2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Issues on Sampling Negative Examples for Predicting Prokaryotic Promoters

Eduardo G. Gusmão
RWTH Aachen University Medical School
Institute for Biomedical Engineering
IZKF Aachen Computational Biology Research Group
Aachen, Germany
Email: eduardo.gusmao@rwth-aachen.de

Marcilio C. P. de Souto
Univ. Orléans
INSA Centre Val de Loire
LIFO EA 4022
FR-45067, Orléans, France
Email: marcilio.desouto@univ-orleans.fr

***Abstract*—Supervised learning methods have been successfully used to build classifiers for the identification of promoter regions. The classifier is often built from a dataset that has examples of promoter (positive) and non-promoter (negative) regions. Thus, a careful selection of the data used for constructing and evaluating a promoter finding algorithm is a very important issue. In this context, experimentally known promoter regions can be safely assumed to be positive training instances. In contrast, since definite knowledge whether a given region represents a non-promoter is not generally available, negative instances are not straightforward to be obtained. To make the problem more complex, for the case of promoter, there is not a unique definition of what a negative instance is. As a consequence, depending on which definition of non-promoter region one assumed to build the data, such a choice could affect significantly the performance of the classifier and/or yield a biased estimate of the performance. We present an empirical study of the effect of this kind of problem for promoter prediction in *E. coli*. As far as we are concerned, up to now, there is no such a kind of study for the context of prokaryotic promoter prediction.**

I. Introduction

The problem with negative sampling is common in many areas of bioinformatics such as prediction of mRNAs that are target of miRNAs, regulatory networks, protein-protein interactions, non-coding RNA finding, among others [1]–[4]. For instance, in the studies of miRNA target the majority of the methods suffer from high rates of false positives or false negatives. This happens mostly because systematic identification of mRNAs not proven to be target of miRNAs is still not addressed properly. Thus, current machine learning methods have to rely on artificially generated negative examples for training [1]. Likewise, regulatory networks modeling and protein-protein interactions share the problem that definite knowledge is typically not available that a given pair of genes, proteins or other products under study do not interact [2], [3]. This problem also arises in the context of prokaryotic promoter prediction. In fact, in this paper, we present an empirical study of the effect of this problem – negative sampling – in the context of prokaryotic promoter prediction.

Marcilio C. P. de Souto is also with Univ. Federal de Pernambuco, Centro de Infomática (CIn), Recife, Brazil. This work was partially supported by the Interdisciplinary Center for Clinical Research (IZKF Aachen), RWTH Aachen University Medical School, Aachen, Germany; and Brazilian research agency FACEPE.

Promoter prediction in prokaryotes is often modeled as a binary classification task: the algorithm should generate a classifier able to discriminate between promoter (positive) and non-promoter (negative) regions. These samples are frequently nucleotide sequences (primary sequence) or features extracted from them.

There is a great deal of work in this context of promoter prediction in prokaryotes as a classification problem. In early studies, Norton [5] made interesting observations about the methodological issues concerning sequence extraction from consensus genomes. He developed a probabilistic method to predict promoters by evaluating uncertainty in the training data. Helmann [6] analyzed 236 *B. subtilis* promoters recognized by σ^A-RNA polymerase and demonstrated interesting characteristics concerning conserved positions and dinucleotide frequency patterns.

More recently, Dhar [7] reviewed the use of features such as curvature, bendability and stability to try to build more accurate classifiers. Wu et al [8] improved the prediction accuracy with a method that considered the correlations between nucleotides at the construction of position weight matrices representing the promoters. In another interesting work, the authors performed neural networks simulations based both on the primary sequences (nucleotide sequences) and other features expressing sequences in terms of their free energy [9].

However, independently of the algorithm used to generate the classifiers or the type of attributes used to represent the instances in the dataset (primary sequence and/or its features), one methodological issue that researchers have to account for is the construction/choice of the negative instances. This is the main of concern of this paper. For example, currently there are many experimentally known promoter regions that can safely be assumed to be positive training instances. In contrast, it is experimentally complex to prove that a sequence does not contain a promoter. As a consequence, negative examples have been obtained in alternative ways, each one of them leading to different "definitions" for what a negative example is.

Thus, depending on which definition (or criterion) of non-promoter region one assumes to build the data, such a choice could affect significantly the performance of the classifier and/or yield a biased estimate of the performance. In this paper, as our main contribution, we present an empirical study of the effect of this kind of problem. In order to do so, we use the

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

promoter prediction in *E. coli* as case study. As far as we are concerned, up to now, there is no such a kind of analysis for the context of prokaryotic promoter prediction.

II. Related Works

As mentioned before, the problem with negative sampling is common in many areas of bioinformatics. One of the strategies to deal with it that has provided good results is the use of positive instances only, with a robust statistical background [2], [4], [10].

In [4], for instance, the authors presented an algorithm, positive sample only learning (PSoL), which combined with support vector machine (SVM) created a powerful tool for finding non-coding RNA genes.

Other interesting way to deal with the problem is the establishment of benchmarks, standardizing the definitions of negative examples and providing high-quality datasets. In the context of protein-protein interactions, for example, a repository (Negatome Database) was constructed containing proteins that have high probability not to interact [11].

So far, in the context of prokaryotic promoter prediction, in the lack of more grounded guidelines, the researchers build their negative samples in different ways. For instance, in order to create the sample of negative instances, the authors in [12] take, randomly, fragments that are not included in the positive sample. In another work, the authors extract at random sequences from coding and non-coding regions [13], [14]. There are also studies in which fragments from a related organism, such as phages, are taken as negatives instances [15], [16].

III. Materials and Methods

We perform an empirical study on the impact of the choice of negative examples in the performance of classifiers built for the task of promoter prediction, using a dataset of *E. coli* as case study. Such a problem is investigated from the perspective in which the *primary DNA sequences*/sequences of nucleotides are given directly as input to the learning algorithm (scenario 1), as well as in the context in which the input for the learning algorithms are *features extracted from the sequences* (scenario 2). These two scenarios were chosen because they span most of the prokaryotic promoter prediction methods available. In addition, using this experiment design we are able to address both categorical and continuous attributes.

In scenario 1, the attributes are categorical and can take values in $\{A, C, G, T\}$, which are the representations of the nucleotides that compose the DNA. In contrast, the attributes in scenario 2 are often continuous. More specifically, in the case of scenario 2, we use the variable-window Z-curve (vw Z-curve) feature extraction method presented in [17]. This method is used because it addresses distribution of purine/pyrimidine, amino/keto and strong/weak H-bonds in a robust manner. Furthermore, it exhibits, as far as we are concerned, the best accuracies up to now.

In terms of machine learning techniques, for both scenarios, we apply a rule based system (decision trees) and statistical learning systems such as naive Bayes classifier and support vector machines. Additionally, for the scenario in which the attributes are the features extracted with vw Z-curve method, we apply the partial least squares algorithm as in [17].

Since the main focus of this work is the comparison of the impact of the different negative sampling methods on the classification, we have chosen these techniques as they provide different learning frameworks. All the learning methods used in our study were obtained from the Matlab and Statistics Toolbox release R2012a.

Next, we present the datasets that we use in the experiments. Then, we briefly introduce the vw Z-curve feature extraction method. Finally, we discuss the methodology used to evaluate the experiments.

A. Datasets

As previously mentioned, the main focus of this study is the evaluation of the impact of specific biases associated with the use of different definitions of negative samples when building a dataset. In order to do so, based on various definitions commonly found in the literature of our case study (prokaryotic promoter prediction), we gathered a representative collection of negative datasets. As general guideline to construct our data, we use the work in [14], [17].

Positive Examples. DNA sequences known to be bound by σ factors (promoters) were built/collected as described in [14], [17]. Briefly, we obtained 812 sequences around transcription start sites (TSSs), representing promoters regions, from RegulonDB [18]. We use promoters from *E. coli* that contain motifs recognized by σ^{19}, σ^{24}, σ^{28}, σ^{32}, σ^{38}, σ^{54} and σ^{70}. All sequences have 80bp length spanning [TSS-60,TSS+19] as used in [17]. Only experimentally verified promoters are used, that is, promoters predicted by transcription initiation mapping or RNA polymerase footprinting, which, according to RegulonDB criteria, are the methods that provide strong evidence. Hereafter, we refer to this dataset as `POS`.

Negative Examples. With respect to the data representing the negative examples, as will be explained in the following, we define different sets taking as basis if the sequences come from coding regions, non-coding regions or from random segments of the genome. For certain cases, some sequences needed to be extracted from *E. coli* complete genome and coding regions. This was accomplished using the database at NCBI [19] for *E. coli* K-12 MG1655. All the negative sequences, as in the case of the positive ones, consist of 80 nucleotides. In order to exclude any problems due to class imbalance and to facilitate specificity comparisons, all negative datasets were restricted to contain 812 sequences. In all cases where the datasets were obtained in previous studies and contained more than 812 examples, sequences were randomly drawn in order to fulfill this criterion.

- **Coding Negative Examples**. As pointed in [14], 81% of known *E. coli* K-12 TSSs are located in the intergenic non-coding regions and 19% in the coding regions. Based on this, we decided to include the coding data present in [14], [17]. Basically, they picked genomic sequences at random from the start of open reading frames (ORFs) in *E. coli* known

coding regions. We denote such dataset as COD1. For reasons that we will discuss later, we propose a second coding dataset (COD2) consisting of sequences randomly drawn from any part of the coding regions, not only from the start of the ORFs as in [14], [17].

- **Non-coding Negative Examples**. Like [14], [17], we also decided to include in our analysis the non-coding examples from a random sample of non-coding convergent intergenic spacers [20]. We will refer to this dataset as NCOD.

- **Random Negative Examples**. Following the methodology in Bland et al [12], we created these data from sequences chosen at random. In order to do so, we extracted at random fragments from the *E. coli* genome, but with the constraint that these sequences cannot have any overlap with the sequences from the positive set (POS). Comparing our methodology to the one in [12], as the latter chooses the sequences completely at random, the sequences could belong to coding or non-coding regions. It is important to point out that since 89% of *E. coli* genome corresponds to coding regions [21], this dataset will be approximately 89% coding and 11% non-coding. Such dataset will be denoted by RAND.

- **Miscellaneous Negative Examples**. The methodology used to generate each of the previous negative datasets, with exception, to a certain degree, of RAND, will not include sequences from different parts of the genome. For example, COD1 and COD2 will not present any fragment from non-coding convergent intergenic spacers (NCOD). However, from a practical point of view, when one scans a genome looking for promoters, the systems will have as input fragments from any part of the genome. Motivated by this, we propose to create two additional negative datasets. The first dataset consists of a random sampling of 50% of the sequences from COD1 and 50% of the sequences from NCOD, which will be denoted by MIX1. The second dataset, denoted by MIX2, follows the same procedure, but using COD2 and NCOD.

- **Control Negative Examples**. To put the results into perspective, we created a "synthetic" negative dataset. Such a dataset is composed of sequences that were generated completely at random, that is, they were not picked from any part of a genome. In order to correct for CG content, the nucleotide frequencies considered to generate this dataset followed the background distribution for *E. coli* genome. We will denote this dataset as CTRL.

The datasets previously defined include most of the definitions of positive and negative examples used in the literature of prokaryotic promoter prediction analysis. A summary of all these datasets is presented in Table I. The first column represents the name of the dataset, the second column provides a short description and the remaining columns represent the frequencies of the nucleotides in each dataset.

B. Variable-window Z-Curve Feature Extraction

The variable-window Z-curve (vw Z-curve) feature extraction, developed by Song [17], consists of a variation of the regular Z-curve approach proposed by Zhang [22]. They can both be used to extract numerical features from a nucleotide sequence. The main idea of the original Z-curve theory is that a 3D curve (or point) representation for a DNA sequence can be created in the sense that each can be uniquely reconstructed given the other. The original Z-curve is calculated from the frequencies of the four bases occurring in the sequence and considers three main components: distribution of purine/pyrimidine, distribution of amino/keto and distribution of strong/weak H-bonds. The regular Z-curve parameters are only derived from the frequencies of mononucleotides occurring in a DNA sequence. Song claims this is a limitation for the case of promoter recognition and modified this idea by introducing a variable-window that allows the point to be in a much higher dimension. More formally, the vw Z-curve method can be defined as follows [17].

Let S_w^i be a string constructed by picking w elements from the set $\{A, C, G, T\}$ with order and repetition, where w is defined as the window length and $i = 1, ..., w^4$. For example, when $w = 2$, $S_2^1 = AA$, $S_2^2 = AC$, ..., $S_2^{16} = TT$. Let the frequency of the pattern $S_w^i X$ be denoted by $p\left(S_w^i X\right)$, where $X \in \{A, C, G, T\}$. The following equation shows the uniform definition of the vw Z-curve variables.

$$
\begin{array}{rcl}
x_{S_w^i} & = & [p\left(S_{w-1}^i A\right) + p\left(S_{w-1}^i G\right)] - \\
 & & [p\left(S_{w-1}^i C\right) + p\left(S_{w-1}^i T\right)] \\
y_{S_w^i} & = & [p\left(S_{w-1}^i A\right) + p\left(S_{w-1}^i C\right)] - \\
 & & [p\left(S_{w-1}^i G\right) + p\left(S_{w-1}^i T\right)] \\
z_{S_w^i} & = & [p\left(S_{w-1}^i A\right) + p\left(S_{w-1}^i T\right)] - \\
 & & [p\left(S_{w-1}^i C\right) + p\left(S_{w-1}^i G\right)] \\
\text{where} & : & w \in \mathbb{N} \quad \text{and} \quad i = 1, 2, ..., 4^{w-1}
\end{array}
\tag{1}
$$

Each window w generates $3 * 4^{w-1}$ features. For example, given a certain sequence (independently of its length), if we generated its respective vw Z-curve, with $w = 1, ..., 6$, such a sequence will be represented by a real-valued vector with 4095 attributes.

C. Evaluation

In order to compare the impact of the datasets on the performance of the classifiers we will perform a 10-fold cross validation using four supervised learning techniques: support vector machine (SVM), naive Bayes classifier (NB), decision trees (DT) and partial least squares (PLS) [17], [23]. In every fold, classifiers are generated by training the supervised learners with a combination of the positive dataset (POS) with every other negative dataset.

We investigate the performance of classifiers in two different scenarios. The first scenario regards the performance results yielded by the usual 10-fold cross validation procedure. For example, given the dataset (POS + COD1), by applying 10-fold cross validation, we can build 10 classifiers. Then, the

TABLE I. LIST OF DATASETS

Dataset	Description	A	C	G	T
POS	Known promoters	29.04	20.48	20.00	30.48
COD1	Start of coding regions	26.62	22.29	24.88	26.21
COD2	Random part of coding region	24.19	24.58	27.21	24.02
NCOD	Non coding region	23.94	25.01	26.78	24.27
RAND	Random non-promoter region	24.46	25.79	25.34	24.41
MIX1	50% COD1 + 50% NCOD	25.47	23.35	25.83	25.35
MIX2	50% COD2 + 50% NCOD	24.02	24.69	27.14	24.15
CTRL	Completely random sequences	24.62	25.42	25.37	24.59

result will be the average of the performance of each one of these classifiers tested with its respective testing set drawn from (POS + COD1).

In the second scenario, we test classifiers generated with a given dataset with different negative examples sets. For example, given the dataset (POS + COD1), by applying 10-fold cross validation, we build 10 classifiers. Then, we can test the performance of these classifiers with, for instance, the dataset POS + NCOD. Obviously that, in terms of the positive examples, the performance will be the same for both cases. However, in terms of negative examples, as the classifiers were trained with COD1, they could have more difficulty in correctly assigning examples coming from completely different dataset (NCOD).

In summary, in terms of challenging the classifiers, the first scenario should be "easier" than the second one, since the testing examples are taken from the same distribution used to create the classifiers. Hereafter, we refer to these two scenarios, respectively, as **Case study 1** and **Case study 2**.

We used three metrics to evaluate the performance of the classifiers: correct rate (Cr), sensitivity (Sn) and specificity (Sp). Table II presents these statistics, assuming TP, FP, TN and FN are, respectively, the number of true positives, false positives, true negatives and false negatives. In order to evaluate whether performance variations between different methods were significant, we applied a two-tailed paired t-test using a confidence level of 95%.

TABLE II. PERFORMANCE METRICS IN TERMS OF TRUE POSITIVES (TP), TRUE NEGATIVES (TN), FALSE POSITIVES (FP) AND FALSE NEGATIVES (FN)

Correct Rate	Sensitivity	Specificity
$\frac{TP+TN}{TP+FP+TN+FN}$	$\frac{TP}{TP+FN}$	$\frac{TN}{TN+FP}$

IV. EXPERIMENTS AND RESULTS

In our experiments we use (1) the primary sequence datasets (categorical attributes) and (2) datasets whose attributes and characteristics were extracted by the vw Z-curve method (numeric attributes). For the latter, following the guidelines in [17], we first generated their respective vw Z-curve datasets, with $w = 1, ..., 6$. This yields 4095 features for each sequence. Then, we selected 300 features from the 4095 using the PLS-based feature selection described in [17]. This number of selected features was chosen based on the accuracies reported by Song under different number of features for *E. coli* datasets.

In terms of machine learning techniques, for both contexts, we apply decision trees (DTs), naive Bayes classifier (NB) and support vector machines (SVMs). Also, for the scenario in which the attributes are the features extracted with vw Z-curve method, we apply the partial least squares (PLS) algorithm as in the work in [17]. In addition, a two-tailed paired t-test at the 5% significance level is performed in order to assess the statistical significance of the results obtained. All the learning methods used in our study were obtained from the Matlab. We use the default parameters for DTs, NB and PLS. For the case of SVMs, after performing some pre-experiments, we chose to use the polynomial kernel with the exponent set to 3 and the complexity factor (C) set to 0.5.

In some further analyses, the Multiple EM for Motif Elicitation (MEME) algorithm was used [24] in order to find enriched motifs within the studied datasets. We searched for the top 5 enriched motifs with length between 3 and 10, which were enriched in a minimum of 100 instances. All other parameters were set to default values.

Case study 1: the performance assessment with the usual 10-fold cross validation procedure

Figure 1 illustrates the performance metrics (correct rate, sensitivity and specificity) for all classifiers obtained by performing the usual 10-fold cross validation procedure on the primary sequence datasets and features extracted with vw Z-curve.

Looking at Figure 1A, at first glance, one can observe that the SVM performed better than the other classifiers (p-value regarding correct rate ranging from 2.5×10^{-3} to 3.4×10^{-9}). Except for COD1, the specificity of the SVM was usually lower than its sensitivity. Differently, for NB and DT, the difference between sensitivity and specificity was not large.

The most noticeable result is the significant (p-value regarding specificity $< 10^{-5}$) large values for the statistics obtained by training and testing with COD1. This can be explained by the fact that COD1 was created by extracting sequences from the start of gene ORFs. That is, such a negative example dataset is the only one that has a highly conserved ATG motif (MEME E-value = 1.3×10^{-8}). It is important

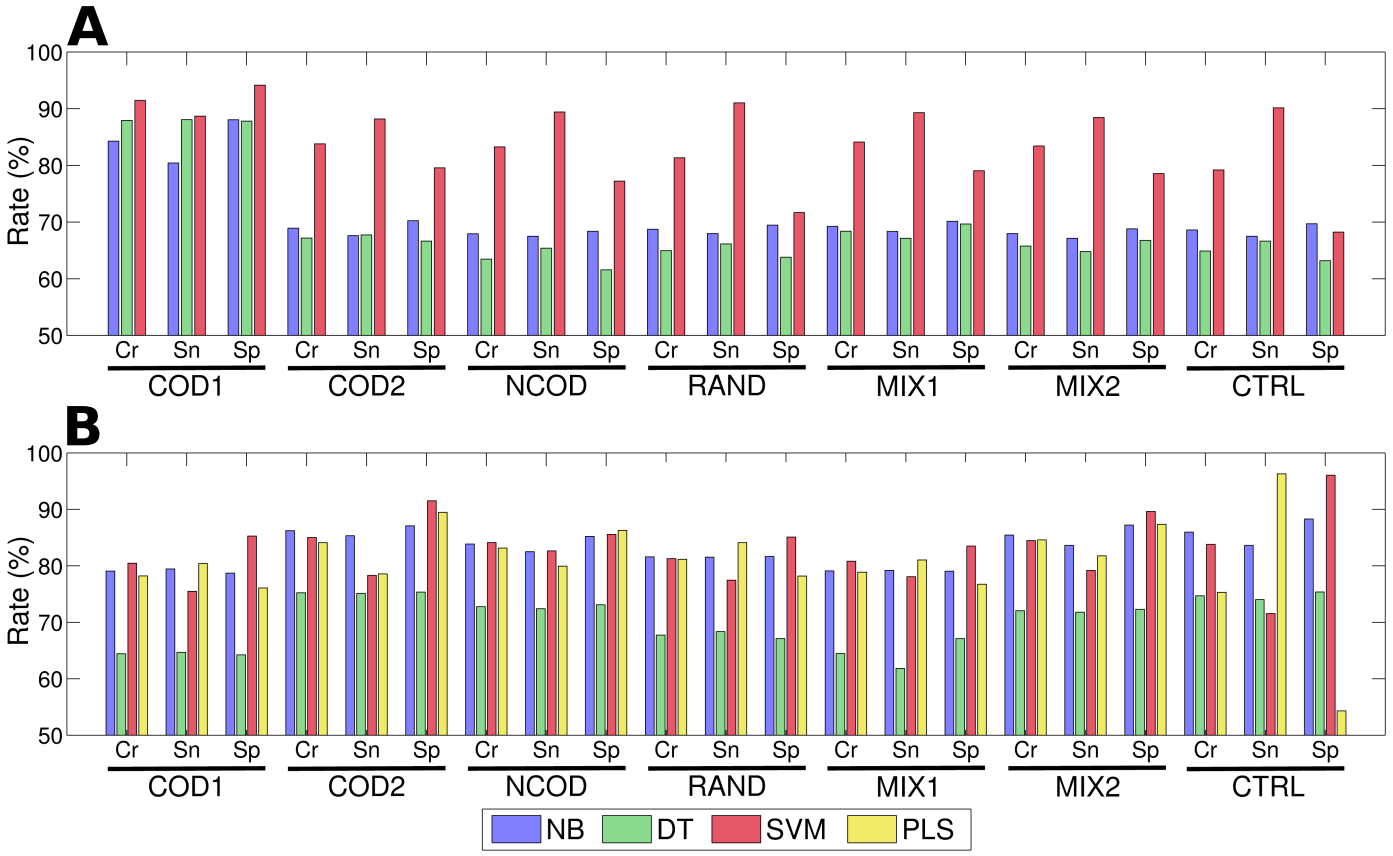

Fig. 1. Correct rate (Cr), sensitivity (Sn) and specificity (Sp) for all classifiers obtained by performing the usual 10-fold cross validation procedure according to case study 1 on the primary sequence datasets (**A**) and features extracted with vw Z-curve (**B**).

to notice that, although the ATG motif is present in the top-scored MEME result for the `POS` dataset, its position varies within the sequence, differently from `COD1`, which explains the large MEME E-value of 1.3×10^{19}. In this categorical scenario, this explains the highly optimistic results for `COD1`, since no other dataset contained an ATG motif among the top 5 enriched motifs based on MEME E-values (see Figure 2).

Now, we turn our analysis to the same data, but, instead of using directly the primary sequences, we consider as attributes the features extracted via the vw Z-curve (Figure 1B). NB, SVM and PLS performed statistically similarly for all the datasets (p-value > 0.1). On the other side, DT had much lower rates in comparison to the other methods (p-values ranging from 4.4×10^{-4} to 5.7×10^{-8}). As DT is not very appropriate to deal with numeric attributes, this is an expected result.

Comparing the results in Figure 1A and B, one can observe slightly lower (p-value regarding specificity $< 10^{-3}$) results for `COD1` (and `MIX1`, which consists of 50% of `COD1`). The explanation for this phenomenon probably also lies in the enriched ATG motif present in `COD1`. Since the vw Z-curve captures features in the DNA sequences, the positional information of this ATG motif is diluted in a numerical representation of such triplet. Since both `POS` and `COD1` contain ATG motifs, the classifiers have more difficulty in separating these datasets with similar features. This result supports our idea that the `COD1` dataset provides biased results.

Besides the above statement, no great variations can be observed. Methods based on the use of different features (properties) extracted from the primary sequence as attributes, instead of using directly the sequence, have been gaining popularity. These results also suggest that, in addition to improving prediction power by providing a wider range of information, this methodology minimizes the biases originating from sequence alignment issues.

Case study 2: the performance assessment with different negative datasets

In this scenario, we compare the performance of the classifiers when the datasets used for training and testing differ. This case study reflects, for instance, exploratory analyses in which not much information is known for a particular organism and researchers usually apply classifiers built with a particular negative dataset in a genome-wide fashion. Since this comparison generates many performance measurements, we will focus our attention to the SVM classifier, which was shown to provide the best performance in our first case study. Table III provides specificity rates obtained by the application of SVM classifier to all combinations of training and testing sets. The upper part of the table regards the primary sequences datasets, whereas the lower part contains the results for vw Z-curve datasets.

For primary sequence datasets, the bias generated by the presence of ATG motif in `COD1` is now evident: the specificity

TABLE III. SPECIFICITY FOR TRAINING AND TESTING WITH SVM CLASSIFIER – CASE STUDY 2

			Testing						
			COD1	COD2	NCOD	RAND	MIX1	MIX2	CTRL
Training	Sequence	COD1	94.14	39.43	38.30	35.83	68.80	41.93	31.26
		COD2	52.79	79.55	78.08	75.11	64.07	87.39	69.93
		NCOD	50.36	77.98	77.21	72.87	73.33	87.87	68.81
		RAND	47.98	75.73	73.67	71.67	58.39	74.80	66.06
		MIX1	95.25	67.61	82.39	63.88	79.04	75.76	58.40
		MIX2	54.90	87.79	87.83	73.88	70.40	78.55	69.47
		CTRL	48.22	73.74	74.76	71.93	59.75	74.27	68.23
	vw Z-curve	COD1	85.29	91.91	84.88	82.14	91.84	88.29	72.98
		COD2	71.44	91.51	85.16	80.11	77.84	92.66	62.13
		NCOD	61.69	84.84	85.58	80.50	79.72	92.08	56.43
		RAND	69.00	87.39	89.25	85.10	78.84	88.81	69.77
		MIX1	87.33	91.06	93.60	84.96	83.49	91.96	68.15
		MIX2	68.76	94.00	91.85	80.25	79.57	89.64	64.84
		CTRL	63.18	75.73	79.52	79.27	70.17	76.12	96.06

rates suffer a dramatic drop when COD1 is used as a training set and other datasets are used as testing set (first row of the upper part of the table). An interesting result is the fact that, for certain contexts, we can observe higher rates for testing with other dataset rather than the one used to build the classifier. For example, in the case of MIX2, the classifiers presented better performance when tested either with COD2 and NCOD. Since the application of novel methodologies will be done in a much noisier environment that the controlled 10-fold cross-validation, this could be seen as an evidence that using multiple datasets is essential for any study.

In addition to the training bias, the use of COD1 also makes clear a testing bias that can be seen by verifying the low specificity rates when training with other datasets and testing with COD1 (first column of the upper part of the table). At this context, mixing datasets methodology has proven to minimize this kind of bias, though it is still noticeable its preference for its own instances when testing.

In contrast to what we have discussed previously for primary sequence datasets, the results of the vw Z-curve datasets did not present the training bias generated by the conserved ATG in COD1. Indeed, the behavior observed was the opposite. When training with COD1, the values for testing with COD2, MIX1 and MIX2 were moderately larger. This unexpected fact led us to conduct a more careful analysis of the structure of these datasets.

In order to do so, we calculated the centroid (average vector) for each dataset in Table I. Then, we computed the Euclidean distance between the centroids of each dataset – inter-dataset distance (see Table IV). Based on this, a probable explanation for the "inverse" COD1 bias for vw Z-curve datasets is that the distance between the centroids of POS and COD2 is greater than that of between POS and COD1 (respectively, 2.77 and 2.13), while the distance between COD1 and COD2 is much lower (1.62). From a geometrical point of view, whatever the decision boundary between POS and COD1 is, this same boundary would be able to correctly classify examples from COD2, which are closer to the instance in COD1, but more distant from the examples in POS.

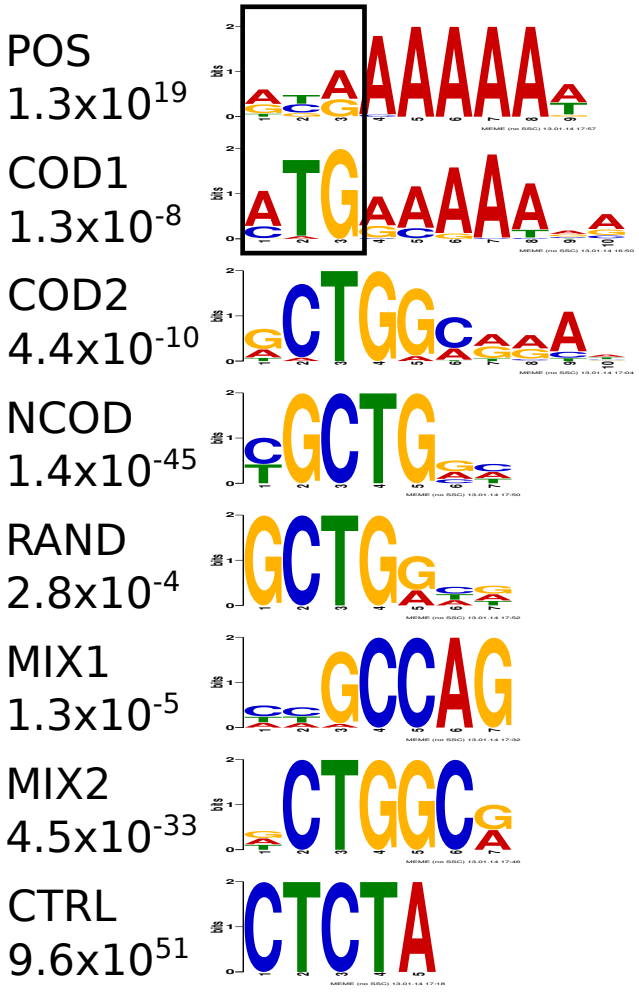

Fig. 2. Top enriched motifs for all datasets studied according to MEME algorithm. Below each dataset name it can be seen the MEME E-value score and, on the right, that dataset's top enriched motif logo. The enriched ATG motifs for the POS and COD1 datasets are marked with a black square.

TABLE IV. DISTANCE BETWEEN THE CENTROIDS OF VW Z-CURVE DATASETS

	POS	COD1	COD2	NCOD	RAND	MIX1	MIX2
COD1	2.13						
COD2	2.77	1.62					
NCOD	2.67	1.78	1.42				
RAND	2.48	1.71	1.55	1.17			
MIX1	2.26	1.00	1.36	1.09	1.31		
MIX2	2.67	1.65	0.89	0.93	1.30	1.16	
CTRL	2.55	2.43	2.75	2.56	2.38	2.42	2.59

Discussion

Having analyzed the results of our experiments, we point out to the fact that when applying any methodology in real biological datasets, the learners face data that do not behave so well as in *in silico* experimental pipelines. "Real world" data consists of general and specific biases and noises that may differ depending on simple actions such as parameter setting. In addition, there are several types of methodological details that, when incorrectly treated, often lead to biased results. For instance, the dataset `COD1`, used in [14], [17], contains a conserved motif. Such a motif, in terms of methods that use primary sequence as input, leads the classifiers generated to classify incorrectly negative examples coming from different distributions.

Moreover, since the goal of any genome-wide prediction technique is to run a classifier and retrieving the regions that are more likely to be enriched for what is expected it to be, using coding and non-coding negative samples in separate does not necessarily addresses the actual problem. In this context, although our analysis has shown no significant bias related to the `RAND` dataset, the original definition made in [12] excluded, from the negative dataset, regions within 50bp from the TSS. These regions are known to have lower stress-induced duplex destabilization (SIDD) profile levels, which is a characteristic of a promoter region. Thus, in generating their negative examples in this way, they somehow prevent getting into the dataset segments of genomes that could make the classification, in this context of SIDD profile features, harder. In fact, as demonstrated in our experiments, when considered out of the SIDD profile context, picking sequences at random from the genome could lead to a dataset that provides a pessimistic view of the classifiers generated.

In summary, we suggest that in studies that the negative examples are hard to determine or supporting evidence is not present, the performance assessment should be accomplished by spanning the largest possible number of scenarios. This includes (1) training and testing with different datasets corresponding to different interpretations of negative samples, (2) generating other datasets, such as combinations of the original datasets, (3) assessing the performance through a wider range of statistics, and (4) exploring other methodological possibilities such as semi-supervised learning and positive only prediction.

V. FINAL REMARKS

In this work, we discussed a number of different biases and misinterpretations that often occur in bioinformatics studies concerning the use of supervised learners when negative instances are hard to obtain or define. More specifically, we present an empirical study of the effect of this kind of problem for promoter prediction in *E. coli*. To support our discussion we created two evaluation scenarios where a 10-fold cross-validation was applied to 14 different datasets, composed of sequences and features extracted from these sequences, using four machine learning methods. We showed specific and general biases that happened under these circumstances and suggested evaluation criteria to assess the actual performance of novel algorithms over a wide range of bioinformatics fields.

This study can be further expanded to encompass other evaluation scenarios. For example, other features extracted from the sequences can be used such as SIDD profile [12] or DNA stability based on free energy [9]. Moreover, the extension of the analyses made in this study to other organisms would probably generate further insights. In this case, more complex negative datasets could be explored such as the ones generated by higher-order Markov chains in order to model nucleotide dependencies. Other possibility is the use of a wider number of classifiers, allowing for a detailed discussion on the specific bias of each learner. One can also compare other approaches, such as non-supervised, semi-supervised or positive sample only learning, to the classical supervised methodology [2], [4], [10].

ACKNOWLEDGMENT

We would like to thank K. Song for providing the code regarding the vw Z-curve method.

REFERENCES

[1] S. Bandyopadhyay and R. Mitra, "TargetMiner: microRNA target prediction with systematic identification of tissue-specific negative examples," *Bioinformatics*, vol. 25, no. 20, pp. 2625–2631, Oct. 2009.

[2] L. Cerulo *et al.*, "Learning gene regulatory networks from only positive and unlabeled data," *BMC Bioinformatics*, vol. 11, no. 1, pp. 228+, May 2010.

[3] Y. Park and E. M. Marcotte, "Revisiting the negative example sampling problem for predicting protein-protein interactions," *Bioinformatics*, vol. 27, no. 21, pp. 3024–3028, Nov. 2011.

[4] C. Wang *et al.*, "PSoL: a positive sample only learning algorithm for finding non-coding RNA genes," *Bioinformatics*, vol. 22, no. 21, pp. 2590–2596, Nov. 2006.

[5] S. W. Norton, "Learning to recognize promoter sequences in *E. coli* by modeling uncertainty in the training data," in *Proceedings of the twelfth National Conference on Artificial Intelligence*. AAAI Press/MIT Press, 1994, pp. 657–663.

[6] J. D. Helmann, "Compilation and analysis of Bacillus subtilis sigma A-dependent promoter sequences: evidence for extended contact between RNA polymerase and upstream promoter DNA," *Nucleic acids research*, vol. 23, no. 13, pp. 2351–2360, Jul. 1995.

[7] P. K. Dhar, "'Promoting' DNA once again," *Nature India*, Dec. 2010.

[8] Q. Wu *et al.*, "An improved position weight matrix method based on an entropy measure for the recognition of prokaryotic promoters," *International journal of data mining and bioinformatics*, vol. 5, no. 1, pp. 22–37, 2011.

[9] S. de Avila E Silva *et al.*, "Rules extraction from neural networks applied to the prediction and recognition of prokaryotic promoters," *Genetics and molecular biology*, vol. 34, no. 2, pp. 353–360, Apr. 2011.

[10] M. Yousef *et al.*, "Learning from positive examples when the negative class is undetermined–microRNA gene identification," *Algorithms for molecular biology : AMB*, vol. 3, pp. 2+, Jan. 2008.

[11] P. Smialowski *et al.*, "The Negatome database: a reference set of non-interacting protein pairs," *Nucleic Acids Research*, vol. 38, no. suppl 1, pp. D540–D544, Jan. 2010.

[12] C. Bland *et al.*, "Promoter prediction in *E. coli* based on SIDD profiles and artificial neural networks," *BMC Bioinformatics*, vol. 11, no. Suppl 6, p. S17, 2010.

[13] G. B. Hutchinson, "The prediction of vertebrate promoter regions using differential hexamer frequency analysis." *Computer Applications in the Biosciences*, vol. 12, no. 5, pp. 391–398, 1996.

[14] L. Gordon *et al.*, "Sequence alignment kernel for recognition of promoter regions," *Bioinformatics*, vol. 19, no. 15, pp. 1964–1971, 2003.

[15] G. G. Towell and J. W. Shavlik, "The extraction of refined rules from knowledge-based neural networks," in *Machine Learning*, 1993, pp. 71–101.

[16] M. I. Monteiro *et al.*, "Machine learning techniques for predicting Bacillus subtilis promoters." in *Brazilian Symposium on Bioinformatics (BSB)*, 2005, pp. 77–84.

[17] K. Song, "Recognition of prokaryotic promoters based on a novel variable-window Z-curve method," *Nucleic Acids Research*, vol. 40, no. 3, pp. 963–971, Sep. 2011.

[18] S. Gama-Castro *et al.*, "RegulonDB version 7.0: transcriptional regulation of Escherichia coli K-12 integrated within genetic sensory response units (Gensor Units)." *Nucleic acids research*, vol. 39, no. Database issue, pp. D98–D105, Jan. 2011.

[19] National Center for Biotechnology Information (NCBI). (2014, Apr.) http://www.ncbi.nlm.nih.gov/.

[20] A. Pallejà *et al.*, "Adaptation of the short intergenic spacers between co-directional genes to the Shine-Dalgarno motif among prokaryote genomes," *BMC genomics*, vol. 10, no. 1, pp. 537+, 2009.

[21] B. Alberts *et al.*, *Molecular Biology of the Cell*, 5th ed. Garland Science, Nov. 2007.

[22] C. T. Zhang, "A symmetrical theory of DNA sequences and its applications," *Journal of Theoretical Biology*, vol. 187, no. 3, pp. 297–306, 1997.

[23] T. M. Mitchell, *Machine Learning*, 1st ed. McGraw-Hill Science/Engineering/Math, Mar. 1997.

[24] T. L. Bailey *et al.*, "MEME SUITE: tools for motif discovery and searching." *Nucleic acids research*, vol. 37, no. Web Server issue, pp. W202–W208, Jul. 2009.