1): Metabolism, Genetic Information Processing, Environmental Information Processing and Cellular Processes. These categories are further subdivided into level 2 and level 3 pathways. The authors of KEGG have used KEGG Orthology (KO) terms to describe each pathway. Given a genome with annotated genes, one can assign KO terms to each gene and determine directly the pathway in which it is involved. The KO terms can be obtained for each organism using the following methods:

(i) KO flatfile contains manually curated data linking KO labels to genes of various organisms. This flatfile was obtained from ftp://ftp.genome.jp/pub/kegg/tarfiles/ko and processed to obtain all genes in the organism with KO annotation.

(ii) KO-Based Annotation System (KOBAS) software (23) was used to annotate the KO terms based on sequence-similarity searches. The program uses BLAST to assign KO labels to genes with an $E$-value $<10^{-5}$ along with some additional information.

(iii) KEGG Automatic Annotation Server (KAAS) was also used to annotate the genes. This annotation uses a bi-directional best hit of BLAST with a threshold BLAST bit score $>60$.

The results of these three programs were combined using the following simple heuristic rule: when the annotation results are different among the three different methods, we use the result by the method with the highest priority score. The KO flatfile annotation is given the highest priority score since the results are manually curated. KOBAS is given the second highest priority since its annotation agrees with the KO flatfile more often than KAAS. Finally, if the KO flatfile and KOBAS cannot assign a gene to a pathway, the annotation result from KAAS is used. By using multiple sources to assign the KO terms, we have increased the coverage of ORFs with KO annotation. In *E.coli*, the number of ORFs in the KO flatfile represents 55% of all protein-coding ORFs. By adding similarity search annotation methods such as KOBAS and KAAS, the coverage improves to 61%. Likewise for *B.subtilis*, the coverage improves from 46 to 54%. For *P.furiosus*, where the number of manually curated ORFs in the KO flatfile represents only 38% of all protein-coding ORFs, the addition of similarity search methods improves the coverage up to 46%.

Once the ORFs are assigned KO annotation, the KEGG pathways can be inferred directly. A KEGG pathway score of 1, 2 or 3, was assigned to a gene pair if they share the same level 1, level 2 or level 3 pathway, respectively. The higher the score, the higher the chance the two gene products are in the same pathway, and hence it is more likely that the two genes belong to the same operon. A score of 1 was assigned to a gene pair if none of them has pathway annotation while a score of 0 was assigned if only one gene has pathway annotation. The score distribution for the {operon, non-operon} and {TP, TN} datasets for *E.coli* and *B.subtilis* are shown in Supplementary Figure S3. Operon gene pairs typically have the same level 3 pathway.

Intergenic distance-based log-likelihood score

Another input to our NN-based predictor is an intergenic distance-based log-likelihood score defined by Equation 1. This score for a gene pair is computed as the log ratio of the probability that their distance belongs to the distance distribution of the TP set and the probability this distance belongs to the distance distribution of the TN set (13). Although this feature is already included in JPOP, OFS and VIMSS, adding it to the input of the NN predictor is shown to improve the prediction in some cases. For our current study, the intergenic distances were computed based on the training sets and applied to score the distances on the test set. This is necessary since the TP and TN distributions are not known. The histogram of intergenic distances for the different sets is shown in Supplementary Figure S4.
There is a clear peak separation between the \{TP,TN\} set in \textit{B.subtilis} whereas in \textit{E.coli}, the TN set is less pronounced and has approximately a uniform distribution. This suggests that the performance in using the intergenic feature will vary between \textit{B.subtilis} and \textit{E.coli} based on our choice of the TP and TN set. In the general case; however, the intergenic feature is beneficial in discriminating between the operon and non-operon gene pairs.

\[
LL(d(g_a, g_b)) = \ln \frac{P(d(g_a, g_b) | \text{TP genepair})}{P(d(g_a, g_b) | \text{TN genepair})}.
\]

**Neural network-based operon predictor**

An artificial NN was implemented to integrate the outputs from three operon prediction programs in order to attain a more robust and efficient tool for operon prediction. Intuitively, by utilizing and consolidating the strengths of these programs into a single operon prediction tool, improved results can be realized. The question is how to best combine the programs to produce the best possible prediction results. NN is a proven technique for combining multiple sources of information, without assuming the underlying relationships among the individual data sources. This technique is robust for noisy data and has been widely used for many biological data analysis problems (24,25). The design of our NN-based predictor is achieved through three main steps: (i) data preprocessing (feature extraction and normalization), (ii) selection of appropriate network architectures (e.g. number of layers, number of neurons) and (iii) training and testing. Iterations of (ii) and (iii) are needed to identify the best network architecture based on the prediction performance.

**Score normalization.** Normalization of the confidence scores of the three prediction programs is required to ensure that the dynamic range of the individual programs does not influence the performance of the NN. For each program, the prediction confidence measure for each gene pair was extracted and normalized to between 0 and 1, where a value >0.5 indicates that the corresponding gene pair belongs to the same operon. A linear mapping was performed to re-center the JPOP confidence scores between [0 1]. This scaling was used to keep it consistent with the probability-based confidence measures from VIMSS and OFS. The GO similarity score, KEGG pathway scores and log-likelihood score of intergenic distance were also linearly normalized into the range [0 1].

**Neural network training.** The idea behind a NN is to train a set of parameters to give a desired output target \(t\), for a given input data \(x\). A trained network can then be applied to new data \(x'\) to predict the outcome \(y'\). In our approach, we present the confidence measures from each of the three prediction programs, \(x_i = [x_{i1}, x_{i2}, x_{i3}]\) for \(i = 1, 2, 3\) to a feed-forward network architecture (see Figure 1). Various combinations of GO similarity, KEGG pathway and intergenic distance scores were also tested as additional inputs into the NN-based predictor. The desired output target is 0/1 \(\{1 = \text{‘gene pair in an operon’}, 0 = \text{‘gene pair not in an operon’}\}\). The training algorithm optimizes the weights \(W=[w_1^T]\) and a bias, \(b\), of the network during the training (supervised learning) phase to minimize the error between the network output, \(a\), and the desired output, \(t\), on the training data. Our NN was trained using MATLAB\textsuperscript{®}'s NN toolbox. The network parameters are optimized using the Levenberg–Marquardt algorithm. Other network training functions were tested but either the results are not as good or the differences are insignificant compared with our selected network (data not shown). The network architecture parameters are (i) the transfer function \(f\), (ii) the number of neurons and (iii) the number of layers. Various network architectures were tried and tested on the \textit{E.coli} and \textit{B.subtilis} datasets. Based on the performance on \textit{B.subtilis} data, an optimal network architecture is selected and applied to predict operons in \textit{E.coli} and \textit{P.furiosus}.

Plots of the receiver operating curve (ROC) (26) were generated to compare our operon predictor with the three programs. The ROC plots the sensitivity versus \(\left(1 - \text{specificity}\right)\) over a range of program thresholds. For each classifier, the area under its receiver operating curve (AUROC) can be computed to give a qualitative measure of the performance not dependent on a specific threshold. Generally, the higher the AUROC, the closer the classifier’s ROC is to the ‘optimal’ performance, i.e. higher sensitivity and higher specificity. We have also calculated the sensitivity, specificity and accuracy values of our predictions, as defined by Equations 2, 3 and 4, respectively, where \(TP\) is the true positive, \(TN\) is the true negative, \(FP\) is the false positive and \(FN\) is the false negative. The overall prediction accuracy takes into consideration both the number of TP and TN correctly predicted to provide for a good comparison of the performance among the individual programs.

**Microarray data analysis**

Microarray gene expression data can be used to verify the operon prediction in \textit{P.furiosus}. Genome-wide cDNA microarray data representing the original 2065 genes in the \textit{P.furiosus} genome have been published in response to changes in carbon source (27) and cold shock (28). Available microarray data in \textit{P.furiosus} are analyzed by two methods. The first one applies a correlation approach to cluster time series microarray data. It is expected that genes within an
operon exhibit similar trends in their expression profiles. The second method analyzes all published microarray datasets to identify significantly expressed gene pairs. This approach has been applied in (27) and (28) to identify groups of putative operons. We have re-analyzed the raw data here in order to standardize the pre-processing procedure among the datasets.

**Kinetic cold shock-microarray data and application.** In this method, the expression trends of genes within an operon over time are analyzed by using available time series (kinetic) cold shock microarray data. In the kinetic cold shock experiment (28), the organism was grown at 95°C and then subjected to cold shock at 72°C starting at time \( t = 0 \). The intensity values of the cDNA microarray were monitored at time \( t = 1, 2 \) and 5 h. The kinetic cold shock experiment consisted of two replicates done on different dates. Each replicate consists of two duplicate slides with each having three copies of the ORF spotted on the cDNA array. The copy, duplicate and replicate terminology is illustrated in Supplementary Figure S5. The raw data were preprocessed as follows. Any signal or reference data point with an intensity value \(<2000\) was set to \( 2000 \) as values under this cutoff are considered too low to be significant. Reference intensities are the initial condition immediately before cold shock at \( t = 0 \). The \( \log_2(\text{signal/reference}) \) ratio is then computed where a positive value indicates upregulation, zero indicates no regulation, and a negative value indicates downregulation. An averaging step then averages the \( \log_2 \) ratios of duplicates with copies that are all non-zero. Non-zero duplicates within the same replicate were averaged and then combined with non-zero replicates to generate a single average \( \log_2 \) ratio to represent the expression of each ORF for time \( t = 1, 2 \) and 5 h, respectively.

The Pearson correlation coefficient \( r \) as defined in Equation 5, where \( x = [x_1, x_2, x_3] \) and \( y = [y_1, y_2, y_3] \) represent the time series profiles for adjacent ORF \( x \) and ORF \( y \) at times \( t_0 = 1 \) h, \( t_1 = 2 \) h and \( t_2 = 5 \) h. Adjacent ORFs of the same direction with a Pearson correlation coefficient \( >0.5 \) and intergenic distance \( \leq 75 \) bp are predicted to be in the same operon. The intergenic distance cutoff was determined from examining a list of 33 known/putative operons published in literature (27–30) where the longest intergenic distance between an operonic gene pair was 74 bp. Using the correlation coefficient and intergenic distance cutoff, we generate a ‘microarray evidence list’ consisting of 357 operons to validate the operon predictions of \( P.furiosus \).

**Generation of putative operon list based on microarray data.** To date, the only studied operons in \( P.furiosus \) have been the lamA (29) and POR/VOR (30) operons, both involved in energy metabolism. Several putative operons have been suggested in (27) and (28) based on peptide + sulfur versus maltose + sulfur and batch/kinetic cold shock data, respectively. The raw data values for these microarray experiments were obtained from the original authors and the pre-processing was standardized among all of these datasets. A one-sample, two-sided unpaired \( t \)-test was performed on the \( \log_2 \) ratios to identify differentially expressed ORFs and the \( P \)-value was determined by applying the Holm’s step-down correction similar to the original papers. Unlike the original paper, the average fold change was computed using only non-zero \( \log_2 \) ratios. This results in less bias of the fold change towards lower values due to the inclusion of noisy data points. Adjacent ORFs with an average fold change \( \geq 2 \), a Holm’s adjusted \( P \)-value \( \leq 0.01 \), and an intergenic distance \( \leq 30 \) bp were predicted as putative operon gene pairs. We refer to this list of operons as the ‘putative operon list’.

An intergenic distance cutoff of 30 bp is used, as it is small enough so as not to contain any internal promoters such as TATA-boxes known to be upstream of the transcriptional start site.

**RESULTS AND DISCUSSION**

Our approach seeks to produce operon predictions with a higher accuracy (overall correct predictions) compared with any of the three individual programs. In order to find the optimal set of input features, we first run 3-fold cross-validation on \( E.coli \) and \( B.subtilis \). Using these features, we evaluate the performance across species by training on one organism and applying it to another. After predicting the operons for \( P.furiosus \), we apply the microarray results to identify a putative list of operons to do further experimental study. We describe the results of each procedure in the following sections.

**Cross-validation on \( E.coli \) and \( B.subtilis \)**

A 3-fold cross-validation was performed on \( E.coli \) and \( B.subtilis \) data using various combinations of inputs into the NN. From the ROC curves, the AUROC was computed to compare the performance of our NN-based predictor with the three existing programs as summarized in Table 1. For comparison and reproducibility, we fix the network to be a single neuron one-layer network with a transfer function \( f = \logsig \). This network will be used unless otherwise specified. By integrating the confidence scores of the three existing programs, we are able to achieve prediction results better than any single program. Even better performance was achieved by incorporating GO, pathway and the intergenic distance information. For \( E.coli \), the most useful features (in order of most to least helpful) are pathway, GO and intergenic distance. On the other hand, for \( B.subtilis \), the intergenic distance was most beneficial, while the pathway and GO information achieved comparable performance. As expected, using the two most helpful features for both organisms produced slightly better results. The two sets of features that performed best can be observed in Supplementary Figures S6 and S7. At virtually all threshold levels, the NN-based predictor achieves both higher sensitivity and higher specificity than any of the three programs. For \( E.coli \), adding the intergenic distance actually decreased the performance compared to using inputs from the three programs with GO and pathway information. In this case, the intergenic distance adds more noise rather than helping. This can be explained by examining the histogram of the intergenic distance for \( E.coli \) compared with \( B.subtilis \) on the \{TP,TN\} set as shown in Supplementary Figure S4. Although the best performance for \( E.coli \) was achieved using additional GO and pathway
information, *B. subtilis* benefitted from using all three additional features. This is analyzed in further detail in the next section. As a side note, increasing to a two-layer NN with more neurons only improves the AUROC by <0.01 (data not shown).

**Validation on *E. coli* using *B. subtilis* training set**

Based on the optimal set of input features determined in the previous section, we train our NN-based predictor using the entire *B. subtilis* dataset and test on *E. coli*. The results of comparing the performance of the NN-based predictor (three programs + GO + pathway) with and without the intergenic distance are shown in Table 2. The overall accuracy on the test set of the NN-based method is higher than any of the three existing programs. Since the test accuracy of the NN-based method is comparable with or without the use of intergenic distance, we decide to use the NN-based predictor with the six inputs including the intergenic distance because of the observed higher training accuracy. Using this architecture, the NN was further optimized to a two-layer (two hidden neurons, one output neuron) to improve the overall accuracy of the test set from 0.8544 to 0.8645 as shown in Table 2. The sensitivity of our NN-based predictor is comparable to OFS and JPOP; however, there is >6–8% improvement in the specificity, respectively. For VIMSS with the highest specificity among the three existing programs, the NN-based predictor was able to improve the specificity slightly and improve the sensitivity by almost 4%. In summary, the NN-based predictor was able to achieve higher sensitivity, specificity and accuracy compared with the three existing programs. The graphical representation and parameters of this optimal network are shown in Supplementary Figure S8 and Supplementary Table S1. The corresponding ROC curves for the training and the testing sets are shown in Figure 2. The AUROC for our predictor on the *E. coli* test set is higher than any of the three other programs examined. We have also tested the scheme by reversing the training and testing data; however, the improvement to overall accuracy was marginal possibly because features in *B. subtilis* are more generalizable to other organisms. As a side note to Figure 2A, using only the intergenic distance in the *B. subtilis* training set seems to outperform the existing programs. As discussed in the previous section, this is not true in the case of *E. coli* so therefore additional features are still needed to aid prediction. Various factors have contributed to differences in performance among the programs, including (i) how other programs define their TP/ TN sets, (ii) gene annotation data available at the time of prediction, (iii) the prior distribution used in generating the intergenic distance scores and (iv) the contribution of the intergenic distance to each program’s prediction. It is worth discussing why VIMSS performs poorly on the *B. subtilis* dataset. We have investigated this by examining the histogram of the confidence scores for each program in Supplementary Figures S9–S11. The distribution of the VIMSS scores for *B. subtilis* seems to have poor discriminatory power between the TP and TN set. Of the reasons mentioned earlier, this is most likely due to the differences in the *B. subtilis* datasets used. Our training dataset consisted of a list of 340 known operons published recently for *B. subtilis* versus the dataset of 100 known operons as used in VIMSS. This and previously mentioned reasons could be investigated in later studies to help explain the poor performance of VIMSS on *B. subtilis* but good performance on *E. coli*.

**Table 1. Three-fold cross-validation results for *E. coli* and *B. subtilis***

| No. of inputs | 3-Fold cross-validation results | *Escherichia coli* | *Bacillus subtilis* |
|---------------|---------------------------------|--------------------|--------------------|
|               | AUROC                           | Escherichia        | Bacillus           |
|               |                                 | coli               | subtilis           |
| ---           |                                 | 0.8967             | 0.8568             |
| ---           | JPOP                             | 0.9105             | 0.8381             |
| ---           | OFS                              | 0.9044             | 0.6207             |
| 3             | 3 only                           | 0.9225             | 0.8787             |
| 4             | 3 + GO                           | 0.9262             | 0.8797             |
| 4             | 3 + pathway                      | 0.9279             | 0.8798             |
| 4             | 3 + intergenic                   | 0.9170             | 0.8926             |
| 5             | 3 + GO + pathway                 | **0.9284**         | 0.8802             |
| 5             | 3 + GO + intergenic              | 0.9218             | 0.8955             |
| 5             | 3 + pathway + intergenic         | 0.9267             | 0.8948             |
| 6             | 3 + GO + pathway + intergenic    | **0.9275**         | **0.8963**         |

The area under the receiver operating curve (AUROC) is given for the three existing programs (JPOP, OFS, VIMSS) and different sets of inputs into the NN. The number of inputs along with the different combinations of inputs is given from 3-fold cross-validation for each organism. The ‘3 only’ represents the use of only the confidence scores of the three existing programs. The ‘3 + GO’ represents the use of the three existing programs and GO similarity score for a total of 4 inputs into the NN. Combinations using the pathway score and the intergenic distance scores are also given similarly. The NN was fixed to be a simple one-layer 1-neuron NN with transfer function $f = \log_{10}$. The majority of the improvement is realized by just combining the confidence scores from the three programs (3 only); however, there is further improvement by including other features such as GO similarity score, KEGG pathway score and intergenic distance. The highest AUROC for each organism is in boldface.

**Table 2. Results of testing on *E. coli* after fixing network parameters and threshold from *B. subtilis***

| Program              | Fixed threshold | TRAIN (*B. subtilis*) | TEST (*E. coli*) |
|----------------------|-----------------|-----------------------|------------------|
|                      |                 | Sn | Sp   | Accuracy | Sn | Sp   | Accuracy |
| JPOP                 | 0.3427          | 0.7962 | 0.8147 | 0.8049 | 0.8819 | 0.7433 | 0.8341 |
| OFS                  | 0.7494          | 0.8025 | 0.7788 | 0.7914 | 0.8819 | 0.7647 | 0.8415 |
| VIMSS                | 0.674           | 0.5892 | 0.6187 | 0.6030 | 0.8453 | 0.8182 | 0.8359 |
| NN (3 + GO + pathway) [1] | 0.4164          | 0.8519 | 0.7788 | 0.8176 | 0.8847 | 0.7647 | 0.8341 |
| NN (3 + GO + pathway + intergenic) [1] | 0.4756          | 0.8662 | 0.8219 | 0.8454 | 0.8903 | 0.7861 | 0.8544 |
| NN (3 + GO + pathway + intergenic) [2,1] | 0.5876          | 0.8328 | 0.8651 | 0.8480 | 0.8847 | 0.8262 | 0.8645 |

The table presents the existing programs and various combinations of inputs into the NN predictor. The number in brackets [ ] following each NN predictor indicates the number of neurons used in each layer. For example, [1] represents a single-layer neuron with one neuron and [2,1] represents a two-layer neuron network with two neurons in the hidden layer and one neuron in the output layer. For each program the following are given: the fixed threshold from *B. subtilis* training, sensitivity (Sn), specificity (Sp) and accuracy. In the *E. coli* testing set, there is improvement in overall accuracy, sensitivity and specificity of the NN-based method over the existing three programs.
Validation on P.furiosus prediction using B.subtilis training set

The parameters trained on B.subtilis were then applied to two datasets from P.furiosus: a ‘known operon’ list consisting of 33 known/putative operons collected from the literature, and the ‘microarray evidence list’ as described in Materials and Methods. The testing results on these two sets are shown in Table 3. It is interesting to note that the prediction sensitivity is much higher at the expense of specificity. This is probably an intrinsic limitation of applying trained parameters from one species to another (more distant) species. The optimal performance for the ‘known operon’ list can be found in Supplementary Figure S12 and Supplementary Table S2. From Supplementary Table S2, using the optimal P.furiosus threshold, the NN-based prediction has highest accuracy compared with any of the three methods. Depending on the tradeoff between the sensitivity and the specificity, a user can best decide which threshold is best for their specific application. Whichever way the threshold is chosen in P.furiosus (whether the default training threshold from

Figure 2. (A) ROC for the B.subtilis training set. (B) ROC for E.coli using trained parameters from B.subtilis. Each plot displays the ROC from the three existing programs [JPOP, OFS, VIMSS], the performance of using only the GO similarity score [GO], the performance of using only the pathway score [pathway], the performance of using only the log-likelihood score of the intergenic distance [intergenic], and the performance of the NN-based predictor incorporating all of the aforementioned (6) features [NN]. The numbers in the legend correspond to the points indicated by an asterisk (*) in the plot showing each program’s threshold that maximizes the (Sensitivity + Specificity) value. For any threshold, the NN-based method has higher performance than any of the existing programs.
Table 3. Results of testing on *P. furiosus* using fixed network parameters and threshold from *B. subtilis*

| Program | Fixed threshold | Known operons | Accuracy | Microarray evidence list | Accuracy |
|---------|-----------------|---------------|----------|--------------------------|----------|
| JPOP    | 0.3427          | 0.8972        | 0.6129   | 0.8333                   | 0.8198   |
| OFS     | 0.7494          | 0.8505        | 0.7419   | 0.8261                   | 0.5545   |
| VIMSS   | 0.674           | 0.8972        | 0.7097   | 0.8551                   | 0.7249   |
| NN      | 0.5876          | 0.9907        | 0.5806   | 0.8986                   | 0.9022   |

For each organism, the number of open reading frames (ORFs) included in the operon prediction, the number of operons, the average operon size and the percent of gene coverage (=100 \times \text{no. of ORFs included in the operon prediction/total no. of ORFs in the organism}) are given.

*B. subtilis* or the optimal threshold), the prediction sensitivity is consistently higher at the expense of lower specificity. A similar trend was observed with the 'microarray evidence list' over a range of threshold values as shown in Supplementary Figure S13. Although the NN-based method achieves the primary goal of our study in having higher overall accuracy compared with other programs, the results tend to overpredict operons and give a higher number of false positives. Between sensitivity and specificity; however, we would prefer a higher sensitivity so as not to miss many operons. Since the objective of our computational prediction is to provide a list of potential operons in *P. furiosus* so that further experimental validation can be performed, this prediction is acceptable in our case since we have microarray data to further filter out the false predictions. It should be noted that these two test sets in *P. furiosus* are by no means complete and this initial attempt to assess the performance of the NN-based method on a new organism is a rough indicator for what one can expect based on these limited data.

Whole-genome operon prediction

The parameters trained on the *B. subtilis* set were applied to the entire genomes of *E. coli*, *B. subtilis* and *P. furiosus*. A summary of the predicted operons for *E. coli*, *B. subtilis* and *P. furiosus* is shown in Table 4. The NN-based method predicted 470 operons covering 1460 ORFs in *P. furiosus*. An average operon consists of 3.1 ORFs. A summary of the number of operons predicted for each organism along with other statistics for the different programs is given in Supplementary Table S3.

Functional annotation of *P. furiosus* operons

The predicted operons that overlap the ‘microarray evidence list’ are annotated and can be found at http://csbl.bmb.uga.edu/~tran/operons. The annotation for *P. furiosus* was obtained from GenBank and the TIGR-Comprehensive Microbial Resource (31). In addition, a subset of this list that overlaps the ‘putative operon list’ can be found in Supplementary Table S4. The number of overlapping gene pairs from these files are summarized in the Venn diagram as shown in Figure 3. The 646 (=489+157) gene pairs common to our predicted operons and the ‘microarray evidence list’ represent 349 unique operons. The 157 gene pairs that overlap all three lists form 98 operons. The novel operons in this set provide biologists a list of targets for further experimental studies.

The 71 (=64+6+1) gene pairs in Figure 3, which were not predicted by the NN-based method may be due to a combination of microarray experimental errors, the methodology used in microarray analysis, or prediction errors. The ‘putative operon list’ has higher specificity than the ‘microarray evidence list’ in terms of applying a lower intergenic distance cutoff and more microarray conditions, which can help to reduce the number of false predictions. As a result, the majority of the gene pairs from the ‘putative operon list’ overlap with the NN-based prediction. The gene pairs in the microarray evidence list not predicted by the NN-based method represent a small fraction of total gene pairs in the list (~10%). These gene pairs may be errors in predictions or could be due to noisy microarray data. The 331 gene pairs predicted by the NN-based method but not present in either of the microarray lists may be due to overprediction or because the condition
for co-expression may not have been tested in the microarray data available.

Our NN-based method was able to detect both of the two operons in \textit{P. furiosus} that are previously known, namely the lamA and POR/VOR operons. In the case of the POR/VOR operon, which consists of three mRNA transcripts: [PF0965 PF0966 PF0967] [PF0968 PF0969 PF0970] [PF0971], our method predicted these as one large operon from PF0965 to PF0971. This is attributed to the fact that all three existing programs predict the POR/VOR operon as one single transcript. This is an intrinsic limitation in combining existing methods though we expect that such cases are rare.

**SUMMARY**

Operon prediction allows for the functional inference of hypothetical and conserved hypothetical genes, and represents a key step in reconstructing biological pathways and networks for prokaryotes. A novel NN-based approach for operon prediction is described herein that integrates the strengths of existing prediction algorithms, which use various sequence features such as codon usage and intergenic distance, conserved gene-order, phylogenetic profiles of genes, and COG functional annotation. By integrating the prediction results of the three programs, we are able to achieve better performance by taking advantage of the complementary information provided by each individual program. By using GO annotation, KEGG pathway and intergenic distance information as additional inputs into our program, we have further improved upon the accuracy. The improvement in performance by our new algorithm is demonstrated through cross-validation on \textit{E.coli} and \textit{B.subtilis}, and also through the test results on \textit{E.coli} using \textit{B.subtilis} data as the training set.

The use of GO annotation and KEGG pathway only improves the NN-based prediction results slightly. This is partially due to the low coverage of GO and KO annotation in each of the involved organisms. Even with a well-studied organism such as \textit{E.coli}, the coverage is only 50% and 61% for GO and KO annotation, respectively. With improved gene annotation, it is expected that the application of such information should provide a higher level of improvement in the NN-based methods.

One major limitation of our method, which is typical of most machine learning approaches, is that it requires the use of training data. In our case, the parameters of the NN and the optimal threshold were fixed based on \textit{B.subtilis} data, and applied to other prokaryotic organisms. This is a general problem with existing operon prediction algorithms which use features that could be species dependent. For example, the intergenic distance to distinguish between operon gene pairs and non-operon gene pairs can vary substantially depending on the species.

For our future work, other machine learning methods such as support vector machines and decision trees can also be investigated as alternative approaches to improve the prediction results. Furthermore, the NN-based prediction method can be expanded to include newer operon prediction programs as they become available. Further analysis to reduce the dimension of the input feature space could also be performed. This could involve testing a NN architecture based on only one or two existing programs in conjunction with combinations of the GO similarity, KEGG pathway and intergenic distance score. With a list of predicted operons in \textit{P. furiosus}, further study can be done to identify potential regulons (groups of operons sharing a common regulatory mechanism). For each operon, we can examine the region ~250 bp upstream and use prediction algorithms such as CUBIC (32) or MEME (33) to predict potential binding sites for transcriptional factors, and use the shared binding motifs as initial indicators for potential regulons.

The operon prediction algorithm presented in this paper coupled with GO, KEGG and microarray analysis brings forth the most comprehensive prediction of operon structure in the organism \textit{P. furiosus} to date. This approach can similarly be applied to other prokaryotic organisms with complete genomes. The computationally predicted operons for \textit{P. furiosus} in this study paves the way for further computational and experimental investigation into a better understanding of the regulatory pathway of this hyperthermophilic archaean.

**SUPPLEMENTARY DATA**

Supplementary Data are available at NAR Online.

**ACKNOWLEDGEMENTS**

We thank the reviewers for providing valuable suggestions and Hongwei Wu, Victor Olman, Huqing Liu, and other members of the Computational Systems Biology Laboratory for their helpful discussions. T.T.T. was supported in part by the US National Science Foundation under the Graduate Research Fellowship Program. P.D., Z.S. and Y.X. were supported in part by the National Science Foundation (#NSF/DBI-0354771, #NSF/ITR-IIS-0407204, #NSF/CCF-0621700, #NSF/DBI-0542119) and a ‘distinguished scholar’ grant from the Georgia Cancer Coalition. The work of F.L.P and M.W.W.A was supported in part by the Department of Energy (FG05-95ER20175) and the National Institutes of Health (GM 60329). Funding to pay the Open Access publication charges for this article was provided by #NSF/ITR-IIS-0407204.

Conflict of interest statement. None declared.

**REFERENCES**

1. Fiala,G. and Stetter,K.O. (1986) \textit{Pyrococcus furiosus} sp. nov. represents a novel genus of marine heterotrophic archaebacteria growing optimally at 100\textdegree C. \textit{Arch. Microbiol.}, 145, 56–61.
2. Adams,M.W.W. and Kelly,R.M. (eds) (2001) \textit{Hyperthermophilic Enzymes, Part C. Methods in Enzymology}. Academic Press, New York, Vol. 334, pp. 3–526.
3. Robb,F.T., Maeder,D.L., Brown,J.R., DiRuggiero,J., Stump,M.D., Yeh,R.K., Weiss,R.B. and Dunn,D.M. (2001) Genomic sequence of the hyperthermophile, \textit{Pyrococcus furiosus}: implications for physiology and enzymology. \textit{Methods Enzymol.}, 330, 134–157.
4. Poole,F.L., II, Gerwe,B.A., Hopkins,R.C., Schut,G.J., Weinberg,M.V., Jenney,F.E., Jr and Adams,M.W. (2005) Defining genes in the genome of the hyperthermophilic archaean \textit{Pyrococcus furiosus}: implications for all microbial genomes. \textit{J. Bacteriol.}, 187, 7325–7332.
5. Chen,X., Su,Z., Dam,P., Palenik,B., Xu,Y. and Jiang,T. (2004) Operon prediction by comparative genomics: an application to the \textit{Synechococcus} sp. WH8102 genome. \textit{Nucleic Acids Res.}, 32, 2147–2157.
