To appear in *Bioinformatics*

A Bayesian Network Approach to Operon Prediction

Joseph Bockhorst[‡†]
joebock@cs.wisc.edu

Mark Craven[†‡]
craven@biostat.wisc.edu

David Page[†‡]
page@biostat.wisc.edu

Jude Shavlik[‡†]
shavlik@cs.wisc.edu

Jeremy Glasner[*]
jeremy@genome.wisc.edu

[†]Dept. of Biostatistics & Medical Informatics
University of Wisconsin
1300 University Avenue
Madison, Wisconsin 53706

[‡]Dept. of Computer Sciences
University of Wisconsin
1210 West Dayton Street
Madison, Wisconsin 53706

[*]Dept. of Genetics
University of Wisconsin
445 Henry Mall
Madison, Wisconsin 53706

Abstract

Motivation: In order to understand transcription regulation in a given prokaryotic genome, it is critical to identify operons, the fundamental units of transcription, in such species. While there are a growing number of organisms whose sequence and gene coordinates are known, by and large their operons are not known.

Results: We present a probabilistic approach to predicting operons using Bayesian networks. Our approach exploits diverse evidence sources such as sequence and expression data. We evaluate our approach on the *E. coli* K-12 genome where our results indicate we are able to identify over 78% of its operons at a 10% false positive rate. Also, empirical evaluation using a reduced set of data sources suggests that our approach may have significant value for organisms that do not have as rich of evidence sources as *E. coli.*

Availability:
Our *E. coli* K-12 operon predictions are available at http://www.biostat.wisc.edu/gene-regulation

Contact: joebock@biostat.wisc.edu

Introduction

The availability of complete genomic sequences and microarray expression data calls for new computational methods for uncovering the regulatory apparatus of a cell. We present a Bayesian network approach to predicting operons in prokaryotic genomes. Our approach is able to take into account several data sources including gene coordinates, codon usage statistics, predicted transcription signals, and expression data. We evaluate our approach using data from the *E. coli* K-12 genome (Blattner *et al.* 1997).

In earlier work (Craven *et al.* 2000), we presented an operon prediction approach that involved two components: a simple probabilistic method for scoring candidate operons, and a dynamic programming algorithm for combining predictions across sequences of candidate operons. With these two components together we are able to predict an *operon map* for an entire genome. The work reported here extends our earlier research in several key directions. First, we describe and evaluate more complex probabilistic models for scoring candidate operons. These models encode more "background knowledge" about the problem than our earlier models and as a result provide better predictive accuracy. Second, we include codon usage statistics in our probabilistic model and show it to be informative. This evidence source is appealing because it requires no additional knowledge beyond gene coordinates and sequence and thus can be used to predict operons in many bacterial genomes. To our knowledge, our method is the first to incorporate codon usage statistics in an operon model. Finally, we evaluate the accuracy of our approach when only limited data sources are available, and demonstrate its value in this situation.

Several other research groups (Overbeek *et al.* 1999; Tamames *et al.* 1997) have addressed the task of predicting *functionally coupled* genes by identifying clusters of genes that are conserved across different genomes. Ermolaeva *et al.* (2001) have presented an approach that uses this kind of information to predict the more specific concept of operons. We consider this method to be complementary to ours in that it is based on cross-genome information, whereas our approach is based on information present within a single genome.

Salgado *et al.* (2000a) and Moreno-Hagelsieb and Collado-Vides (2002) have investigated an approach to predicting operons in *E. coli* using gene coordinates and functional annotation data. Our work differs from theirs in several key respects. First, our learned models use a richer representation of the problem. Whereas their predictions are based on likelihood ratios, ours are based on Bayesian networks that represent key dependencies among various pieces of information used in the predictions. Second, our models use additional data sources – codon usage, expression data and predicted transcription signals – in their predictions. Third, our experiments measure accuracy using held-aside test instances, whereas theirs do not.

Other operon prediction approaches use just a single type of evidence. Yada *et al.* (1999) use hidden Markov models to predict genes as well as operons from DNA sequence alone, Tjaden *et al.* (2002) use expression data from both genes and non-coding regions and Zheng *et al.* (2002) use biochemical pathway information.

Another fundamental way in which our approach differs from some previous work (Salgado *et al.* 2000a; Ermolaeva, White, & Salzberg 2001; Moreno-Hagelsieb & Collado-Vides 2002; Tjaden *et al.* 2002) is that ours predicts *complete* operons whereas others predict only *pairs* of genes belonging to the same operon.

Problem Domain

The task we consider here is to predict operons in the *E. coli* genome, although our approach is applicable to other prokaryotic organisms. The genome of *E. coli*, which was sequenced at the University of Wisconsin (Blattner *et al.* 1997), consists of a single circular chromosome of double-stranded DNA. The chromosome of the particular strain of E. coli (K-12) in our data set has 4,639,221 base pairs. *E. coli* has approximately 4,400 genes, which are located on both strands.

The definition of *operon* that we use throughout the article is a sequence of one or more genes that, under some conditions, are transcribed as a unit. There are several aspects of this definition that are important to note. First, genes that are transcribed individually are included in this definition; we refer to these special cases as *singleton* operons. Second, our definition treats as multiple operons those cases (such as *rpsU-dnaG-rpoD* in *E. coli*) in which multiple promoters and/or terminators result in different subsequences of a larger gene sequence being transcribed under different conditions. We consider each of the distinct gene sequences that can be transcribed as a unit to be an operon.

Figure 1 illustrates the concept of an operon. The transcription process is initiated when RNA polymerase binds to a *promoter* before the first gene in an operon. The RNA polymerase then moves along the DNA using it as a template to produce an RNA molecule. When the RNA polymerase gets past the last gene in the operon, it encounters a special sequence called a *terminator* that signals it to release the DNA and cease transcription.

The data that we have available for learning a model of operons, some of which come from the RegulonDB (Salgado *et al.* 2000b), include the following:

- complete DNA sequence of the genome,

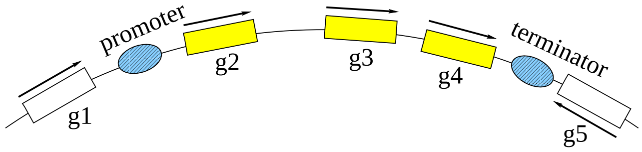


Figure 1: The concept of an operon. The curved line represents part of the *E. coli* chromosome and the rectangular boxes on it represent genes. An operon is a sequence of genes, such as $[g2, g3, g4]$ that is transcribed as a unit. Transcription is controlled via an upstream sequence, called a *promoter*, and a downstream sequence, called a *terminator*. Each gene is transcribed in a particular direction, determined by which of the two strands it is located. The arrows in the figure indicate the direction of transcription for each gene.

- beginning and ending positions of 3,033 genes and 1,372 putative genes,
- positions and sequences of 438 known promoters, and 289 known terminators (147 rho-dependent and 142 rho-independent),
- gene expression data characterizing the activity levels of the 4,097 genes and putative genes across 39 experiments and
- 365 known operons.

It is estimated that there are thousands of undiscovered operons in *E. coli* (Salgado *et al.* 2000a; Wolk *et al.* 2001). Our goal is to predict these operons using a model learned from the data described above. We assume that operons do not include RNA genes and that they do not "bridge" genes on the opposite strand.

An interesting aspect of our learning task is that we do not have a set of known non-operons to use as negative examples. The nature of scientific inquiry is such that several hundred operons have been identified in *E. coli*, but little attention has been focused on identifying sequences of genes that *do not* constitute operons.

We are able, however, to assemble a set of 6633 putative non-operons by exploiting the fact that operons rarely overlap with each other. Given this rule, we generate a set of negative examples by enumerating every sequence of consecutive genes, from the same strand, that contains at least one gene from a known operon but is itself not a known operon. Some of these generated non-operons might actually be true operons, because operons do overlap in some cases. However, the probability of any particular negative example being a true operon is small,and our learning algorithms are robust in the presence of noisy data.

Problem Representation

In this section we describe the features that our learning method uses to assess the probability that a given candidate (that is, sequence of genes) actually is an operon. Other than the codon usage based features, these features have been used in previous work (Craven *et al.*

2000). Space considerations dicate that the discussion of some of these features be condensed.

Length and Spacing Features

We use several features that relate to the length and inter-genic spacing of operons and non-operons:

- Operon length: The number of genes in the candidate operon.
- Within-operon spacing: The *mean* and the *maximum* spacing (number of DNA base pairs) between the genes contained in a candidate operon, e.g., the distances between *g2* and *g3* and between *g3* and *g4* in Figure 1. Since the genes in an operon are transcribed together, there might be constraints on intergene spacing. These features are not defined for singleton operons (operons consisting of one gene).
- Distance to neighboring genes: The distances to the *preceding* (*g1* in Figure 1) and *following* (*g5* in Figure 1) genes. Notice that these genes are not part of the candidate operon.

Codon Usage Features

The codon usage characteristic of a gene is influenced by a variety of factors (S. Karlin 1998) including some, such as gene function, expression level and evolutionary history, that also influence the grouping of genes into operons. To decide whether or not a sequence of genes constitutes an operon, it may be helpful to consider the codon usage of the genes.

We associate with each gene g_k a set of codon bias vectors $\{\vec{b}^k_a\}$, one for each amino acid. Let n_{uvw} be the number of times codon uvw appears in g_k Then, the elements of the bias vectors are

$$b^k_{a,uvw} = \hat{f}_{(uvw|a)} - \bar{f}_{(uvw|a)}$$

where uvw is a codon that codes for a, $\bar{f}_{(uvw|a)}$ is the frequency with which a is encoded by uvw (relative to other codings for a) over the whole genome and

$$\hat{f}_{(uvw|a)} = \frac{n_{uvw} + \bar{f}_{(uvw|a)}}{\sum_{xyz \in \mathrm{codons}(a)} n_{xyz} + 1}$$

is the smoothed frequency with which a is coded for by uvw in the gene. The sum in the denominator ranges over all codons that code for amino acid a.

We define the codon usage similarity between two genes as

$$Sim(g_k, g_l) = \sum_a \vec{b}^k_a \cdot \vec{b}^l_a.$$

This measure is symmetric and reflects both the consistency and degree to which the bias vectors are correlated.

We derive four codon usage based features for a candidate operon c: the codon usage similarity between the first gene in c and the previous gene, the codon usage similarity between the last gene in c and the following gene, and the mean and minimum of the codon usage similarity among all pairs of genes in c.

Transcription Signal Features

Other types of evidence associated with operons are transcription control signals, such as promoters and terminators. Thus, to decide if a given sequence of genes represents an operon or not, we would like to look upstream from the first gene in the sequence to see if we find a promoter, and to look downstream from the last gene to see if we find a terminator. The task of recognizing promoters and terminators, however, is not easily accomplished. Although there are known examples of both types of sequences, the sufficient and necessary conditions for them are not known. Thus, to use promoters and terminators as evidence for operons, we first need some method that can be used to predictively identify them.

Our approach is to use the known examples of these two types of signals to learn statistical models for predicting them. Specifically, we induce *interpolated Markov models* (IMMs) (Jelinek & Mercer 1980) that characterize the known promoters and terminators. Features associated with a candidate operon are constructed by scanning the trained promoter and terminator IMMs along the sequence upstream and downstream of the candidate operon, and retaining the largest predicted probability for each scan.

Gene Expression Features

The expression data we use comes from 39 microarray experiments conducted by the Wisconsin *E. coli* Genome Project. Since our expression data comes from cDNA arrays, we have two measurements (fluorescence intensities) for each gene in each experiment: the relative amount of mRNA under some experimental condition versus the relative amount under some baseline condition. We employ the common practice of using the ratio of these two values as the expression intensity for a single gene under the condition being measured.

We associate with each gene a vector of its expression ratios over all conditions. The expression based features used to classify candidate operons are based on a correlation metric, used previously for analyzing gene expression data (Eisen *et al.* 1998), between the expression vectors of certain pairs of genes.

Specifically, we use following four expression based features to characterize a candidate operon c: (i) the correlation between the first gene in c and the previous gene in the sequence, (ii) the correlation between the last gene in c and the next gene in the sequence. (iii) the mean correlation of all pairs of genes in c and (iv) the minimum correlation among all pairs of genes in c.

Making the Feature Values Discrete

The operon model we employ, which we describe in the next section, is more easily defined and accurately trained if features are restricted to a relatively small number of discrete values. We make all our feature values discrete by partitioning each feature's range into ten non-overlapping bins. These bins are labeled one through ten and thresholds between bins are chosen so that an equal number of positive training examples fall into each. We use this method of define bin boundaries because our training sets have many more negative than positive examples and we want to avoid the case when very few training examples of some class fall in any one bin due to sampling effects. The discrete value of a feature is the index of the bin into which the associated raw value falls.

Bayesian Network Operon Model

In this section we show how the features just described are used by a machine learning approach to induce a probabilistic model, called a *Bayesian network* (Pearl 1988), of operons. A Bayesian network is way of representing the joint probability distribution of a set of random variables that exploits the conditional independence relationships among the variables, often greatly reducing the number of parameters needed to represent the full joint probability distribution. Also, Bayes nets give a powerful and natural way to represent the dependencies that do exist. This contrasts to naive Bayes models, which we used in previous work (Craven *et al.* 2000), where it assumed that all non-class variables are conditionally independent of each other given the class.

The ability to model dependencies between variables is important in our task for pairs of features that both relate to a certain aspect of an operon; for example, the 5' spacing and promoter features both refer to the 5' edge. The values of these features are clearly not independent of each other for negative examples. For example, if the promoter feature indicates there is high probability of a promoter upstream of some negative example, it is likely its 5' edge is the 5' edge of some actual operon thus changing our distribution over 5' spacing. These types of dependencies cannot be represented by a naive Bayes model, but are naturally handled with general Bayes nets.

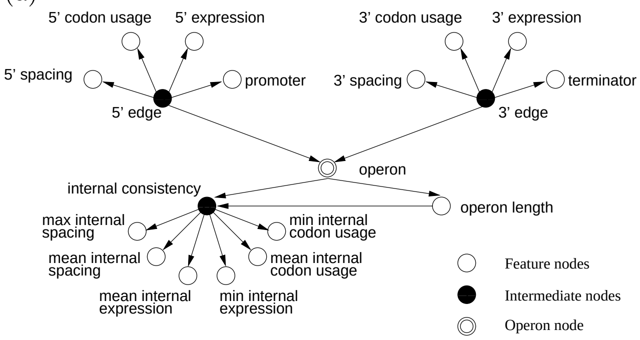
A Bayes net consists of two components: a qualitative one (the *structure*) in the form of a directed acyclic graph whose nodes correspond to the random variables and a quantitative component consisting of a set of conditional probability distributions. The structure of the graph encodes a set of conditional independence assertions through the absence of arcs between nodes. In particular, a node is conditionally independent of all non-parent ancestors given its parents. These assertions allow the full joint probability distribution to be compactly represented by storing a conditional probability distribution at each node conditioned on its parents, as can be readily seen through a rewriting of the chain rule

$$\Pr(X_1,\ldots,X_n) = \prod_i^n \Pr(X_i|X_1,\ldots,X_{i-1}) \quad (1)$$

$$= \prod_i^n \Pr(X_i|Parents(X_i)). \quad (2)$$

Here the X_i's are the random variables and $Parents(X_i)$ is the set of X_i's parents. The last line exploits the conditional independence relationships and the ordering from 1 to n is such that a node is preceded by all of its parents.

(a)



(b)

5' edge	3' edge	Pr(¬operon)	Pr(operon)
false	false	0.999	0.001
false	true	0.999	0.001
true	false	0.999	0.001
true	true	0.580	0.420

Figure 2: (a) The structure of our Bayes net operon model. (b) An example conditional probability distribution for the operon node given its parents nodes 5' edge and 3' edge.

We construct by hand the structure of our Bayes net operon model, shown in Figure 2, from knowledge of the domain and with an effort to have arcs correspond with causes. This correspondence usually results in a network with fewer arcs and therefore fewer parameters to estimate. Our network contains three types of nodes: those that represent the features just described (feature nodes, the open in circles in Figure 2), those that represent intermediate concepts important in describing operons (intermediate nodes, the filled in circles in Figure 2) and one, the operon node, that represents whether a candidate is an operon or not.

The intermediate nodes and the operon node are Boolean valued. The 5' edge node represents the case that the first gene of the candidate is the first gene in *some* actual operon; an analogous statement holds for the 3' edge node and the last gene. The node internal consistency represents the case that all of the genes in a candidate are in the same actual operon. Note that with the exception of operon length, the feature nodes only influence operon through one of the intermediate nodes and if that intermediate node is observable, the child feature nodes' values have no impact on operon. This matches the intuition that if, say, it is known that, for some candidate, 5' edge is true, the distance to the nearest gene upstream (5' spacing) has no influence on whether or not the candidate is a true operon.

Since all of the variables are discrete, each node's conditional probability distribution can be represented by a conditional probability table (CPT). An example of the CPT for the operon node is shown in Figure 2(b). Each row of this CPT refers to a state of the operon node's parents and gives its probability distribution given this parental state.

Training

Training our Bayes net involves the setting of the probability parameters. To do this, we first compute a table of counts for each node. This table has the same dimensions as its CPT; an example CPT is shown in Figure 2(b). A cell in the count table for X_i indicates the total number of times over the training set that X_i took on the value given by the cells' column when its parents were in the state given by its row. To create a CPT from this count table we add 1 to each cell and normalize each row so it sums to 1. That is, we compute Laplace estimates (Mitchell 1997). In some training examples the values of one or more intermediate nodes are unobservable while their child nodes' values are observable. Although, the standard approach in these cases is to use EM or Gibbs' sampling to fully utilize these training examples, the number of these cases is small so, for simplicity, we keep the tables of counts for the unobservable nodes and their children unchanged.

Classification

Given a new candidate operon we use the following procedure to decide whether or not a candidate is an actual operon. First, we set the values of any observable intermediate nodes. In particular, since all genes of an operon are transcribed in the same direction, if the orientations of the first gene and the previous gene do not match, the 5' edge node is observable (and true) and if the orientations of the last gene and the following gene do not match the 3' edge node is true. Also, internal consistency is true for singleton candidate operons. Next, we set the feature nodes to the candidates' feature values. The values of the feature nodes are always observable with the exception that for singletons, the children of internal consistency are undefined. However, in this case internal consistency is directly observable. Then, we apply the variable-elimination inference procedure (Russell & Norvig 1995) to compute the probability distribution of the operon node given the values just fixed. Although inference in Bayesian networks is in general a computationally demanding problem, it can be performed very quickly in our case because both the network size and the number of hidden nodes are relatively small. When we want to classify a candidate we choose a threshold, such as 0.5, above which we call a candidate an actual operon, or we may use the probabilities themselves if we want to do further inference.

Empirical Evaluation

In this section we examine the predictive accuracy of our Bayes net operon model. We have conducted a set of experiments designed to answer the following questions:

- What is the overall accuracy of our model?
- What is the predictive value of individual data sources?
- What is the predictive value of the network structure?

We run ten-fold cross-validation experiments with a data set consisting of 365 known operons and 6633 runs of genes thought not to be operons. For each training/test set split, besides learning new histograms for each of our features, we also learn new promoter and terminator IMMs, leaving out of these models' training sets those promoters and terminators that are associated with test-set operons. In this way, we can ensure that our operon predictions are not biased by using information that is closely linked to a given test case (i.e., a known promoter or terminator), that we would not have in the case of a currently undiscovered operon.

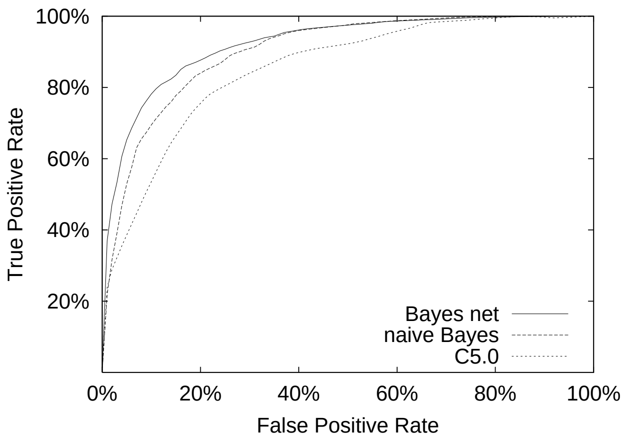


Figure 3: ROC curves for our Bayes net model, a naive Bayes model, and C5.0.

To answer the first question above, we treat our Bayes net operon model as a classifier and generate an ROC (receiver operating characteristic) curve (Eagen 1975). An ROC curve is a plot of false positive (FP) rate versus true positive (TP) rate where

$$\text{FP rate} = \frac{\#\text{false positives}}{\#\text{negatives}} \tag{3}$$

and

$$\text{TP rate} = \frac{\#\text{true positives}}{\#\text{positives}}. \tag{4}$$

Points on the curve are generated by varying the threshold on the posterior probability of the operon node that divides positive from negative predictions. A model that guessed randomly would result in an ROC "line" defined by: TP rate = FP rate. For comparison, we consider two alternatives, the naive Bayes models developed in our earlier work (Craven *et al.* 2000) and decision trees induced using the C5.0 algorithm (Quinlan 1999), where both alternatives are given as features

Table 1: Area under ROC curves and p-values from a test comparing areas under the ROC curve to that of the Bayes net model.

method	ROC area	p-value
Bayes net	0.921	
naive Bayes	0.901	0.020
C5.0	0.851	< 0.001

the values of the Bayes net's feature nodes. We generate points on the C5.0 ROC curve by varying misclassification costs. All ROC curves shown are averages over all folds. That is, the TP Rate at a given FP Rate is the mean of the TP Rate at that FP Rate over all folds.

The ROC curves of the Bayes net and alternative models are shown in Figure 3. The significant deviation of all ROC curves from the 45° random-guess ROC line (not shown) indicates that each model has substantial predictive value. Comparing the Bayes net ROC curve to the others, we see that at any given FP Rate, its TP Rate is equal to or greater than the TP Rates of the other methods.

The area under an ROC curve can be used as an measure of model quality. For the probabilistic classifiers, an interpretation of this measure is the probability that a randomly selected positive instance will have a higher probability than a randomly selected negative instance. Table 1 shows this measure for the three models. The Bayes net ranks first followed by the naive Bayes and C5.0 models.

We perform a statistical test to determine if the difference in the areas under the ROC curves of two models is significant. For any two models, we define the null hypothesis to be the case that the expected difference in the areas under their ROC curves is zero. We make the assumption that the area under the ROC curve for a model on a single fold is a normally distributed random variable whose mean is the expected area and whose variance is constant for all folds. The p-values from a two-tailed, paired t-test comparing the Bayes net model with the alternatives are shown in the last column of Table 1. If we use a standard threshold of 0.05 on the p-values, there exists significant evidence that our Bayes net model is more accurate than both of the alternative models.

Although the alternative models employ different inductive biases than the Bayes net, their predictive value as indicated by their ROC curves indicates that the features we have defined are valuable and suggests that much of the predictive value of the Bayes net model is due to them. However, the differences between the curves and areas under them suggests that the structure of the Bayes net has value as well.

Evaluating the Data Sources

In this section we examine the predictive power of groups of features. We perform this analysis to de-

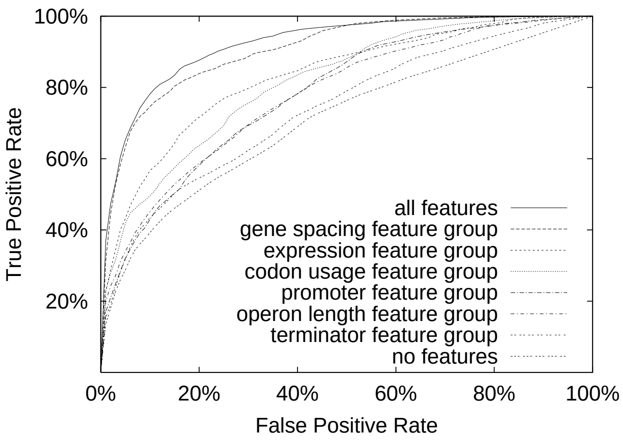


Figure 4: ROC curves for our Bayes net model using single feature groups. Note that the order of the key matches the order of the curves. The ROC curves for models using all features and no features are included for reference.

termine the relative value of the data sources and the predictive value of our approach under conditions of reduced data sources. This final condition is especially important because it provides an estimate of the applicability of our approach to organisms that do not have as diverse of data sources as *E. coli*.

We assign each feature node to a group based on its data source. The operon length, promoter, and terminator groups contain a single feature and the gene spacing, codon usage, and expression groups each contain multiple features. We conduct a set of experiments with reduced Bayes net models where subsets of the feature groups are removed from the Bayes net. The intermediate nodes, however, are not removed even in cases where they do not have any child feature nodes.

Figure 4 shows the ROC curves for reduced Bayes net models containing single feature groups. For reference, the ROC curve with all features is repeated and the ROC curve with no features is shown. The "no features" model contains just the three intermediate nodes and the operon node. The gene spacing group is the clear winner as it dominates over the other groups and is close to the "all features" curve. Next, the codon usage and expression groups have simliar ROC curves as do the promoter and operon length groups, each with substantial improvement over the "no features" curve. Finally, we have the terminator group with little improvement.

The robustness of the model using only the gene spacing group is encouraging because it indicates that our approach is applicable to other genomes that currently do not have the rich data sources that are available for *E. coli*. Furthermore, as long as gene boundaries and sequence are known, the operon length and codon usage features can be easily obtained as well. In fact, experiments on the Bayes net model with the gene spacing, operon length and codon usage features yield perfor-

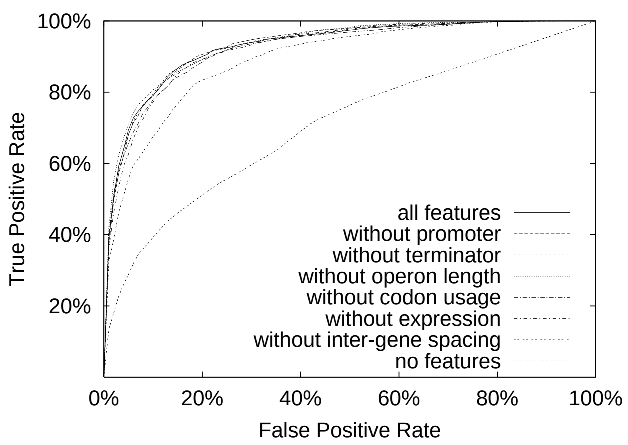


Figure 5: ROC curves for our Bayes net model leaving individual feature groups out. Note that the order of the key matches the order of the curves.

mance similar to the full model (last line in Table 2). Of course, making predictions presupposes the existence of a model trained from a training set of known operons and non-operons. If known operons are not available for a genome of interest, a model trained on another organism, for example *E. coli*, may be applied. Although we expect performance to degrade somewhat as a learned model is transferred across genomes, less so for closely related genomes, the precise implications of such a transfer are not well understood and is an area of current research.

Two of the weakest features are those derived from the models of the transcription control sequences. The relatively poor performance of these features, especially the terminator feature, highlights the difficulty of finding good models for these signals and suggests that considering alternate models (Bockhorst & Craven 2001) may be useful.

Figure 5 shows the ROC curves for models where single feature groups have been *removed*. These results support the findings from using only a single feature group. The most significant loss occurs when the gene spacing features are withheld. The only other groups whose withholding results in a noticeably lower curve are the expression and codon usage groups.

We are especially interested in the value of the codon usage features because they are easily obtained and have not previously been used in making operon predictions. To measure their value, we perform two paired t-tests of ROC curve area as above. The first compares the model trained with all features to the model trained with all features except codon usage and the second compares the model trained with operon length, gene spacing and codon usage to the model trained with just operon length and gene spacing (ROC curves not shown). We designed the second test to measure the value of the codon usage features for a situation where only gene coordinates and sequence are known. The p-values for thses two tests are 0.15 and 0.05, from which we conclude that the codon usage features can have significant value, at least in cases of limited data.

Evaluating the Structure

In order to more thoroughly evaluate the benefit of the Bayes net structure, we also run the single-feature group experiments using naive Bayes models. In addition we run experiments with models containing the features derivable from only gene coordinates and sequence data (the gene spacing, operon length and codon usage features). Table 2 shows the area under the ROC curves of Bayes net and naive Bayes models using the different feature groups. Also shown are the p-values from a test of equal areas. For all feature groups, the area under the Bayes net curves are greater than those of the naive Bayes curves and the differences are statistically significant.

Although the addition of features beyond the easily obtained operon length, gene spacing and codon usage features offers at best a slight improvement in performance, we present results with the other more complicated features as well for three reasons. First, on their own, all feature types have predictive value. Second, the value of the gene expression features may grow as the number and quality of the experiments grow. Third, and most importantly, it is not clear how well the results reported here will transfer to other genomes, (although there is indication that genome-wide gene spacing statistics are relatively consistent across a wide range of prokaryotic genomes (Moreno-Hagelsieb & Collado-Vides 2002)) and having a variety of evidence sources in these cases may prove useful.

Conclusions

We have presented a computational method, based on a Bayesian network, for predicting operons in prokaryotic genomes. The network structure is manually crafted from background knowledge and its parameters are set by a machine learning method. Our method takes advantage of a variety of data sources including gene coordinates, predicted transcription signals, gene expression experiment results and codon usage statistics. We believe this to be the first use of codon usage statistics for the task of operon prediction.

From our empirical evaluation we make the following conclusions:

- The Bayes net model has significant predictive value.
- The Bayes net model significantly outperforms naive Bayes and Quinlan's C5.0 algorithm in terms of area under ROC curves.
- The codon usage features add significant predictive value, especially in the case of limited data sources.
- Under conditions where the available data sources are limited, the Bayes net model still has significant predictive value. In particular, our model is accurate

Table 2: Comparison of our Bayes net and naive Bayes models. Results for models constructed using all features, individual feature groups and those features derivable from only gene coordinates and sequence (gene spacing, operon length and codon usage) are presented. The first column indicates the features used to construct the model. The second and third columns show the areas under the ROC curves (AUC) of our Bayes net model and a naive Bayes model respectively. The fourth column shows the p-values from a test of the null hypothesis that the AUCs of the two models are the same.

group	AUC Bayes net	AUC naive Bayes	p-value
all features	0.929	0.911	0.006
spacing	0.915	0.896	0.003
expression	0.831	0.751	0.001
codon usage	0.815	0.662	< 0.001
promoter	0.781	0.715	0.001
operon length	0.778	0.713	< 0.001
terminator	0.739	0.568	< 0.001
length, spacing, & codon usage	0.924	0.883	< 0.001

given only sequence and gene coordinates. The implication of this result is that our approach is generally applicable to organisms that have not been as heavily studied as *E. coli*.

The relative weakness of the features based on transcription control signals, especially the terminator feature, indicate an area for improvement. We are currently exploring more appropriate models for these important signals.

Finally, we have used our Bayes net model along with a dynamic program for combining predictions (Craven *et al.* 2000) to construct an operon map for *E. coli* K-12. This map can be obtained from http://www.biostat.wisc.edu/gene-regulation.

Acknowledgments

This work was supported by NSF Grant IIS-0093016 (JB and MC), NIH Grant 1T15 LM07359-01 (JB), NIH Grant 1R01 LM07050-01 (MC and JS), NSF Grant IIS-9987841 (DP), NIH Grant 2P30 CA14520-29 (JS), and NIH Grant 1R01 GM35682-15A1 (JG). Thanks to Fred Blattner, Christina Kendziorski, Nicole Perna and Craig Richmond for helpful discussions regarding this research. Thanks to Samuel Karlin for suggesting the incorporation of codon usage information.

References

Blattner, F. R.; Plunkett, G.; Bloch, C. A.; Perna, N. T.; Burland, V.; Riley, M.; Collado-Vides, J.; Glasner, J. D.; Rode, C. K.; Mayhew, G. F.; Gregor, J.; Davis, N. W.; Kirkpatrick, H. A.; Goeden, M. A.; Rose, D. J.; Mau, B.; and Shao, Y. 1997. The complete genome sequence of Escherichia coli K-12. *Science* 277:1453–1474.

Bockhorst, J., and Craven, M. 2001. Refining the structure of a stochastic context-free grammar. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 1315–1320. Seattle, WA: Morgan Kaufmann.

Craven, M.; Page, D.; Shavlik, J.; Bockhorst, J.; and Glasner, J. 2000. A probabilistic learning approach to whole-genome operon prediction. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, 116–127. La Jolla, CA: AAAI Press.

Eagen, J. P. 1975. *Signal Detection Theory and ROC analysis.* New York: Academic Press.

Eisen, M. B.; Spellman, P. T.; Brown, P. O.; and Botstein, D. 1998. Cluster analysis and display of genome-wide expression patterns. *PNAS* 95(25):14863–14868.

Ermolaeva, M.; White, O.; and Salzberg, S. 2001. Prediction of operons in microbial genomes. *Nucleic Acids Research* 29:1216–1221.

Jelinek, F., and Mercer, R. L. 1980. Interpolated estimation of Markov source parameters from sparse data. In Gelsema, E. S., and Kanal, L. N., eds., *Pattern Recognition in Practice.* North Holland. 381–397.

Mitchell, T. M. 1997. *Machine Learning.* New York: McGraw-Hill.

Moreno-Hagelsieb, G., and Collado-Vides, J. 2002. A powerful non-homology method for the prediction of operons in prokaryotes. *Bioinformatics* 18(Suppl. 1):S329–S336.

Overbeek, R.; Fonstein, M.; D'Souza, M.; Pusch, G. D.; and Maltsev, N. 1999. The use of gene clusters to infer functional coupling. *Proceedings of the National Academy of Science (USA)* 96:2896–2901.

Pearl, J. 1988. *Probabalistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, CA: Morgan Kaufmann.

Quinlan, J. R. 1999. C5.0 decision tree software. http://www.rulequest.com/.

Russell, S., and Norvig, P. 1995. *Artificial Intelligence.* New Jersey: Prentice Hall.

S. Karlin, J. Mrázek, A. C. 1998. Codon usages in different gene classes of the *escherichia coli* genome. *Molecular Microbiology* 6:1341–1355.

Salgado, H.; Moreno-Hagelsieb, G.; Smith, T. F.; and Collado-Vides, J. 2000a. Operons in Escherichia coli: Genomic analyses and predictions. *Proceedings of the National Academy of Sciences, USA* 97(12):6652–6657.

Salgado, H.; Santos-Zavaleta, A.; Gama-Castro, S.; Millán-Zárate, D.; Blattner, F. R.; and Collado-Vides, J. 2000b. RegulonDB (version 3.0): Transcriptional regulation and operon organization in Escherichia coli K-12. *Nucleic Acids Research* 28(1):65–67.

Tamames, J.; Casari, G.; Ouzounis, C.; and Valencia, A. 1997. Conserved clusters of functionally related genes in two bacterial genomes. *Journal of Molecular Evolution* 44:66–73.

Tjaden, B.; Haynor, D.; Stolyar, S.; Rosenow, C.; and Kolker, E. 2002. Identifying operons and untranslated regions of transcripts using Escherichia coli RNA expression analysis. *Bioinformatics* 18(Suppl. 1):S337–S344.

Wolk, Y.; Rogozin, I.; Kondrashov, A.; and Koonin, E. 2001. Genome alignment, evolution of prokaryotic genome organization, and prediction of gene function using genomic context. *Genome Research* 11(3):356–372.

Yada, T.; Nakao, M.; Totoki, Y.; and Nakai, K. 1999. Modeling and predicting transcriptional units of Escherichia coli genes using hidden Markov models. *Bioinformatics* 15(12):987–993.

Zheng, Y.; Szustakowski, J.; Fortnow, L.; Roberts, R.; and Kasif, S. 2002. Computational identification of operons in microbial genomes. *Genome Research* 12(8):1221–1230.