PLOS | ONE

RESEARCH ARTICLE

Assessing the Effects of Data Selection and Representation on the Development of Reliable *E. coli* Sigma 70 Promoter Region Predictors

Mostafa M. Abbas[1], **Mostafa M. Mohie-Eldin**[2], **Yasser EL-Manzalawy**[3,4]*

1 KINDI Center for Computing Research, College of Engineering, Qatar University, Doha, Qatar, **2** Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt, **3** Systems and Computer Engineering, Al-Azhar University, Cairo, Egypt, **4** College of Information Sciences, Penn State University, University Park, United States of America

* yme2@psu.edu

Abstract

As the number of sequenced bacterial genomes increases, the need for rapid and reliable tools for the annotation of functional elements (e.g., transcriptional regulatory elements) becomes more desirable. Promoters are the key regulatory elements, which recruit the transcriptional machinery through binding to a variety of regulatory proteins (known as sigma factors). The identification of the promoter regions is very challenging because these regions do not adhere to specific sequence patterns or motifs and are difficult to determine experimentally. Machine learning represents a promising and cost-effective approach for computational identification of prokaryotic promoter regions. However, the quality of the predictors depends on several factors including: i) training data; ii) data representation; iii) classification algorithms; iv) evaluation procedures. In this work, we create several variants of E. coli promoter data sets and utilize them to experimentally examine the effect of these factors on the predictive performance of E. coli σ^{70} promoter models. Our results suggest that under some combinations of the first three criteria, a prediction model might perform very well on cross-validation experiments while its performance on independent test data is drastically very poor. This emphasizes the importance of evaluating promoter region predictors using independent test data, which corrects for the over-optimistic performance that might be estimated using the cross-validation procedure. Our analysis of the tested models shows that good prediction models often perform well despite how the non-promoter data was obtained. On the other hand, poor prediction models seems to be more sensitive to the choice of non-promoter sequences. Interestingly, the best performing sequence-based classifiers outperform the best performing structure-based classifiers on both cross-validation and independent test performance evaluation experiments. Finally, we propose a meta-predictor method combining two top performing sequence-based and structure-based classifiers and compare its performance with some of the state-of-the-art E. coli σ^{70} promoter prediction methods.

OPEN ACCESS

Citation: Abbas MM, Mohie-Eldin MM, EL-Manzalawy Y (2015) Assessing the Effects of Data Selection and Representation on the Development of Reliable *E. coli* Sigma 70 Promoter Region Predictors. PLoS ONE 10(3): e0119721. doi:10.1371/journal.pone.0119721

Academic Editor: Dinesh Gupta, International Centre for Genetic Engineering and Biotechnology (ICGEB), INDIA

Received: June 12, 2014

Accepted: January 26, 2015

Published: March 24, 2015

Data Availability Statement: Data is included in Supporting Information files.

Funding: This publication was made possible by NPRP grant No. 4-1454-1-233 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

Introduction

Transcription initiation is the first and key step leading to gene expression [1]. The process starts with the binding of RNA polymerase (RNAP) to a specific segment in DNA (called promoter region) located upstream of the transcription start site (TSS). Understanding how RNAP locates and recognize promoter regions remains an active research area in molecular biology and a challenging task in bioinformatics. In bacteria, transcription initiation requires an additional subunit called σ factor, which associates with the core RNA polymerase to form a holoenzyme [2, 3]. Different σ factors interact with distinct consensus promoter sequences. Each σ factor is labeled according to its molecular weight (e.g., σ^{24}, $\sigma,^{28}$ σ^{32}, σ^{38}, σ^{54}, and σ^{70}). The accurate identification of σ-specific promoter regions is tiresome and technically exacting [4–6]. Therefore, computational methods for reliably identifying promoter sequences are highly desirable.

Many computational methods for predicting promoter regions in prokaryotes have been proposed in literature (e.g., [7–24]). Among these methods, several machine learning algorithms have been used in developing prokaryotic promoter region prediction methods. For example, support vector machine (SVM) [7, 8, 12, 25, 26], artificial neural networks (ANNs) [16, 17, 20, 27–29], partial least square (PLS) [18], and quadratic discriminant analysis (QDS) [14]. Some methods are based on probabilistic approaches (e.g., hidden Markov models (HMMs) [30] and a combination of HMMs and ANNs [31]). In such prediction methods, the promoter identification problem is viewed as a binary classification problem such that the given test sequence is predicted to be a promoter or a non-promoter [27].

In general, the quality of these prediction methods depends on several factors including: i) Training data: Does the data include redundant sequences? Is negative data experimentally validated? If not, how is negative data generated?; ii) Data representation: The vast majority of machine learning algorithms do not accept DNA sequence as input. Hence, some technique has to be applied to map each sequence into a vector of (often numeric) features such that the machine learning algorithm can efficiently discriminate between positively labeled and negatively labeled sequences; iii) Classification algorithms: Typically, the developer has to apply several machine learning algorithms to the target data and use the one with the best performance as the final predictor; iv) Evaluation procedures: There exist two widely-used evaluation procedures, cross-validation and blind test evaluations. In k-fold cross-validation experiments, the data is randomly partitioned into k subsets of equal size. Then, $k - 1$ folds are used to train the classifier and the hold away fold is used for testing. This step is repeated k times such that in each time a different fold is hold for testing the classifier. Leave-one-out evaluation procedure is a cross-validation procedure where k is set to the number of instances in the data. In blind test set experiments, an independent test set is prepared and used for evaluating the trained classifiers.

For each factor, several design choices have been made by the developers of prokaryotic promoter regions prediction methods. Following are some examples: i) Training data: Due to lack of sufficient data, the vast majority of methods did not remove redundant data (i.e., promoters or non-promoter sequences that share high similarity scores). No experimentally validated non-promoter sequence data source exists. Therefore, developers have to generate their non-promoter sequences. Several strategies for generating non-promoter sequences have been used including: randomly generated sequences [16, 17, 28]; sequences extracted from intergenic or coding regions [7, 11, 12, 14, 15, 18, 25, 28]; ii) Feature extraction: Several sequence and structure-based feature representations have been used for developing prokaryotic promoter region prediction methods. Examples of sequence based features include: k-mer representation [7, 12, 28, 32], variable-window Z-curve [18], and nucleotide identity (NID) [17]. Examples of structure based features include: stress induced duplex destabilization (SIDD), DNA curvature and stacking energy explored in [13], roll, tilt, twist and average free energy used in [14], and DNA stability proposed in [23]; iii) Classification algorithms: support vector machines (SVMs) and artificial neural networks (ANNs) are widely used for this classification task; iv) Evaluation procedures: the vast majority of prediction methods [7, 8, 10, 12, 13, 15–17, 19, 26, 33] have been evaluated using cross-validation experiments. However, few methods (e.g., [11, 15]) employed blind test sets in addition to cross-validation procedure to assess the performance of their predictors.

Against this background, we conduct extensive experiments to analyze the influence of these factors on prokaryotic promoter region predictors. One of our major aims is to guide the developers of future prokaryotic promoter region predictors to make appropriate design decisions (e.g., how to generate non-promoter sequences? how to get an accurate estimate of the performance of your classifier? how to avoid misleading conclusions?). Our results suggests that good representative non-promoter sequences should be extracted using multiple strategies (e.g., a combination of randomly generated sequences and sequences extracted from coding and non-coding regions). Our results also demonstrate that cross-validation estimates might be misleading (especially, when the non-promoter sequences are randomly generated or extracted from coding regions). We show that a more accurate estimate of performance could be obtained using high-quality independent test set or by averaging over multiple versions of the cross-validation data. Finally, we propose a meta-predictor combining two sequence-based and structure-based predictors for predicting E. coli σ^{70} promoter regions and compare it with some state-of-the-art prediction methods.

Materials and Methods

Data sets

We used experimentally validated promoters from E. coli, a well studied prokaryotic model organism. RegulonDB [34, 35] is a rich resource for curated information on transcriptional regulation in E. coli K-12. The latest version, RegulonDB 8, has been enriched with a large number of promoters and TSS mapped using high-throughput technology. In our experiments, we used promoters extracted from RegulonDB 7 for constructing our cross-validation data sets and promoters extracted from RegulonDB 8 but not in RegulonDB 7 to construct our independent test sets. We limited our experiments to σ^{70} promoters due to the lack of sufficient training for other σ factors dependent data sets after removing redundant sequences. Both cross-validation and test sets are included in the Supporting Information section (S1 Dataset).

Generation of negative data sets. Like many bioinformatics classification tasks, predicting prokaryotic promoter region is challenged by the lack of experimentally validated negative (i.e., non-promoter) data. To study how different approaches for generating non-promoter sequences might affect the performance of the classifier, we explored three approaches that randomly select non-promoter regions from: i) Randomly generated DNA sequences. DNA segments of length 81 were randomly selected from a DNA sequence of length 1000,000 that was randomly generated with frequencies 0.28, 0.22, 0.22, and 0.28 for T, G, C, and A (respectively); ii) Coding regions in E. coli K12 genome downloaded from NCBI GenBank [36]; iii) Intergenic regions in E. coli K12 genome downloaded from Ecogene database [37] and categorized into convergent, divergent, Codirectional+, and Codirectional-.

Cross-validation data sets. We extracted 741 σ^{70} promoters from RegulonDB 7. After removing sequences that share more than 45% similarity, our final set of positive data consists of 579 promoter sequences. Seven versions of the cross-validation data set has been constructed by combining the positive set with seven non-redundant, by removing sequences that share more than 45% similarity, sets of 579 sequences (none of them share more than 45% sequence similarity with any positive sequence). To the best of our knowledge, this is the first E. coli promoter region data set that establishes some criteria for reducing sequence similarity. Although a more stringent similarity cutoff might be preferred, this choice was not applicable as the number of the promoter sequences drops to 100 sequences at 35% similarity cutoff. The seven versions of the cross-validation data share the same positive set but each data set version has different negative set (see Table 1).

Independent test sets. We downloaded 1790 σ^{70} promoters from RegulonDB 8. Then, we discarded from this list of sequences any sequence that share more than 45% sequence similarity with at least one promoter sequence in the cross-validation data. Our final independent test sets consist of seven versions: TS_Random; TS_Coding; TS_Convergent; TS_Divergent; TS_CoPos; TS_CoNeg, and TS_Mixed. All versions share the same 792 promoter sequences but each one has its own negative set of 792 non-promoters generated using the same procedure used with the cross-validation data. None of the negative data sequences share more than 45% sequence similarity with any corresponding negative cross-validation data set sequence.

Features extraction

In our data sets, each promoter sequence is 81 bp long region [TSS-60. . .TSS+20], with the mapped TSS at position 0. Non-promoter sequences are also 81 pb long regions with no validated TSS at position 0. The vast majority of machine learning algorithms can not be applied directly to such input. Instead, a per-processing step (called features extraction) has to be performed in order to map each DNA sequence into a feature vector. For instance, 1-mer features representation maps each DNA sequences into four numeric features, that are typically the frequency of the four types of nucleotides in the target DNA sequence.

In our experiments, we evaluated several features extraction methods, which have been widely used for promoter classification tasks (e.g., [7, 12, 14, 28, 32]) and for DNA classification tasks (e.g., [38–40]). The features extraction methods that we evaluated could be categorized into two main categories: i) sequence-based features; ii) structure-based features.

For sequence-based features, we used k-mer features [7, 12, 28, 32] with k = 1, 2, . . ., 5. k-mer features representation maps each DNA sequence into 4^k numeric features representing the normalized counts of each k-mer substring in the sequence. A major limitation of k-mer features is that some sequence order information is lost. We evaluated two sequence-based

Table 1. Summary of cross-validation data sets.

Data set	Source of negative data
CV_Random	Randomly extracted from a single long sequence that is generated with frequencies 0.28, 0.22, 0.22, and 0.28 for T, G, C and A (respectively), according to Silva et al., [17]
CV_Coding	Randomly extracted from coding regions extracted form E.coli K-12 genome downloaded from NCBI GenBank [36]
CV_Convergent	Randomly extracted from convergent intergenic regions downloaded from EcoGene 3.0 database [37]
CV_Divergent	Randomly extracted from divergent intergenic regions downloaded from EcoGene 3.0 database [37]
CV_CoPos	Randomly extracted from codirectional positive spacer regions downloaded from EcoGene 3.0 database [37]
CV_CoNeg	Randomly extracted from codirectional negative spacer regions downloaded from EcoGene 3.0 database [37]
CV_Mixed	Six equal subsets of negative sequences extracted from negative sequences in CV_Random, CV_Coding, CV_Convergent, CV_Divergent, CV_CoPos, and CV_CoNeg

doi:10.1371/journal.pone.0119721.t001

features that preserves the sequence order information: i) nucleotide identity (NID) features; ii) dinucleotides identity (DNID) features. In nucleotide identity features, each 81 nucleotides long DNA sequence is represented with 81 features. Each feature is a nominal attribute which can take any value from the set {A, C, G, T}. In dinucleotides identity features, each 81 nucleotides long DNA sequence is represented with 80 features. Each feature is a nominal attribute which can take any value from the set {AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT, TA, TC, TG, TT}. It should be noted that some classifiers (e.g., Naive Bayes and Random Forests) works directly with nominal (i.e., categorical) attribute values, while others (e.g., Support Vector Machines and Neural Networks) do not support this type of attributes and requires the conversion into numeric features (i.e., using orthogonal codification to convert each nominal value into a binary string).

For structure-based features, we evaluated several DNA structure-based features derived from twelve dinucleotides scales including: A-philicity (M1) [41]; Ohler B-DNA twist (M2) [42]; Olson B-DNA twist (M3) [43]; DNA bending stiffness (M4) [44]; DNA denaturation temperature (M5) [45]; Z-DNA free energy (M6) [46]; duplex disruption free energy (M7) [47]; duplex stability free energy (M8) [48]; protein-induced deformability (M9) [43]; propeller twist (M10) [49]; protein-induced DNA twist (M11) [43]; and base stacking energy (M12) [50]. Using these scales, each DNA sequence in our data is represented with 80 numeric features using the procedure described in [51] and a smoothing window of size equals three.

All sequence and structure-based features are implemented as part of the Genome Annotation Toolkit (Gennotate) [52]. Gennotate is an extension of WEKA [53], a widely used machine learning workbench supporting many standard machine learning algorithms. Most of these algorithms could not be applied directly to DNA sequence data sets. Developers have to pre-process their data for feature extraction and then apply WEKA implemented algorithms to the data set in its numerical representation. Gennotate integrates the DNA feature extraction step into WEKA and allows on-the-fly mapping of DNA sequences into feature vectors. This simplifies and expedites the process of building machine learning based models for different genome annotation tasks and facilitates sharing offline versions of developed models and the development of consensus and hybrid models on top of existing ones [52].

Classification algorithms

We evaluated three machine learning algorithms that are widely used in bioinformatics sequence classification tasks: i) Naive Bayes (NB) [54]; ii) Random Forest (RF) [55]; iii) Support Vector Machines (SVM) [56]. Each of these algorithms has some strengths and weaknesses. For example, NB is superior in terms of training speed, training simplicity (i.e., no parameter tuning is needed), and scalability. On the other hand, NB algorithm relies on a strong assumption regarding attribute independence. We discuss the strengths and limitations of each algorithm in predicting E. coli σ^{70} promoters in the Results and Discussion section. In the following paragraphs, we summarize the three algorithms.

The NB classifier [54] is a direct and straightforward application of Bayes Theorem. The main assumption of NB classifier is the conditional independence of all attributes given the class label. In spite of the unrealistic assumption of independence, the performance of NB classifier is competitive with sophisticated classifiers for many real-world classification tasks. The NB classifier takes the random variable $X = (x_1, x_2, x_3, \ldots ..., x_n)$, promoter sequence features, as input and produce the binary class $C \in \{1, 0\}$ as output, where '1' denotes a promoter and '0' denotes a non-promoter. For a query instance, X, NB classifier returns '1' (promoter) if:

$\frac{P(C=1|X=x_1,x_2,x_3,.....,x_n)}{P(C=0|X=x_1,x_2,x_3,.....,x_n)} = \frac{P(C=1)\prod_{i=1}^{n} P(x_i=x|C=1)}{P(C=0)\prod_{i=1}^{n} P(x_i=x|C=0)} \geq 1$ and returns the class label ‘0’ (non-promoter) otherwise.

The RF classifier [55] is a combination of random decision tree base classifiers. It integrates *bagging* [57] with the random selection of subset feature for training decision trees as following: i) Generating bootstrap samples with n training instances (i.e., randomly selecting with replacement n instances from the training data); ii) Randomly selecting m variables from the set of M input features, $m \ll M$, and using the sampled training data for generating individual base decision trees. The k tree classifiers will be constructed by repeating this two-step procedure. The RF classifier reports the average prediction of all decision tree classifiers for given query instance. In our experiments, we used RF classifier with $k = 100$.

The SVM classifier [56] is based on the concept of decision planes that define decision boundaries. The SVM classifier maps the input features into feature vectors in a high-dimensional feature space. In the training stage, the data is separated into positive and negative in the feature space by finding a hyperplane that maximizes the margin of separation. In case of non-linearly separable training data in the input space, SVM classifier uses a kernel function K to map non-linearly separable data in the input space into a typically high-dimensional feature space where the data are assumed to be linearly separable without explicitly mapping each training example from the input space into the feature space. The selection of the kernel function is a critical factor in training SVM classifiers. Therefore, the performance of SVM classifier depends on selecting a suitable kernel and tuning the kernel parameters (if any). In our experiments, we applied two widely-used kernel functions: i) Linear kernel; ii) Radial bias kernel (RBF).

The input of the above classifiers is a variety of sequence or structural features of the promoter and non-promoter sequences. On the other hand, we tried an additional classifier (based on HMM algorithm [58]) that takes the DNA sequence itself as input. The HMM classifier is a stochastic generative model classifier based on Markov chain. The HMM classifier assumes that labels are hidden and its goal is to predict these hidden labels given the input sequence. The HMM is composed of two stochastic processes. The first process is characterized by hidden states (with three types: match, delete, and insert states) and probabilities of transition such that each state depends only on the previous state (i.e., Markov property is established). The second process produces emissions observable at each moment, based on a state-dependent probability distribution. The parameters of an HMM will be iteratively modified during the training phase. We used a java implementation of HMM algorithm that is provided in Gennotate tool [52].

Performance evaluation

The predictive performance of promoter region prediction classifiers was assessed using Accuracy (ACC), Sensitivity (S_n), Specificity (S_p), and Mathew Correlation Coefficient (MCC) measures defined as follows [59, 60]:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$S_n = \frac{TP}{TP + FN} \tag{2}$$

$$S_p = \frac{TN}{TN + FP} \tag{3}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TN + FN)(TN + FP)(TP + FN)(TP + FP)}} \tag{4}$$

where TP, FP, TN, and FN are the numbers of true positive(promoter sequence classified as promoter), false positive(non-promoter sequence classified as promoter), true negative(non-promoter sequence classified as non-promoter), and false negative(promoter sequence classified as non-promoter), respectively.

All these metrics are determined using a specific threshold value, which could be viewed as a trade off between S_n and S_p. The Receiver Operating Characteristic (ROC) curve [61] provides a wide comprehensive picture of the performance of the predictor where it describes the performance of the classifier over all possible thresholds. The ROC curve is a two-dimensional graph in which the true positive rate is plotted on the *Y* axis and the false positive rate is plotted on the *X* axis. Each point on the ROC curve represents the behavior of the classifier at a specific choice threshold value, and hence a particular choice of tradeoff between true positive rate and false positive rate. The area under ROC curve (AUC) is equivalent to the probability that a randomly chosen positive example will be ranked higher than a randomly chosen negative example. Swets [62] suggested evaluation grades for the classifiers based on AUC scores (see Table 2) Here, we limit our discussion to the AUC metric and report the performance of classifiers using threshold-dependent metrics in the Supporting Information section (S1 Text).

Statistical analysis

For comparing several classifiers on multiple data sets, we used the non-parametric statistical test proposed by Demšar [63]. First, classifiers are ranked based on their observed performance (e.g., AUC) for each data set separately (i.e., for each data set, the best classifier is assigned a rank of 1, the second best classifier is ranked 2, and so on). Second, the Friedman test is applied to determine whether the measured average ranks are significantly different from the mean rank under the null hypothesis. Third, if the null hypothesis can be rejected at a significance level of 0.05, the Nemenyi test is used to determine whether significant differences exist between any given pair of classifiers.

Results and Discussion

Misleading cross-validation performance estimates

Tables S1-S5 in S1 Text report the average AUC of several sequence-based classifiers obtained using the average of 10 runs of 10-fold cross-validation experiments. Classifiers with excellent performance ($AUC \geq 0.90$) are observed more oftenly when classifiers are evaluated using

Table 2. Grading scale for classifiers based on their AUC scores.

AUC score	Grade
0.90–1.00	Excellent
0.80–0.89	Good
0.70–0.79	Fair
0.50–0.69	Poor

doi:10.1371/journal.pone.0119721.t002

CV_Random and CV_Coding data sets. However, such classifiers with excellent (or good) cross-validation performance estimates might perform poorly in real world scenarios. For example, the classifier NB_3-mer_Coding has AUC score equals 0.91. However, when this classifier is evaluated using TS_Coding, TS_Convergent, and TS_Mixed test sets its AUC scores are 0.90, 0.59, and 0.68 (respectively). Another example is the classifier NB_4-mer_Random, which has AUC score equals 0.87 while its performance on TS_Coding, TS_Convergent, and TS_Mixed test sets is 0.91, 0.59, and 0.68 (respectively).

For structure-based classifiers (Tables S6-S9 in S1 Text), the performance obtained using CV_Coding data set is significantly higher than the performance of the classifiers evaluated on other versions of the cross-validation data (including CV_Random). These results suggest that, for cross-validation experiments, there exists some combination of classifier design choices (e.g., randomly generated negative data combined with k-mer features) that could produce a classifier with excellent performance estimates. However, this classifier will perform poorly on independent test sets. This finding underscores the necessity of employing independent test sets for evaluating methods for predicting prokaryotic promoters.

A hypothesis for identifying good predictors

A careful examination of results reported in Tables S1-S9 (S1 Text) suggests the following hypothesis for identifying good predictors (i.e., predictors which perform well on cross-validation and independent test experiments): A good predictor is a predictor that performs well on cross-validation data regardless how negative data is generated. In our experiments, we used the average AUC over the seven cross-validation data sets to indicate the overall classifier performance. We also used the standard deviation (STD) to indicate how sensitive the classifier to different choices of negative data. Using this hypothesis, we chose the following representative set of good predictors: NB_DNID, RF100_M7, and HMM which have $AUC \pm STD$ equal 0.85 ± 0.04, 0.80 ± 0.04, and 0.82 ± 0.05 (respectively). We also chose the following representative set of bad performing classifiers: NB_4-mer and NB_M1 with $AUC \pm STD$ equal 0.73 ± 0.12 and 0.70 ± 0.09 (respectively). Table 3 shows the performance of these five representative classifiers trained using CV_Mixed data and tested on the seven independent set versions. Interestingly, all classifiers could discriminate between promoter and coding sequences. Another interesting observation is that the AUC using TS_Mixed is within ±0.03 of the AUC obtained using CV_Mixed cross-validation data set. This suggest that the average cross-validation performance estimate obtained using different versions of the cross-validation data,

Table 3. AUC scores for selected classifiers (trained using CV_Mixed data) and tested on different versions of independent test set (e.g., TS_Random and TS_Coding).

Data set	NB_DNID	RF100_M7	HMM	NB_4-mer	NB_M1	meta-predictor
TS_Random	0.83(1.5)	0.77(5.0)	0.80(3.5)	0.76(6.0)	0.80(3.5)	0.83(1.5)
TS_Coding	0.89(2.5)	0.87(5.0)	0.89(2.5)	0.88(4.0)	0.86(6.0)	0.91(1.0)
TS_Convergent	0.80(2.5)	0.80(2.5)	0.78(4.0)	0.64(6.0)	0.66(5.0)	0.82(1.0)
TS_Divergent	0.80(2.0)	0.79(3.0)	0.78(4.0)	0.61(6.0)	0.65(5.0)	0.82(1.0)
TS_CoPos	0.79(2.0)	0.78(3.0)	0.76(4.0)	0.58(6.0)	0.66(5.0)	0.81(1.0)
TS_CoNeg	0.82(2.0)	0.80(3.5)	0.80(3.5)	0.68(5.5)	0.68(5.5)	0.84(1.0)
TS_Mixed	0.83(2.0)	0.82(3.0)	0.81(4.0)	0.70(6.0)	0.71(5.0)	0.85(1.0)
Average	0.82(2.0)	0.80(3.4)	0.80(3.5)	0.69(5.5)	0.72(4.9)	0.84(1.1)
STD	0.03	0.03	0.04	0.10	0.08	0.03

See Methods Section for more information about these test sets. For each data set, the rank of each classifier is shown in parentheses.

doi:10.1371/journal.pone.0119721.t003

created using different techniques for generating non-promoter sequences, is a good estimate of performance estimates obtained on independent test sets.

Sequence-based versus structure-based predictors

Tables S1-S4 (S1 Text) show that sequence-based classifiers evaluated using NID and DNID features representation outperform classifiers evaluated using k-mer features representation (in terms of higher AUC scores and lower standard deviations). The top two classifiers are SVMRBF_DNID and NB_DNID with $AUC \pm STD$ equal 0.86 ± 0.05, 0.85 ± 0.04 (respectively).

Tables S6-S9 (S1 Text) show that the top performing structure-based classifiers (with $AUC = \sim 0.80$) are obtained using Random Forest algorithm and DNA bending stiffness (M4), duplex disruption free energy (M7), duplex stability free energy (M8), or base stacking energy (M12) features representation. Interestingly, the vast majority of structure-based classifiers seem to be less sensitive to the design choice of the non-promoter sequences.

Although the cross-validation experiments suggest a superior performance of sequence-based classifiers over structure-based ones, results on independent test sets (see Table 3) narrow the gap in performance between top performing sequence-based and structure-based classifiers. A meta-predictor combining NB_DNID and RF100_M7, using average of predicted probabilities, results in 0.02 improvement in AUC over NB_DNID. Further improvements in performance could be achieved by: i) including more divergent base classifiers (e.g., HMM or classifiers using the same feature representation but different classification algorithms); ii) using more sophisticated approaches for combining base classifiers (e.g., using second stage meta-classifier). An alternative approach for integrating DNA sequence and structure-based features is to concatenate them and train a single classifier. For example, a novel DNA feature representation, pseudo dinucleotide composition, combines dinucleotide composition with six local DNA structure properties into a single feature vector has been proposed [64]. Pseudo k-tuple nucleotide composition combines k-tuple (k-mer) features with DNA structure features [65].

For identifying the statistically significant differences in the performance of selected classifiers, we applied Demšar's three-step procedure to the results obtained on the independent test data sets. Table 3 shows the AUC scores associated with the rank for each classifier on each data set and the average of them. At a significance of 0.05, the application of Friedman test suggests the existence of statistically significant differences between the selected methods. Hence, there is at least two classifiers such that the difference between their average ranks is statistically significant at 0.05 level of significance. The significantly different pair-wise comparisons, obtained using Nemenyi test, are summarized in Fig. 1.

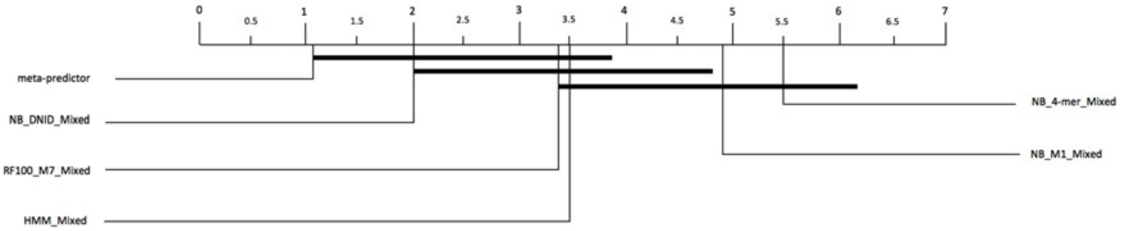

Fig 1. Pair-wise comparison of selected classifiers with Nemenyi test applied to results on independent test data sets. Groups of classifiers that are not significantly different (at p-value = 0.05) are connected.

doi:10.1371/journal.pone.0119721.g001

Analysis of discriminative features

Table 4 summarizes the performance of NB, RF100, SVMLnr, and SVMRBF classification methods using 12 different structure-based representations of the cross-validation data set. We observed that Olson B-DNA twist (M3) representation leads to consistent ($STD = 0.00$) average poor performance ($0.5 \leq AUC \leq 0.69$) while DNA bending stiffness (M4) representations leads to consistent ($STD = 0.01$) average fair performance ($0.7 \leq AUC \leq 0.79$). Therefore, for each of the four classification methods, M3 representation leads to a classifier with poor performance while M4 representation leads to a classifier with fair performance. To understand the differences between these two representations, we used the cross-validation data to plot the profiles for M3 and M4 (see Fig. 2). Briefly, cross-validation data (using M3 and M4 representation, respectively) were grouped into 8 groups: Promoter, Mixed, Coding, CoNeg, Convergent, CoPos, Divergent, and Random. For each group, we got a profile by averaging the values in each attribute feature. Fig. 2 shows the profiles for promoter and non-promoter sequences using M3 (top) and M4 (bottom) feature representations. Interestingly, using M4 feature representation, the profile of negative coding data can be easily discriminated from other profiles. Also, the profile of promoter data has distinguishable peak signal in the region 45–55 which allows for discriminating promoter profile from other non-promoter profiles. To validate this observation, we applied WEKA's InfoGainAttributeEval feature selection method to rank the attributes in CV_Mixed data set using 10-fold cross-validation experiment. The top 10 ranked attributes are attributes corresponding to positions: 49, 48, 50, 51, 52, 47, 53, 54, 29, and 46. Interestingly, 9 out of top 10 ranked attributes lie in the region 45–55. On the other hand, the signal for discriminating the promoter profile from other profiles using M3 feature representation is not as strong as the one using the M4 feature representation.

Influence of negative data on performance estimates

To examine the influence of negative data on the estimated performance of predictors, we performed the following experiment. First, we decided to estimate the performance using blind

Table 4. Summary of the performance of NB, RF100, SVMLnr, and SVMBRF classifiers on cross-validation data using twelve different structure-based feature representations.

Features	NB	RF100	SVM(Lnr)	SVM(RBF)	Average	STD
M1	0.70(9.0)	0.72(9.0)	0.67(11.0)	0.68(10.0)	0.69(9.8)	0.02
M2	0.65(12.0)	0.70(10.0)	0.61(12.0)	0.61(12.0)	0.64(11.5)	0.04
M3	0.68(10.5)	0.68(11.5)	0.69(9.5)	0.68(10.0)	0.68(10.4)	0.00
M4	0.78(1.5)	0.80(2.5)	0.78(1.5)	0.78(2.0)	0.79(1.9)	0.01
M5	0.76(5.5)	0.78(7.0)	0.76(5.0)	0.76(6.0)	0.77(5.9)	0.01
M6	0.78(1.5)	0.79(5.5)	0.77(3.0)	0.78(2.0)	0.78(3.0)	0.01
M7	0.74(7.5)	0.80(2.5)	0.76(5.0)	0.77(4.5)	0.77(4.9)	0.03
M8	0.77(3.5)	0.80(2.5)	0.78(1.5)	0.78(2.0)	0.78(2.4)	0.01
M9	0.74(7.5)	0.79(5.5)	0.71(7.5)	0.72(7.5)	0.74(7.0)	0.04
M10	0.76(5.5)	0.77(8.0)	0.71(7.5)	0.72(7.5)	0.74(7.1)	0.03
M11	0.68(10.5)	0.68(11.5)	0.69(9.5)	0.68(10.0)	0.68(10.4)	0.00
M12	0.77(3.5)	0.80(2.5)	0.76(5.0)	0.77(4.5)	0.78(3.9)	0.02

A-philicity (M1) [41]; Ohler B-DNA twist (M2) [42]; Olson B-DNA twist (M3) [43]; DNA bending stiffness (M4) [44]; DNA denaturation temperature (M5) [45]; Z-DNA free energy (M6) [46]; duplex disruption free energy (M7) [47]; duplex stability free energy (M8) [48]; protein-induced deformability (M9) [43]; propeller twist (M10) [49]; protein-induced DNA twist (M11) [43]; and base stacking energy (M12) [50]. For each data set, the rank of each classifier is shown in parentheses.

doi:10.1371/journal.pone.0119721.t004

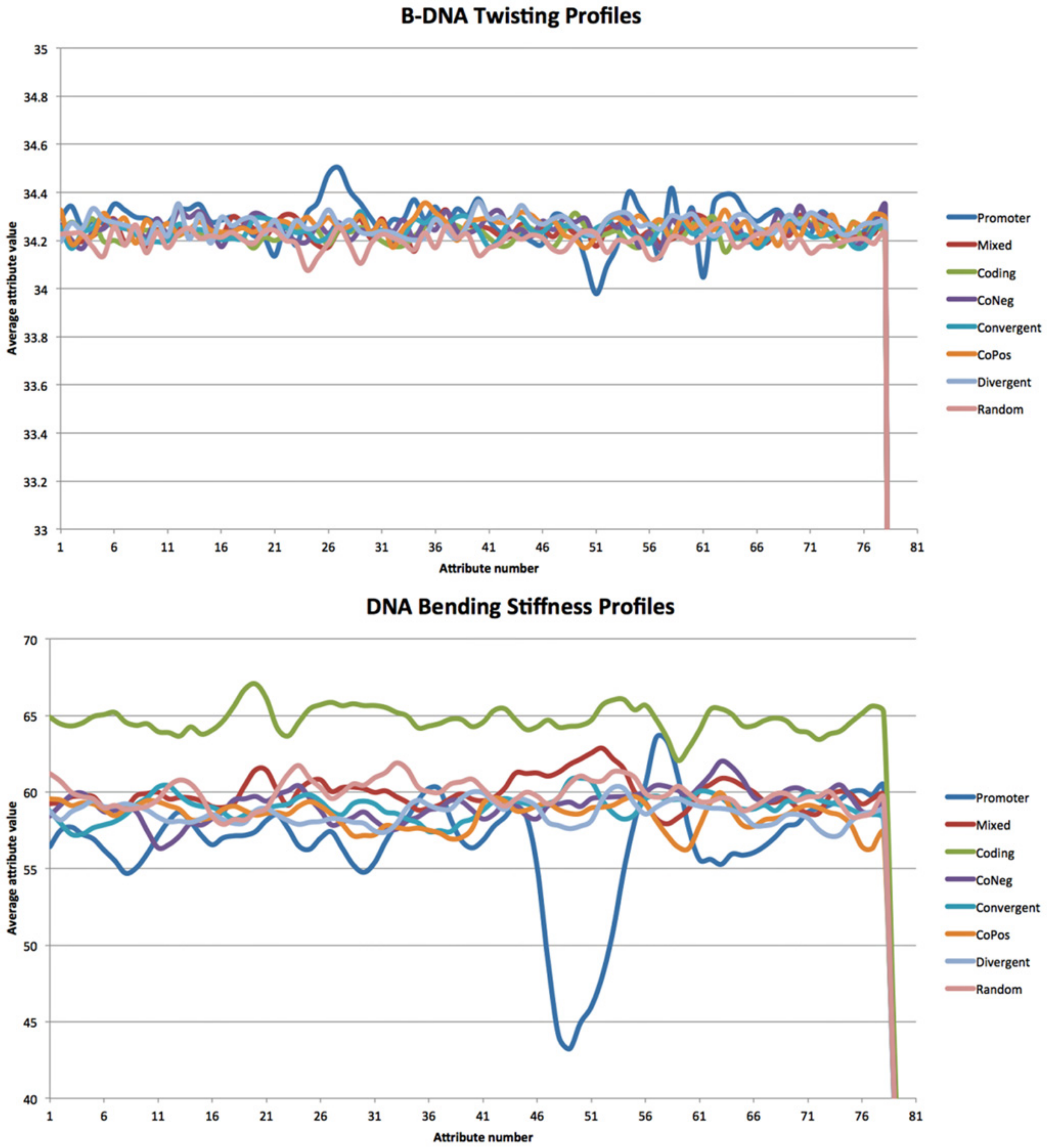

Fig 2. B-DNA twisting profiles (top) and DNA bending stiffness profiles (bottom) generated from cross-validation data.

doi:10.1371/journal.pone.0119721.g002

Table 5. AUC scores for Naive Bayes classifier with DNID features (NB_DNID) trained using seven versions of CV data and in each time tested on the seven versions of the independent test data.

Training data	TS_Random	TS_Coding	TS_Convergent	TS_Divergent	TS_CoPos	TS_CoNeg	TS_Mixed
CV_Random	0.87(2.0)	0.90(1.0)	0.80(5.0)	0.79(6.5)	0.79(6.5)	0.83(4.0)	0.84(3.0)
CV_Coding	0.80(2.5)	0.93(1.0)	0.74(6.5)	0.75(5.0)	0.74(6.5)	0.78(4.0)	0.80(2.5)
CV_Convergent	0.82(4.5)	0.84(1.0)	0.82(4.5)	0.80(6.5)	0.80(6.5)	0.83(2.5)	0.83(2.5)
CV_Divergent	0.80(4.5)	0.86(1.0)	0.80(4.5)	0.79(6.0)	0.77(7.0)	0.82(2.0)	0.81(3.0)
CV_CoPos	0.80(5.0)	0.84(1.0)	0.81(2.5)	0.80(5.0)	0.79(7.0)	0.80(5.0)	0.81(2.5)
CV_CoNeg	0.80(4.5)	0.86(1.0)	0.80(4.5)	0.79(6.0)	0.76(7.0)	0.83(2.0)	0.82(3.0)
CV_Mixed	0.83(2.5)	0.89(1.0)	0.80(5.5)	0.80(5.5)	0.79(7.0)	0.82(4.0)	0.83(2.5)
Average	0.82(3.6)	0.87(1.0)	0.80(4.7)	0.79(5.8)	0.78(6.8)	0.82(3.4)	0.82(2.7)
STD	0.03	0.03	0.03	0.02	0.02	0.02	0.01

Each row corresponds to a specified training set while each column corresponds to a specified test set.

doi:10.1371/journal.pone.0119721.t005

test set experiments in order to avoid over-optimistic performance estimates reported using cross-validation experiments. Second, based on our suggested hypothesis, we picked up two classification methods: NB_DNID and NB_4-mer to represent good and bad predictors (respectively). Third, we trained these two classifiers using the seven versions of training data sets and tested them on the seven versions of blind test data. Tables 5 and 6 report the performance of NB_DNID and NB_4-mer in this experiment, respectively.

Our first observation is that the coding regions should not be used to generate negative data for blind test sets. TS_Coding test set shows an over-optimistic performance of both classifiers. However, such low-quality test data can discriminate between good and bad predictors since NB_DNID seems to be less sensitive to the choice of training data (STD = ±0.03) while NB_4-mer seems to be more sensitive to the choice of training data version (STD = ±0.13).

For test sets with negative data generated from non-coding regions (e.g., TS_CoNeg, TS_CoPos, TS_Convergent, TS_Divergent), they are all successful in discriminating between good and bad predictors. Both NB_DNID and NB_4-mer have consistent good and poor performances (respectively) regardless which version of training data has been used for training the classifiers. TS_CoPos test set seems to be the most challenging test set because both classifiers have their lowest performance when this test data is used (see TS_CoPos columns in Tables 5 and 6).

Table 6. AUC scores for Naive Bayes classifier with 4-mer features (NB_4-mer) trained using seven versions of CV data and in each time tested on the seven versions of the independent test data.

Training data	TS_Random	TS_Coding	TS_Convergent	TS_Divergent	TS_CoPos	TS_CoNeg	TS_Mixed
CV_Random	0.87(1.0)	0.82(2.0)	0.63(5.0)	0.60(6.0)	0.58(7.0)	0.65(4.0)	0.69(3.0)
CV_Coding	0.73(2.0)	0.91(1.0)	0.59(6.0)	0.60(5.0)	0.56(7.0)	0.63(4.0)	0.68(3.0)
CV_Convergent	0.62(3.5)	0.56(6.5)	0.74(1.0)	0.56(6.5)	0.57(5.0)	0.64(2.0)	0.62(3.5)
CV_Divergent	0.64(4.5)	0.83(1.0)	0.61(6.0)	0.64(4.5)	0.55(7.0)	0.68(2.0)	0.66(3.0)
CV_CoPos	0.57(7.0)	0.65(2.0)	0.63(3.5)	0.59(6.0)	0.66(1.0)	0.61(5.0)	0.63(3.5)
CV_CoNeg	0.58(6.0)	0.74(1.0)	0.62(4.0)	0.60(5.0)	0.55(7.0)	0.71(2.0)	0.63(3.0)
CV_Mixed	0.76(2.0)	0.88(1.0)	0.64(5.0)	0.61(6.0)	0.58(7.0)	0.68(4.0)	0.70(3.0)
Average	0.68(3.7)	0.77(2.1)	0.64(4.4)	0.60(5.6)	0.58(5.9)	0.66(3.3)	0.66(3.1)
STD	0.11	0.13	0.05	0.02	0.04	0.03	0.03

Each row corresponds to a specified training set while each column corresponds to a specified test set.

doi:10.1371/journal.pone.0119721.t006

For the test set with randomly generated sequences, Tables 5 and 6 show that TS_Random can always discriminate between good predictors (e.g., NB_DNID) and poor predictors (e.g., NB_4-mer) except when the two predictors are trained using CV_Coding or TS_Random. Therefore, results reported on blind test set where negative data are fragments of randomly generated DNA sequences should be handled with caution, especially when the negative training data was also generated using the same way.

Finally, for TS_Mixed where negative data has been obtained by mixing six subsets of equal numbers of negative data generated using the six approaches for generating negative data explored in this study, we noted that: i) TS_Mixed can successfully discriminate between good and bad predictors regardless which training data version is used for training the classifiers; ii) Performance of both classifiers is remarkably less sensitive to the type of training data (e.g., small STD value is reported for both classifiers).

In summary, any version of the training data sets (including the versions with randomly generated negative data and negative data extracted from coding regions) could produce a good classifier (e.g., a classifier with AUC score between 0.80 and 0.89 on the blind test set) when sequence data is represented using discriminative features (e.g., DNID features). On the other hand, test sets with negative data randomly generated or extracted from coding regions should be avoided.

Comparison with existing prokaryotic promoter prediction servers

Although our main goal is not to develop a predictor that outperforms the state-of-the-art methods for predicting prokaryotic promoters (in the sense that no attempts have been made to tune the parameters of any classifier considered in this study), it is of interest to figure out how the performance estimates of our identified good predictors compare with some existing methods for predicting σ^{70} promoter regions in E. coli. To address this question, we compared NB_DNID_Mixed, HMM_Mixed, RF100_M7_Mixed, and meta-predictor classifiers trained using CV_Mixed data set with IPMD [15], BacPP [17], and variable-window Z-curve (VWZ) [18] methods. None of these methods has a Web server. However, the source code and the data sets used to evaluate VWZ method [18] can be downloaded at: http://www.csssk.net/publications. We adapted this code to return prediction scores instead of predicted binary labels and to train and test on two separate data sets instead of performing jackknife test on a provided data set. The modified code is provided in the Supporting Information section (S1 Code). For IPMD and BacPP methods, the TS_Mixed test set has been submitted to the authors of the two methods who kindly agreed to apply their methods to our test data and returned predicted probabilities to us. The predictions returned by these three methods were compared with the predictions of our four classifiers (in terms of AUC) in Fig. 3.

BacPP encodes nucleotides as 4-bit binary strings and uses artificial neural networks [66] for training its predictors using promoter data extracted from RegulonDB 6.1 and non-promoter sequences extracted from intergenic (non-coding) regions. The lacking performance of BacPP might be due to the low quality of the data set (e.g., insufficient training data and no attempts to remove highly similar sequence have been tried).

IPMD combines increment of diversity and position weight matrices for predicting eukaryotic and prokaryotic promoters. For training and evaluating the IPMD σ^{70} promoter predictor, the authors used 1400 non-promoter sequences (700 coding and 700 convergent intergenic sequences). The observed IPMD performance (AUC = 0.84) is competitive with NB_DNID (AUC = 0.83) and meta-predictor (AUC = 0.85).

VWZ method extracts Z-curve descriptors [67] of k-mer features (for k = 1, . . ., 6) and uses a partial least squares (PLS) classifier combined with an iterative feature selection procedure to

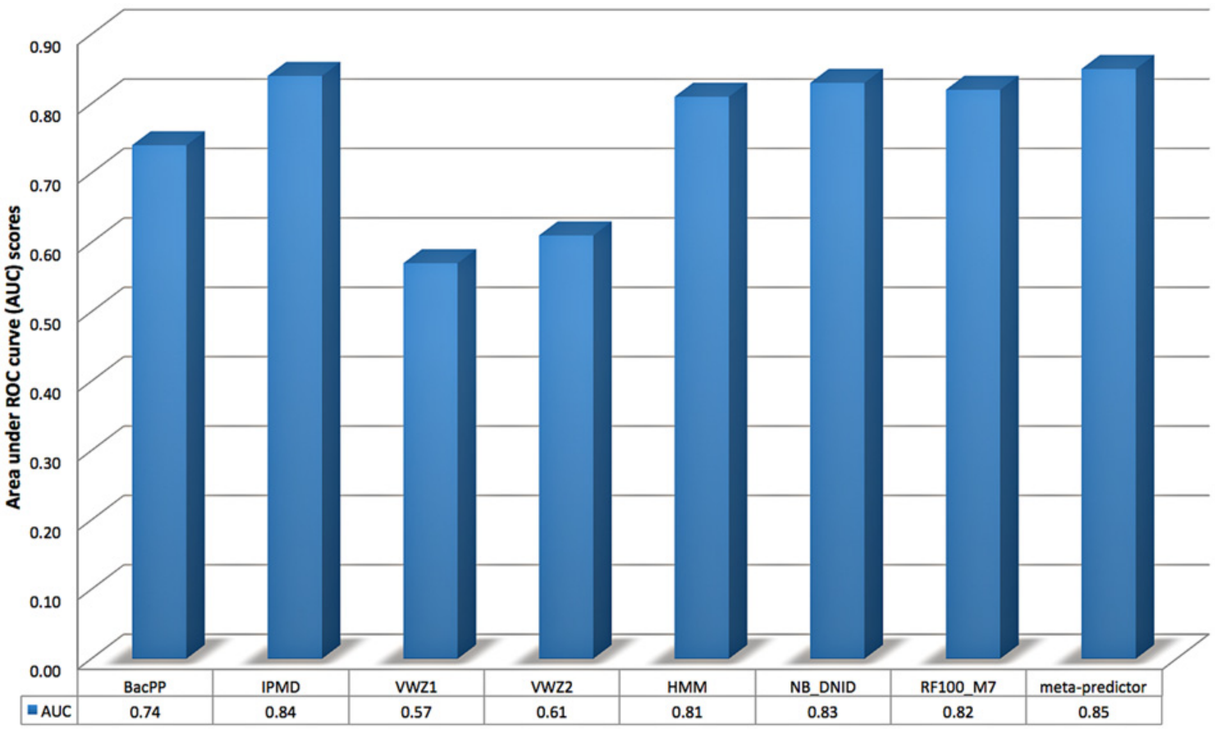

Fig 3. Performance comparison of BacPP, IPMD, and two variable-window Z-curve models, VWZ1 and VWZ2, trained using Datatset-1 and Datatset-2 (respectively) with four selected classifiers (NB_DNID, RF100_M7, HMM, and meta-predictor) using TS_Mixed independent test set.

doi:10.1371/journal.pone.0119721.g003

eliminate irrelevant and highly-correlated features. The method was evaluated on two data sets: Dataset-1 contains 576 σ^{70} promoters (positive samples) and 836 coding fragments (negative samples) of E. coli; Dataset-2 contains the same 576 σ^{70} promoters (positive samples) and 825 non-coding fragments (negative samples) of E. coli. Using jackknife tests, an excellent performance, accuracy $\geq$ 90% using top ranked 330 and 220 features for Dataset-1 and Dataset-2 (respectively) was reported [18]. Fig. 3 shows that the AUC scores for the two VWZ classifiers trained using Dataset-1 and Dataset-2 using optimal number of features and tested on TS_Mixed data are 0.57 and 0.61 (respectively). The huge discrepancy between the cross-validation (jackknife test is an *n*-fold cross-validation test, where *n* is the number of instances) performance reported in [18] and the independent test reported in this study might be in part justified by the low quality of the training data (i.e., no similarity reduction have been applied to Dataset-1 and Dataset-2). To test this hypothesis, we trained one more VWZ model using CV_Mixed data set and top ranked 220 features. The AUC for such model on the TS_Mixed test data is 0.70. This result emphasize the importance of independent test sets to confirm the performance estimates of cross-validation tests and to avoid misleading cross-validation results that might be observed due to the redundancy in the data or due to the possibility that some classifiers might overfit the data and produce a model with impressive cross-validation performance and a poor generalization performance on other independent data sets.

Conclusions

The development of reliable prokaryotic promoter region prediction methods is highly desirable for improving the accuracy of microbial genomes annotation tools. A major limitation in developing reliable prokaryotic promoter region predictors is the lack of experimentally validated non-promoter data. We evaluated several strategies for generating non-promoter sequences and showed that a more accurate estimate of the classifier performance could be obtained using negative data consisting of equal size subsets of sequences generated using multiple strategies or by generating multiple versions of the cross-validation data (each with negative data generated with a different strategy) and use the average cross-validation performance over these data sets as the estimated cross-validation performance of the classifier. This approach would be very useful in cases where it is hard to obtain more experimental data to be used for independent test experiments. We also showed that a good predictor and/or good feature representation should allow for the discrimination between promoter sequences and all types of non-promoter sequences.

Cross-validation experiments are widely used for estimating the performance of classifiers developed for different bioinformatics classification tasks. In this work, we showed that for some combination of developers' design decisions (e.g., randomly generated non-promoter sequences with k-mer features), cross-validation estimates might be misleading regarding the performance of the classifier. For example, a Naive Bayes classifier using 4-mer features has AUC = 0.87 on such cross-validation data, while its performance drops to AUC = 0.56 when evaluated using a high-quality test set. To avoid such misleading conclusions, independent test sets (when possible) should be used to evaluate the performance of the proposed prediction methods.

Sequence-based approaches for developing prokaryotic promoter region predictors are highly competitive to structure-based approaches evaluated in this study. However, a slight improvement in performance is observed when combining predictors based on the two approaches. More improvement could be obtained using: i) more sophisticated approaches for combining classifiers [68]; ii) building a single classifier using combined sequence and structure features and using feature selection algorithms to remove irrelevant or redundant features.

Finally, due to the lack of sufficient experimental data, we limited our experiments to E. coli σ^{70}promoter region predictions. Our future work aims at extending this work to cover other σ factors and explore the influence of the four factors considered in this study on the development of related DNA sequence prediction methods. Given that obtaining negative data is a challenge for most bioinformatics classification tasks, we conjecture that our findings apply not only to the problem of predicting E. coli σ^{70} promoter regions but also to other bioinformatics sequence classification tasks.

Supporting Information

S1 Dataset. Data sets used in this study.
(ZIP)

S1 Text. Detailed results on cross-validation and independent test sets.
(XLSX)

S1 Code. Matlab code for evaluating variable-window Z-curve method.
(ZIP)

Acknowledgments

The authors are grateful to Dr. Scheila de Avila e Silva of University of Caxias do Sul (Brazil) and Dr. H. Lin of University of Electronic Science and Technology (China) for kindly applying their methods to our independent test data and sharing predicted probabilities with us.

Author Contributions

Conceived and designed the experiments: YE. Performed the experiments: MMA YE. Analyzed the data: MMA MMM YE. Contributed reagents/materials/analysis tools: MMA YE. Wrote the paper: MMA YE.

References

1. Reznikoff WS, Siegele DA, Cowing DW, Gross CA (1985) The regulation of transcription initiation in bacteria. Annual review of genetics 19: 355–387. doi: 10.1146/annurev.ge.19.120185.002035 PMID: 3936407
2. Paget M, Helmann JD (2003) The sigma70 family of sigma factors. Genome Biology 4: 203. doi: 10.1186/gb-2003-4-1-203 PMID: 12540296
3. McClure WR (1985) Mechanism and control of transcription initiation in prokaryotes. Annual review of biochemistry 54: 171–204. doi: 10.1146/annurev.bi.54.070185.001131 PMID: 3896120
4. Tjaden B, Saxena RM, Stolyar S, Haynor DR, Kolker E, et al. (2002) Transcriptome analysis of escherichia coli using high-density oligonucleotide probe arrays. Nucleic acids research 30: 3732–3738. doi: 10.1093/nar/gkf505 PMID: 12202758
5. Sharma CM, Hoffmann S, Darfeuille F, Reignier J, Findeiß S, et al. (2010) The primary transcriptome of the major human pathogen helicobacter pylori. Nature 464: 250–255. doi: 10.1038/nature08756 PMID: 20164839
6. Wang C, Lee J, Deng Y, Tao F, Zhang L (2012) ARF-TSS: an alternative method for identification of transcription start site in bacteria. BioTechniques 2012: 1. doi: 10.1016/j.jbiotec.2011.06.034
7. Gordon J, Towsey M (2005) SVM based prediction of bacterial transcription start sites. In: Proceedings of the 6th international conference on Intelligent Data Engineering and Automated Learning. Springer-Verlag, pp. 448–453.
8. Gordon JJ, Towsey MW, Hogan JM, Mathews SA, Timms P (2006) Improved prediction of bacterial transcription start sites. Bioinformatics 22: 142–148. doi: 10.1093/bioinformatics/bti771 PMID: 16287942
9. Huerta AM, Collado-Vides J (2003) Sigma70 promoters in escherichia coli: Specific transcription in dense regions of overlapping promoter-like signals. Journal of molecular biology 333: 261–278. doi: 10.1016/j.jmb.2003.07.017 PMID: 14529615
10. Maetschke S, Towsey M, Hogan J (2006) Bacterial promoter modeling and prediction for E. coli and B. subtilis with beagle. In: Proceedings of the 2006 workshop on Intelligent systems for bioinformatics. Australian Computer Society, Inc., volume 73, pp. 9–13.
11. Wang H, Benham CJ (2006) Promoter prediction and annotation of microbial genomes based on DNA sequence and structural responses to superhelical stress. BMC bioinformatics 7: 248. doi: 10.1186/1471-2105-7-248 PMID: 16677393
12. Towsey MW, Gordon JJ, Hogan JM (2006) The prediction of bacterial transcription start sites using svms. International Journal of Neural Systems 16: 363–370. doi: 10.1142/S0129065706000767 PMID: 17117497
13. Towsey M, Hogan J, Mathews S, Timms P (2007) The in silico prediction of promoters in bacterial genomes. In: International Conference on Genome Informatics. volume 19, pp. 178–189.
14. Du Y, Wu T (2008) A novel method of prokaryotic promoter regions prediction with feature selection: quadratic discriminant analysis approach. In: 7th Asian-Pacific Conference on Medical and Biological Engineering. Springer, pp. 608–614.
15. Lin H, Li QZ (2011) Eukaryotic and prokaryotic promoter prediction using hybrid approach. Theory in Biosciences 130: 91–100. doi: 10.1007/s12064-010-0114-8 PMID: 21046474
16. Silva SdA, Gerhardt GJ, Echeverrigaray S (2011) Rules extraction from neural networks applied to the prediction and recognition of prokaryotic promoters. Genetics and molecular biology 34: 353–360. doi: 10.1590/S1415-47572011000200031
17. de Avila e Silva S, Echeverrigaray S, Gerhardt GJ (2011) BacPP: Bacterial promoter prediction?a tool for accurate sigma-factor specific assignment in enterobacteria. Journal of theoretical biology 287: 92–99. doi: 10.1016/j.jtbi.2011.07.017 PMID: 21827769
18. Song K (2012) Recognition of prokaryotic promoters based on a novel variable-window z-curve method. Nucleic acids research 40: 963–971. doi: 10.1093/nar/gkr795 PMID: 21954440
19. Bockhorst J, Qiu Y, Glasner J, Liu M, Blattner F, et al. (2003) Predicting bacterial transcription units using sequence and expression data. Bioinformatics 19: i34–i43. doi: 10.1093/bioinformatics/btg1003 PMID: 12855435

20. Burden S, Lin YX, Zhang R (2005) Improving promoter prediction improving promoter prediction for the NNPP2.2 algorithm: a case study using escherichia coli DNA sequences. Bioinformatics 21: 601–607. doi: 10.1093/bioinformatics/bti047 PMID: 15454410

21. Rangannan V, Bansal M (2007) Identification and annotation of promoter regions in microbial genome sequences on the basis of DNA stability. Journal of biosciences 32: 851–862. doi: 10.1007/s12038-007-0085-1 PMID: 17914227

22. Zhou X, Li Z, Dai Z, Zou X (2013) Predicting promoters by pseudo-trinucleotide compositions based on discrete wavelets transform. Journal of theoretical biology 319: 1–7. doi: 10.1016/j.jtbi.2012.11.024 PMID: 23211833

23. Kanhere A, Bansal M (2005) A novel method for prokaryotic promoter prediction based on DNA stability. BMC bioinformatics 6: 1–10. doi: 10.1186/1471-2105-6-1 PMID: 15631638

24. Vanet A, Marsan L, Sagot MF (1999) Promoter sequences and algorithmical methods for identifying them. Research in Microbiology 150: 779–799. doi: 10.1016/S0923-2508(99)00115-1 PMID: 10673015

25. Gordon L, Chervonenkis AY, Gammerman AJ, Shahmuradov IA, Solovyev VV (2003) Sequence alignment kernel for recognition of promoter regions. Bioinformatics 19: 1964–1971. doi: 10.1093/bioinformatics/btg265 PMID: 14555630

26. Polat K, Güneş S (2007) A novel approach to estimation of E. coli promoter gene sequences: Combining feature selection and least square support vector machine (FS_LSSVM). Applied mathematics and computation 190: 1574–1582.

27. Mahadevan I, Ghosh I (1994) Analysis of E. coli promoter structures using neural networks. Nucleic Acids Research 22: 2158–2165.

28. Rani TS, Bhavani SD, Bapi RS (2007) Analysis of E. coli promoter recognition problem in dinucleotide feature space. Bioinformatics 23: 582–588. doi: 10.1093/bioinformatics/btl670 PMID: 17237059

29. Demeler B, Zhou G (1991) Neural network optimization for E. coli promoter prediction. Nucleic acids research 19: 1593–1599. doi: 10.1093/nar/19.7.1593 PMID: 2027766

30. Pedersen AG, Baldi P, Brunak S, Chauvin Y (1996) Characterization of prokaryotic and eukaryotic promoters using hidden markov models. In: Ismb. Citeseer, volume 4, pp. 182–191.

31. Mann S, Li J, Chen YPP (2007) A pHMM-ANN based discriminative approach to promoter identification in prokaryote genomic contexts. Nucleic acids research 35: e12–e12. doi: 10.1093/nar/gkl1024 PMID: 17170007

32. Rani TS, Bapi RS (2009) Analysis of n-gram based promoter recognition methods and application to whole genome promoter prediction. In silico biology 9: S1–S16. PMID: 19537162

33. Li QZ, Lin H (2006) The recognition and prediction of σ^{70} promoters in escherichia coli k-12. Journal of theoretical biology 242: 135–141. doi: 10.1016/j.jtbi.2006.02.007 PMID: 16603195

34. Gama-Castro S, Salgado H, Peralta-Gil M, Santos-Zavaleta A, Muñiz-Rascado L, et al. (2011) RegulonDB version 7.0: transcriptional regulation of escherichia coli k-12 integrated within genetic sensory response units (gensor units). Nucleic acids research 39: D98–D105. doi: 10.1093/nar/gkq1110 PMID: 21051347

35. Salgado H, Peralta-Gil M, Gama-Castro S, Santos-Zavaleta A, Muñiz-Rascado L, et al. (2013) RegulonDB v8. 0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. Nucleic acids research 41: D203–D213. doi: 10.1093/nar/gks1201 PMID: 23203884

36. Benson DA, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, et al. (2013) Genbank. Nucleic Acids Research 1: 1–6. doi: 10.1093/nar/gkt559

37. Zhou J, Rudd KE (2013) Ecogene 3.0. Nucleic acids research 41: D613–D624. doi: 10.1093/nar/gks1235 PMID: 23197660

38. Sonnenburg S, Schweikert G, Philips P, Behr J, Rätsch G (2007) Accurate splice site prediction using support vector machines. BMC bioinformatics 8: S7. doi: 10.1186/1471-2105-8-S10-S7 PMID: 18269701

39. Chen W, Feng PM, Lin H, Chou KC (2014) iSS-PseDNC: Identifying splicing sites using pseudo dinucleotide composition. BioMed Research International 2014: 623149. doi: 10.1155/2014/623149 PMID: 24967386

40. Fletez-Brant C, Lee D, McCallion AS, Beer MA (2013) kmer-SVM: a web server for identifying predictive regulatory sequence features in genomic data sets. Nucleic acids research 41: W544–W556. doi: 10.1093/nar/gkt519 PMID: 23771147

41. Ivanov VI, Minchenkova LE, Chernov BK, McPhie P, Ryu S, et al. (1995) CRP-DNA complexes: Inducing the a-likeform in the binding sites with an extended central spacer. Journal of molecular biology 245: 228–240. doi: 10.1006/jmbi.1994.0019 PMID: 7844815

42. Ohler U, Niemann H, Liao Gc, Rubin GM (2001) Joint modeling of DNA sequence and physical properties to improve eukaryotic promoter recognition. Bioinformatics 17: S199–S206. doi: 10.1093/bioinformatics/17.suppl_1.S199 PMID: 11473010

43. Olson WK, Gorin AA, Lu XJ, Hock LM, Zhurkin VB (1998) DNA sequence-dependent deformability deduced from protein—DNA crystal complexes. Proceedings of the National Academy of Sciences 95: 11163–11168. doi: 10.1073/pnas.95.19.11163

44. Sivolob AV, Khrapunov SN (1995) Translational positioning of nucleosomes on DNA: the role of sequence-dependent isotropic DNA bending stiffness. Journal of molecular biology 247: 918–931. doi: 10.1006/jmbi.1994.0190 PMID: 7723041

45. Blake R, Delcourt SG (1998) Thermal stability of DNA. Nucleic acids research 26: 3323–3332. doi: 10.1093/nar/26.14.3323 PMID: 9649614

46. Ho PS, Ellison MJ, Quigley GJ, Rich A (1986) A computer aided thermodynamic approach for predicting the formation of Z-DNA in naturally occurring sequences. The EMBO journal 5: 2737–2744. PMID: 3780676

47. Breslauer KJ, Frank R, Blöcker H, Marky LA (1986) Predicting DNA duplex stability from the base sequence. Proceedings of the National Academy of Sciences 83: 3746–3750. doi: 10.1073/pnas.83.11.3746

48. Sugimoto N, Nakano Si, Yoneyama M, Honda Ki (1996) Improved thermodynamic parameters and helix initiation factor to predict stability of DNA duplexes. Nucleic acids research 24: 4501–4505. doi: 10.1093/nar/24.22.4501 PMID: 8948641

49. El Hassan M, Calladine C (1996) Propeller-twisting of base-pairs and the conformational mobility of dinucleotide steps in DNA. Journal of molecular biology 259: 95–103. doi: 10.1006/jmbi.1996.0304 PMID: 8648652

50. Ornstein RL, Rein R, Breen DL, Macelroy RD (1978) An optimized potential function for the calculation of nucleic acid interaction energies I. base stacking. Biopolymers 17: 2341–2360. doi: 10.1002/bip.1978.360171005 PMID: 24624489

51. Gan Y, Guan J, Zhou S (2012) A comparison study on feature selection of dna structural properties for promoter prediction. BMC bioinformatics 13: 4. doi: 10.1186/1471-2105-13-4 PMID: 22226192

52. EL-Manzalawy Y, Bui N, Sridharan K, Brendel V, Honavar V (2015). Gennotate: Genome annotation toolkit. Available: http://ailab.ist.psu.edu/gennotate. Accessed 6 February 2015.

53. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, et al. (2009) The WEKA data mining software: an update. ACM SIGKDD explorations newsletter 11: 10–18. doi: 10.1145/1656274.1656278

54. Mitchell TM (1997) Machine learning. McGraw-Hill Boston, MA.

55. Breiman L (2001) Random forests. Machine learning 45: 5–32. doi: 10.1023/A:1010933404324

56. Vapnik VN (1995) The nature of statistical learning theory. Springer-Verlag New York, Inc.

57. Breiman L (1996) Bagging predictors. Machine learning 24: 123–140. doi: 10.1023/A:1018094028462

58. Rabiner L (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77: 257–286. doi: 10.1109/5.18626

59. Baldi P, Brunak S, Chauvin Y, Andersen CA, Nielsen H (2000) Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 16: 412–424. doi: 10.1093/bioinformatics/16.5.412 PMID: 10871264

60. Fawcett T (2006) An introduction to ROC analysis. Pattern recognition letters 27: 861–874. doi: 10.1016/j.patrec.2005.10.010

61. Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern recognition 30: 1145–1159. doi: 10.1016/S0031-3203(96)00142-2

62. Swets JA (1988) Measuring the accuracy of diagnostic systems. Science 240: 1285–1293. doi: 10.1126/science.3287615 PMID: 3287615

63. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research 7: 1–30.

64. Chen W, Feng PM, Lin H, Chou KC (2013) iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition. Nucleic Acids Research: gks1450.

65. Guo SH, Deng EZ, Xu LQ, Ding H, Lin H, et al. (2014) iNuc-PseKNC: a sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition. Bioinformatics 30: 1522–1529. doi: 10.1093/bioinformatics/btu083 PMID: 24504871

66. Yegnanarayana B (2009) Artificial neural networks. PHI Learning Pvt. Ltd.

67. Zhang R, Zhang CT (2014) A brief review: The z-curve theory and its application in genome analysis. Current genomics 15: 78–94. doi: 10.2174/1389202915999140328162433 PMID: 24822026

68. Wozniak M (2013) Hybrid Classifiers: Methods of Data, Knowledge, and Classifier Combination, volume 519 of *Studies in Computational Intelligence*. Springer.