2007

Operon Prediction with Bayesian Classifiers

Natalia Khuri
San Jose State University

OPERON PREDICTION WITH BAYESIAN CLASSIFIERS

A Project Report
Presented To
The Faculty of the Department of Computer Science

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Natalia Khuri
May 2007

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

_____

Dr. John Avila

_____

Dr. Agustin Araya

_____

Dr. Chris Pollett

ABSTRACT

OPERON PREDICTION WITH BAYESIAN CLASSIFIERS

by Natalia Khuri

In this work, we present an approach to predicting transcription units based on Bayesian classifiers. The predictor uses publicly available data to train the classifier, such as genome sequence data from Genbank, expression values from microarray experiments, and a collection of experimentally verified transcription units.

We have studied the importance of each of the data source on the performance of the predictor by developing three classifier models and evaluating their outcomes. The predictor was trained and validated on the *E. coli* genome, but can be extended to other organisms. Using the full Bayesian classifier, we were able to correctly identify 80% of gene pairs belonging to operons.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Bioinformatics is a discipline that brings together scientists from different fields to gain a better understanding of the biological processes. Advances in this field have resulted in numerous important discoveries and generated a lot of data. General and specialized databases have been developed to store the information. New experimental techniques, such as microarray technologies, are expected to shed light on many processes not yet understood. Many tools can be developed to mine the data and test different hypotheses. In this work, we are interested in developing a tool to predict transcription units in the prokaryotic organism, *E. coli* bacterium.

In the next section we introduce the biology needed to understand the processes occurring in a prokaryotic cell. In Section 3, we present a field of functional genomics and describe different experimental techniques to study expression of genes in an organism. Bioinformatics methods for the discovery of the transcription units are discussed in Section 4. In Section 5, we state the problem we try to solve in this work and present the solution developed, a predictor based on Bayesian classifiers. The process of estimating parameters of the model is given in Section 6 and the details of our implementation in Section 7. We present our results in Section 8 and conclude this report in Section 9.

2 BIOLOGICAL BACKGROUND

A *deoxyribonucleic acid* (DNA) of an organism encodes all of the *ribonucleic acid* (RNA) and protein molecules that are needed for its functioning. All the cells of an organism, except blood and reproductive cells contain DNA. The entire DNA of an organism is called a *genome*. *Prokaryotes* are unicellular or multi-cellular organisms, such as bacteria, whose genomes are contained in a single double-stranded circular DNA molecule. Some prokaryotic organisms also have smaller DNA molecules called *plasmids*. The sizes of sequenced microbial genomes range from 0.49 million base pairs (Mbps) in hyperthermophilic archaeal parasite *Nanoarchaeum equitans* Kin4-M [42] to 9.12 Mbps in Gram-positive bacterium *Streptomyces avermitilis* MA-4680 [19].

All cells in an organism contain exactly the same DNA. Exactly the same DNA is also found in cells in different stages of development [1]. But different portions of the DNA are transcribed and translated under different conditions or in different cells of an organism. In general, when a cell needs new proteins a transcription process is activated. The DNA is copied (transcribed) into a more unstable RNA molecule. The segment of the DNA that is transcribed into RNA is called a *gene*. The RNA that codes for a protein is called *messanger* RNA (mRNA) and the DNA segment that provides that code is known as *open reading frame* (ORF). When read in the 5′ to 3′ direction, the portion of the DNA before an ORF is called *upstream*, and the portion following an ORF is called *downstream*. Although about 90% of all genes in a prokaryotic organism are protein coding, only about 4% of cell's total RNA codes for proteins [22]. The majority of the RNA, such as ribosomal RNA (rRNA) and transfer RNA (tRNA), is used to aid the translation process. Cells also have many types of small RNAs whose function is under rigorous investigation in major laboratories today.

The process shown in Figure 1, by which a gene produces its product and the product carries out its function is called *gene expression*. G*ene regulation* refers to a mechanism that controls the synthesis of a particular gene product. Gene expression in prokaryotes is regulated mainly through transcription. At any given time, only a fraction of genes in an organism is expressed, and cells have the ability to change the expression of their genes in response to external signals. Many of the genes are always expressed, while some become active only when the cell needs their products. Interestingly, even though gene expression is said to occur when gene products are needed, cells always maintain the minimum amount of RNA from every gene in the genome.

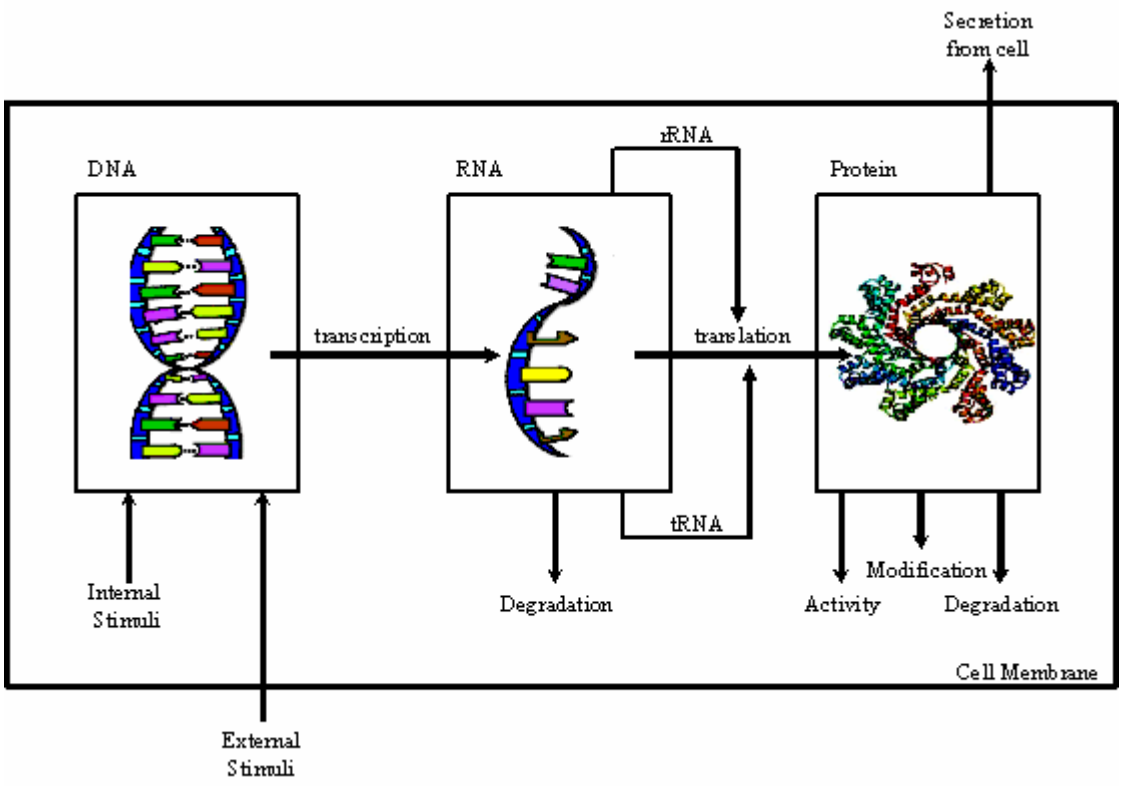

Figure 1. Gene expression is a multi-step process. [Adopted from [22]].

Internal and external factors trigger transcription of protein coding genes. The DNA of these genes is transcribed into mRNA. The mRNAs are translated into proteins, which have different roles within and outside of cells. Three types of RNA molecules, mRNA, rRNA and tRNA participate in the translation process. The translation products are folded, modified, and sent to their final destinations. In prokaryotes, mRNA is degraded within a few minutes after translation. The protein structure shown is of a protein of unknown function from a pro-phage integrated into genome of *Bacillus cereus* bacterium [2].

Experimental studies of the *Escherichia coli* (*E. coli*) bacterium by F. Jacob and J. Monod in the 1950s revealed a special type of genes that are co-expressed under the same condition, such as the availability of food sources [20]. These genes are organized into multi-gene clusters, called operons. Operon genes often have the same cellular function and their products form complex molecules [29].

Formally, an *operon* is defined as a *transcription unit* (TU) consisting of a promoter followed by two or more genes and a transcription terminator. A *promoter* is a DNA sequence located upstream of a gene recognized by an RNA polymerase [1]. The genes in an operon are usually transcribed from the same promoter into a single primary transcript, which contains coding regions for the synthesis of multiple proteins (see Figure 2). The mRNA of this type is called a *polycistronic* mRNA; mRNAs coding for a single protein are called *monocistronic*, for example, **amyA** gene in Figure 2. The same ribosome translates all of the proteins coded by the polycistronic mRNA. The actual quantity of each of the proteins synthesized from a polycistronic mRNA can differ. These differences are partly due to the failure of ribosome to reinitiate with the mRNA when translating downstream genes. There are operons with several promoters, some of which are found between operon genes. These alternative promoters are used by RNA polymerases in certain conditions. Thus, sometimes all of operon genes are transcribed and other times, only a subset. Within each promoter lies an *operator*, a short region of a regulatory DNA, used for binding of a special protein, called *regulator* that can either repress or induce transcription of an operon. The gene coding for the regulator protein does not have to be adjacent to the genes in the operon. Operons can be induced or repressed under different conditions or by different regulators. A *terminator* is sequence in DNA that signals to the RNA polymerase the termination of the transcription process.

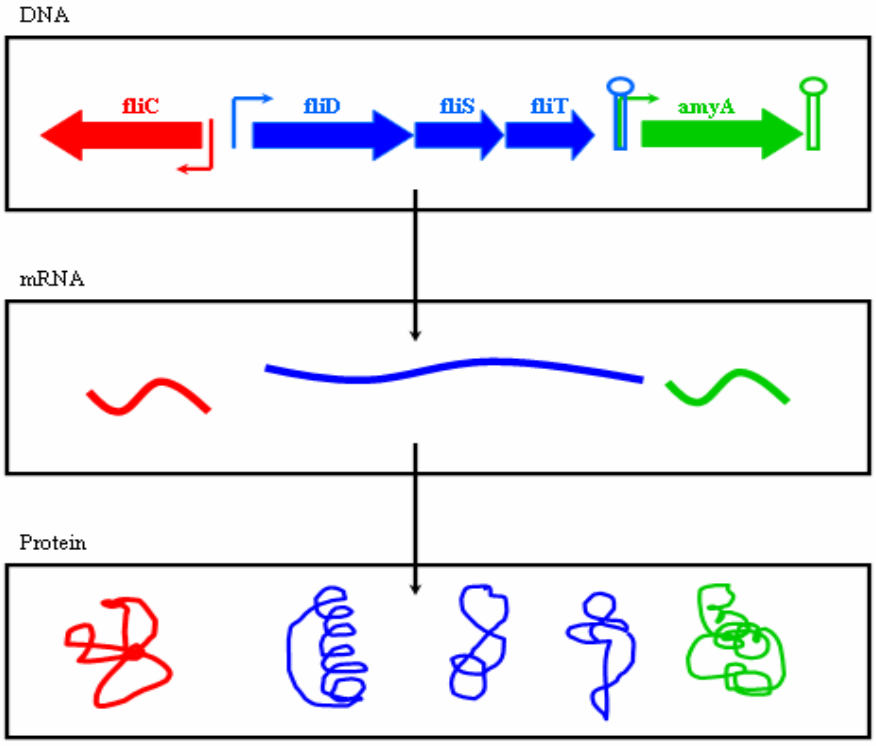

Figure 2. The fliDST operon in *E. coli*.

The region of the *E. coli* genome between base pairs 2,000,134 and 2,005,667 is shown. Three genes **fliD**, **fliS**, **fliT** encode the filament-cap protein of the flagellar apparatus, which facilitates the polymerization of endogenous flagellin at the tips of the growing filaments [30]. The operon's sigma-38 promoter sequence starts at base pair 2,001,841. The rho-independent terminator sequence is identified at positions 2,004,135-2,004,189. The operon terminator and the promoter of the **amyA** gene overlap. The **amyA** gene is transcribed and translated as a single unit, while the status of **fliC** is still unknown (no transcription

terminator has been identified) [30]. Genes on the forward and reverse DNA strands are shown as block arrows. Promoters are shown as bent arrows; terminators as cylinders with round tops.

The origin of the operons has not been established. Some researchers believe that they arise when new bacterial phenotypes are developing. Bacteria often exchange genes and keeping multiple genes with the same function as a cluster makes such an exchange easier. Others propose that operons originated in thermophilic organisms as the means of facilitating the association of functionally related products and protecting these products from thermal degradation [17]. Originally, operons were thought to exist in prokaryotic organisms only. In 1990, operons have been discovered in a eukaryote, nematode *Caenorhabditis elegans* (*C. elegans*). It has been suggested that about 15% of the genes in the *C. elegans* genome are organized into around 1,000 operons [7].

3 FUNCTIONAL GENOMICS

At the end of last century, a lot of effort has been put into finding the exact DNA sequence (sequencing) of different organisms. The human genome is now known and sequencing of microbial genomes has become a routine task. Computer programs have been developed to predict the location of genes in the genomes and assign a putative function to these genes. However, as Alberts points out "a complete DNA sequence of an organism would no more enable us to reconstruct the organism than a list of English words would enable us to reconstruct a play by Shakespeare. In both cases the problem is to know how elements in the DNA sequence or the words on the list are used" [1]. As a consequence, new ways of studying genetics appeared. The field of *functional genomics* attempts to reconstruct the patterns of gene expression and gene regulation in an organism. The methods in functional genomics consist of traditional gene expression studies, large scale microarray experiments, and bioinformatics. Each of the methods can be evaluated in terms of precision and cost (see Figure 3). The best results are expected when all three methods are used. For example, bioinformatics tools can be used to identify genes of interest for the microarray study and laboratory experiments to validate the results.

3.1 Traditional Expression Studies

Polycistronic (and monocystronic) transcripts are experimentally identified using Northern blot, reverse transcription-polymerase chain reaction (RT-PCR) and primer extension analysis [34]. These experiments can only be done with very few genes at a time. Also, most of the experimental work to identify operons concentrated on *E. coli* and *Bacillus subtilis* and hundreds of microbial genomes still remain uncharacterized. Traditional methods for gene expression studies, in general and operons, in particular are most accurate, reproducible and well-documented in the laboratory notebooks and publications. On the negative side, these methods are most costly, labor-intensive and slow.

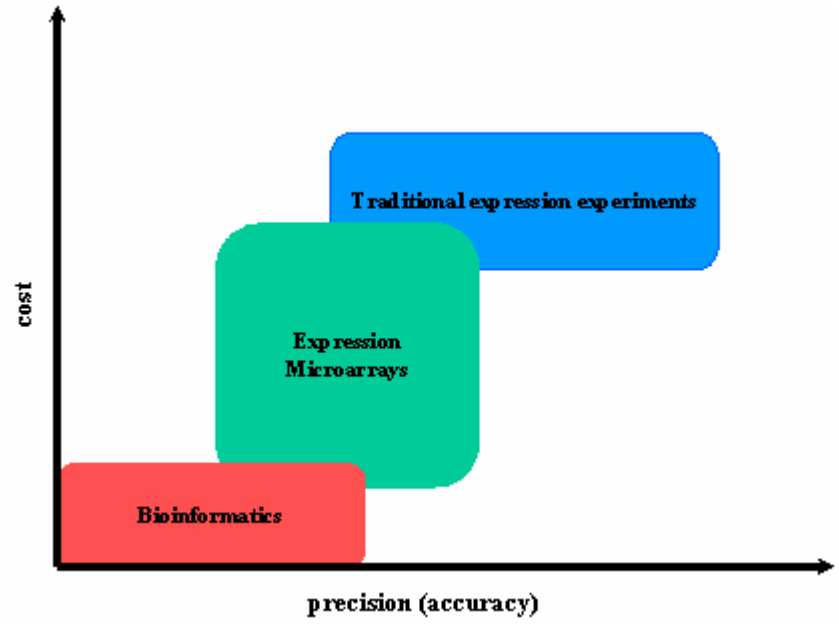

Figure 3. Accuracy of results versus cost of different methods in functional genomics.

Both, accuracy of bioinformatics predictions and their cost are low. Once bioinformatics programs are written they are used to predict gene expression and regulation patterns in newly sequenced organisms. Microarray expression studies are more expensive. The researchers are still trying to access the accuracy of microarray results. Traditional laboratory experiments give the most accurate results, but they are very slow and costly.

3.2 Microarray Technology

Microarrays were invented in Pat Brown's laboratory in 1995. They allow researchers to study which genes are expressed by detecting the amounts of mRNA in cells [31]. In a typical microarray experiment researchers can study thousands of genes at the same time. A single DNA microarray (or chip) is as small as a postage stamp with 10,000-100,000 distinct spots, organized as a matrix. Each spot on the chip contains a unique DNA sequence that can hybridize with either DNA or RNA isolated from cells grown under different conditions. Two major types of microarray technologies used today are cDNA arrays and the high-density oligonucleotide arrays [26].

3.2.1 The robotically spotted cDNA microarray technology

The robotically spotted cDNA microarrays are used to measure global gene expression levels in an experimental sample relative to a control sample, for example, in *E.coli* cells grown with and without glucose. In the cDNA microarrays, DNA fragments of coding sequences of interest, called *probes*, are amplified by a polymerase chain reaction (PCR) and then synthesized onto a glass slide in a high-density grid pattern. Messenger RNA is extracted from cells grown under two different conditions and copied to *complementary DNA* (cDNA) with the help of a reverse transcriptase enzyme. The cDNA from each sample is labeled with either green (Cy3) or red (Cy5) fluorescent dye.

Next, both labeled cDNA samples are mixed together and hybridized onto the slide. Because the samples are mixed, hybridization is competitive. This means that the density of green or red cDNA strands hybridized to microarray probes is proportional to the concentration of red and green cDNA molecules in the mixture [18]. Special scanners detect the amounts of both fluorescent labels in each microarray spot. The ratio of the two labels is determined and represented by color intensity, e.g. green, red, or yellow. This ratio of the expression color intensities indicates the ratio of the amounts of mRNA in the two samples. The color intensity of genes that are overexpressed in the sample labeled with Cy5 will be red, those that are underexpressed will be green. Genes that have equal expression levels in both samples will be yellow. The $\log_{10}$ of the ratio (called log-ratio) is a typical metric used in cDNA microarray data analysis. A log-ratio value close to zero indicates that gene in question is expressed in similar levels in the two conditions compared in the experiment [28].

3.2.2 The oligonucleotide microarray technology

In the oligonucleotide array method, developed by the Affymetrix [23], for each DNA region of interest a probe set is synthesized onto the slide. The *probe set* consists of 10-16 probe pairs. Each *probe pair* consists of a *perfect match* (PM) probe and a *mismatch probe* (MM), shown in Figure 4. Matching probes are 25 base pairs in length and exactly match the target sequence. The probes in the MM group have the same sequences as they counterparts in the PM set except for a complementary base in the middle (at position 13). The MM probes are mainly used as controls for nonspecific hybridization [22].

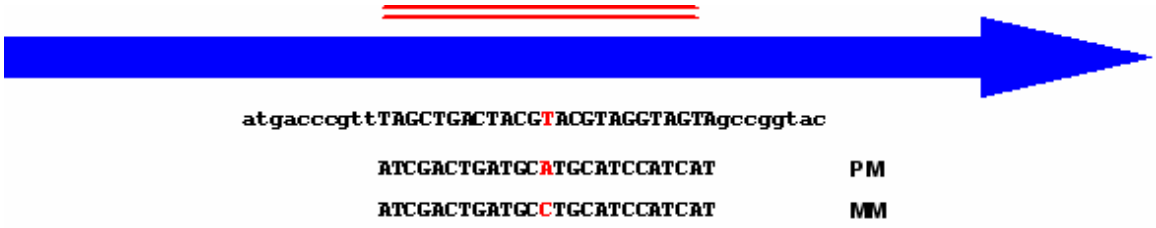

Figure 4. Probe pair for high-density oligonucleotide microarray.

One of the gene's probe pairs is shown. The perfect match (PM) probe is an exact complement of the target gene sequence of length 25 bp, the mismatch probe (MM) has one base pair mismatch in position 13 shown in red. Block arrow shows the direction of transcription.

The photolithographic masking technique, similar to fabrication of integrated circuits, is used to construct wafers of 40-400 oligonucleotide microarrays. Each wafer is a glass slide, on which probes are built one base at a time. All probes are constructed in parallel. After the first base is placed on the glass slide, the slide is exposed to ultraviolet light, then the next base is added and the slide is again exposed to ultraviolet light. The process is repeated until all 25 bases are added. At the end, the wafer is broken into the individual microarrays.

For the gene expression study, the mRNA is extracted from a single sample, copied to cDNA labeled with a fluorescent dye, and hybridized to the oligonucleotides on the slide. Each slide is scanned to obtain an image, in which each probe is represented by a small rectangular area. This area is called a *probe cell*. Each cell is made of several pixels. When the image file is analyzed, the intensity of each probe cell is computed as the 75th percentile of the intensities of all its pixels, excluding the pixels at the border of the cell. Thus, for each DNA region whose expression is measured, there are between 20 and 32 probe cells, i.e. 10-16 probe cells representing PM probes and 10-16 probe cells representing MM probes. After correction for background noise, the expression level of each gene is determined by averaging the amounts of labels from matching probes and correcting for less specific binding to the mismatching probes as shown in Figure 5 [22].

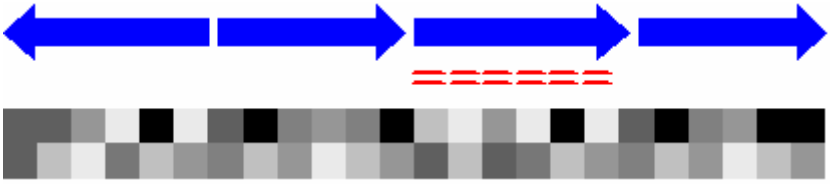

Figure 5. Determining the expression level of a gene in a high-density oligonucleotide microarray experiment.

Four genes and the corresponding two rows of microarray cells are shown. The PM probe cells are shown as the top row and the MM probe cells as the bottom row. To determine the expression level of the gene of interest, the number and the intensity of all PM and MM probes in a set are compared and a decision is made about whether the transcript is present or absent. The probe set in the example consists of six probe pairs. The amount of hybridization to MM probes is higher than to PM probes (the bottom row in this region contains larger number of darker cells than the top row) and most software packages will mark the expression call for this gene as "absent" or unreliable. Genes on the forward and reverse DNA strands are shown as block arrows. The probe pairs corresponding to probe cells are shown in red as parallel lines above the gene of interest.

A single oligonucleotide microarray assays more genes than robotically spotted cDNA microarray. However, to compare cells grown in two different conditions researchers need two separate oligonucleotide microarrays versus one cDNA chip. In general, experiments using oligonucleotide microarrays are more expensive and require more sophisticated data processing and interpretation, e.g. an inter-array normalization.

The main disadvantage of using microarray technology in functional genomics is that the experiments are only done under a handful of different conditions and the data from the microarray has very few replicates. Cost is the main hindering factor. Many microarray experiments under different conditions have to be performed to assure that the expression of all the genes is captured, but each experiment has to be replicated (repeated) to obtain statistically sound data and reduce the noise. Additionally, there are no agreed upon validation techniques. Current practice is to validate some of the microarray expression data using slow and labor-intensive traditional techniques described above, such as RT-PCR. This absence of standard computational and biological validation for microrarray studies lead to publications that could not be confirmed by the follow-up studies [22]. As the costs of microarrays go down, researchers are improving their results, but the number of different possible experiments is still limited due to the paucity of biological samples, collected under different conditions.

3.3 Role of Bioinformatics in Functional Genomics

Unlike previously described functional genomics methods, bioinformatics methods do not produce new expression values but rather make predictions based on already available data, such as DNA sequence, gene maps, protein structures, gene expression, metabolic pathways and vast literature in the PUBMED database [4].

Computational methods can help reduce the gap between the genomic data coming out of the sequencing projects and experimental studies of transcriptional regulation. Among these methods, operon prediction is very important not only because it provides the prediction about which genes are co-regulated, but also because the prediction of other regulatory elements, such as transcription binding sites, promoters, etc., often relies on operon predictions [38]. Additionally, operon prediction can improve computer annotation of genomes and help infer possible function for uncharacterized proteins as suggested by Strong et al. [33], since genes in the putative operon are expected to have similar function.

In what follows, we will summarize different bioinformatics techniques to predict operons in prokaryotic organisms.

4 BIOINFORMATICS APPROACHES TO OPERON PREDICTION

The very first operon predictors used the fact that genes are flanked by the transcriptional regulatory signals, such as promoters and terminators. Since genes in an operon are transcribed as a group, no regulatory signals should be present between the genes within an operon. Yada et al. constructed a hidden Markov model (HMM) and trained it with 205 known N-terminal start sites, 441 σ^{70}–dependent promoters and 145 ρ-independent terminators [41]. This HMM-based method exactly predicted 60% of known transcription units in the *E. coli* genome. Two factors hinder the success of this method. First, the HMM depends on the availability of well-studied promoter and terminator sequence patterns in different organisms. Unfortunately, only few such patterns are known. Second, some operons have internal promoters and terminators, i.e. regulatory signals are present between the genes transcribed into a monocistronic RNA [16]. Due to these limitations, this method did not gain wide popularity, although other predictors can be coupled with promoter and terminator signals recognition [10].

Current prediction methods use either experimental data (such as microarray expression data) or genome sequence. Among the methods that use only annotated genome sequences, two approaches have been cited most frequently in the literature. The first method predicts operons from the distances between adjacent genes [29]. The second predicts operons based on the conserved gene order and orientation in multiple genomes [16]. Both approaches combine computational and statistical techniques to assign each gene pair a probability of being in an operon. In evaluating published results, it is very difficult to make comparisons (see Table 1). Most publications report gene pairs that are

most likely to be co-transcribed, while others determine entire transcription units, monocistronic as well as polycistronic. Additionally, there is no agreed upon metric for evaluating the predictions. One of the three: accuracy, sensitivity and specificity, is typically reported.

Table 1. Summary of operon prediction methods.

Year	Author	Results	Data source	Gene pairs/Operons
1999	Yada et al	60% accuracy	Genome annotation and known regulatory elements	Transcription units
2000	Salgado et al	88% accuracy	Genomic sequence data	Gene pairs
2002	Sabatti et al	88% sensitivity and 88% specificity	cDNA microarray expression data	Gene pairs
2002	Tjaden et al	99% sensitivity and 63% specificity	Oligonucleotide microarray expression data	Gene pairs
2003	Strong et al	89% sensitity	Genome annotation	Transcription units
2005	Price et al.	88.3% sensitivity and 79.9% specificity	Genome annotation and expression	Gene pairs

4.1 Distance-based operon prediction

A distance-based operon prediction technique was first described by Salgado et al. [29]. In their analysis, the authors used the annotated genomic sequence of *E. coli* K-12 and the data set of 361 experimentally verified transcription units. Transcription units were divided into two subsets, polycistronic (237 genes) and monocistronic (124 genes). The authors compared the distributions of distances between gene pairs within operons with those found at the borders of transcription units that are transcribed in the same direction. The gene pairs at the borders of transcription units consist of one gene that belongs to that transcription unit and one that does not. From the total number of adjacent gene pairs, 572 pairs were found in operons and 346 pairs at the borders of transcription units. Distances between two consecutive genes (intergenic distances) were calculated as follows:

$$distance = gene_{i+1}(start) - gene_i(end + 1)$$

The *start* and *end* refer to coordinates of the genes. The subscripts of the genes represent the order in which they occur in the genome sequence. The authors found that the distribution of distances between adjacent genes in operons differs from the distribution of distances between adjacent genes at the boundaries of transcription units. The former has clear peaks at short distances, while the later appears almost flat. The authors compute the log-likelihood of a pair of adjacent genes to be in the same operon as a function of the distance between genes:

$$Log - Likelihood\,[\text{distance}] = \log \frac{N_{op}[\text{distance}]/TN_{op}}{N_{nop}[\text{distance}]/TN_{nop}},$$

where N_{op} and N_{nop} are pairs of genes in operons and at transcriptional boundaries, respectively, separated by [*distance*] base pairs (in 10 bp intervals), and TN_{op} and TN_{nop} are the total number of gene pairs in operons and at the transcription unit boundaries, respectively. Contiguous gene pairs are said to belong to the same operon if their log-likelihood score is above some given threshold. Using the log-likelihood scores, Salgado et. al. predicted around 630 to 700 operons in *E. coli* [29].

Moreno-Hagelsieb and Collado-Vides [24] provided evidence that the distance-based method can be used to predict operons in any prokaryotic genome. They verified that the *E. coli* log-likelihood scores can differentiate between gene pairs within operons and at the transcription unit boundaries in a data set of 100 experimentally confirmed operons of *B. subtilis*. The *B.subtilis* data set consisted of 310 gene pairs in operons and 123 gene pairs at transcription unit boundaries. The results were then used to determine the sensitivity (true positives detected per known gene pairs within operon) and specificity (true negatives per known gene pairs at the transcription unit boundaries).

Sensitivity, specificity and accuracy curves under different log-likelihood thresholds were plotted for *E. coli* and *B. subtilis* (Fig. 1. in [20]). The authors concluded that since the results were almost identical in both genomes, the distance-method works equally well in either one of these evolutionarily distant organisms. The estimated accuracy of operon prediction is 88% in *E.coli* and 82% in *B.subtilis*. Furthermore, the authors analyzed the frequency distributions of distances between all gene pairs transcribed in the same direction in 50 prokaryotic genomes and determined that almost all genomes show the characteristic peak between -20 to 30 bp, with the prevalent overlap of 4 bp. The frequency distribution of intergenic distances in *E.coli* operons shows similar characteristic peaks [24]. Among the examples, where the intergenic distances do not follow *E.coli* distribution are two Cyanobacteria, *Nostoc* sp. PCC 7120 and *Synechocystis* sp. PCC 6803. Both genomes exhibit very low peaks, and consequently, have very few predicted operons. Aside from annotation problems, this could be an indication that these two genomes either contain very few operons or that there is a different distance distribution pattern in Cyanobacteria than in other organisms. A later study by Rogozin et al. found that intergenic distances between genes in operons vary in different species and, thus, the distance model built from *E. coli* data may not always be as effective as previously thought [39].

Strong et al. [33] evaluated distance-based operon prediction in the pathogenic *Mycobacterium* tuberculosis H37Rv. The gene orientation and distances between them were used to determine functional links between genes. Two genes were considered functionally linked if they were transcribed in the same direction and the nucleotide distance between them was less or equal to a predetermined distance threshold (see Figure 6). Multiple genes are functionally linked if they were all transcribed in the same direction and all have intergenic distances less than or equal to a threshold.

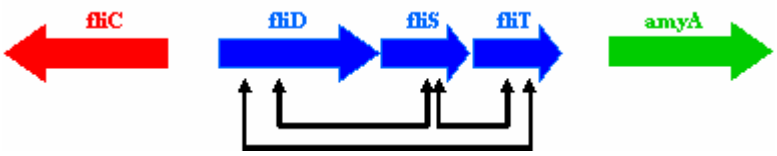

Figure 6. Functional linkages as described in Strong et. al. [6].
Genes **fliC** and **fliD** are not linked because they are transcribed in the opposite directions. Genes **fliT** and **amyA** are not linked because the distance between them is greater than the threshold. The distance between genes **fliD** and **fliS** and **fliS** and **fliT** is less than a predetermined threshold resulting in two functional links. Since the distances between both **fliD/fliS** and **fliS/fliT** pairs are less than the threshold we have a total of three functional links.

At a distance threshold of 0 bps, the authors reported 1,279 genes (25% of *M*. tuberculosis genes) and 2,034 functional links in *M*. tuberculosis genome. The expectation is that a large percentage of these genes would be true operon linkages. At the 100 bps threshold, ~75% of the genes had one or more links. To access the accuracy of their predictions the authors used an updated *E.coli* transcription units dataset obtained from RegulonDB [30] as well as a keyword recovery scheme described in [25]. Applying distance threshold of 0 bps to determine functional links in the dataset of known transcription units in *E.coli*, Strong et al. that 89% of these correspond to true operon links.

Keyword recovery means that identical keywords are found in the SWISS-PROT [3] annotations for proteins connected by the functional link. The authors reported a 50% keyword recovery for two linked genes separated by 0 bp, i.e. half of the total keywords in the SWISS-PROT annotations of these two genes are shared between the linked pairs. As the distance between linked genes increases from 0 to 100 bps, keyword recovery drops to about 45%, indicating that some of the gene pairs may not be true operon links.

The keyword recovery was also evaluated at different thresholds for the combined intergenic distances between genes of a directon. The authors linked all gene pairs in the same directon and assigned to this link a value equal to the sum of all the intergenic distances in that directon. The keyword recovery of links with cumulative intergenic distances less or equal to 150 bps is 34-52%, steadily decreasing as the cumulative intergenic distances increase above 150 bps.

Since the absolute threshold values are needed for linking genes, the authors attempted to create a distance profile (frequency distribution) that would be indicative of the distances between genes in operons. To create such a profile, two separate data sets were generated from all adjacent genes in the same orientation in *M*. tuberculosis genome. The first data set consisted of gene pairs that were functionally linked by either Rosetta Stone [25], Phylogenetic Profile [36] or Conserved Gene Neighbor [25], [35] method. The second data set contained gene pairs not linked by any of the three methods. Frequency of distances at 10 bp intervals was computed for both data sets. The mean of the linked data set was 27 bps and the mean of the non-linked gene pairs was 94 bps. The χ-square statistical test verified with 95% confidence level that these two samples are different from each other. The distance profile determined by the authors was in

agreement with the frequency distribution of the distances between gene pairs within operons in the *E.coli* genome. The authors extrapolate that at the distance thresholds of 50 bps and 100 bps, more than 80% and 90% of true operons, respectively, would be recovered in the *M*. tuberculosis genome.

It should be pointed out that the accuracy of operon prediction using intergenic distances depends on the accuracy of genome annotation. Three problems in genome annotations can affect the outcomes: incorrect start codons, missing genes or including non-existing genes, which, if corrected, can improve prediction of operons using distance-based method. Despite its dependence on genome annotation, the distance-based method has been widely accepted and is frequently used in integrative predictors [10], [33].

4.2 Prediction based on the conservation of gene order

The proponents of the conservation method base their analysis on the assumption that gene clusters shared by two genomes assert with high probability that these gene clusters are indeed operons. Analysis of completely sequenced microbial genomes and several strains of the same organism revealed that some genes tend to be located together even in distantly related organisms while others undergo rearrangement in two strains of the same bacteria. The authors mention for any two genomes, four different explanations that can account for the conservation of the gene pair (two adjacent genes separated by ≤200 bp) [16]:

1. genes belong to the same operon;
2. genes were inherited from a common ancestor and have maintained their adjacent locations;
3. a later gene transfer occurred whereby the gene pair was moved from one genome into the other;
4. the conserved genes are adjacent by chance.

The authors use the first explanation to predict operons in *E. coli*. If a gene pair is conserved in multiple genomes it is most likely to belong to an operon.

4.3 Prediction based on the expression data

Sabatti et al [28] studied to what extend microarray gene expression can be used to predict operons. The authors utilized the gene expression data from 72 *E.* coli cDNA microarray experiments and a training set of 257 known operons and 102 single transcriptional units. A Bayesian classifier was constructed from the 604 known operon pairs and 151 known non-operon pairs, i.e. either gene pairs containing either a single-gene transcript or genes in front of known promoters. The bootstrap technique was used to assess the variability of data [28]. A series of bootstrap samples was created from the observed data sets using sampling with replacement. Statistics of interest (i.e. standard deviation of average correlation of gene pairs) were computed for each bootstrap sample and the distribution of statistics across all samples was taken as the representative of the distribution of the statistics across experiments. The classifier was constructed from a

prior probability of being in an operon for each adjacent gene pair and a distribution for the correlation of expression values in known operons and known non-operon gene pairs. The classifier predicted operon gene pairs with 82% sensitivity and 70% specificity. The addition of the intergenic distances improved the results to 88% sensitivity and 88% specificity. For comparison, the distance-only classifier had 84% sensitivity and 82% specificity.

To assess the validity of using log-ratios of the microarray expression values for prediction of operonic gene pairs, the authors compared expression correlations from known operons, known non-operons and a set of 200 randomly selected gene pairs. The results showed that gene pairs in known operons (mean value 0.632, bootstrap standard deviation 0.017) have higher correlation than known non-operon gene pairs (mean 0.177 and bootstrap standard deviation 0.027). Known non-operon genes have higher correlation coefficients than randomly selected gene pairs (mean and standard deviation not reported). However, the difference between known operon and known non-operon gene pairs is not statistically significant. The surprising results are explained as follows. First, the mRNA is an unstable molecule and its degradation could produce different correlation patterns within the same operon. For example, the correlation between the first two genes in an operon could be closer to 1 than the correlation between the last two genes. Second, operons often contain internal promoters that are active under certain conditions. As a result, different genes of the operon can be transcribed in different experiments. Third, the microarray experiments are not designed with the goal of assessing the global regulatory network of *E. coli* but rather the activity of a subset of all the genes. For example, in an experiment measuring the expression of genes of *E. coli* growing on plus/minus sucrose, only a subset of genes will show changes in their expression values. The expression of the majority of the genes will remain the same, thus resulting in the correlation ratios close to zero for either operon or non-operon gene pairs. Lastly, the variability in microarray measurements can be very high due to errors. These results suggest that care should be taken when microarray data is used for attempting to construct a global picture of organism's regulatory network. Sabatti et al. propose a way to increase the information content of their data set by eliminating genes whose expression values do not show perturbation beyond the noise level. This, however, drastically reduces the number of gene pairs in the operon training set.

Tijaden et al. [34] used expression data from 28 Affymetix (2 replicates for 14 different conditions) high-density oligonucleotide microarrays. These arrays measure expression of both coding sequence and intergenic regions (the segment of DNA between two adjacent annotated coding sequences). The expression data was used to construct an HMM-based predictor of 5′ untranslated regions and operon gene pairs. The results of the study were validated against experimentally known transcripts. The authors report 99% specificity and 63% sensitivity in predicting operon gene pairs. *E.* coli oligonucleotide arrays used in this study contain 295,936 probes or 147,968 (295,936/2) probe pairs. Each probe pair consists of a perfect match (PM) oligo and a mismatch (MM) oligo. The PM oligos are sequences of 25 nucleotides exactly matching the target sequence. Each MM oligo is exactly the same as its corresponding PM except for the base in the middle (base 13) that is complementary to the target sequence. Every coding sequence were assayed by a probe set (~15 probe pairs) and an expression vector $\Theta = (\Theta_1\Theta_2 \ldots \Theta_{28})$,

was computed calculated using expectation-maximization algorithm. Here, Θ_i is the expression index of a coding sequence in experiment *i*. Every intergenic region of at least 40 base pairs was assayed in both orientations by a probe set. Since the intergenic distances between operon genes are very short (shorter than 40 bp), only 154 intergenic probe sets could be used in the positive operon data set. An expression vector for each probe set was calculated using expectation-maximization. Reasoning similar to Sabatti was used in the analysis of the expression values of the intergenic probe sets. The correlation between probe sets assaying intergenic regions within operons and the genes on either side of that intergenic region should be close to 1. Likewise, the correlation of intergenic probe sets with the genes not within an operon should be close to 0.

The authors constructed a 2-state Hidden Markov Model (HMM) and tested it with the *E.coli* genome. They report 99% sensitivity and 63% specificity in predicting gene pairs belonging to operons.

5 PROBLEM DOMAIN AND PREDICTOR OVERVIEW

The aim of this project is to develop and evaluate a predictor of transcription units in the *E.coli* genome. More precisely, we want to identify all transcription units in *E. coli*. Three points are important to mention. First, the definition of transcription units includes genes transcribed individually as well as operons (this is consistent with the approaches described in the literature [30]). Second, our definition of transcription units refers to genes only and does not include the transcriptional regulatory signals, such as promoters or terminators. Third, our definition does not take into account alternatively transcribed operons, or subsets of genes that are transcribed differently under different conditions (see Section 2).

Our approach takes into account gene location and orientation and the expression values from the microarray experiments and outputs the transcription status of each gene. This is done in a two-phase process. In the first stage all gene pairs in the organism are assigned to either operon or non-operon class with a Bayesian classifier described in the next section. In the second stage, adjacent gene pairs classified as operon gene pairs are joined into longer transcription units in the manner similar to functional linkages described in Section 4.1. After the predictor is trained and evaluated on a test data set, we will use it to predict the transcription units in the whole *E. coli* genome.

We train and test our predictor on the data from the free-living bacterium *E. coli*. The decision to use this organism in this work is due to the fact that *E. coli* is a model organism and many *E. coli* experimental studies in gene function and gene regulation have been performed by the researchers around the word. A number of databases have been created to store the results of these experiments. The most comprehensive database of transcription units, RegulonDB, contains information about 1,254 experimentally verified transcription units. Even though *E. coli* is one of the best studied prokaryotic organisms, there are still thousands of undiscovered transcription units [9] making it an interesting case study not only for training of the predictor but also as a target of the predictor.

5.1 The Bayesian classifier

Classification is a technique to assign objects to a particular class based on some distinct features. In a classical classification problem, we are given a training set of features along with class labels and we want to output a classifier. A classifier can be viewed as a set of discriminant functions, one for each class [14]. Given an unclassified data, the classifier will assign it to a class whose function outputs the maximum value. One such classifier is the Bayesian classifier originating from the Bayesian theory of probability. *Bayesian classifiers* are probabilistic models, robust to data noise and missing values [13]. These classifiers are simple, fast in learning and classifying and do not require a lot of storage space. They show very good results even when the sample size is small. These classifiers are sometimes called naive Bayesian classifiers. The term "naïve" refers to the assumption that given the class, the features are independent. In practice, however, the features are rarely independent given a class. This later fact was the reason for which naïve Bayesian classifiers were largely ignored by the machine learning community up to about the 1980s. The interest in the Bayesian classifiers started to pick up after several articles were published showing that they can perform well in many complex areas, including those where there are clear attribute dependences [13]. Bayesian classifiers have been used in document classification and in spam reduction. In bioinformatics, Bayesian classifiers have been successfully applied in many domains. In 2005, about 50 bioinformatics research articles mention naïve Bayesian classifier [13]. Our work is largely influenced by the article by De Hoon et al [8].

The following is a formal probabilistic model of the naïve Bayesian classifier. Given a set of features (or variables), $X = \{X_1, X_2,\ldots,X_n\}$, we want to determine the posterior probability for the event C_i among a set of possible outcomes or classes $C = \{C_1,C_2,...,C_m\}$. Using Bayes rule:

$$p(C_i \mid X_1, X_2,..., X_n) = \frac{p(C_i)\,p(X_1, X_2,..., X_n \mid C_i)}{p(X_1, X_2,..., X_n)},$$

where $p(C_i|X_1,X_2,\ldots,X_n)$ is the *posterior probability*, i.e., the probability that X belongs to C_i. Also, note that the denominator in the equation is not dependent on C and remains constant for all classes. The numerator is a joint probability and can be rewritten as follows (by the definition of conditional probability):

$$\begin{aligned} p(C_i)p(X_1,X_2,\ldots,X_n|C_i) &= p(C_i)p(X_1|C_i)p(X_2,\ldots,X_n|C_i,X_1) \\ &= p(C_i)p(X_1|C_i)p(X_2|C_i,X_1)p(X_3,\ldots,X_n|C_i,X_1,X_2) \\ &= p(C_i)p(X_1|C_i)p(X_2|C_i,X_1)\ldots p(X_n|C_i,X_1,X_2,\ldots X_{n-1}) \end{aligned}$$

By the definition of independence, every feature Xi is independent of every other feature X_j for all $i \neq j$ and $p(X_i|C_i,X_j) = p(X_i|C_i)$. The joint model can now be rewritten as:

$$p(C_i \mid X_1, X_2,..., X_n) = \frac{p(C_i)\,p(X_1 \mid C_i)\,p(X_2 \mid C_i),..., p(X_n \mid C_i)}{p(X_1, X_2,..., X_n)}.$$

To construct a Bayesian classifier, we combine the probability model derived above with a decision model. Two approaches have been commonly used in the literature.

1. *Maximum a posteriori (MAP)* decision rule assigns an unseen example to a class with the highest posterior probability. In other words, we define a function *classify* as follows:

$$classify(x_1, x_2, ...x_n) = \arg\max_c \frac{p(C_i = c)\prod_i^n p(X_i = x_i \mid C_i = c)}{p(X_1, X_2, ...X_n)}.$$

   The denominator is often ignored, since it remains constant for all possible classes.

2. The second decision rule can be used when we only have two mutually exclusive classes. It assigns an unseen example to a class if the posterior probability exceeds a predetermined threshold value, for example, $p_{threshold}$. In other words, we define a function *classify* as follows:

$$classify(x_1, x_2, ...x_n) = c \quad if \quad p(C_i = c)\frac{\prod_i^n p(X_i = x_i \mid C_i = c)}{p(X_1, X_2, ...X_n)} > p_{threshold}$$

In summary, every Bayesian classifier requires two important parts. The first part is the *prior distribution*, which can be derived from previous data or any other relevant information. It is a subjective measure and reflects investigator's knowledge about the system under study. In the equation above, the prior is $p(C_i=c)$. Often uninformative prior (p=0.5) is used. The second part deals with the type of data being analyzed and results in a *likelihood function* (the second term in the equation above).

In constructing our Bayesian classifier, we collected and used the following information about the *E.coli* genome:

1. Distribution of the sizes of known transcription units.
2. Distribution of the distances that separate genes in the operon gene pairs.
3. Distribution of the distances that separate genes in the non-operon gene pairs.
4. Distribution of the Pearson correlation coefficients between genes in the operon gene pairs.
5. Distribution of the Pearson correlation coefficients between genes in the non-operon gene pairs.

We then build the statistical model as follows. Let $X = \{d,r\}$ be the set of features, where d is the distance between adjacent genes and r is the correlation coefficient of the

expression values of two adjacent genes. Let $C = \{OP, \neg OP\}$ be the set of two classes, operon class and non-operon class. The posterior probability of a pair being in operon is shown below.

$$p_{posterior}(OP \mid d, r) = \frac{p(OP)\,p(d \mid OP)\,p(r \mid OP)}{p(OP)\,p(d \mid OP)\,p(r \mid OP) + (1 - p(\neg OP))\,p(d \mid \neg OP)\,p(r \mid \neg OP)}.$$

Using uninformative prior, i.e. $p(OP) = p(\neg OP) = 0.5$, and the second decision rule described above, we classify a given gene pair as an operon gene pair if the pair's posterior probability is greater than some threshold. Otherwise, we classify it as a non-operon pair. We determine $p_{threshold}$ during the evaluation phase (described below) as the value that maximizes the accuracy of our Bayesian classifier.

5.2 The Linkage Phase

After the gene pairs have been classified, the linkage step builds longest possible continuous runs from operon gene pairs. We join together adjacent operon gene pairs until either non-operon or unclassified pair is discovered. At the end of this step we will output a collection of monocistronic and polycistronic transcription units.

5.3 Empirical Evaluation

We evaluate our Bayesian classifier using a leave-one-out analysis [8]. In this analysis, we repeatedly remove one of the gene pairs from the training data set, train our predictor with the remaining data and then classify the removed gene pair. The analysis is repeated for different $p_{threshold}$ values. We compute false positives (FP) and true positives (TP) fractions as shown below (where N stands for the number of occurrences, *positives* refers to operon gene pairs and *negatives* to non-operon gene pairs) [43] and generate a receiver operating characteristic curve (ROC).

$$FP = \frac{N_{\text{false positives}}}{N_{negatives}}$$

and

$$TP = \frac{N_{\text{true positives}}}{N_{positives}}$$

An ROC curve in our work is a plot of FP versus TP fractions under different thresholds for the posterior probability. A predictor that randomly assigns genes to transcription units would have equal FP and TP rates and will appear as a diagonal line on the plot, i.e. FP rate = TP rate [43]. This means that equal number of true and false positives is found at each threshold for the posterior probability (see Figure 7). An ROC curve that is well above the diagonal random line represents a significant predictive power and a curve below the diagonal suggests that the predictor consistently gives wrong results. The latter can be fixed by simply inverting the classifier's decisions [43].

The accuracy of the predictor is measured as the area below the curve. An optimal predictor will have an area of 1.

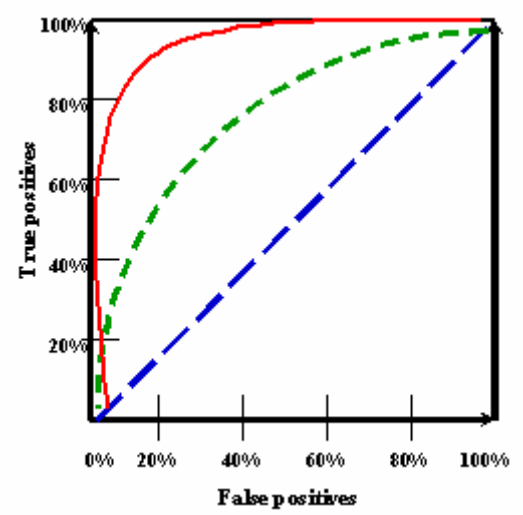

Figure 7. Interpreting an ROC curve.
A hypothetical receiver operating characteristic curve is shown. The percentage of false positives is shown on the x-axis and the percentage of true positives is shown on the y-axis. A straight 45°line from the bottom left to top right corner represents a predictor that randomly classifies the gene pairs. This line (shown in blue) is called the "line of no-discrimination" [43]. The curve above the no-discrimination line (shown in green) represents a predictor that is able to discriminate between true and false positives. Two ROC curves are shown in red (solid) and green (dotted). The predictor shown by the solid red curve is better because its accuracy or the area under the curve is larger. If the ROC curve appears below the no-discrimination line, the predictor consistently gives incorrect results.

Besides assessing the accuracy of the joint Bayesian classifier, we also evaluate the predictive power of each of the three observations. We perform this analysis in order to evaluate the contribution of each of the data source to the predictor's overall outcome. For these tests, we repeatedly leave out one of the observations from the Bayesian classifier and generate corresponding ROC curves. This analysis is of great value if we were to use the predictor with other organisms for which one of the observations is not available.

6 CLASSIFIER PARAMETER ESTIMATION

The operon predictor parameters were estimated from the data from three sources.

1. ***E. coli* K-12 MG1655 sequence and annotation.** The GenBank file (accession number U00096, September 8, 2006 update) containing whole genome sequence and annotation was downloaded from the National Center for Biotechnology Information (NCBI) web server [4]. The file is found in Appendix A. The annotation includes the location on either forward or reverse strand and the beginning and ending positions of genes.

2. ***E. coli* transcription units.** The dataset includes description of both monocistronic and polycistronic experimentally verified transcription units in the *E. coli* genome. An ASCII text file (September 28, 2006 update) containing transcription unit name, size, orientation and gene names was downloaded from the RegulonDB web server [30]. The file is found in Appendix B.

3. ***E. coli* gene expression.** Gene expression and probe responses from 28 Affymetrix oligonucleotide microarray experiments were obtained from Tjaden et al [34]. For each experiment, 2 replicates are reported along with standard errors. The file is found in Appendix C. The conditions of experiments are summarized in Table 2.

Table 2. Summary of the conditions of the microarray experiments.

Experiment Number	Experiment Description
1	MM + glucose, aerobic, mid log phase
2	MM + glucose, aerobic, midway between log phase and stationary
3	MM + glucose, aerobic, early stationary phase
4	MM + glucose, aerobic, late (24 hours) stationary phase
5	MM + glycerol, aerobic, mid log phase
6	MM + glucose, first time point during switch anaerobic -> aerobic (15 min)
7	MM + glucose, second time point during switch anaerobic -> aerobic (30 min)
8	MM + glucose, third time point during switch anaerobic -> aerobic (60 min)
9	MM + glucose, aerobic, temperature = 42 degrees, mid log phase
10	MM + glucose, aerobic, temperature = 20 degrees, mid log phase
11	MM, aerobic, starvation (withdrawing of glucose at mid log phase)
12	Broth, aerobic, mid log phase
13	Broth, anaerobic, mid log phase
14	MM + glucose, aerobic, mid log phase (replicate of Exp #2)

Since the first part of the predictor, the Bayesian classifier uses the knowledge from the three data sources to classify each gene pair in the *E. coli* genome as either operon pair or non-operon pair, we start with some definitions. A *gene pair* refers to two genes located next to each other in the genome (see Figure 8). The gene pairs located within operons of size greater than one are called *operon gene pairs*. There is no consensus in the literature in how to define non-operon gene pairs. For the purpose of this work, a *non-operon gene pair* is defined as one of the following:

a) adjacent genes transcribed in opposite directions or neighboring genes located on opposite strands in the genome,
b) adjacent genes that include first gene in the transcription unit and the gene upstream of it, and

c) adjacent genes that include last gene in the transcription unit and the gene downstream of it.

Figure 8. Operon and non-operon gene pairs.
The region of the *E. coli* genome between base pairs 1,999,094 and 2,006,114 is shown. A gene pair consists of two genes located next to each other in the genome, for example **fliA** and **fliC**, **fliC** and **fliD**, etc. Three genes, **fliD**, **fliS**, and **fliT** are co-transcribed and two gene pairs **fliD/fliS** and **fliS/fliT** are, therefore, referred to as operon gene pairs. Two gene pairs, **fliC/fliD** and **amyA/yedD** consist of genes on opposite strands and are referred to as non-operon genes (case a) above). Case b) is represented by gene pairs **fliD/fliC**, **fliC/fliD**, **fliA/fliC**, and **amyA/fliT**. Case c) is represented by **fliC/fliA**, **fliT/amyA**, and **amyA/yedD**.

In what follows, we summarize the results of our genomic analyses of the three data sets.

6.1 Genomic Analysis of the E. coli bacterium

The genome *Escherichia coli* bacterium strain K12 substrain MG1655 was sequenced in 1997 [6]. The genome consists of a single double-stranded circular chromosome of length 4,639,675 bps. The *E.coli* Genbank record contains annotation of 9,033 features. A *feature* in the GenBank annotation refers to a region in the DNA that has some known characteristics, such as gene, repeat, etc. Table 3 summarizes all annotated features in the *E. coli* genome. In the current GenBank file, 4,488 features are annotated as genes. Of these, 2,218 genes (~49.42%) are on the forward strand and 2,270 (~50.58%) on the reverse strand. The average length of the protein coding gene is 949 bp. Two genes are only 45 bp in length: trp operon leader peptide (accession number b1265) and phenylalanyl-tRNA synthetase operon leader peptide (b1715). The longest (7,104 bp) annotated gene is adhesin (b1978).

From the parsed Genbank annotation, a data set was constructed containing 4,488 gene pairs. The Table 4 summarizes the number of gene pairs in each orientation category. Since convergently or divergently transcribed genes cannot be transcribed into a single transcription unit, 29.5% of the gene pairs in the *E. coli* genome can be classified as non-operon gene pairs without further analysis.

The average distance between genes in the *E. coli* is 104.24 bps, the longest is 1,604 bps and the longest overlap between two genes is of size 8,622 bps. The frequency distribution of distances between gene pairs is shown in Figure 9. Although the majority of the distances fall in the range between -500 and 500 bps, ~4% of gene pairs have intergenic distance greater than 500 bps and ~0.5% overlap by more than 500 bps. The long overlaps reported here are artifacts of the genome annotation. Upon closer examination of the gene pairs with long overlaps, we determined that these are annotated as alternative pseudo-genes. *Pseudo-genes* are non-functional stretches of the DNA that resemble known genes. Often the same stretch of the DNA resembles multiple known genes and each match is annotated individually. Since Genbank format does not have a

separate feature type for pseudo-genes, they are reported as /gene or even /CDS (see Table 3). Since these genes are never expressed in the cell, they will not have microarray expression values and they will be excluded from the analysis. The status of these genes will be reported as *unclassified* by our predictor.

Table 3. Summary of features in the *E. coli* Genbank file (accession number U00096)

GenBank feature	Description	Number of occurrences
/source	Whole genomic sequence	1
/gene	All genes, including protein coding, tRNA, rRNA, pseudo-genes and other RNA coding genes	4,488
/CDS	Coding sequence, including protein coding genes and pseudo-genes	4,331 (4322 protein coding, 8 pseudo-genes and 1 frame-shift)
/rRNA	Genes coding for ribosomal RNA	22
/tRNA	Genes coding for transfer RNA	86
/misc_RNA	Other RNA coding genes, e.g. small RNA	49
/repeat_region	Genomic repeat	44
/misc_feature	Insertion sequences and other genomic regions with known function	11
/rep_origin	Origin of replication	1
Total		**9,033**

Table 4. Summary of pair orientation.

Orientation	Description	Number of pairs
Convergent	→←	661
Divergent	←→	661
Reverse strand	←	1,609
Forward strand	→	1,557
Total pairs		4,488

Based on the literature review (see Section 4), operon genes tend to be separated by shorter distances than non-operon genes. We would then expect to find the majority of the genes among 3,539 gene pairs separated by the distances less than 200 bps or overlapping by fewer than 100 bps. The frequency distribution of the gene pair falling in this category is shown in Figure 10. About 41% of these distances are between -20 bps and 20 bps. The two most frequently occurring distances are overlaps of 4 and 1 bps

found in 310 and 169 gene pairs, respectively. In both cases, these represent overlaps between the stop codon of the first gene and the start codon of the next gene.

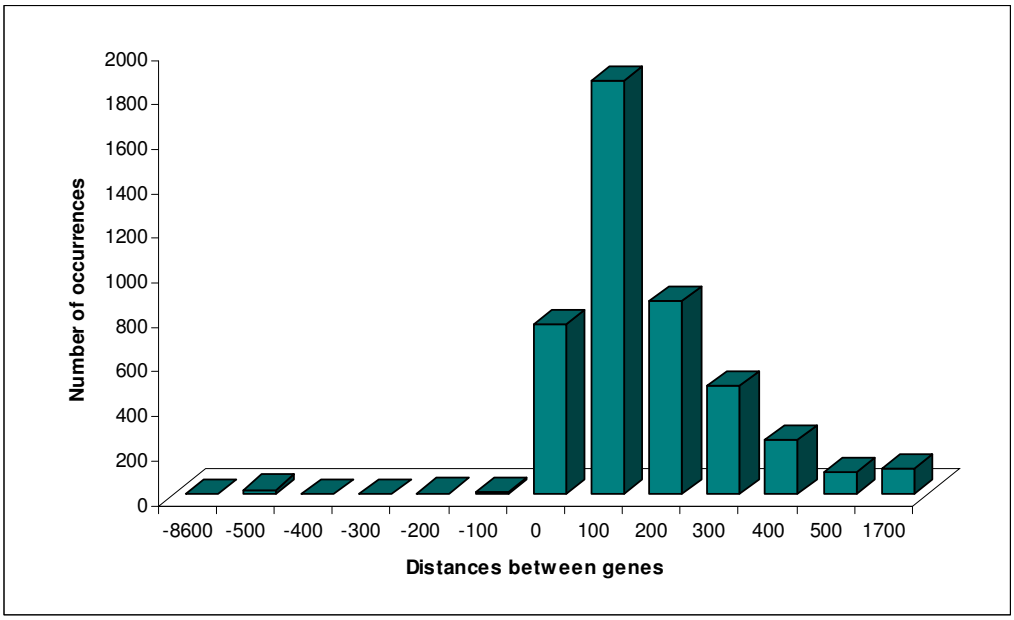

Figure 9. Frequency distribution of distances between gene pairs in 100 bps increments.

Distances between two adjacent gene pairs in the *E. coli* genome are shown on the x-axis and the corresponding number of occurrences (or counts) on the y-axis. The distances in the range [-500, 500] are shown in 100 bp increments. The intervals [-8600, -500] and [500, 1700] include all distances falling within these ranges.

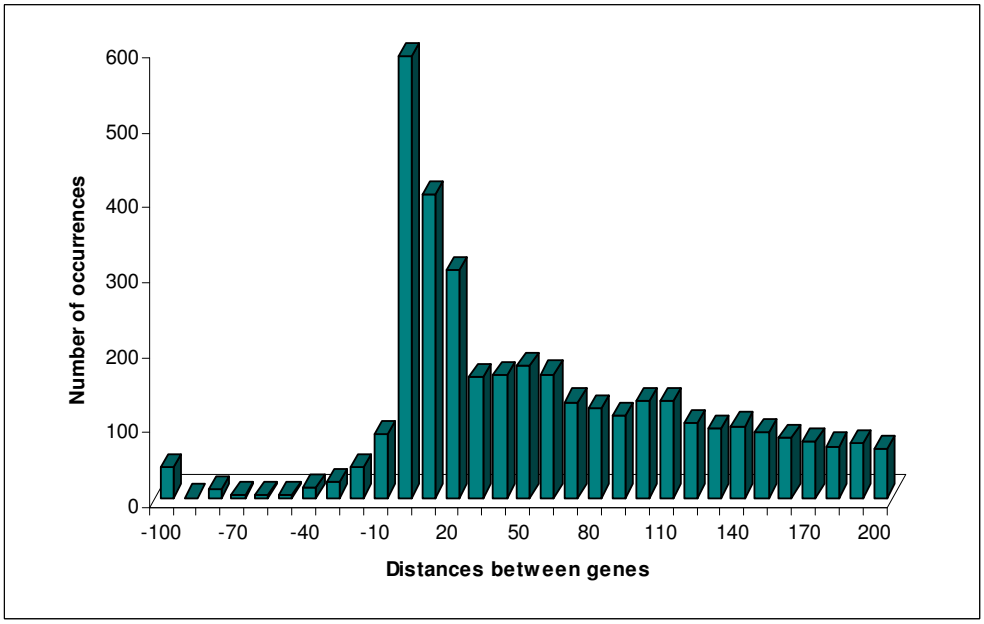

Figure 10. Frequency distribution of gene pairs with integenic distances between -100 and 200 bps.

Distances between two adjacent gene pairs in the *E. coli* genome are shown on the x-axis in 10 bp-increments and the corresponding number of occurrences (or counts) on the y-axis. The cluster of distances in the range [-20, 20] represents ~41% of all observations.

6.2 Analysis of the E. coli Microarray Expression Data

The expression data used in this work comes from 28 microarray experiments representing 14 different conditions. The microarrays used in this work are Affymetrix high-density oligonucleotide microarrays and the expression values are represented by the expression levels computed with the Affymetrix software. A total of 4,243 genes are represented by the microarray dataset. For each gene represented on the microarray, we construct an expression vector denoted by $E = (E_1, E_2, \ldots E_n)$, where n=28, the number of experiments. The Bayesian classifier uses as one of its features the correlation coefficient between expression vectors of adjacent genes. The Pearson correlation coefficient measures the extent to which two expression vectors are linearly related. Thus, given two expression vectors E_i and E_j, the Pearson correlation coefficient, denoted by r is computed as follows [43]:

$$r = \frac{\sum E_i E_j - \frac{\sum E_i \sum E_j}{N}}{\sqrt{\left(\sum E_i^2 - \frac{\left(\sum E_i\right)^2}{N}\right)\left(\sum E_j^2 - \frac{\left(\sum E_j\right)^2}{N}\right)}}$$

If there is a perfect linear correlation between two expression vectors then r = 1. Biologically, it means that both genes are either both expressed or both not expressed in the same conditions. If two genes exhibit expression pattern opposite from each other, i.e. the expression of one gene is up and the expression of the other is down, the correlation coefficient will be equal to -1. Correlation coefficient $r = 0$ represents situation, where no linear relationship between two expression vectors can be determined. These three scenarios are illustrated in Figure 11. Of course, the correlation coefficients will rarely be 1 or -1 due to the limitations in the microarray technology (see Section 3.2).

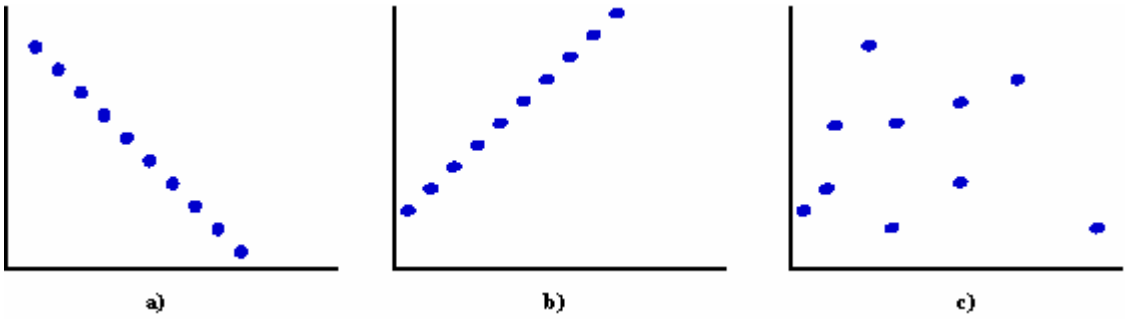

Figure 11. Interpreting Pearson correlation coefficients.

Positive correlation is shown in a), where the expression levels of both neighboring genes either go up or down in similar conditions. Negative correlation is shown in b). Negative correlation means that when the expression of one gene goes up, the expression of the gene next to it goes down. Example in c) shows the situation where there is no relationship between the expression levels of two neighboring genes.

We computed the correlation coefficients for 3,936 pairs in the *E. coli* genome (552 gene pairs do not have expression values). The majority of the adjacent genes show positive correlation. Thirty two percent of all gene pairs in the *E. coli* genome have correlation coefficients greater than 0.6. The most frequently occurring correlation coefficient is 0.4, followed by 0.3. Figure 12 shows the distribution of the correlation coefficients in the data set.

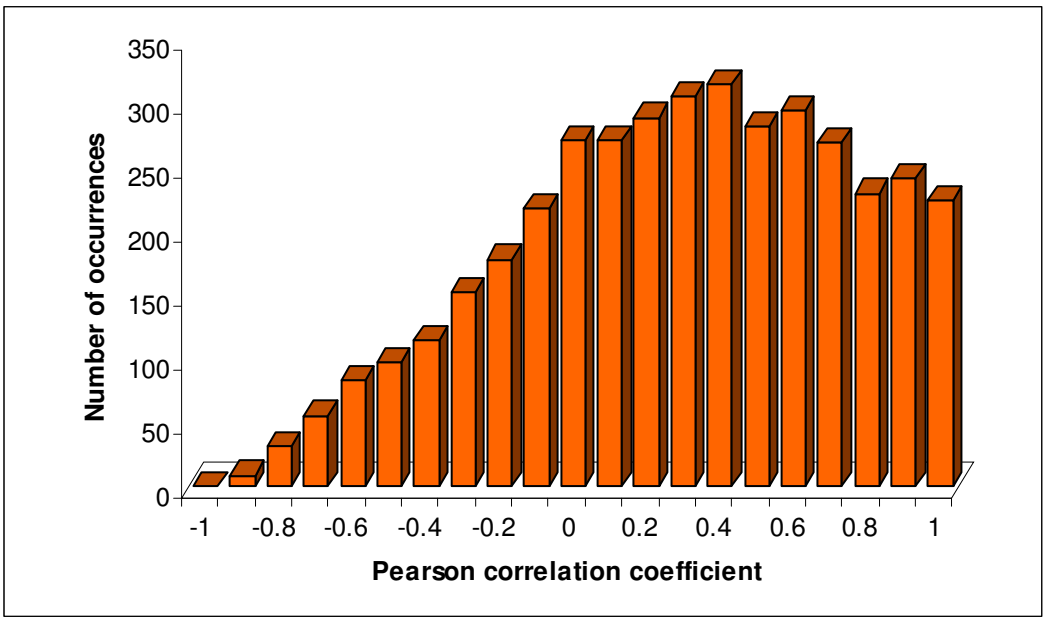

Figure 12. Frequency distribution of Pearson correlation coefficients.

Pearson correlation coefficients in 0.1-increments are plotted along the x-axis. Number of occurrences (or counts) of gene pairs are shown on the y-axis. The distribution has a peak corresponding to r=0.4-0.5.

6.3 Genomic Analysis of the RegulonDB Data

As of September 28, 2006, the RegulonDB transcription unit dataset contained 1,254 experimentally verified transcription units and represented the transcriptional status of 2,293 *E. coli* genes. About 37% of all transcription units in the dataset or 849 transcription units are monocistronic; 406 of these are on the forward strand and 443 on the reverse. Of the 405 polycistronic transcription units (~63% of all transcription units), 190 were found on the forward strand and 215 on the reverse strand. The average size of the transcription unit in the dataset is 1.83 and the average size of the polycistronic transcription unit (defined as the number of genes transcribed as a single mRNA molecule) is 3.57 genes. The longest 3 operons are of size 15. Table 5 summarizes the distribution of the length of the 1,254 transcription units in the training data set.

Approximately 18.3 % of protein coding genes are known to be transcribed into single transcripts and 31.4% into polycistronic transcripts. All of *E. coli*'s ribosomal genes are transcribed into polycistronic RNA. The status of 50.3% of CDS, 9 tRNA genes and 24 small RNA genes is unknown. The summary of transcript representation by gene type is shown in Table 6.

Table 5. Number of genes per transcription unit in the RegulonDB data set.

Size	Number of transcription units
1	849
2	172
3	83
4	60
5	34
6	23
7	10
8	7
9	7
10	1
11	2
12	2
13	1
15	3

Table 6. Number of moncistronic and polycistronic transcripts in the RegulonDB data set.

Gene Type	Number of Genes in Monocistronic Transcripts	Number of Genes in Polycistronic Transcripts
CDS	792	1355
rRNA	0	22
tRNA	20	57
other RNA	24	1
Total	**836**	**1435**

Our operon data set consists of 1,444 genes. Only 957 gene pairs had both expression vectors. These 957 gene pairs were actually used as our operon training data set. Analysis of gene pairs within operons revealed that the average distance is 33.65 bps (versus ~104 bps between all genes in the *E.coli* genome), the longest distance is 559 bps and the longest overlap is 142 bps. Approximately 93% of the gene pairs have intergenic distance between -20 and 130 bps.

The first non-operon data set consisted of 2,194 gene pairs. This data set was reduced by removing 586 duplicate gene pairs. The final collection of non-operon gene pairs is of size 1,608. The average distance between non-operon gene pairs is 181.13 bps, the longest distance between genes at the boundaries of transcription unit is 1,455 bps, and the longest overlap is 527 bps. Figure 13 shows distribution of distances between genes in operon versus non-operon data set. Although the average distance between operon genes is less than between non-operon gene pairs, two distributions overlap in the region

of 40 to 70 bps. The classifier based on the intergenic distances alone would have tough time distinguishing between operon and non-operon data sets.

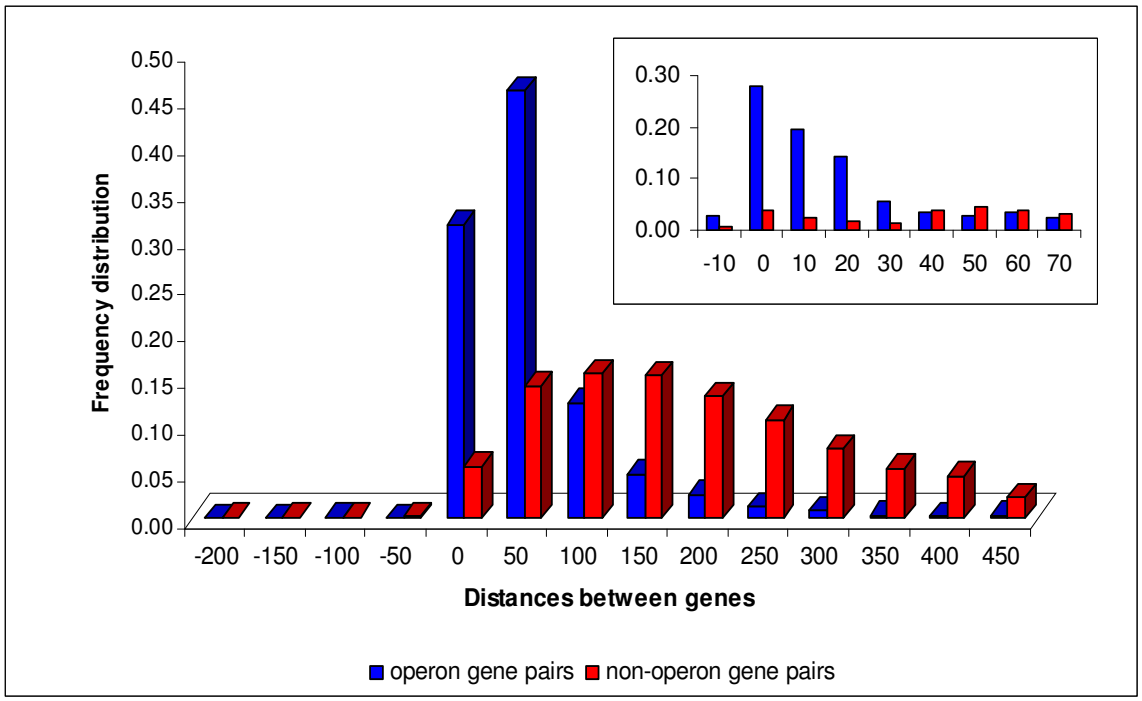

Figure 13. Frequency distribution of the distances in the two training data sets.
The partial frequency distribution is shown for intergenic distances between -200 and 450 bps. Distances between gene pairs are shown on the x-axis in 50 bp-increments. Frequencies are shown on the y-axis. Frequency was computed as the number of occurrences of each distance divided by the total number of gene pairs in a class. The insert shows frequency distribution in 10 bp-incrrements for distances between -10 bps and 70 bps. An overlap between two distributions is seen in the [40, 70] region. Operon gene pairs are shown in blue and non-operon gene pairs in red.

Frequency distribution of Pearson correlation coefficients was computed for the two datasets (see Figure 14). Although, the frequency of correlation coefficients greater than 0.9 is higher in known operons than in known non-operons, the correlation coefficients in the range 0.3-0.5 are almost as likely to come from non-operons as from operons.

6.4 Training Data: Challenges and Problems

To train our Bayesian classifier we created a training set consisting of 2,565 known operon and non-operon gene pairs. This task was very challenging due to the fact that the data sets described above have many inconsistencies and missing values. For example, some genes found in the RegulonDB file are missing from the Genbank file, or there is no expression data for some of the genes in either Genbank or RegulonDB files. We merged together three data sets, removed entries with missing values and used only gene pairs that had all three values present: intergenic distance, expression vector and transcription unit status.

The second challenge in creating the training set is how to treat zero counts. *Zero counts* occur when a class and a feature value do not occur together in the training set. This scenario will cause a problem for the classifier because the probability $p(X_i=x_i|C_i=c)$ is zero and thus, the posterior probability will be zero as well. To avoid zero probabilities we can add pseudo-counts to the frequencies. The solution used in this work is Laplace's rule: we add one to each frequency [15].

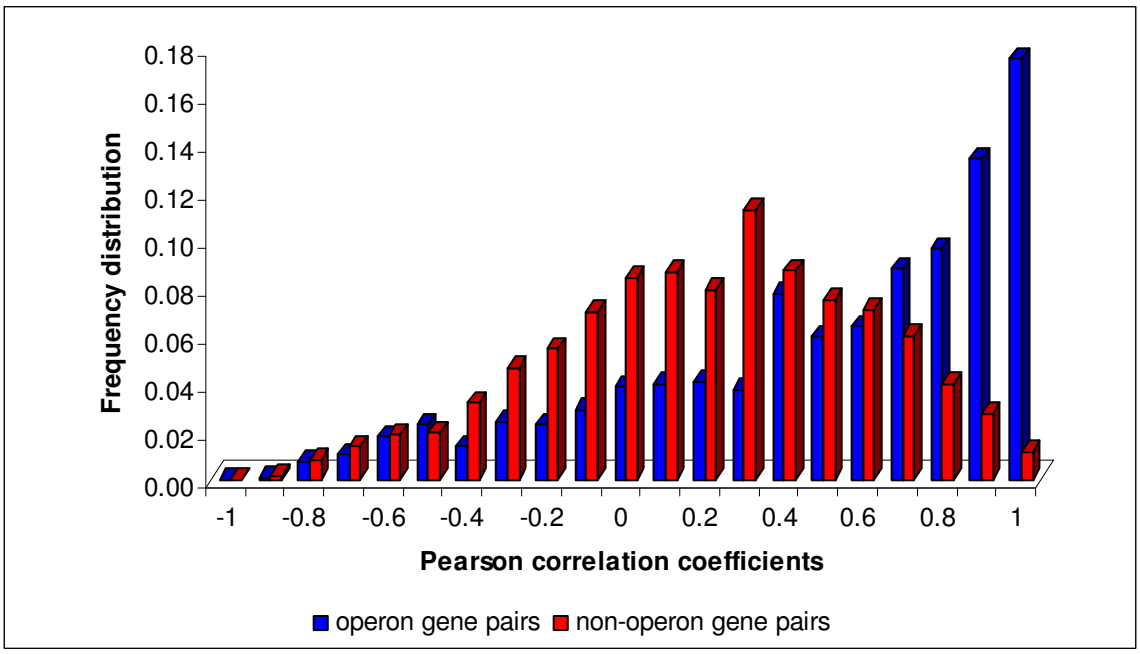

Figure 14. Frequency distribution of Pearson correlation coefficients in two training data sets.

Operon gene pairs are shown in blue and non-operon gene pairs in red. Pearson correlation coefficients are shown in 0.1 increments. Frequencies are shown on the y-axis. Frequency was computed as the number of occurrences of each distance divided by the total number of gene pairs in a class.

7 IMPLEMENTATION DETAILS

7.1 Computer Technologies

To complete the work two open-source computer technologies were used, MySQL and Perl. MySQL is an open source relational database management system based on a client/server model [27]. It is owned by the Swedish company MySQL AB. It is very popular for web applications and runs on many different hardware platforms, including Solaris, Linux, Mac OS, and Windows. MySQL version 5.0.24a running on Windows XP was used in this work.

The second technology used in this work is the Perl programming language [37]. Perl was created by Larry Wall as an alternative to UNIX shell scripting. It was initially used for system administration and text-processing, but has since grown into a powerful, general purpose programming language. It runs on almost all hardware platforms, is free and well maintained by a group of enthusiasts. There is a large number of modules

written to support database access, CGI programming and other tasks. Perl 5.8 was used in this work. Three Perl modules, DBI, BioPerl, and Statistics, were downloaded from the Comprehensive Perl Archive Network (CPAN) and used in this work.

Perl's database support is provided in form of the two-step architecture. First, a generic SQL access to databases is provided through the database interface (DBI) module. The second step requires the database driver (DBD) specific to MySQL [32].

BioPerl is a free collection of Perl modules for bioinformatics written by programmers around the world [5]. The modules are written in an object-oriented style and support most common bioinformatics tasks. The modules used in this work allow for an easier parsing of the Genbank files.

The Statisistcs Perl module was obtained from CPAN to provide methods for frequency computations [11].

7.2 Database Overview

To facilitate data analysis and store the results of our predictor, we designed and developed the database called **ecoli**. The database currently contains data about the *E.coli* genome only, but can very easily be populated with data from other genomes. The database consists of 10 tables summarized in Table 7. The following is a brief description of the database tables (full information about each table and its attribute can be found in Appendix D).

- Experiment table stores information about different microarray experiments, including the description and array type (cDNA or oligonucleotide). Each experiment is uniquely identified and is linked to the organism via the organism identifier.
- Expression table contains expression values, standard errors for each gene and each experiment. Each expression result is uniquely identified. The expression table is related to the gene and organism table in a one-to-many and many-to-one fashion.
- Gene table contains information about genes parsed from Genbank file described in Section 6. Each gene is given a unique database identifier. The table contains gene name(s), gene type, gene annotations and the translated protein sequence, if applicable.
- Gene_tu_map is the table mapping the genes from the gene table to transcription units in the tu table. This table is needed to avoid a many-to-many relation that exists between the tu and gene tables.
- Organism table contains the basic information about the organism, such as taxonomic name, genome size, and a unique identifier.
- Pair table contains pairs of adjacent genes along with the distance between them, their orientation, and computed Pearson coefficients. Pairs were created by connecting adjacent genes starting from base pair 1 in the genome. Each pair is uniquely identified by an id number. Gene identifiers of both genes in the pair can

be used to retrieve information about them from other tables. Gene pairs can be associated with an organism via the organism identifier.

Table 7. Summary of the ecoli database.

Table Name	Records	Type	Size	Comments	
experiment	14	MyISAM	3.2 KB	Creation:	Feb 25, 2007 at 09:07 PM
				Last update:	Feb 25, 2007 at 09:07 PM
expression	118888	MyISAM	5.2 MB	Creation:	Feb 25, 2007 at 09:07 PM
				Last update:	Feb 25, 2007 at 09:09 PM
gene	4488	MyISAM	2.4 MB	Creation:	Feb 25, 2007 at 09:07 PM
				Last update:	Feb 25, 2007 at 09:07 PM
gene_tu_map	2293	MyISAM	45.8 KB	Creation:	Feb 25, 2007 at 09:07 PM
				Last update:	Mar 03, 2007 at 08:58 PM
organism	1	MyISAM	2.0 KB	Creation:	Feb 25, 2007 at 09:07 PM
				Last update:	Feb 25, 2007 at 09:07 PM
pair	4488	MyISAM	243.8 KB	Creation:	Feb 25, 2007 at 09:07 PM
				Last update:	Mar 03, 2007 at 06:28 PM
pair_status_map	3151	MyISAM	28.7 KB	Creation:	Feb 25, 2007 at 09:07 PM
				Last update:	Mar 04, 2007 at 02:22 PM
status	10	MyISAM	2.4 KB	Creation:	Feb 25, 2007 at 09:07 PM
				Last update:	Feb 25, 2007 at 09:07 PM
tu	1254	MyISAM	52.2 KB	Creation:	Feb 25, 2007 at 09:07 PM
				Last update:	Mar 03, 2007 at 08:58 PM
tu_status_map	2508	MyISAM	23.0 KB	Creation:	Feb 25, 2007 at 09:07 PM
				Last update:	Mar 03, 2007 at 08:47 PM
10 table(s)	**137095**	**--**	**8.0 MB**		

- Pair_status_map is a utility table eliminating a many-to-many relationship between the pair table and the status table. The table stores the status of each of the gene pair.
- Status table is a utility table to describe the status of gene pairs and transcription units as operon or non-operon. For each status, we also keep the information about the experimentally identified transcription units, transcription units predicted using distances and expression values, distances only, or expression only.
- Tu table contains the information extracted from the RegulonDB file. Each transcription unit is uniquely identified and the record contains transcription unit, size, and orientation and relates to the organism table in a many-to-one fashion. In addition, this table contains the description of the transcription units predicted by our program.
- Tu_status_map is a utility table mapping the transcription units to their status. This table was created to avoid the many-to-many relationship between the tu and status tables.

The database and the tables were created using a SQL script found in Appendix E. Four input files described in Section 6 were parsed with Perl scripts and the data was loaded into the databases. Ten Perl object-oriented packages were written to store objects parsed from input files. Four Perl programs were written to parse the data and load them into the database. The source code of the packages and programs are available in Appendix F. In addition, the results of the predictions, i.e. output of the Bayesian classifier and output from the predictor were loaded into the database for an easy retrieval in the future.

7.3 Data Analysis

Genomic data analyses were performed using SQL statements. Two separate Perl utility scripts were written to compute Pearson correlation coefficients between genes and frequency distributions. The graphs were created using Microsoft Excel.

7.4 Program Overview

The predictor consists of two Perl programs: BayesianClassifier.pl and PredictTU.pl. Both programs make use of the DBI module to connect to the **ecoli** database, retrieve required data and load new results. Both programs can be found in Appendix F. The following is a brief description of the programs:

BayesianClassifier.pl is a program to train the Bayesian classifier, evaluate the test results, and to classify unknown gene pairs. The program runs in one of the two modes: test or predict. In either mode, the classifier is first trained using the training set of known operon and non-operon gene pairs. In the next step in the test mode, the accuracy of the classifier is evaluated using leave-one-out method described in Section 5.3. The process

is performed three times, one for each classifier, i.e. distance-based, expression-based and distance-and-expression-based. In the predict mode, all gene pairs in the genome are retrieved from the database and status of each is predicted using each of the classifiers with their optimal $p_{threshold}$ values. If the feature values of unseen gene pairs are missing in the training data set, the status of the gene pair will remain unclassified. The results of the predict mode are stored in the pair and pair_status_map tables in the **ecoli** database.

PredictTU.pl program extracts predicted gene pairs from the database along with the information about their location, sorts gene pairs based on their location and then joins adjacent gene pairs into longer runs. The results are stored in the tu, tu_status_map and gene_tu_map tables.

8 RESULTS

We implemented the operon predictor using Bayesian classifiers. We trained and tested our predictor on the data set derived from experimentally known transcription units in the *E.coli* genome. We performed the leave-one-out analysis to validate the predictor and access the predicting power of features used to construct Bayesian classifiers. This analysis is very important, since it gives us a way to measure the contribution of each of the data source to the overall prediction accuracy. The latter outcome, in turn, provides an estimate of how well this predictor would perform with other organisms for which one of the features is missing.

We repeated validation tests with the reduced Bayesian classifier where one of the features was removed from the model. Figure 15 shows the ROC curves for each test. Note that all curves deviate from the "no-discrimination line" or the 45-degree diagonal of a random predictor. The farther away from this line is ROC curve, the better is model's predictive power. The area under the curve is typically interpreted as "the measure of the probability that a randomly selected positive instance will have a higher probability than a randomly selected negative instance" [9]. Better classifiers will have larger areas under the curve.

From Figure 15, we see that the reduced Bayesian classifier with expression-only feature has the lowest performance. The curves for the full Bayesian and distance-only classifiers are very close to each other. To evaluate the hypothesis that the full model comprised of two features, distance and microarray expression values, performs better than the corresponding reduced models, we conducted a two-tailed, paired t-test [21] with a standard threshold of 0.05 on p-values. This statistical test is typically performed when comparing two or more alternatives and finding the best one.

A p-value less than 0.05 indicates that there is a significant difference between models in terms of their predictive power. A p-value greater than 0.05 means that statistically no difference exists between the two models. We compared each of the reduced Bayesian classifiers to the full model and determined that a significant improvement over expression-only classifier can be gained by combining distances and expressions as features in the full Bayesian classifier (p-value 0.01). However, adding expression correlation coefficients does not seem to improve the distance-only Bayesian predictor (p-value 0.39). The summary of the t-tests are shown in Table 8.

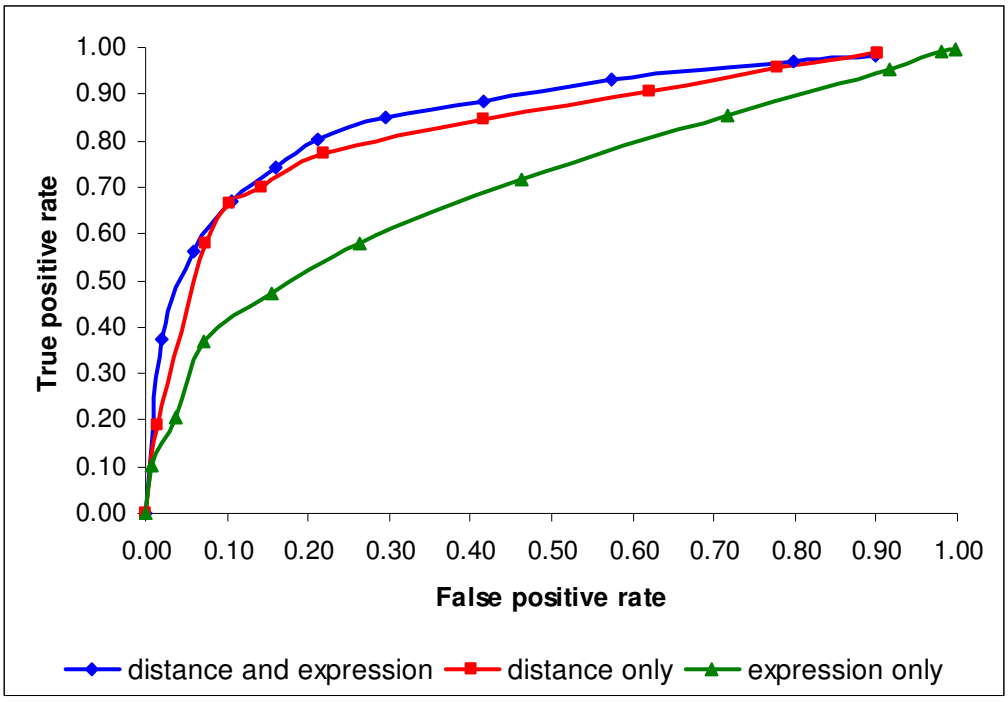

Figure 15. ROC curves for our Bayesian classifier with distance and expression features, with distance only and expression only.
The ROC for Bayesian classifier with two features, distance and expression, is shown in blue. The ROC for Bayesian classifier with distance only is shown in red and the ROC for the Bayesian classifier with expression only is shown in green. Note that the curves appear in the same order as the labels in the legend. The points on each curve were obtained by varying the $p_{threshold}$ for the posterior probability that separates positive from negative predictions.

The results can be interpreted in the following fashion. Distance between genes has a higher predictive power than the correlation of expression values. This fact is very important if the Bayesian classifier were applied to other prokaryotic genomes that do not have microarray data. Distances between genes can very easily be computed from the genome annotation. Of course, it is possible that the model trained on the *E.coli* will not perform well on other genomes due to differences in the distribution of the intergenic distances.

Table 8. Summary of two-tailed, paired t-tests.
A p-value < 0.05 indicates that two models are significantly different from each other and a p-value ≥ 0.05 indicates that there is no statistical difference between two models. Statistics were computed using Microsoft Excel.

Statistic	**Distance and Expression versus**	
	Distance-only	**Expression-only**
Pearson Correlation	0.99	0.92
T Stat	0.91	3.10
P(T<=t) one-tail	0.19	0.01
t Critical one-tail	1.81	1.81
P(T<=t) two-tail	0.39	0.01
t Critical two-tail	2.23	2.23

The results might also suggest that the performance of the expression-only classifier would improve if a different set of microarray data was used. This does not seem to be the case. Published results consistently rate expression-only predictors below other methods. The problem seems to lie in the way microarrays are designed. As mentioned in Section 4.3, most of the microarrays experiments perturb the expression of a small number of genes. The expression of most genes in these experiments remains the same. This lack of change in expression values over the number of experiments results in the correlation coefficient equal to 0 (no correlation between neighboring genes). What is needed for a robust expression-only classifier is data from microarray experiments that cause changes in the expression values of many genes.

We would also like to point out that our Naïve Bayesian classifier is able to recover 80% of operon gene pairs when both expression and distance are used as features. While this seems lower than 88% sensitivity reported by some of the authors (see Table 1), we cannot really make a fair comparison. The results published usually do not report how the performance metrics were calculated. In addition, the training data set grows with every release of RegulonDB (quarterly), and we cannot compare the results computed using 2000 data with ours.

We used our predictor to identify transcription units in the whole *E. coli* genome. In Table 9, we report the number of pairs classified as operon and non-operon with the full and reduced Bayesian classifiers. In all three classifiers, $p_{threshold} = 0.5$ was used. The results of each prediction are stored in the **ecoli** database. Distance-only classifier predicts larger number of gene pairs in both categories, which is probably due to the fact that some of the gene pairs in the *E.coli* genome have missing expression vectors, whereas all gene pairs have integenic distances.

Table 9. Prediction results of Bayesian classifiers on the whole E. coli genome.
The results shown are the number of gene pairs classified as either operon or non-operon by the three Bayesian classifiers: full, distance-only and expression-only. The $p_{threshold}$ for the posterior probability was set to 0.5. Prior probability was set to 0.5.

Classifier Type	**Number of predicted gene pairs**	
	Operon	**non-operon**
Distance and expression	1,613	2,469
Distance-only	1,744	2,632
Expression-only	1,234	2,892

In the final step, we linked classified gene pairs into transcription units. Table 10 shows the final results of the operon predictor. The results indicate that combining both distances and expression values results in an increased power of the predictor. All three predictors correctly identify transcription units of size 1. Correct identification of polycistronic transcription units remains a challenge for all three predictors. A point to make here is that a true positive in the prediction of whole transcription units is considered a sequence of genes exactly matching confirmed monocistronic or polycistronic transcription units. This criterion is very strict: either the whole transcription unit is predicted or not. Correctly identifying the first three genes in a

transcription unit of size four, but missing the last one will count this transcription unit as a negative.

Table 10. Overall prediction results.

Predictor Type	Number of predicted transcription units		
	monocistronic	**polycistronic**	**unclassified**
Distance and expression	3,373	397	728
Distance only	4,206	101	181
Expression only	4,398	11	79

Upon closer examination of the polycistronic transcription units' prediction, we notice that our predictor often breaks true transcription units into two. This occurs because the gene pair connecting these two parts either did not have posterior probability greater than the threshold or the value of one of the features, most often the intergenic distance, was missing from our training data set. The result might often indicate the presence of an alternative operon.

9 CONCLUSION

In this work, we obtained the sequence, annotations, microarray expression data and a set of completely characterized operons of the *E. coli* genome. We performed genomic analyses of all gene pairs in the organism as well as the gene pairs belonging to or at the boundaries of the transcription units. Gene pairs belonging to operons differ from non-operon gene pairs in terms of intergenic distances as well as and in terms of microarray expression values. Therefore, these features, intergenic distances and expression values can be used to predict operons in the *E. coli* genome and, potentially, in other sequenced prokaryotic genomes.

We have developed an operon predictor based on the intergenic distances and expression values of neighboring genes transcribed in the same direction. The predictor finds transcription units by first classifying each gene pair as either operon or non-operon with the Bayesian classifier and then extends gene pairs into longer runs. We evaluated three versions of the predictor, one that uses distances between genes only, one that uses correlation of expression values of two neighboring genes, and one that combines two features into one model. From the empirical evaluation of our method we conclude that all three predictors have significant predictive value shown by their distance from the 'no-discrimination line' in the ROC plots. The full model outperforms two reduced models. Intergenic distance is a significant feature and has a significantly higher predictive power than the correlation of expression values. We also propose that the reduced, distance-only model can be applied to other genomes when microarray data is not available. The outcomes of such application have not yet been evaluated, but can be a nice extension of this work in the feature. In addition, using other important transcriptional signals as features can be explored as well.

10 REFERENCES

[1] Alberts, B., et al., *Molecular biology of the cell*. 3rd ed. 1994: Carland Publishing.

[2] Berman, H. M., et al., *The Protein Data Bank*. Nucleic Acids Research, 2000. **28**:pp. 235-242.

[3] Bairoch, A. and R. Apweiler, *The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000.* Nucleic Acids Res, 2000. **28**(1): p. 45-8.

[4] Benson, D.A., et al., *GenBank.* Nucleic Acids Res, 1999. **27**(1): p. 12-7.

[5] BioPerl home page: www.bioperl.org

[6] Blattner,F.R., et. al., *The complete genome sequence of Escherichia coli K-12*. Science 1997. **277** (5331): pp. 1453-1474.

[7] Blumenthal, T. and K.S. Gleason, *CAENORHABDITIS ELEGANS OPERONS: FORM AND FUNCTION.* Nature Reviews Genetics, 2003. **4**(2): p. 110-118.

[8] de Hoon, M.J.L., et al., *Predicting the operon structure of Bacillus subtilis using operon length, intergenic distance, and gene expression information*, Proc. of the Pacific Symposium on Biocomputing, 2004. pp. 276--287.

[9] Bockhorst, J., et al., *Predicting bacterial transcription units using sequence and expression data.* Bioinformatics, 2003. **19**(90001): p. i34-43.

[10] Chen, X., et al., *Operon prediction by comparative genomics: an application to the Synechococcus sp. WH8102 genome.* Nucleic Acids Res, 2004. **32**(7): p. 2147-57.

[11] Comprehensive Perl Archive Network (CPAN) home page: www.cpan.org

[12] Dandekar, T., et al., *Conservation of gene order: a fingerprint of proteins that physically interact.* Trends Biochem Sci, 1998. **23**(9): p. 324-8.

[13] De Ferrari, L. and Aitken, S. *Mining housekeeping genes with a Naive Bayes classifier*. BMC Genomics 2006. p.7:277

[14] Domingos, P. and Pazzani, M. J., *On the the optimality of the simple Bayesian classifier under zero-one loss*. Machine Learning, 1997. **29**(2-3):pp.103--130.

[15] Durbin, R. et al., *Biological sequence analysis: Probabilistic Models of Proteins and Nucleic Acids*. 1998. Cambridge University Press.

[16] Ermolaeva, M.D., O. White, and S.L. Salzberg, *Prediction of operons in microbial genomes.* Nucleic Acids Res, 2001. **29**(5): p. 1216-21.

[17] Glansdorff, N., Journal of Molecular Evolution, 1999. **49**: p. 432-438.

[18] Hartl, S.L., et al., *Genetics: An Analysis of Genes and Genomes*. Fifth Edition ed. 2001: Jones and Bartlett Publishers, Inc.

[19] Ikeda, H., et al., *Complete genome sequence and comparative analysis of the industrial microorganism Streptomyces avermitilis.* Nat Biotechnol, 2003. **21**(5): p. 526-31.

[20] Jacob, F. and J. Monod, *Genetic regulatory mechanisms in the synthesis of proteins.* Journal of Molecular Biology, 1961. **3**: p. 318-56.

[21] Jain, R., *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. 1991. John Wiley & Sons, Inc.

[22] Kohane, I. S., et al., *Microarrays for an Integrative Genomics*. 2003: The MIT Press.

[23] Lockhart, D.J., et al., *Expression monitoring by hybridization to high-density*

oligonucleotide arrays. Nat Biotechnol, 1996. **14**: p. 1675-1680.
[24] Moreno-Hagelsieb, G. and J. Collado-Vides, *A powerful non-homology method for the prediction of operons in prokaryotes.* Bioinformatics, 2002. **18 Suppl 1**: p. S329-36.
[25] Marcotte, E.M., et al., *A combined algorithm for genome-wide prediction of protein function.* Nature, 1999. **402**(6757): p. 83-86.
[26] Mount, D.W., *Bioinformatics: Sequence and Genome Analysis*. Second Edition ed. 2004: Cold Spring Harbor Laboratory Press.
[27] MySQL home page: www.mysql.com
[28] Sabatti, C., et al., *Co-expression pattern from DNA microarray experiments as a tool for operon prediction.* Nucleic Acids Res, 2002. **30**(13): p. 2886-93.
[29] Salgado, H., et al., *Operons in Escherichia coli: genomic analyses and predictions.* Proc Natl Acad Sci U S A, 2000. **97**(12): p. 6652-7.
[30] Salgado H., et al., *RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions*. Nucleic Acids Res. 2006. Jan 1;**34**(Database issue):D394-7
[31] Schena, M., et al., *Quantitative monitoring of gene expression patterns with a complementary DNA microarray.* Science, 1995. **270**: p. 467-470.
[32] Sebesta, R.W., *Programming the World Wide Web.* 3rd edition. 2006. Addison-Wesley.
[33] Strong, M., et al., *Inference of protein function and protein linkages in Mycobacterium tuberculosis based on prokaryotic genome organization: a combined computational approach.* Genome Biol, 2003. **4**(9): p. R59.
[34] Tjaden, B., et al., *Identifying operons and untranslated regions of transcripts using Escherichia coli RNA expression analysis.* Bioinformatics, 2002. **18**: p. S337-S344.
[35] Overbeek, R., et al., *The use of gene clusters to infer functional coupling.* Proc Natl Acad Sci U S A, 1999. **96**(6): p. 2896-901.
[36] Pellegrini, M., et al., *Assigning protein functions by comparative genome analysis: protein phylogenetic profiles.* Proc Natl Acad Sci U S A, 1999. **96**(8): p. 4285-8.
[37] Perl home page: www.perl.org
[38] Price, M.N., et al., *A novel method for accurate operon predictions in all sequenced prokaryotes.* Nucl. Acids Res., 2005. **33**(3): p. 880-892.
[39] Rogozin, I.B., et al., *Congruent evolution of different classes of non-coding DNA in prokaryotic genomes*. Nucl. Acids Res. 2002. **30**: pp. 4264-4271.
[40] Russell, P.J., *Essential iGenetics*. 2003, San Francisco, CA 94111: Benjamin Cummings. p. 614.
[41] Yada, T., et al., *Modeling and predicting transcriptional units of Escherichia coli genes using hidden Markov models.* Bioinformatics, 1999. **15**(12): p. 987-93.
[42] Waters, E., et al., *The genome of Nanoarchaeum equitans: Insights into early archaeal evolution and derived parasitism.* PNAS, 2003. **100**(22): p. 12984-12988.
[43] Zweig, M.H. and Campbell, G. *Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine*. Clinical chemistry, 1993. **39** (8): p. 561–577