A Comparative Study of Machine Learning Methods for Detecting Promoters in Bacterial DNA Sequences

Leonardo G. Tavares, Heitor S. Lopes*, and Carlos R. Erig Lima

Bioinformatics Laboratory
Federal University of Technology Paraná (UTFPR),
Av. 7 de setembro, 3165 80230-901, Curitiba (PR), Brazil
leonardo.tavares@up.edu.br, hslopes@pesquisador.cnpq.br,
erig@utfpr.edu.br

Abstract. Machine Learning methods have been widely used in bioinformatics, mainly for data classification and pattern recognition. The detection of genes in DNA sequences is still an open problem. Identifying the promoter region laying prior the gene itself is an important aid to detect a gene. This paper aims at applying several Machine Learning methods to the construction of classifiers for detection of promoters in the DNA of Escherichia coli. A thorough comparison of methods was done. In general, probabilistic and neural network-based methods were those that performed better regarding accuracy rate.

Keywords: Bioinformatics; Data classification; Machine Learning; DNA.

1 Introduction

The DNA (Deoxyribonucleic acid) is a organic molecule responsible for the coordination and functioning of all living beings. It not only carries the genetic information of the organism, but also stores all the necessary information for synthesizing proteins.

The DNA is composed by simple elements (monomers) forming a long polymer. Monomers, in turn, are called nucleotides and are formed by chemical compounds and one of four types of molecules (bases): adenine (A), cytosine (C), guanine (G) and thymine (T).

Within the cell, the DNA is organized into structures called chromosomes, and the set of chromosomes is the genome of the organism. Most part of the DNA of eukaryotes (organisms having a membrane surrounding the nucleus of their cells) is not expressed in an amino acid chain of a protein is known as "junk DNA". For instance, only 1.5% of human genome is known to be protein-coding [16]. Although it is speculated that junk DNA may be involved in regulatory or catalytic activities, their function is still unknown. The remaining segments

* This work was partially supported by the Brazilian National Research Council – CNPq, under research grant no. 309262/2007-0 to H.S. Lopes.

D.-S. Huang et al. (Eds.): ICIC 2008, LNAI 5227, pp. 959–966, 2008.

of DNA that are related to the genetic information of the organism are called genes. Many effort has been spent in developing methods for identifying genes in DNA, from both Biochemistry/Biology and Computer Science communities.

Proteins, the product of genes, are synthesized after transcription and traduction of stretches of DNA. The beginning of the transcription of the gene takes place when the enzyme RNA-polymerase binds to given regions of the DNA known as promoters. These regions indicate to the enzyme that the genetic information to be transcribed is about to come in the linear sequence of nucleotides. Therefore, the following stretch of DNA is transcribed into mRNA (messenger Ribonucleic Acid), from which introns will be later extracted, and then will map a protein [7]. The rules that dictate the behavior of RNA-polymerase are not precisely known. Therefore, many studies have been carried out on this subject, basing on known examples. Using such approach, supported by Computational Intelligence methods, researchers aim at improving the current knowledge of this important biological process.

After the genome of an organism be sequenced, the following task is the identification of genes present in its DNA. Actually, gene detection is still an open problem and many methodologies have been proposed. The most usual approach for finding a gene is detecting signals, that is, particular sequences with biological meaning in the DNA. Among them, searching for promoter regions is one of the most important tasks [9], since a promoter indicates precisely where a gene will start in the DNA sequence.

Currently, the techniques most frequently used for recognizing promoter regions in DNA are based on machine learning methods, such as neural networks, decision trees and others.

Machine learning is a branch of Computer Science related to the development of algorithms and techniques that allow computers to learn from data. Basically, there are two types of computational learning: inductive and deductive. The later type is focused on logic. On the other hand, the methods of the former type are the most popular and they extract rules or formatted knowledge from data. The objective of this work is to evaluate and compare several machine learning methods applied to the detection of promoter regions in the DNA of a common bacteria, Escherichia coli.

2 Methodology

2.1 The Data Set

In this work we use a data set compiled by Towell et al. [13] with 106 instances, half of which having promoter sequences (positive cases) and the remaining being intragenic sequences (negative cases). The positive cases were withdrawn from another data base created by Harley and Reynolds [5], whereas the negative cases were obtained by selecting contiguous substrings from a 1500-base sequence (for more details, see [13]).

Each instance is 57 nucleotide long, and promoter regions correspond to positions -50 to +7 from the beginning of a gene. That is, the 50 nucleotides that

precede the starting point of a gene, together with the 7 first nucleotides of the gene. For simplicity purposes, in this work we did not consider positions from +1 to +7, considering only the nucleotides that precede the gene.

2.2 Computational Methods

Two computational tools for machine learning were used in this work. The first software is known as Weka [15] and is frequently used by the data mining community. It has a large number of conventional methods for data analysis in a single tool. HMMER version 2.3.2 [2] is the other software, and is a tool used by the bioinformatics community for the analysis of sequences.

The input file for Weka was formatted to ARFF model. This is a text file in which attributes are described and corresponding instances are listed. The convention used for data was: $m_{50}, m_{49}, ..., m_1$, corresponding to positions from -50 to -1 in the nucleotide sequence, followed by the target attribute, that identifies the class of each instance.

HMMER was used for evaluating the Hidden Markov Model (HMM) method. HMMs have been used for pattern recognition in many domains and, in special, in bioinformatics for the detection of patterns in protein and DNA sequences [2][14]. HMMs are considered in this work due to its efficiency in dealing with the probabilistic nature of biological sequences. A HMM is an statistical method considered a simple Bayesian network. A system modeled by a HMM is considered as a Markovian process which parameters are unknown. Consequently, the objective is to estimate the value of the parameters from known instances, and then, use the model for analyzing unknown instances and detecting patterns [3].

The input file for HMMER must be a pre-aligned set of sequences. The 53 positive sequences (that is, corresponding to a promoter sequence) were already aligned in the original database.

3 Computational Experiments and Results

A total of 30 different methods for supervised learning in Weka were used. These methods were grouped as Bayesian, neural networks, meta-learners, trees, lazy-learners and rules.

The default training/testing methodology included a 10-fold cross-validation [6]. That is, the data set is divided into 10 parts. In the first round, nine parts are used for training and the remaining for testing. Next, another other nine partitions are chosen for training and one is set apart for testing. This procedure is repeated until all 10 possible combinations have been tested. The reported result is the average of the 10 runs. The purpose of cross-validation is to avoid biased results when using a small sample of data.

Each HMM was constructed using the tool *hmmbuild* and, later calibrated using *hmmcalibrate* [2]. In both cases the running parameters were configured to the standard values. The detection threshold was empirically set to -8.4, in such a way to minimize the number of errors.

The main objective of this work is to compare the performance of classifiers in detecting promoters. Therefore, performance is measured according to the predictive accuracy of the classifiers and other parameters, such as processing time or memory requirements were not taken into account.

Table 1 shows the results for the 31 supervised classification methods. Methods are divided by category and, within categories they are ordered by descending order of predictive accuracy. In this table it is also shown parameters used in the evaluation of supervised classifiers and, from which, some metrics can be derived. These parameters are drawn from a confusion matrix (or contingency table). For a two-class prediction problem, outcomes can be binary labelled as positive (class "promoter") or negative (class "non-promoter"). When applying a given classifier method to a set of instances, depending on the outcomes, four different parameters can be computed:

- tp: true positive - number of positive instances that were correctly classified as positive;
- fn: false negative - number of positive instances that were wrongly classified as negative;
- fp: false positive - number of negative instances that were wrongly classified as positive;
- tn: true negative - number of negative instances that were correctly classified as negative.

By combining these parameters, it is possible to compute several metrics commonly used in machine learning, such as sensitivity (Se), and specificity (Sp), defined in Eq. 1. Another measure of quality for two-class problems is the Matthews Correlation Coefficient (MCC) (Eq. 2), regarded as a balanced measure and frequently used in bioinformatics [8][1]. These measures are also shown in Table 1 for all classifiers.

$$Se = \frac{tp}{(tp + fn)} \qquad Sp = \frac{tn}{(tn + fp)} \tag{1}$$

$$MCC = \frac{tp.tn - fp.fn}{\sqrt{(tp + fp).(tp + fn).(tn + fp).(tn + fn)}} \tag{2}$$

A ROC (Receiver Operating Characteristics) graph is a useful technique for comparing classifiers and observing visually their performance. This kind of graph is commonly used no only in decision making, but also in machine learning, data mining and bioinformatics [12]. In a ROC graph axes x and y are defined, respectively, as $1 - specificity$ and $sensitivity$. These axes can be interpreted as the relative trade-offs between the benefits and costs of a classifier. In this work, all classifiers were run once since they used the standard parameters. Therefore, the ROC graph can be represented by a single ROC point for each non-parametric classifier, corresponding to their $(1 - Sp, Se)$ pairs [4].

When comparing classifiers using a ROC graph, the best possible prediction method would be that lying as close as possible to the upper left corner (coordinates $(0, 1)$), representing 100% sensitivity and 100% specificity. A completely

Table 1. Comparative performance of 31 machine learning methods

Group	Method	tp	fn	fp	tn	MCC	accuracy rate (%)
HMM	HMM	50	3	5	48	0.850	92.45
Bayes	CNB	49	4	3	50	0.868	93.40
	NaiveBayes	48	5	3	50	0.850	92.45
	NaiveBayesSimple	48	5	3	50	0.850	92.45
	NaiveBayesUpdateable	48	5	3	50	0.850	92.45
	AODE	50	3	7	46	0.814	90.56
Neural Net	MultiplayerPerceptron	49	4	3	50	0.968	93.40
	SMO	49	4	4	49	0.849	92.45
	RBFNetwork	48	5	6	47	0.793	89.62
	Logistic	45	8	5	48	0.756	87.73
	VotedPerceptron	46	7	10	43	0.680	83.96
Meta	LogitBoost	47	6	5	48	0.793	89.62
	MultiBoostAB	47	6	7	46	0.755	87.73
	MultiClassClassifier	45	8	5	48	0.756	87.73
	ThresholdSelector	44	9	5	48	0.738	86.79
	ADABoost	46	7	8	45	0.717	85.84
Trees	NBTree	47	6	5	48	0.793	89.62
	LMT	47	6	6	47	0.774	88.67
	ADTree	47	6	8	45	0.736	86.79
	J48	45	8	12	41	0.624	81.13
	ID3	44	7	14	37	0.594	76.41
Lazy	LBR	48	5	3	50	0.850	92.45
	IB1	49	4	15	38	0.656	82.07
	Kstar	48	5	14	39	0.651	82.07
	IBk	49	4	16	37	0.639	81.13
	LWL	41	12	14	39	0.510	75.47
Rules	PART	44	9	11	42	0.623	81.13
	DecisionTable	41	12	11	42	0.566	78.30
	Ridor	41	12	11	42	0.566	78.30
	JRip	42	11	13	40	0.548	77.35
	NNge	31	22	2	51	0.591	77.35

random guess would give a point along a diagonal line (the so-called line of no-discrimination) from coordinates $(0, 0)$ to $(1, 1)$ the left bottom to the top right corners.

Figure 1 shows the ROC graph for the classifiers evaluated in this work. Since the performance of all classifiers were above the no-discrimination line, this figure shows only the upper left quadrant of the graph. This is done only for comparison purposes, so as to amplify the differences between classifiers. The top classifiers are identified in the ROC space: HMM (Hidden Markov Model [2]), SMO (Sequential Minimal Optimization algorithm for support vector machine [10]), MLP (Multilayer Perceptron neural network [15]), LBR (Lazy Bayesian Rules classifier [15]), CNB (Complement class Naive Bayes [11], and Bayes. This last one comprehends three different versions: Naive Bayes, NaiveBayesSimple and

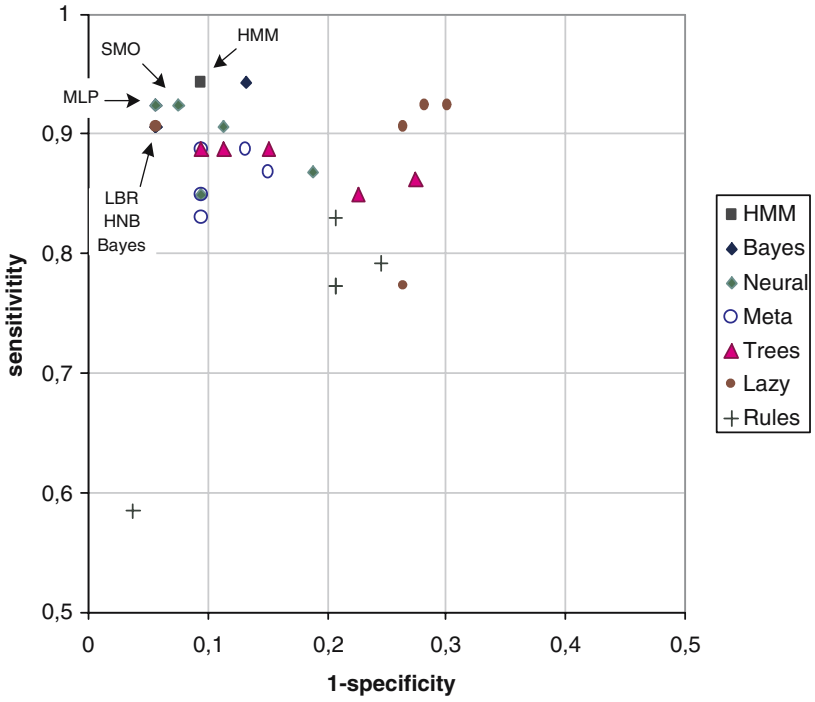

Fig. 1. ROC space for the classifiers tested. Each symbol represents the classifiers of a given group (see text).

NaiveBayesUpdateable. It should be noted that all classifiers tested are represented in the ROC space. Since some of the classifiers have achieved the same (or near the same) performance, there are several points superimposed in the graph.

Besides comparing the classifiers with themselves, our results were also compared with other published works. Using the same data set, Towell et al. [13] proposed a hybrid approach mixing neural networks and symbolic rules, named KBANN (Knowledge Based Neural Network). This approach was compared with a multilayer perceptron neural network, a decision tree induced with ID3 algorithm, a clustering algorithm k-NN and the technique known as O'Neill. These results are shown in table 2 [13].

Table 2. Comparison of different methods for the same data set, by Towell et al. [13]

Method	**accuracy rate (%)**
KBANN	96.22
Multilayer Perceptron	92.45
O'Neill's method	88.68
k-NN	87.74
Decision tree (ID3)	82.08

4 Discussion and Conclusions

Several methods for detecting promoters in nucleotide sequences were compared. Table 1 shows that, considering only the accuracy rate, HMM and the Bayesian methods (including LBR and excluding AODE) were those that performed better. The results of another methods (table 2) are approximately similar to those obtained in this work. Also, they obtained the best results with neural network-based approaches, a fact that was confirmed in our work. Recall that no effort was done to fine-tune parameters in our experiments. Results were obtained using the default parameters, except for HMM where the threshold was set empirically. Therefore, it could be reasonable to expect slight better performances.

The ROC graph shows the difference between methods more clearly than table 1. Possibly, the small differences in performance of the methods tested may be due to the small number of instances of the data set. A single misclassified instance may lead to an error of almost 2%.

In general, the probabilistic methods, including HMM and Naive Bayes classifiers achieved better results than other classifiers. Possibly, this is due to the specific nature of biological sequences, where uncertainty is also present. That is why Bayesian methods have been traditional in bioinformatics, for tasks such as sequence alignment, pattern recognition and others. Similarly, some neural network-based methods, namely SMO and MLP, obtained results competitive to those of Bayesian methods, possibly due to its ability to establish complex hiperplanes in the problem space. This shows its adequacy for dealing with biological sequences. Regarding MCC, again a Bayesian and a neural-network method, respectively, CNB and MLP, were those that performed better.

Traditional methods for induction of decision-trees, such as ID3 and C45, are considered baseline in many data mining applications. However, for the data set and standard parameters used in this work they did not performed so well as others. This was confirmed by analyzing the MCC of those classifiers.

Future work include the use of ensemble methods for associating classifiers and the application of these methods to the detection of promoters in DNA sequences of eukariotes, as part of an automatic gene detection system.

References

1. Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A.F., Nielsen, H.: Assessing the Accuracy of Prediction Algorithms for Classification: an Overview. Bioinformatics 16, 412–424 (2000)
2. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, Cambridge (1998)
3. Ephraim, Y., Merhav, N.: Hidden Markov Processes. IEEE T. Inform. Theory 48, 1518–1569 (2002)
4. Fawcett, T.: An Introduction to ROC Analysis. Pattern Recogn. Lett. 27, 861–874 (2006)
5. Harley, C., McClure, W.: Compilation and Analysis of Escherichia coli Promoter DNA Sequences. Nucleic Acids Res. 11, 2237–2255 (1983)

6. Kohavi, R.: A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. In: 14th Int. Joint Conf. on Artificial Intelligence, pp. 1137–1143 (1995)
7. Nelson, D.L., Cox, M.M.: Lehninger Principles of Biochemistry, 4th edn. W.H. Freeman, Chicago (2006)
8. Matthews, B.W.: Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme. Biochim. Biophys. Acta 405, 442–451 (1975)
9. Mount, D.W.: Bioinformatics: Sequence and Genome Analysis. CSHL Press, Woodbury (2001)
10. Platt, J.: Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In: Schoelkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge (1998)
11. Rennie, J., Shih, L., Teevan, J., Karger, D.: Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In: Fawcett, T., Mishra, N. (eds.) 20th Int. Conf. on Machine Learning, pp. 616–623. AAAI Press, Menlo Park (2003)
12. Sing, T., Sander, O., Beerenwinke, N., Lengauer, T.: ROCR: Visualizing Classifier Performance in R. Bioinformatics 21, 3940–3941 (2005)
13. Towell, G., Shavlik, J., Noordewier, M.: Refinement of Approximate Domain Theories by Knowledge-based Artificial Neural Networks. In: 8th National Conference on Artificial Intelligence, pp. 861–866. AAAI Press, Menlo Park (1990)
14. Weinert, W., Lopes, H.S.: Neural Networks for Protein Classification. Appl. Bioinformatics 3, 41–48 (2004)
15. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
16. Wolfsberg, T., McEntyre, J., Schuler, G.: Guide to the Draft Human Genome. Nature 409, 824–826 (2001)