A Comparative Study on Feature Selection for E.coli Promoter Recognition

Paul C. Conilione and Dianhui Wang

Department of Computer Science and Computer Engineering
La Trobe University, Melbourne, VIC 3086, Australia

csdhwang@ieee.org

Abstract

This paper explores the application of feature selection by the Correlation based Feature Selection (CFS) algorithm on the problem of classification of E.coli promoters using neural networks, Support Vector Machines (SVM) and Extreme Learning Machines (ELM). It was found that even though in general the classification accuracy can be reduced by a statistically significant amount, in real terms this was only a few percent. The results also indicate some interesting characteristics of the features used in E-coli promoters. A comparative study with three typical classifiers was carried out in this study.

Keyword: E.coli, promoters, classification, pattern recognition, neural networks, SVM, ELM, feature selection

I. Introduction

The number of features representing data depends on what is being observed, from a few features for a mechanical system, to several thousands for biological sequences [1]. As the number of features increases, the volume of the feature space grows exponentially, this is called the *Curse of Dimensionality* [2].

Consequently, significant effort has been put into the area of feature selection algorithms (FSA's), which aim to reduce the number of features that are needed to perform the tasks, but still maintain or even improve the learners' performance. In general the goal is to remove irrelevant and redundant features from the data. By reducing the feature space, it can increase learning speed, increase learner performance (e.g. classification accuracy and generalization), can make the learners model more easily understood, and reduce the learners' storage requirements [3]. FSA's can be grouped into two main categories, i.e., classifier independent and classifier dependent. A classifier independent approach is the filter approach which preprocesses the training data to determine the *best* feature subset to use. Then the classifier is trained and tested on the data set only using the features found by the FSA. The advantage of the filter methods is that they are generally computationally efficient.

The wrapper approach is a classifier dependent FSA and uses a specific learning algorithm, such as decision trees, and ANN's, to evaluate the feature subset via the performance of the learner. This has the advantage of selecting features that are suited to the specific learner, and hence generally result in higher learner performance, e.g. accuracy. However as the learner needs to rerun for each new subset, it is computationally costly, and does not scale well to large numbers of features. Consequently we only examine a filter type FSA.

Research into the application of FSA's to biological sequences has been growing, for example they have been applied to DNA sequence classification [4], splice site prediction [5] and gene expression profiles [6].

One biological problem is the identification of a promoter region within a DNA sequence. A promoter is a region of DNA, recognised by and a binding target for Ribonucleic Acid (RNA) polymerase, which then starts transcription of the coding region at the Transcription Start Site (TSS). Using biochemical or genetic means to identify the promoter regions and pinpoint the binding site(s) at which the RNA polymerase comes into contact with the DNA is difficult. For this reason, previous techniques for identification of the promoter regions are based on statistical and alignment techniques. Research by [7], [8] and [9] compiled increasingly larger number of promoter regions of E.coli. Using statistical methods, they identified two major consensus sequences, which consist of two hexamers (6 base pairs (bps)) long. The first consensus sequence is TATAAT and is approximately 35 bps upstream from the TSS, (labelled -35 hexamer). The second consensus sequence is TTGACA and is approximately 10 bps upstream from the TSS (labelled -10 hexamer).

Previous researchers have applied ANN's to the problem of promoter recognition [10], [11] and [12], achieving promoter recognition in the 90% range and false positive rates of around 5-10%. However, there is not a significant amount of work on the application of FSA's to the problem of promoter recognition in E.coli.

In this paper, we applied four different feature extraction methods to the E.coli DNA, namely CODE-4, 19 High Level Features, and the structural DNA profiles GC Trinucleotide frequency and Stacking Energy. We then compare the classification results of three different classifiers, backpropogation neural network, Support Vector Machines (SVM) and the Extreme Learning Machine (ELM), using the full feature set and the feature subset.

II. Methods

A. Data

We used a pool of 872 E.coli (K12 strain) promoter sequences [13]. The promoter sequences were taken from 60 bases upstream of the TSS, to 21 bases downstream of the TSS. Three different types of non-promoter DNA sequences were used. The first type was randomly generated DNA sequences with the same base frequency as the target DNA (random-prom). The second type was taken from the gene coding regions of the E.coli K12 strain [13], with 872 sequences selected from the pool of approximately 4400 known genes, starting 100 bps downstream of the TSS (gene). The third type used was randomly generated sequences, but using the same base frequencies of occurrences as the 872 gene DNA sequences, (random-gene). From these non-promoters （NPs）, six data sets are created which are given in Table I.

Table 1: Details of the data sets used in this paper, where size is the number of bases.

Label	**Prom Region**	***N*(bps)**	**Size**	**Non-prom**	**Size**	**Total**
Random Promoter	-60 to +21	81	872	rand-prom	872	1774
Gene	-60 to +21	81	872	gene	872	1774
Random-Gene	-60 to +21	81	872	rand-gene	872	1774
Random-Gene Half	-60 to +21	81	872	1/2 rand-gene	436	1308
All NPs	-60 to +21	81	872	All NPs	2616	3488
All NPs Third	-60 to +21	81	872	1/3 All NPs	870	1742

B. Feature Extraction

1. CODE 4

The nucleotides of DNA are represented by four symbols, {A, T, C, G}, and are encoded using four binary bits, where *A*→0001, *T*→0010, *G*→0100 and *C*→1000. This scheme is commonly referred to as CODE-4 encoding. As each base is represented by four binary bits, to represent a sequence of length *N*, requires 4*N* input nodes of the ANN.

2. 19 High Level Features

The following are definitions for the *high level* features of a DNA sequence as outlined in [14], and formally defined in [15];

Features 1 to 12 - Helical Parameters.

Table. 2 lists the 12 different patterns as defined in [16], where R and Y denotes purine (A and G) and pyrimidine (C and T) respectively and each feature takes on the number of times a non-overlapping pattern occurs.

Features 13 and 14 - Site Specific Information.

Feature 13 is the number of times the gtg_motif occurs in a DNA sequence *S*, where it does not overlap. Feature 14 is the number of times the gtg_pair motif occurs, where the *spacer* is a multiple of 10±1 bases from the beginning of each motif.

Table 2: Features 13 and 14

Feature	Label	Pattern
13	gtg_motif	GTG or CAC
14	gtg_pair	gtg_motif *spacer* gtg_motif

Features 15 to 16 - Local Secondary Structure.

The *local secondary structures* are characterised by the presences of *tandem* and *invert* repeats. Let *S* be a sequence of *N* bases, drawn from an alphabet of {*A*,*T*,*C*,*G*}. $S = s_1, s_2, ..., s_m$, where s_i is a base at position *i* in *S*. The reverse of *S* is denoted S^{-1}. The complement of a base is the nucleotide that binds to it on the opposite strand of the DNA sequence and is denoted as s_i, e.g. if $s_i = A$, then s_i=*T*. The complement of a sequence is denoted $\bar{S}$.

Feature 15 - Tandem repeats.

A tandem repeat is a sequences of nucleotides that occurs twice on the same DNA strand. We define an imperfect tandem repeat with no gaps between repeating sequences as *T*=*UV*, where the subsequences *U* and *V* are expressed as $U = u_1, u_2, ..., u_m$ and $V = v_1, v_2, ..., v_m$. The period *p* of *T* is the minimum integer such that $u_i = v_{i+p}$ for some *i* [17]. The mismatch between subsequences *U* and *V* is given by the hamming distance, $d(U, \bar{V}) = c$, where *c* is the number of mismatches. The no-gap condition is met iff $u_1 = v_1$ and $u_m = v_m$.

Feature 16 - Inverted repeats.

An inverted repeat is a sequence of nucleotides that is found to be repeated in the reverse order on the opposite strands of the DNA double helix. We define an imperfect inverted repeat as *I* = *UV*, where

$U = u_1, u_2, ..., u_m$, $V = v_1, v_2, ..., v_m$, and the number of mismatches is given by the hamming distance $d(U, \bar{V}^{-1}) = c$.

Given a sequence *S*, all repeats of the same type are found and the size of the repeat *n* and number of matches *b*=*n*-*c* are recorded. Then the probability of one or more of the repeats being found is calculated using the process detailed in [15], and the smallest probability is used for the feature value.

Features 17 to 19 - DNA compositions

The AT content, AG/TC ratio and AC/TG ratio are given in (1), (2) and (3) respectively.

$$AT\_content = \frac{A+T}{N} \tag{1}$$

$$AT\_TC\_content = \begin{cases} \dfrac{A+G}{T+C} & T+C \neq 0 \\ 0 & T+C \neq 0 \end{cases} \tag{2}$$

$$AC\_TG\_content = \begin{cases} \dfrac{A+C}{T+G} & T+G \neq 0 \\ 0 & T+G \neq 0 \end{cases} \tag{3}$$

where *A*, *C*, *G* and *T* are the number of adenines, cytosines, guanines and thymines respectively, and *N* is the total number of nucleotides in the sequence window.

3. DNA Structural Profiles.

For a sequence $S = s_1, ..., s_N$, with *N* bases, its profile *P*(*S*) is given by $\{p(s_i,..,s_{i+k})\}$ where *p*(.) is the DNA property value for a given set of bases, 1≤*i*≤*N*-*k*+1 and *k* is the number of nucleotides used to calculate its value. So for properties based on dinucleotides, *k*=2 and for trinucleotide properties, *k*=3. We used the DNA structural profiles GC trinucleotide frequency count and Stacking energy [18], please see [19] for further information.

C. Feature Selection

After examining a very wide range of filter type FSAs, we selected the Correlation bases Feature Selection (CFS) algorithm [20] which has the advantage using both class-feature and feature-feature correlations to measure the *merit* of a given subset of features using a heuristic. The CFS method, first calculates the correlation coefficient between both the feature to class and features to features. Then using a search algorithm, explores the feature subspace and evaluates the optimality of each subset by using a heuristic, given in Eq. (4).

$$J = \frac{k\bar{r}_{cf}}{\sqrt{k + k(k-1)\bar{r}_{ff}}} \tag{4}$$

where r_{cf} and r_{ff} are the average class-feature correlation and feature-feature correlation values respectively. Hence, the merit function will have larger values for feature subsets that have features with strong class-feature correlation and weak feature-feature correlation. However, even if a set of features has strong class-feature correlation, if there is strong feature-feature correlation the merit value will be degraded.

We used the best-first search algorithm as given in [21], using forward search. The search stops if it does not find a subset with a better merit value after 5 branch expansions. As CFS can use any correlation function we tried Symmetrical Tau (ST), Symmetrical Uncertainty (SU) and Relief-F (RF). To begin the definition of these correlation functions, we first define the contingency table from which they are calculated. The aim of statistical methods, such as chi-square test, is to determine if a variable *B* is correlated with variable *A*. A contingency table, Tab. 3, is used to relate the two variables, where variable *A* has α categories, *B* has β categories, and a_i and b_j are particular category of *A* and *B* respectively.

Table 3: Contingency table.

A	*B*				
	b_1	b_2	...	b_β	Total
a_1	c_{11}	c_{12}		$c_{1\beta}$	c_{1+}
a_2	c_{21}	c_{22}		$c_{2\beta}$	c_{2+}
⋮	⋮	⋮	⋮	⋮	⋮
a_α	$c_{\alpha 1}$	$c_{\alpha 2}$		$c_{\alpha\beta}$	$c_{\alpha +}$
Total	c_{+1}	c_{+2}	...	$c_{+\beta}$	n

Hence, the probability for a given value of $a \in A$ or $b \in B$ can be expresses as follows.

$$\Pr(a_i) = \frac{c_{i+}}{n}$$

$$\Pr(b_j) = \frac{c_{+j}}{n}$$

$$\Pr(b_j \mid a_i) = \frac{c_{ij}}{c_{i+}}$$

$$\Pr(a_i \mid b_j) = \frac{c_{ij}}{c_{+j}}$$

1. Symmetrical Uncertainty

The information measure usually determines the information gain from using a feature. The information gain (IG) is the difference between the prior uncertainty and the expected posterior uncertainty when including variable a_i. The problem with information gain is that it is biased towards features with more values, as well as needing the values to be normalised [22]. A measure that overcomes these problems is *symmetrical uncertainty* [23] and is defined below:

$$SU(A,B) = 2\left[\frac{IG(A \mid B)}{H(A) + H(B)}\right] \quad (5)$$

where *H*(*A*), *H*(*B*), and *IG* is defined below.

$$H(A) = -\sum_{a \in A} \Pr(a) \log_2 \Pr(a)$$

$$H(B) = -\sum_{b \in B} \Pr(b) \log_2 \Pr(b)$$

$$IG(A|B) = H(A) + H(B) - H(A,B)$$

Hence, Eq. (6) can be written as,

$$SU(A,B) = 2\left[\frac{IG(A|B)}{H(A)+H(B)}\right] \qquad (6)$$

In practice, *H*(*A*), *H*(*B*) and *H*(*A*,*B*) can be rewritten to use the values in the contingency table.

$$H(A) = -\sum_{i=1}^{\alpha} \Pr\left(\frac{c_{i+}}{n}\right)\log_2 \Pr\left(\frac{c_{i+}}{n}\right)$$

$$H(B) = -\sum_{j=1}^{\beta} \Pr\left(\frac{c_{+j}}{n}\right)\log_2 \Pr\left(\frac{c_{+j}}{n}\right)$$

$$H(A,B) = -\sum_{i=1}^{\alpha}\sum_{j=1}^{\beta} \Pr\left(\frac{c_{ij}}{n}\right)\log_2 \Pr\left(\frac{c_{ij}}{n}\right)$$

Variable *A* and *B* that has a *SU*=1 have strong correlation, whilst if *SU*=0 there is no correlation.

2. Symmetrical Tau

The problem with the most commonly used statistical, and information theory based feature selection methods, such as Chi-square criterion, Asymmetrical Tau, Information Gain and Gini indexing criterion, is that they tend to favour features with more values. To overcome this problem, [24] proposed the Symmetrical Tau. In the case of a multinomial sampling model, the maximum likelihood estimator of τ is given in (7).

$$\tau = \frac{n\left[\sum_{j=1}^{\beta}\sum_{i=1}^{\alpha}\frac{(c_{ij})^2}{c_{+j}} + \sum_{j=1}^{\beta}\sum_{i=1}^{\alpha}\frac{(c_{ij})^2}{c_{i+}}\right] - \sum_{i=1}^{\alpha}(c_{i+})^2 - \sum_{j=1}^{\beta}(c_{+j})^2}{2n^2 - \sum_{i=1}^{\alpha}(c_{i+})^2 - \sum_{j=1}^{\beta}(c_{+j})^2} \qquad (7)$$

When there is perfect association between variables *A* and *B*, *T*=1. Whilst if *T*=0 there is no association.

3. Relief-F

The Relief algorithm can be reformulated so that it can be applied to any two variables [20].

$$\mathrm{Re}\,lief_A = \frac{Gini' \times \sum_{\alpha\in A}\Pr(a)^2}{\left(1-\sum_{b\in B}\Pr(b)^2\right)\sum_{b\in B}\Pr(b)^2} \qquad (8)$$

where $Relief_A$ can have a value between 1 for strongly correlated and -1, strongly uncorrelated, and *Gini*' is defined as:

$$Gini' = \left[\sum_{b\in B}\Pr(b)(1-\Pr(b))\right] - \sum_{a\in A}\left(\frac{\Pr(a)^2}{\sum_{a\in A}\Pr(a)}\sum \Pr(b|a)(1-\Pr(b|a))\right) \qquad (9)$$

These formulas are rewritten use the contingency table,

$$\mathrm{Re}\,lief_A = \frac{Gini' \times \sum_{i=1}^{\alpha}\left(c_{i+}\right)^2}{\left(1-\sum_{j=1}^{\beta}\left(\frac{c_{+j}}{n}\right)^2\right)\sum_{j=1}^{\beta}\left(c_{+j}\right)^2} \tag{10}$$

Then for the *Gini*' equation:

$$Gini' = \left[\sum_{j=1}^{\beta}\frac{c_{+j}}{n}\left(1-\frac{c_{+j}}{n}\right)\right] - \sum_{i=1}^{\alpha}\left(\frac{\left(c_{i+}\right)^2}{\sum_{i=1}^{\alpha}\left(c_{i+}\right)^2}\sum_{j=1}^{\beta}\frac{c_{ij}}{c_{i+}}\left(1-\frac{c_{ij}}{c_{i+}}\right)\right) \tag{11}$$

As relief is asymmetrical, the correlation between two variable are calculated where one variable is *A* and the other is *B*, this is then reversed and the average taken for symmetrical relief.

D. Classifiers

1. Neural Network

We used a feed-forward neural network and explored the optimal topology of the network by varying the number of neurons in the single hidden layer by varying the ration between the number of hidden neurons and the number of features. All networks were fully connected, with logarithmic sigmoid activation functions for the hidden neurons and the single output neuron. All weights and biases were randomly initialised in the range of [-0.01,0.01] and training was performed using the batched Scaled Conjugate Gradient (SCG) algorithm. We trained the ANN until it reached or exceeded a particular classification accuracy as measured by the F-measure on the training data, for this we tried an accuracy target of 0.85, 0.90 and 0.95.

2. Support Vector Machine

For the SVM classifier, we used an implementation by [25], using a radial basis function (RBF) for the kernel we explored the effect of the *cost* and γ on the test accuracy by using a grid search of the *cost* and γ parameters as suggested by [26] to find the optimal values.

3. Extreme Learning Machine

The Extreme Learning Machine (ELM) is a single hidden layer feedforward neural network using a unique training algorithm that allows the ELM to learn classification problems many orders of magnitude faster than tradition ANN training algorithms [27]. We used a sigmoidal activation function and explored the effect of the number of neurons in the hidden layer by using different ratios of the number of features in the training and test data.

E. Performance Evaluation

The performance of the three classifiers was measured using a confusion matrix and derived F-measure (8), and accuracy (9) metrics. From the confusion matrix, Table 4, a promoter that is correctly classified is called a *true positive* (TP), whilst a promoter classified as a non-promoter is called a *false negative* (FN). A non-promoter that is correctly classified is called a *true negative* (TN) and an incorrectly classified non-promoter is called a *false positive* (FP). The definition of the confusion matrix was found in a number of sources and checked with the reference [28].

Table 4: Confusion matrix

Actual Class	Predicted Class	
	Promoter	Non-Promoter
Promoter	TP	FN
Non-promoter	FP	TN

$$F = \frac{2 \times precision \times recall}{precision + recall} \qquad (8)$$

$$acc = \frac{TP + TN}{TP + FP + TN + FN} \qquad (9)$$

Where recall and precision are defined in equations (10) and (11) respectively.

$$recall = \frac{TP}{TP + FN} \qquad (10)$$

$$precision = \frac{TP}{TP + FP} \qquad (11)$$

The accuracy of classifying each class is given by the promoter and non-promoter sensitivity in (12) and (13).

$$S_P = \frac{TP}{TP + FN} \qquad (12)$$

$$S_{NP} = \frac{TN}{TN + FP} \qquad (13)$$

F. Training and Testing Process

The six data sets were processed using the four different feature extraction methods as outlined above. For each feature type, the data was equally split randomly into training and testing sets (ie, hold-out method), which was repeat ten times so that each classifier can be trained and tested ten times to gain a more reliable estimate of performance. CFS was applied to each data set to determine an optimal feature subset using each of the correlation functions, finally each classifier was then trained and tested on the full feature set and the subset of features for comparison.

III. Results and Discussion

A. Features Selection Results

Tables 5, 6, 7 and 8 presents the ratio between r_{cf} and r_{ff} over *all* features, the number of features selected and the merit value calculated from Eq. (4), for all six data sets and the three correlation functions.

In the next sections we examine more closely which features were selected for each data set by looking at the histogram of features selected for each data set. The maximum number of times a given feature can be selected is ten, as there are ten sets of training and test data.

Table 5: CFS results for feature type: CODE 4

NP	SU			ST			RF		
	r_{cf}/r_{ff}	Merit	|X|	r_{cf}/r_{ff}	Merit	|X|	r_{cf}/r_{ff}	Merit	|X|
Random-Promoter	0.3827	0.0599	28.20	0.4991	0.0755	27.40	0.6277	0.0655	28.20
Gene	1.4788	0.2057	58.50	1.9004	0.2549	58.20	2.4151	0.2207	69.00
Random-Gene	1.0058	0.1168	69.30	1.3089	0.1463	69.70	1.6729	0.1270	83.90
Random-Gene Half	0.9701	0.1218	60.20	1.1917	0.1447	63.50	2.1557	0.3063	76.20
All types	0.4724	0.0647	29.40	0.5595	0.0733	28.20	0.2286	0.2291	56.50
All Types Third	1.1575	0.1450	62.60	1.5015	0.1813	63.50	1.9224	0.1592	72.90

Table 6: CFS results for feature type: 19 high level

NP	SU			ST			RF		
	r_{cf}/r_{ff}	Merit	$\lvert X\rvert$	r_{cf}/r_{ff}	Merit	$\lvert X\rvert$	r_{cf}/r_{ff}	Merit	$\lvert X\rvert$
Random-Promoter	0.1552	0.1709	1.00	0.2730	0.3060	2.00	0.2043	0.2131	2.00
Gene	0.1954	0.1713	1.00	0.3291	0.3015	1.10	0.2487	0.2071	1.10
Random-Gene	0.2068	0.1802	2.00	0.3427	0.3082	1.00	0.2584	0.2164	1.00
Random-Gene Half	0.2003	0.1720	2.00	0.3152	0.2911	1.30	0.2128	0.2003	1.20
All types	0.1127	0.1205	1.00	0.1988	0.2248	1.90	0.1391	0.1328	2.00
All Types Third	0.1968	0.1713	1.40	0.3301	0.3003	1.30	0.2481	0.2061	1.70

Table 7: CFS results for feature type:: GC trinucleotide

NP	SU			ST			RF		
	r_{cf}/r_{ff}	Merit	$\lvert X\rvert$	r_{cf}/r_{ff}	Merit	$\lvert X\rvert$	r_{cf}/r_{ff}	Merit	$\lvert X\rvert$
Random-Promoter	0.2154	0.0495	17.10	0.3019	0.0606	18.60	0.2644	0.0513	16.30
Gene	0.7746	0.1499	26.60	1.0584	0.1826	35.10	0.9161	0.1522	35.30
Random-Gene	0.9055	0.1669	29.10	1.2244	0.1998	39.10	1.0400	0.1640	33.90
Random-Gene Half	0.7975	0.1659	19.80	0.9974	0.1810	22.10	0.7651	0.1701	12.20
All types	0.2288	0.0555	5.70	0.2779	0.0602	6.00	0.3448	0.0682	33.00
All Types Third	0.8716	0.1664	26.80	1.1795	0.1997	33.90	1.0017	0.1648	33.10

Table 8: CFS results for feature type: Stacking energy

NP	SU			ST			RF		
	r_{cf}/r_{ff}	Merit	$\lvert X\rvert$	r_{cf}/r_{ff}	Merit	$\lvert X\rvert$	r_{cf}/r_{ff}	Merit	$\lvert X\rvert$
Random-Promoter	0.2130	0.0495	26.20	0.3903	0.0679	35.10	0.3257	0.0546	35.70
Gene	0.4632	0.1112	22.70	0.8357	0.1489	32.80	0.7015	0.1206	33.10
Random-Gene	0.4497	0.0996	28.10	0.8336	0.1408	39.40	0.7799	0.1271	44.30
Random-Gene Half	0.4032	0.1087	13.40	0.7194	0.1366	31.30	1.0879	0.1929	40.10
All types	0.2135	0.0433	17.10	0.3073	0.0489	26.50	-0.0901	0.0217	13.30
All Types Third	0.4231	0.1005	22.40	0.7763	0.1367	30.20	0.6940	0.1169	35.40

From the four tables, it can be seen that using the three types of correlation function, CFS successfully reduced the number of features for all data sets and feature types. However, the RF correlation function has a tendency to cause CFS to select a larger number of features and is evident from the histograms it is also less consistently selects features compared to SU and ST. Given the merit function essentially aims to find sets of features that have strong correlation with the class and low correlation between features, it is reasonable to use the r_{cf}/r_{ff} ratio over *all* features to indicate the general *strength* of a given data set to the classification problem. Hence, one would expect that a high ratio would indicate that the features have an overall strong correlation with the class and so the CFS should select more features as there is lower r_{ff} and so should gain higher merit. On the other hand, given a lower ratio, CFS would select smaller feature sets as the high feature inter-correlation would make it difficult to find larger sets of features as the merit would be reduced. This trend is found to be true for the datasets as seen in the tables. Data sets with higher r_{cf}/r_{ff} ratios tend to have a larger number of features selected compared to those that have a smaller r_{cf}/r_{ff} ratios. Furthermore, SU and ST have smaller ratios compared to RF, and tend to select few features than RF.

Looking at the result more carefully, the *Random-Promoter* consistently has a lower r_{cf}/r_{ff} ratio than the other non-promoter types. This suggests that the random DNA sequences with the same base frequency as the promoter, provide poor differentiation between target and non-target classes. *All Types* also has a low ratio, however given *All Types 872* has a fairly high ratio, this low value would be due to the significantly larger proportion of non-promoter examples in the data set, hence *washing out* the distinction between promoter and non-promoter in the feature space.

For the high level encoding the CFS regularly selected feature 15 (tandem repeats), occasionally feature 16 (inverted repeats) and really feature 17(AT content). This suggests that the repeat structures are useful features with a strong class correlation, however, also with a strong correlation.

For CODE-4, GC Trinucleotide frequency and Stacking Energy, it is evident that that the FSA tends to select features predominately around the biologically significant -10 region, and to a lesser extent the -35 region and TSS, indicating their importance at distinguishing promoters from non-promoters.

B. Classification Results

In this subsection, we report the performance of the three classifiers, and show the statistically significant improvement or degradation in test classification accuracy by ’+’ or ’-’ respectively, using a two-tailed t-test with $\alpha=0.05$. Due to space consideration, the standard deviation of the results could not be shown. For the Neural network and ELM tables, we present the average ratio between the number of features and the number of hidden neurons over the 10 hold-out data sets.

It is clear that the CFS algorithm drastically reduces the number features for all features types and data sets. However, the merit function has been noted to be too aggressive in removing features that maybe valuable for classification [29]. This is evident for the 19 High Level features, where only one or two features are selected and the resulting classification accuracy is significantly reduced. In our previous work, we found that between 15 to 17 features were needed to gain an improvement in the classification accuracy [30], and the reason why the merit function of CFS selects so few features is because features 15, 16 and 17 generally have significantly larger correlations with the class than the other features for most data sets.

For the three other feature types (CODE-4, GC-Trinucleotide frequency and Stacking Energy), even though the classification accuracy degrades, there are numerous cases of a statistically significant improvement in classification accuracy for different data sets.

Table 9: Best test classification results for each feature type and classifiers, comparing to the full feature set with the subset chosen by CFS

Feature Type		**Neural Network**			**SVM**			**ELM**		
		NP	**Acc**	**|X|**	**NP**	**Acc**	**|X|**	**NP**	**Acc**	**|X|**
CODE-4	**Full**	G	89.07	324	G	88.92	324	G	78.23	324
	CFS	G	87.83	69 (RF)	G	88.13	69 (RF)	G	83.64	124.7 (RF)
19 High	**Full**	RG	80.54	19	RG	80.88	19	RG	79.77	19
	CFS	RG	79.21	2 (SU)	RG	79.84	2 (SU)	RG	79.28	2 (SU)
GC Tri-nucleotide	**Full**	RG	80.93	79	RGH	84.13	79	RGH	79.60	79
	CFS	RG	80.37	39.1 (ST)	RG	80.84	29.1 (SU)	RG	80.21	39.1 (ST)
Stacking Energy	**Full**	G	79.44	80	G	81.01	80	AT	77.07	80
	CFS	AT	79.27	36.5 (ST)	RG	78.34	44.3 (RF)	AT	77.35	26.5 (ST)

From Table 9, the best results overall was with the neural network, on the Gene data set with CODE-4, achieving a test classification accuracy of 89.07% with the full feature set. The best classification accuracy using a subset of features was again with the Gene data set with CODE-4 using the SVM classifier, with a test classification accuracy of 88.13%. The best classification accuracy is generally achieved with either the Gene or Random-Gene data sets, which indicate that the gene DNA sequence can be more easily differentiated from the promoter compared to other types of non-promoters.

The application of the CFS to CODE-4 feature type provided the best results, gaining the most number of statistically significant improved test classification accuracy compared to the full feature set. The reason for the other feature types suffering generally lower classification accuracy with the subset of features selected by CFS, is that CODE-4 had a significantly larger number of features to begin with and so the reduction of redundant and irrelevant features had a more dramatic effect on the classification accuracy of the classifiers.

Poor classification performance was seen for the All-Types data set as the promoter sensitivity S_p was very low for all feature types and classifiers. This would be due to the higher ratio of non-promoters to promoters, hence the classifiers are able to classify non-promoters more strongly, than promoters. Furthermore, All-Types 872 does not suffer such low S_p for all classifiers, which shows the importance of the ratio between the target and non-target training examples.

The type of non-promoter training examples strongly determines the classification accuracy. Gene and rand-gene non-promoters produced the best classification results, as we have found in previous studies, [15] and [19].

Examining the ratio between the number of features and the number of hidden neurons for ELM, it is evident that there is a higher ratio of hidden neurons to number of features for the features subsets compared to the classifier using all the features. This indicates that the features selected by the CFS contain possibly more complex information that requires a more complex model compared to the number of features. The neural network classifier showed a similar trend, however not as clearly.

IV. Conclusions

From the experimental results, features selection can drastically reduce the number of features that are needed to express the characteristics of E.coli promoters using a number of different feature types. We showed that in the case of CODE-4, GC-Trinucleotide and Stacking Energy structure profiles, the CFS algorithm selected features that had biologically significant importance, namely around the -10, -35 and TSS regions. In general we found that the classification performance for the three classifiers we used, feed-forward neural network, SVM, and ELM, had a statistically significant reduction in accuracy, however this was only in the range of a few percent and at the benefit of quicker training and smaller classification models. We found that the merit function was very aggressive in selecting small subsets, at the expensive of classifier performance. In future work, we look at modifications to the merit function and alternative functions that do not have this problem.

Acknowledgements

The authors would like to thank Heladia Salgado for supplying the RegulonDB data set files.

References

[1] E. P. Xing, M. I. Jordan, and R. M. Karp, “Feature Selection for High-Dimensional Genomic Microarray Data,” in *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001, pp. 601–608. [Online].

[2] R. E. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[3] L. C. Molina, L. Belanche, and A. Nebot, “Feature Selection Algorithms: A Survey and Experimental Evaluation.” in *Proc. of the International Conference on Data Mining (ICDM’02)*. Maebashi City, Japan: IEEE Computer Society, Dec 2002, pp. 306–313, ISBN 0-7695-1754-4.

[4] N. Chuzhanova, A. Jones, and S. Margetts, “Feature selection for genetic sequence classification,” *Bioinformatics*, vol. 14, no. 2, pp. 139–143, 1998.

[5] Y. Saeys, S. Degroeve, D. Aeyels, Y. Van de Peer, and P. Rouze, “Fast feature selection using a simple Estimation of Distribution Algorithm: A case study on splice site prediction.” *Bioinformatics*, vol. 19, no. 2, pp. 179–188, 2003.

[6] C. Park and S.-B. Cho, “Genetic Search for Optimal Ensemble of Feature-Classifier Pairs in DNA Gene Expression Profiles,” in *Neural Networks, 2003. Proceedings of the International Joint Conference on*, vol. 3, July 2003, pp. 1702–1707.

[7] D. K. Hawley and W. R. McClure, “Compilation and analysis of Escherichia coli promoter DNA sequences,” *Nucl. Acids. Res.*, vol. 11, no. 8, pp. 2237–2255, 1983.

[8] C. B. Harley and R. P. Reynolds, “Analysis of E. coli promoter sequences,” *Nucl. Acids. Res.*, vol. 15, no. 5, pp. 2334–2361, 1987.

[9] S. Lisser and H. Margalit, “Compilation of E.coli mRNA promoter sequences,” *Nucl. Acids. Res.*, vol. 21, no. 7, pp. 1507–1516, April 1993.

[10] I. Mahadevan and I. Ghosh, “Analysis of E.coli Promoter Structures using Neural Networks,” *Nucl. Acids. Res.*, vol. 22, no. 11, pp. 2158–2165, June 1994.

[11] Q. Ma, J. T. L. Wang, D. Shasha, and C. H. Wu, “DNA Sequence Classification via an Expectation Maximization Algorithm and Neural Networks: A Case Study,” *IEEE Transactions on Systems, Man and Cybernetics, part c*, vol. 31, no. 4, pp. 468–475, November 2001.

[12] Q. Ma, J. T. L. Wang, and J. R. Gattiker, *Mining Biomolecular Data Using Background Knowledge and Artificial Neural Networks in Handbook of Massive Data Sets*. Kluwer Academic Publishers, 2002, ch. 30, pp. 1141–1168.

[13] H. Salgado, A. Santos-Zavaleta, S. Gama-Castro, D. Millán-Zárate, E. Díaz-Peredo, F. Sánchez-Solano, E. Pérez-Rueda, C. Bonavides-Martínez, and J. Collado-Vides, “RegulonDB (version 3.2): Transcriptional Regulation and Operon Organization in Escherichia coli K-12,” *Nucl. Acids. Res.*, vol. 29, no. 1, pp. 72–74, 2001.

[14] H. Hirsh and M. Noordewier, “Using background knowledge to improve inductive learning of DNA sequences,” in *Artificial Intelligence for Applications, 1994., Proceedings of the Tenth Conference on*, March 1994, pp. 351–357.

[15] P. C. Conilione and D. Wang, “Effect of Non-Target Examples on E.coli Promoters Recognition Using Neural Networks,” in *Proceedings of International Joint Conference on Neural Networks IJCNN 2005*.Montreal, Canada: IEEE, 2005, pp. 310–315.

[16] G. G. Lennon and R. Nussinov, “Homonyms, synonyms and mutations of the sequence/structure vocabulary.” *J Mol Biol.*, vol. 175, no. 3, pp. 425–430, May 1984.

[17] R. M. Kolpakov and G. Kucherov, “Finding Approximate Repetitions under Hamming Distance,” in *ESA*, ser. Lecture Notes in Computer Science, F. M. auf der Heide, Ed., vol. 2161.Springer, 2001, pp. 170–181.

[18] R. L. Ornstein, R. Rein, D. L. Breen, and R. D. Macelroy, “An optimized potential function for the calculation of nucleic acid interaction energies I. Base stacking,” *Biopolymers*, vol. 17, no. 10, pp. 2341–2360, 1978.

[19] P. C. Conilione and D. Wang, “Neural Classification of E.coli Promoters Using Selected DNA Profiles,” in *The Fourth IEEE International Workshop on Soft Computing and Transdisciplinary Science and Technology*. Muroran, Japan: Springer, 2005, pp. 51–60.

[20] M. A. Hall and L. A. Smith, “Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper.” in *FLAIRS Conference*, 1999, pp. 235–239.

[21] G. F. Luger, *Artificial Intelligence*. Addison Wesley, 2002.

[22] L. Yu and H. Liu, “Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution.” in *ICML*, 2003, pp. 856–863.

[23] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, 1988.

[24] X. J. Zhou and T. S. Dillion, “A Heuristic - Statistical Feature Selection Criterion For Inductive Machine Learning In The Real World,” in *Systems, Man, and Cybernetics, 1988. Proceedings of the 1988 IEEE International Conference on*, vol. 1, Aug 1988, pp. 548–552.

[25] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[26] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, “A Practical Guide to Support Vector Classification,” 2005.

[27] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks,” in *2004 International Joint Conference on Neural Networks (IJCNN’2004)*.IEEE, July 2004.

[28] R. Kohavi and F. Provost, “Glossary of Terms,” *Mach. Learn.*, vol. 30, no. 2-3, pp. 271–274, 1998.

[29] M. A. Hall, “Correlation-Based Feature Selection for Machine Learning,” Ph.D. dissertation, The University of Waikato, Department of Computer Science, April 1999.

[30] P. C. Conilione and D. Wang, “E-coli Promoter Recognition Using Neural Networks with Feature Selection,” in *International Conference on Intelligent Computing, Lecture Note in Computer Science LNCS3645*, vol. 2, Hefei, China, August 23-26 2005, pp. 61–70.

Paul C. Conilione received his BSc (Honours) with a major in physics, and honours in wireless ad-hoc networks from Monash University, Australia, in 2003. He is currently a Masters of Science candidate at La Trobe University, Australia where he is undertaking research into neural based classifiers, features extraction and feature selection of biological sequences.

Dianhui Wang received his PhD from the Northeastern University, China in 1995. Since July 2001, he has been with the Department of Computer Science and Computer Engineering at the La Trobe University, Australia. He holds adjunct professorship and visiting professorship in China and South Korea, respectively.

Dr. Wang’s current working areas include data mining and soft computing techniques for bioinformatics, multimedia information processing, and intelligent diagnosis and control of engineering systems. He has published more than 110 technical papers in applied mathematics, control engineering and computer science.