Signal Detection in Genome Sequences Using Complexity Based Features

Mehdi Kargar, Aijun An, Nick Cercone, Kayvan Tirdad and Morteza Zihayat
Department of Computer Science and Engineering
York University, Toronto, Canada
{kargar,aan,nick,tirdad,zihayatm}@cse.yorku.ca

ABSTRACT

In this work, we tackle the problem of evaluating complexity methods and measures for finding interesting signals in the whole genome of three prokaryotic organisms. In addition to previous complexity measures, new measures are introduced for representing Open Reading Frames (ORF). We apply different classification algorithms to determine which complexity measure results in better predictive performance in discriminating genes from pseudo-genes in ORFs. Also, we investigate whether positions and lengths of windows in ORFs have significant impact on distinguishing between genes and pseudo-genes. Different classification algorithms are applied for classifying ORFs into genes and pseudo-genes[1].

General Terms

Classification Algorithms, Measurement

Keywords

Signal Detection, Biological Sequences, Complexity Measures

1. INTRODUCTION

One of the challenges of modern molecular biology is to analyze large amounts of experimental data. Due to the complexity of biological systems, it is not a trivial task. For acquiring complete and comprehensive knowledge, different techniques should be used for analyzing the input data and capture different types of signals in biological sequences. These techniques can be used to recognize specific regions with particular structures and specific functions in a long sequence. For example, they can be used to analyze phylogenetic networks, discriminating coding, non-coding and regulatory regions and recognizing repeated sequences [12, 15, 3, 2].

[1]This paper is an extended version of [7].

BIOKDD'13, Chicago, IL, USA.

The genome of an organism is a sequence of a four letter alphabet. The alphabet consists of four nucleotides: A (Adenine), C (Cytosine), G (Guanine) and T (Thymine). For detecting different signals and patterns in the genome sequence, different methods can be applied. These methods involve the use of a variety of complexity and entropy measures [5]. One of the most important properties of DNA sequences is their high repetitiveness. Different combinations of frequencies and repetitiveness in specific parts of genomic sequences indicate the presence and density of biological messages. Thus, diverging from an expected frequency or repeated pattern in both directions of DNA is assumed to be a possible presence of a biological signal [16].

In this work, the performance of different methods and measures in discriminating genes from pseudo-genes in the genome sequences of three prokaryotic organisms is evaluated. These three organisms include *Escherichia coli*, *Staphylococcus epidermidis* and *Streptococcus pyogenes*. In prokaryotic organisms genes and pseudo-genes belong to Open Reading Frames (ORFs). We will describe how to represent ORFs using different complexity measures. The measures applied in this work include frequency based, distance based, Pearson's chi-square, linguistic complexity and entropy measures. Previous work focused on linguistic complexity and entropy measures. We propose novel ORF representation methods that use a vector to present an ORF (such as the frequency-based and distance-based methods) and compare such representation methods to single value representation methods (such as the linguistic complexity and entropy measures). We evaluate the effectiveness of different representation methods by applying statistical and machine learning classification methods to the represented ORFs to distinguish genes from pseudo-genes.

Recently, it has been shown that the area around the start codon position of ORFs has the best performance for discriminating genes from pseudo-genes among other positions of an ORF [12]. Thus, we only consider this specific region of ORFs for our discriminating test. However, we test three parts of the area near the start codon position (i.e. before the start codon, in the middle of the start codon and after the start codon) to show which part has the best performance among the three parts. In order to observe the changing pattern of measures, we also apply the measures in some specific genome sequences.

The contributions of the paper can be summarized as follows:

1. We propose novel methods for representing ORFs using vectors. The vectors are defined based on frequen-

cies of nucleotide combinations of different lengths or the positions of the nucleotide combinations in a window.

2. We compare the proposed measures with the linguistic complexity and entropy-based measures by applying classification learning algorithms to ORFs to see which representation methods (i.e., measures) can better distinguish genes from pseudo-genes.

3. Shannon and CM entropy measures are considered using words of three different lengths (i.e. lengths 1, 2, 3).

4. We investigate which part of the start codon (before, middle and after) is more suitable for discriminating genes from pseudo-genes.

5. We investigate the effect of the size of the window for representing ORFs on discriminating genes from pseudo-genes.

6. To improve the reliability of the results, the methods are applied to three prokaryotic organisms.

7. We show that a good performance in statistical tests such as Mann-Whitney, does not necessarily imply a good performance in discrimination when the measure is used in a classification algorithm.

The rest of this paper is organized as follows. Section 2 reviews the related work. General structure of the methods are presented in section 3. Different methods for representing ORFs are introduced in section 4. Statistical and classification learning algorithms are briefly presented in section 5. Experimental evaluations and discussions are provided in section 6. Section 7 concludes the paper.

2. RELATED WORK

Structure and dynamics of living organisms are affected by the evolution, properties and complexity of genome sequences [8]. The complexity of sequences is useful in reproducing phylogenetic trees, compacting biological sequences, identifying the genomic structures and studying genomic evolution [9]. Since there is no unique definition for sequence complexity, various complexity measures have been defined for analyzing biological sequences. In contrast to complexity measures, low complexity regions have a well defined definition. Low complexity zones are produced in the presence of dispersed or tandem repeats, palindromic structures, biased nucleotide composition and also a combination of these properties [14].

Menconi et al. represented the complexity of a DNA sequence as the information content per nucleotide [8]. It is calculated using Lempel-Ziv data compression algorithm. They distinguished among genomes of different domains of life using the statistics of the complexity values of the functional regions. They also demonstrated that three domains of life (Archaea, Bacteria and Eukarya) might be plotted in separate zones within the two dimensional space. In that case, the axes are the skewness coefficient and the kurtosis coefficient of the aforementioned distribution.

The application of statistical methods such as Pearson's chi-square test to detect the signals in the whole genome of the *Escherichia coli* is presented in [12]. The efficiency of the method is evaluated by comparing the Pearson's chi-square test with linguistic, CE and CWF complexity on the complete genome of E. coli. They showed that Pearson's chi-square test distinguishes genes (coding regions) from pseudo-genes (non-coding regions). They also demonstrated which parts of the ORF have significant effect on discriminating genes from pseudo-genes. These parts are 100 nucleotides before the start codon position, around the start codon position, the middle position of an ORF and around the stop codon position. They concluded that the region around the start codon has the best performance in discriminating genes from pseudo-genes.

Sequence	A	C	A	T	G	G	T	C	A	T
Position	1	2	3	4	5	6	7	8	9	10

Figure 1: An example sequence of size 10.

The efficiency of detrended fluctuation analysis and rescaled range analysis in discriminating coding DNA, regulatory DNA and non-coding non-regulatory DNA is evaluated in [15]. The authors applied the measures on the genome sequence of *Drosophila melanogaster*. They estimated the degree of sequential persistence or dependence among nucleotides. They demonstrated that these methods can distinguish the three types of DNA in eukaryotes. It is also reported that the performance of rescaled range analysis is better than detrended fluctuation analysis. In addition, the region of coding DNA is classified as anti-persistent and the regulatory regions have intermediate sequential dependency. The degree of sequential persistence of non-coding, non-regulatory DNA is higher than the other DNA types. They concluded that rescaled range and detrended fluctuation analysis are useful tools for refined functional and structural segmentation of DNA in eukaryotic organisms.

None of the above work has used a vector-based measure in comparison to complexity measures. Also, none of them has applied chi-square on weighted distance vectors. In addition, in this work we consider the words of length 1, 2 and 3 for calculating Shannon and CM entropy measures.

3. FINDING AND REPRESENTING ORFS

In this section we describe our method for identifying ORFs from a DNA sequence and our process of computing the measures based on identified ORFs.

3.1 Finding Open Reading Frame Regions

In molecular biology and genetics, an open reading frame (ORF) is a portion of an organism's genome which contains a sequence of bases that could potentially encode a protein. If an ORF codes a protein, it is a gene; otherwise it is a pseudo-gene. Theoretically, the DNA sequence can be read in six reading frames in organisms with double-stranded DNA; three on each strand. Each frame contains some genes and proteins. In prokaryotes, ORFs start with the start codon (ATG) and end with one of three stop codons (TAA, TAG and TGA). The longest sequence without a stop codon usually determines the ORF. In our work, we simply use this rule (which was used in [12]) to identify ORFs in the genome sequences of three prokaryotic organisms. Note that this

simple method is not applicable to eukaryotes. The process of finding ORFs in eukaryotic genomes is difficult because of the existence of introns, exons and gaps in their genomes.

3.2 Process of Computing the Measures

There are several positions in an ORF which might have important signals that could be useful for discriminating genes from pseudo-genes. These positions are the start codon position, the middle of the ORF and the stop codon position. Since authors in [12] showed that the region near the start codon has the best performance in discriminating genes from pseudo-genes, we concentrate on this position in this work. However, considering the region near the start codon position, there are three areas to study. These areas are: before the start codon, in the middle of the start codon and after the start codon. We consider these three areas to see which one is more effective in representing an ORF than others. For each area, we consider a window of size n, where n is the number of nucleotides in the window (i.e., the subsequence or substring in the area). A window S of size n is shown as $S = (s_1, s_2, \ldots, s_n)$.

For each ORF and each area, a measure of interest is calculated in the window (as described in the next section). It is expected that the distribution and order of nucleotides are different in genes and pseudo-genes. Thus, the measure can be used to divide ORFs to two categories: genes and pseudo-genes.

4. REPRESENTATION OF ORFS WITH DIFFERENT MEASURES

In this section, we present how we use different complexity measures to represent an ORF. The measures include frequency based, distance based, Pearson's chi-square, linguistic complexity and entropy measures.

4.1 Frequency-based Measures

4.1.1 Vector-based Frequency Measure

The frequency of a word X of length one in window S is simply defined as $freq_X = \sum_{i=1}^{i=N}\{1|(s_i) = X\}$. It is extendable to words of length two or three. The frequency of word Y of length two and word Z of length three are defined as $freq_Y = \sum_{i=1}^{i=n-1}\{1|(s_i s_{i+1}) = Y\}$ and $freq_Z = \sum_{i=1}^{i=n-2}\{1|(s_i s_{i+1} s_{i+2}) = Z\}$. Since the alphabet size is four ($\{A, C, G, T\}$), there are 4 combinations of nucleotides of length one, 16 combinations of nucleotides of length two and 64 combinations of nucleotides of length three. Thus, we can use a 4 dimensional vector of length-1 word frequencies to represent an ORF in a window as follows:

$$\overrightarrow{freq1} = \langle freq_A, freq_C, freq_G, freq_T \rangle. \quad (1)$$

Similarly, a 16 or 64 dimensional vector of length-2 or length-3 word frequencies can be used to represent an ORF as well:

$$\overrightarrow{freq2} = \langle freq_{AA}, freq_{AC}, freq_{AG}, \ldots, freq_{TT} \rangle \quad (2)$$

$$\overrightarrow{freq3} = \langle freq_{AAA}, freq_{AAC}, \ldots, freq_{TTT} \rangle. \quad (3)$$

4.1.2 Single Value-based Frequency Measure

Existing complexity measures for ORFs produce single values instead of vectors. To make a comparison, we also define single-value measures based on Equations 1, 2 and 3. We will compare the single-value measures with the vector-based measures in the experiment section. To define a single-value frequency-based measure, we use the angle between each vector and the average vector. The average vector is the vector whose frequencies are the average frequency of windows of length n in the whole genome sequence. The average vector of words of length one, two and three are represented by $\overrightarrow{Avg1}$, $\overrightarrow{Avg2}$ and $\overrightarrow{Avg3}$ respectively. This measure is called frequency ratio and is denoted by $freq\_ratio1$, $freq\_ratio2$ and $freq\_ratio3$ for words of length one, two and three respectively. Since the purpose is to see how far each vector is from the average vector, the measure, which is based on the cosine of the angle between two vectors, is defined as follows:

$$freq\_ratioi = 1 - \frac{\overrightarrow{freqi} \bullet \overrightarrow{Avgi}}{\|\overrightarrow{freqi}\| \times \|\overrightarrow{Avgi}\|} \quad (4)$$

where i ranges from 1 to 3 representing the word length, $\overrightarrow{freqi} \bullet \overrightarrow{Avgi}$ is the dot product of two vectors $\overrightarrow{freqi}$ and $\overrightarrow{Avgi}$, and $\|\overrightarrow{freqi}\|$ and $\|\overrightarrow{Avgi}\|$ are the magnitude of the vectors.

4.2 Weighted Distance Measures

In this section we define new measures based on where a word occurs in a window. These measures are based on the distances from the beginning of the window to each occurrence of a word in the window.

4.2.1 Weighted Distance

Assume that the first nucleotide in a window is at position 1 , the second nucleotide is at position 2, and so on (see Figure 1). We define the *distance of a nucleotide X at position i* to be the number of the nucleotides from the beginning of the window to position i, which is i. To measure the distribution of a nucleotide X in the window, a summed distance weighted by the frequency of X in the window can be used:

$$WD_X = freq_X \times (\sum_{i=1}^{i=n}\{i|(s_i) = X\}). \quad (5)$$

We call this measure *weighted distance* of X. [2] For example, Figure 1 shows a window of size 10. Nucleotide A occurs three times at positions 1, 3 and 9. The weighted distance of A is $WD_A = 3 \times (1 + 3 + 9) = 39$.

Similarly, the weight distance of a length-2 word Y is defined as

$$WD_Y = freq_Y \times (\sum_{i=1}^{i=n-1}\{i|(s_i s_{i+1}) = Y\}) \quad (6)$$

and the weighted distance of a length-3 word Z is defined as

$$WD_Z = freq_Z \times (\sum_{i=1}^{i=n-2}\{i|(s_i s_{i+1} s_{i+2}) = Z\}). \quad (7)$$

4.2.2 Vector-based Distance Measure

Since there are four nucleotides, i.e., $X \in \{A, C, G, T\}$ in Equation 5, we can use a 4 dimensional vector of the

[2]This measure is inspired by a measure of divergence defined in [1].

weighted distances of the 4 nucleotides to represent the distribution of the length-one words in a window as follows:

$$\overrightarrow{WD1} = \langle WD_A, WD_C, WD_G, WD_T \rangle \quad (8)$$

For example, the weighted distance vector of length-one words for the sequence in Figure 1 is $\langle 39, 20, 22, 63 \rangle$ because $WD_A = 3 \times 13 = 39$, $WD_C = 2 \times 10 = 20$, $WD_G = 2 \times 11 = 22$ and $WD_T = 3 \times 21 = 63$.

Similarly, the weighted distance vector of length-2 or length-3 words is defined using a 16 or 64 dimensional vector, respectively, as follows:

$$\overrightarrow{WD2} = \langle WD_{AA}, WD_{AC}, \ldots, WD_{TT} \rangle \quad (9)$$

$$\overrightarrow{WD3} = \langle WD_{AAA}, WD_{AAC}, \ldots, WD_{TTT} \rangle \quad (10)$$

4.2.3 Single-valued Distance Measure

Similar to the frequency-based measures, we define single-value weighted distance measures as follows. Let $\overrightarrow{WDAvg1}$, $\overrightarrow{WDAvg2}$ or $\overrightarrow{WDAvg3}$ denote the average weighted distance vector for length 1, 2 or 3 words respectively, which is computed by taking the average values of the weighted distance vectors for windows of size n in the whole genome sequence. We define the weighted distance ratio of length-i words to be the cosine of the angle between the weighted distance vector and the average vector of length-i words:

$$WD\_ratioi = 1 - \frac{\overrightarrow{WDi} \bullet \overrightarrow{WDAvgi}}{\|\overrightarrow{WDi}\| \times \|\overrightarrow{WDAvgi}\|} \quad (11)$$

where i ranges from 1 to 3.

4.3 Pearson's Chi-square Measure

Pearson's chi-square test (χ^2) is a statistical measure which tests a null hypothesis that the relative frequencies of occurrence of observed events have an expected frequency distribution. In addition, the events should be independent and follow the same distribution [12]. In the previous section, the frequency vector and the average vector were introduced for words of length one, two and three. The purpose is to find the regions that the distribution of nucleotides is different from the ones for whole genome sequence. For comparing these two frequencies, Pearson's chi-square test is applied and calculated as follows:

$$\chi^2 = \sum_{i=1}^{M} \frac{(freq_i - Avg_i)^2}{Avg_i} \quad (12)$$

where $freq_i$ and Avg_i are the observed and the average frequencies of the i-th word. M is total number of words and equals 4, 16 and 64 for combinations of nucleotides of length one, two and three respectively. Pearson's chi-square test for words of length one, two and three are shown by χ_1^2, χ_2^2 and χ_3^2 respectively. The chi-square test for weighted distance measures is defined in the same manner and is represented by $WD\chi_1^2$, $WD\chi_2^2$ and $WD\chi_3^2$ for words of length one, two and three respectively.

4.4 Linguistic Complexity Measure

One of the complexity based measures is linguistic complexity. Linguistic complexity is represented by the ratio of the number of subsequences that occur in the sequence of interest to the maximum number of subsequences for a sequence of the same length over a same alphabet [16]. Since the sequence is a DNA, the alphabet is defined as A, C, G and T. The maximum number of subsequences (also called *maximum vocabulary*) can be computed as follows:

$$maximum\ vocabulary = \sum_{L=1}^{N} min(A^L, N - L + 1) \quad (13)$$

where A is the alphabet size, L is the subsequences length and N is the size of the window. The complexity measure (referred to as *Ling* in this paper) is defined as

$$Ling = \frac{n}{maximum\ vocabulary}, \quad (14)$$

where n is the number of subsequences that occur in the sequence of interest.

4.5 Entropy Measures

Shannon's entropy is defined based on the probability of occurrence of the symbols [9]. The entropy of a subsequence (window) S is calculated based on the following relation:

$$Shannon's\ entropy = -\sum_{i=1}^{M} [m_i/(N-m+1)] \log_2 [m_i/N-m+1] \quad (15)$$

where N is the window size, m is the length of the word (in this work one, two or three), $M = A^m$ is the total number of words with length m, A is the alphabet size (in this work it equals 4) and m_i is the number of i-th word in a window.

In this work, we used words of lengths one, two and three. Thus, the Shannon's entropy for these different word lengths are called $Shannon1$, $Shannon2$ and $Shannon3$ respectively.

The entropy of high order Markov model (CM) which is similar to Shannon's entropy is given as [10]:

$$CM = -\sum_{i=1}^{M} [m_i/(N - m + 1)] \log_M [m_i/N - m + 1] \quad (16)$$

the parameters are the same as Shannon's entropy in relation 15. Like Shannon's entropy, the CM entropy for these different word lengths are called $CM1$, $CM2$ and $CM3$.

5. CLASSIFICATION ALGORITHMS

To determine which of the above measures can better distinguish gene and pseudo-genes, we apply classification learning algorithms to learn classification models from the ORFs represented by the measures and use the learned models to classify some other ORFs into genes or pseudo-genes. In this section we briefly introduce the classification learning algorithms that we use for evaluating different complexity measures.

Logistic Regression. In statistics, logistic regression predicts the probability of occurrence of an event by fitting data to a logistic curve. It is a generalized linear model used for binomial regression. It uses one or more predictor variables. The variables can be either numerical or categorical. Unlike some other statistical methods such as discriminant analysis, logistic regression does not assume the data follow the normal distribution. Since the values of the complexity measures for genes and pseudo-genes do not usually have normal distribution, we use logistic regression in this work.

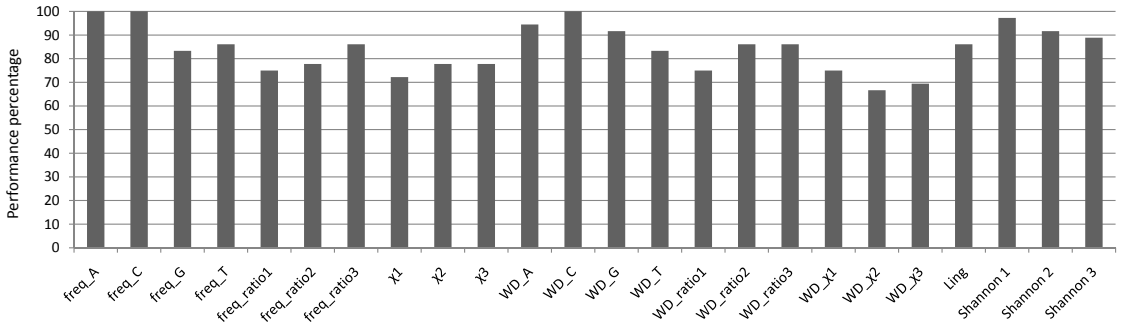

Figure 2: The percentage of the datasets that have significantly different distribution between genes and pseudo-genes using Mann-Whitney test.

C4.5 Algorithm. C4.5 is a decision tree learning algorithm [13]. A decision tree is a flowchart-like tree structure, where each internal node represents a test on an attribute, each branch represents an outcome of the test, and each leaf node is assigned to a class label. A decision tree can be used for classification. In C4.5, an information theoretic measure, *information gain ratio* is used as a criterion for attribute selection in the decision tree induction process.

RIPPER Algorithm. RIPPER is an inductive rule learner [4]. It learns a set of propositional rules from a set of data by first generating overfitting rules from a subset of training data and then post-pruning the rules using an optimized incremental reduced error punning technique based on a remaining set of the training data. The algorithm was shown to be an efficient rule learner that produces competitive results with C4.5 with respect to predictor error rates [4].

K-Nearest Neighbors Algorithm. The k-nearest neighbors algorithm (k-NN) is a classification method that classifies a new example x by finding the k examples that are closest to x from the training data and uses the most common class among the k examples to classify x. The method does not learn a general model; instead it stores all the training examples for use in the classification phase. In k-NN, parameter k is usually set to a small odd number such as 3, 5 or 7 [6].

Bayesian Belief Networks. A Bayesian belief network [11] is a direct acyclic graph whose nodes represent variables and edges represent probabilistic dependencies. Each variable in the network is conditionally independent of its nondescendants in the graph, given its parents. Each node is associated with a probability function that takes as input a particular set of values for the node's parent variables and gives the probability of the variable represented by the node. A Bayesian network can be learned from a set of training data and used to predict the class membership probabilities for a given object.

Support Vector Machines. Support vector machines (SVMs) make use of a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it finds a decision boundary that separates the objects of one class from other classes. Using a proper nonlinear mapping to a sufficiently high dimension, different objects that belongs to different classes can be separated by a hyperplane. The SVM produces this hyperplane using margins and support vectors [6].

Neural Networks. A neural network consists of a set of interconnected computing units. Each connection (edge) has a weight associated with it. A neural network can be used to predict class labels for the input objects. In the learning phase, the network tries to improve its performance by adjusting the weights of the edges of the network. Training a neural network may take a long time, but a neural network usually has good predictive performance in the presence of noise. There are different types of neural networks and neural network training algorithms. In our experiment, we use the backpropagation algorithm [6].

6. RESULTS AND DISCUSSION

In this section the results of different complexity measures for discriminating genes from pseudo-genes are presented. The results are obtained from three different prokaryotic organisms. These three organisms include *Escherichia coli*, *Staphylococcus epidermidis* and *Streptococcus pyogenes*. The complexity measures are calculated in three different locations near the start codon position of ORFs. These locations include before the start codon, in the middle of the start codon and after the start codon. In addition, the complexity measures are calculated in windows of size 50, 100, 200 and 500.

Two evaluation methods are used to evaluate the complexity measures. First, we determine whether a measure can result in values whose distributions are significantly different among genes and pseudo-genes. The results of this evaluation are presented in Section 6.1. Second, we apply classification methods to the ORFs represented by the complexity measures and use cross-validation to determine whether a complexity measure can lead to a good classification performance. The results of this evaluation are presented in Sections 6.2-6.6. In the evaluation, we also investigate whether locations and window sizes in an ORF have significant impacts on the predictive performance. Since the number of genes is much less than the number of pseudo-genes (for example, the ratio of genes to pseudo-genes in ecoli are around 1/5), we apply down-sampling to the sets of ORFs to balance the data before learning classification models from ORFs. This is to improve the performance of classification algorithms. Down-sampling is performed by randomly removing some pseudo-genes from the dataset until we have the same number of genes and pseudo-genes for each organism.

It should be noted that all of the results are obtained using appropriate statistical methods. All of the statistical tests and the logistic regression method are performed using

SPSS 15.0 for Windows. Other classification algorithms are performed using Weka 3.6.

6.1 Value Distribution of Complexity Measures

To see how values of the complexity measures are distributed among genes and pseudo-genes, one-sample Kolmogorov-Smirnov test is performed. This is a non parametric test. This test suggests that none of the complexity measures, when applying to any of the three organisms with any window size at any location, has a normal distribution (p-value < 0.05). Thus, to determine whether the values from a complexity measure have significantly different distributions among genes and pseudo-genes, we use Mann-Whitney test. This is also a non parametric test. The significance level is set to 0.05. Since there are three organisms, four different window sizes and three different locations, we have 36 different pairs of value sets for each measure, one set for genes and the other for pseudo-genes in each pair. For each measure the percentage of pairs that have significant different distributions between genes and pseudo-genes is used as a criterion for evaluating the performance of the measure. We call such a percentage *difference percentage*. A higher difference percentage implies a better ability for the measure to distinguish genes and pseudo-genes through its values. Figure 2 illustrates the *difference percentages* of all the measures. For example, for measure *count_ratio*1, 9 out of 36 pairs of value sets for genes and pseudo-genes do not have significant different distributions between genes and pseudo-genes. Thus, the difference percentage of *count_ratio*1 is 75%. The results in Figure 2 show that some frequency, distance and Shannon measures have better performance than others in terms of *difference percentage*. In particular, $freq\_A$, $freq\_C$ and WD_C always have significantly different distributions for genes and pseudo-genes.

To show the difference between the average value of a measure on genes and the one on pseudo-genes, we illustrate the average values of each measures for genes and pseudo-genes for *E. coli* with window size 100 and the middle of the start codon location in Figure 3. Since different complexity measures have different value ranges, all the average values are normalized to fall between 0 and 1.

6.2 Predictive Performance of Different Complexity Measures

In Figure 4, the performance of different complexity measures in discriminating genes from pseudo-genes using logistic regression is presented. Since 3 organisms, 4 different window sizes and 3 different locations are examined, we obtained a set of 36 performance results (in terms of classification accuracy) for each complexity measure. As this figure suggests, the performance of vector measures ($\overrightarrow{freqi}$ and $\overrightarrow{WDi}$) are significantly better than the performance of other measures (p-value < 0.05).

By increasing the size of the frequency and weighted distance vectors, the performance of them increase too. On the other hand, by increasing the vector size, the running time of the algorithms increases. The performance of $freq\_ratio1$, $freq\_ratio2$, $freq\_ratio3$, WD_ratio1, WD_ratio2, WD_ratio3, χ_1^2, χ_2^2, χ_3^2, $WD\_\chi_1^2$, $WD\_\chi_2^2$ and $WD\_\chi_3^2$ is poor and is around 55%. Increasing the length of words does not change the performance very much. The performance of the linguistic method is also poor. The performance of Shannon and CM are very close together. Their performance are around 60%. By increasing the length of word, the performance of Shannon and CM decreases.

Table 1: Classification accuracy of different window sizes using logistic regression method

Measure name	50	100	200	500
$\overrightarrow{freq1}$	74.73 %	73.83 %	72.81 %	71.72 %
$\overrightarrow{freq2}$	77.62 %	76.57 %	76.32 %	75.68 %
$\overrightarrow{freq3}$	81.88 %	81.38 %	80.68 %	79.21 %
$freq\_ratio1$	55.42 %	54.53 %	53.32 %	51.80 %
$freq\_ratio2$	54.63 %	53.72 %	53.53 %	52.11 %
$freq\_ratio3$	58.52 %	58.58 %	56.94 %	53.67 %
$\overrightarrow{WD1}$	73.79 %	73.63 %	73.10 %	72.42 %
$\overrightarrow{WD2}$	75.50 %	75.87 %	76.12 %	76.19 %
$\overrightarrow{WD3}$	78.93 %	79.56 %	79.99 %	79.52 %
WD_ratio1	54.04 %	53.21 %	52.93 %	52.01 %
WD_ratio2	56.20 %	55.71 %	54.63 %	53.20 %
WD_ratio3	57.93 %	57.98 %	56.96 %	54.77 %
χ_1^2	56.26 %	55.40 %	53.56 %	51.49 %
χ_2^2	54.63 %	52.91 %	52.86 %	51.92 %
χ_3^2	57.47 %	51.38 %	54.70 %	53.08 %
$WD\_\chi_1^2$	55.01 %	53.72 %	53.21 %	51.62 %
$WD\_\chi_2^2$	53.54 %	52.96 %	52.69 %	51.93 %
$WD\_\chi_3^2$	53.03 %	52.19 %	52.54 %	52.16 %
$Ling$	56.30 %	56.10 %	56.09 %	52.21 %
$Shannon1$	62.78 %	61.50 %	59.42 %	55.74 %
$Shannon2$	59.89 %	59.82 %	58.84 %	55.68 %
$Shannon3$	57.48 %	57.67 %	57.62 %	55.78 %

6.3 The Effect of Different Window Sizes on the Performance of Complexity Measures

The performance of different window sizes using logistic regression is presented in Table 1. Since the performance of $Shannon$ and CM are very close, the CM results are not shown. For analyzing the results, one-way analysis of variance (one-way ANOVA) is used. This statistical test suggests that except measures $freq\_ratio3$, χ_1^2 and χ_3^2 which have significant different performance for different window sizes (p value < 0.05), other measures does not have significant different performance for different window sizes.

For comparing a pair of window sizes, we used Bonferroni's method. This method is useful for small datasets. This method reveals that there is not a significant difference between any pair of window sizes except for windows of size 50 and 500. For windows of size 50 and 500, there is a significant difference between measures χ_1^2 and χ_3^2 (p value < 0.05). Other measures for these window sizes have the same performance distribution. Although the average performance of windows of size 50 and 100 are higher than that of windows of size 200 and 500, in most cases there is not a significant difference between them.

6.4 The Effect of Different Locations on the Performance of Complexity Measures

The performance of different locations using logistic regression is presented in table 2. Since the performance of $Shannon$ and CM are very close, the CM results are not shown. For analyzing the results, one-way analysis of variance (one-way ANOVA) is used. For comparing a pair of locations, we used Bonferroni's method. The results of one way ANOVA and Bonferroni's method are shown in Table 4. The significant differences are shown in bold numbers. The

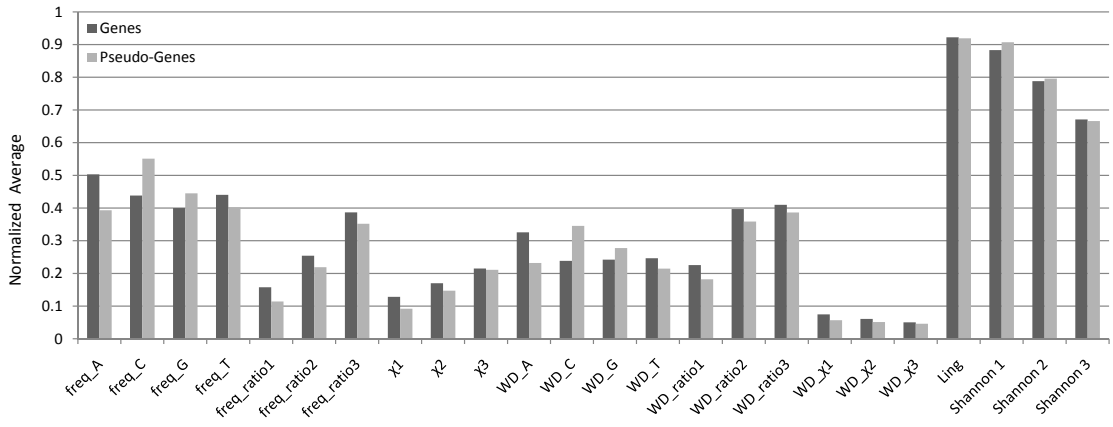

Figure 3: The normalized average of different complexity measures for organism *E. coli*, window size 100 and the middle of the start codon location.

results suggest that the middle and before locations have better performance than the after location for discriminating genes from pseudo-genes.

Table 2: Classification accuracy of different locations using logistic regression method

Measure name	before	middle	after
$\overrightarrow{freq1}$	73.09 %	76.76 %	69.96 %
$\overrightarrow{freq2}$	75.93 %	79.97 %	73.73 %
$\overrightarrow{freq3}$	79.83 %	83.49 %	78.98 %
$freq\_ratio1$	54.38 %	54.97 %	51.97 %
$freq\_ratio2$	53.16 %	53.73 %	53.61 %
$freq\_ratio3$	57.25 %	58.67 %	54.85 %
$\overrightarrow{WD1}$	75.01 %	76.12 %	68.56 %
$\overrightarrow{WD2}$	77.21 %	78.67 %	71.86 %
$\overrightarrow{WD3}$	80.92 %	81.42 %	76.15 %
WD_ratio1	53.96 %	53.14 %	52.04 %
WD_ratio2	55.48 %	55.95 %	53.39 %
WD_ratio3	58.11 %	58.62 %	53.98 %
χ_1^2	55.34 %	55.43 %	51.75 %
χ_2^2	53.11 %	53.22 %	52.90 %
χ_3^2	54.35 %	55.24 %	52.87 %
$WD\_\chi_1^2$	54.90 %	53.95 %	51.33 %
$WD\_\chi_2^2$	54.36 %	52.80 %	51.16 %
$WD\_\chi_3^2$	53.43 %	52.68 %	51.32 %
$Ling$	56.35 %	56.46 %	52.70 %
$Shannon1$	61.50 %	62.55 %	55.53 %
$Shannon2$	59.95 %	61.04 %	54.67 %
$Shannon3$	58.23 %	59.47 %	53.70 %

6.5 The Performance of Windows with Small Size

The performance of vectors $\overrightarrow{freq1}$, $\overrightarrow{freq2}$, $\overrightarrow{WD1}$ and $\overrightarrow{WD2}$ in windows with small sizes are presented in Table 3. The performance of windows of size 10 and 20 are compared to windows of size 50 and 100. Since the vector measures have better performance, we only evaluated them here. The results suggest that by decreasing the window size 20 or 10, we lose some information since the predictive performance decreases significantly. On the other hand, increasing the window size to 200 or 500 also decreases the predictive performance although not significantly (as shown in Table 2).

Table 3: The Performance of Windows with Small Size

Measure name	10	20	50	100
$\overrightarrow{freq1}$	65.53 %	73.63 %	74.76 %	73.83 %
$\overrightarrow{WD1}$	63.41 %	71.74 %	73.78 %	73.60 %
$\overrightarrow{freq2}$	69.37 %	76.87 %	77.62 %	76.57 %
$\overrightarrow{WD2}$	67.32 %	73.92 %	75.50 %	75.87 %

Thus, for discriminating genes from pseudo-genes, a suggested window size is between 50 and 100.

6.6 The Performance of Different Classification Algorithms in Discriminating Genes from Pseudo-Genes

The performance of different classification algorithms in discriminating genes from pseudo-genes is presented in Figure 5. The methods include C4.5, RIPPER, kNN, Bayesian network, SVM and neural network. The input data are taken from organism *E. coli*, with window size 100 and the location is set to the middle of the start codon. Similar to the result for the logistic regression method, the performance of vector-based measures is higher than the performance of single values. In C4.5, by increasing the size of vector, from 4 to 64, the performance of the algorithm decreases. By increasing the word length from 1 to 3, the performance of $freq\_ratio$, WD_ratio and χ^2 decreases. On the other hand, the performance of WD_χ^2 increases. In addition, $Shannon2$ and $Shannon3$ has worst performance.

It should noted that Shannon measures have good performance in the Mann-Whitney test as presented in Section 6.1. Their poor performance in classification accuracy suggests that a good performance in Mann-Whitney does not imply good performance with classification algorithms.

7. CONCLUSION

The performance of different complexity measures for discriminating genes from pseudo-genes is presented in this work. First, we introduce new vector-based measures based on frequency and weighted distance. We also introduce some new single-valued measures by converting a vector into a sin-

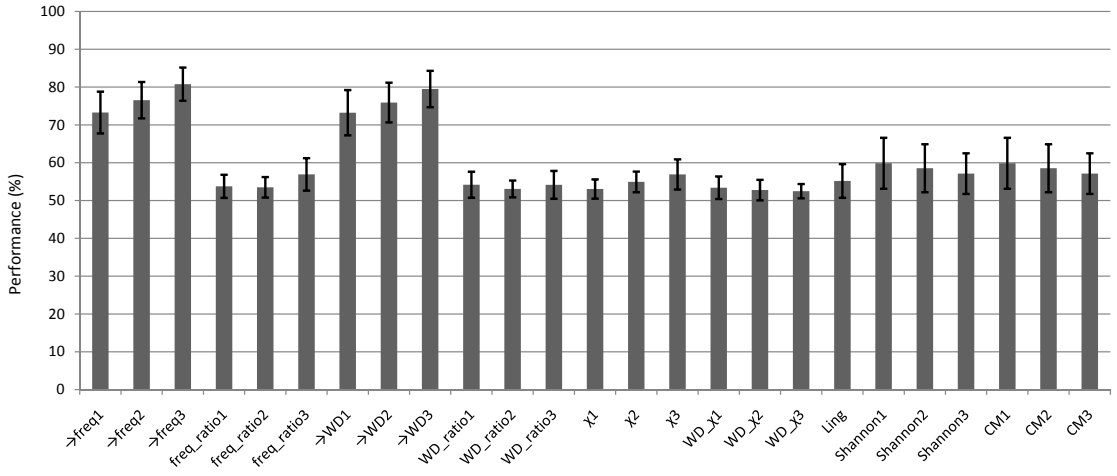

Figure 4: The classification accuracy of different complexity measures in discriminating genes from pseudo-genes using the logistic regression method.

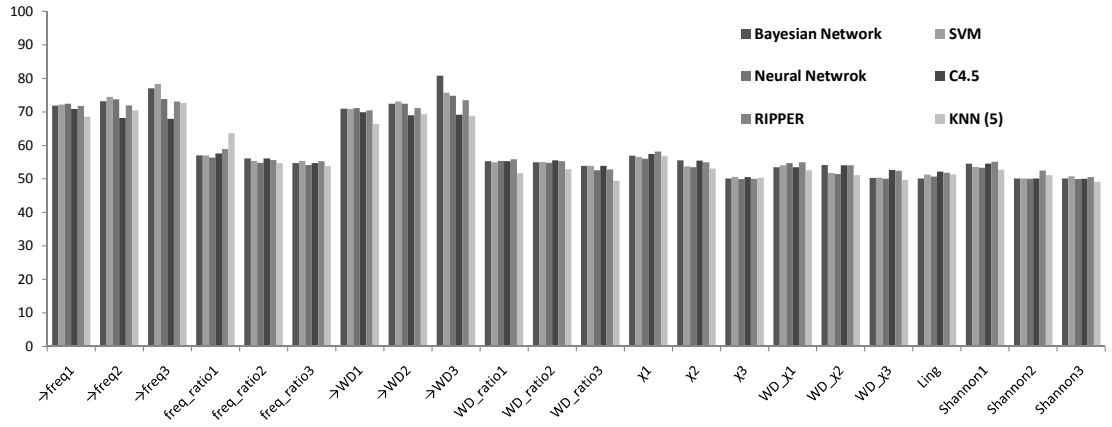

Figure 5: The classification accuracy of Bayesian network, SVM, neural network, C4.5, RIPPER and kNN (5) in discriminating genes from pseudo-genes for organism *E. coli*, window size equals 100 and the middle of the start codon location is used

gle value using cosine or chi-square statistics. These measures are compared with linguistic complexity, Shannon and CM entropy measures. In addition, Shannon and CM entropies are calculated using words of length 1, 2 and 3. To evaluate the accuracy and reliability of the results, the methods are applied on three prokaryotic organisms. All of these measures are applied on different locations near the start codon positions of ORFs with different window sizes. Statistical tests such as Mann-Whitney suggest that the distribution of most of these measures are different in genes and pseudo-genes. However, when classification algorithms are applied to classify ORFs to two groups, the performance of single value measures is poor. On the other hand, the performance of vector-based measures is significantly better than the performance of single value measures. By increasing the size of a vector from 4 to 16 and from 16 to 64, the performance of classification algorithms increases. These results suggest that the high significant level of statistical tests does not imply a reasonable performance when the data are used as input to classification algorithms. In addition, the statistical analysis reveals that in most cases, there is not a significant difference between windows of size 50, 100, 200 and 500. However, the performances of windows of size 50 and 100 are better than those of the windows of size 200 and 500. In addition, the performance of windows with small sizes (such as 10 and 20) is not comparable to that of windows of size 50 and 100. Using statistical tests, we conclude that the location before the start codon position or in the middle of the start codon position has better performance in comparison to the location after the start codon position. In this work, we used several classification algorithms with different approaches to see the performance of the proposed methods. The results show that the performance of different algorithms are very closet o each other.

8. REFERENCES

[1] A. An, Q. Wan, J. Zhao, and X. Huang. Diverging patterns: discovering significant frequency change dissimilarities in large databases. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1473–1476. ACM, 2009.

Table 4: The significant difference between different locations. One-way ANOVA is used to compare the the combination of total locations. Bonferroni's method is used to compare pairs of locations(Before vs Middle, Before vs After and Middle vs After). The significant level is set to 0.05.

Measure name	Total	Bef-Mid	Bef-Aft	Mid-Aft
$\overrightarrow{freq1}$	**0.01**	0.23	0.38	**0.01**
$\overrightarrow{freq2}$	**0.00**	0.07	0.61	**0.00**
$\overrightarrow{freq3}$	**0.02**	0.11	1.00	**0.03**
$freq\_ratio1$	**0.04**	1.00	0.14	**0.04**
$freq\_ratio2$	0.87	1.00	1.00	1.00
$freq\_ratio3$	0.09	1.00	0.49	0.09
$\overrightarrow{WD1}$	**0.00**	1.00	**0.01**	**0.00**
$\overrightarrow{WD2}$	**0.00**	1.00	**0.02**	**0.00**
$\overrightarrow{WD3}$	**0.01**	1.00	**0.03**	**0.02**
WD_ratio1	0.18	1.00	0.20	0.86
WD_ratio2	**0.05**	1.00	0.17	0.06
WD_ratio3	**0.00**	1.00	**0.02**	**0.01**
χ_1^2	**0.01**	1.00	**0.02**	**0.02**
χ_2^2	0.94	1.00	1.00	1.00
χ_3^2	0.29	1.00	0.99	0.37
$WD\_\chi_1^2$	**0.01**	1.00	**0.01**	0.06
$WD\_\chi_2^2$	**0.01**	0.38	**0.01**	0.32
$WD\_\chi_3^2$	**0.02**	0.88	**0.02**	0.19
$Ling$	0.06	1.00	0.13	0.11
$Shannon1$	**0.02**	1.00	0.07	**0.03**
$Shannon2$	**0.03**	1.00	0.10	**0.04**
$Shannon3$	**0.02**	1.00	0.09	**0.02**

[2] W. Ashlock and S. Datta. Distinguishing endogenous retroviral ltrs from sine elements using features extracted from evolved side effect machines. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(6):1676–1689, 2012.

[3] W. Ashlock and S. Datta. Evolved features for dna sequence classification and their fitness landscapes. *IEEE Transactions on Evolutionary Computation*, 14(2):185–198, 2013.

[4] W. W. Cohen. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine learning*, 1995.

[5] F. F. Dedus, L. I. Kulikova, S. A. Makhortykh, N. N. Nazipova, A. N. Pankratov, and R. K. Tetuev. Analytical recognition methods for repeated structures in genomes. *Doklady Mathematics*, 24(3):926–929, December 2006.

[6] J. Han and M. Kamber. *Data Mining: Concepts and Techniques, Second Edition*. Morgan Kaufmann Publisher, 2006.

[7] M. Kargar and A. An. Evaluation of different complexity measures for signal detection in genome sequences. In *Proceedings of the First ACM International Conference On Bioinformatics and Computational Biology (ACM-BCB'10)*, 2010.

[8] G. Menconi, V. Benci, and M. Buiatti. Data compression and genomes: A two-dimensional life domain map. *J. Theor. Biol.*, 253(2):281–288, July 2008.

[9] F. Nan and D. Adjeroh. On complexity measures for biological sequences. In *Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference*, pages 522–526. TeX Users Group, 2004.

[10] Y. L. Orlov and V. N. Potapov. Complexity: an internet resource for analysis of dna sequence complexity. *Nucleic Acids Res.*, 23:628–633, 2004.

[11] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Publisher, 1988.

[12] L. Pirhaji, M. Kargar, A. Sheari, H. Poormohammadi, M. Sadeghi, H. Pezeshk, and C. Eslahchi. The performances of the chi-square test and complexity measures for signal recognition in biological sequences. *J. Theor. Biol.*, 251(2):380–387, March 2008.

[13] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publisher, 1993.

[14] A. Sheari, M. Kargar, A. Katanforoush, S. Arab, M. Sadeghi, H. Pezeshk, C. Eslahchi, and S.-A. Marashi. A tale of two symmetrical tails: Structural and functional characteristics of palindromes in proteins. *BMC Bioinformatics*, 9(274), 2008.

[15] R. te Boekhorst, I. Abnizova, and C. Nehaniv. Discriminating coding, non-coding and regulatory regions using rescaled range and detrended fluctuation analysis. *Biosystems*, 91(1):183–194, January 2008.

[16] O. G. Troyanskaya, O. Arbell, Y. Koren, G. M. Landau, and A. Bolshoy. Sequence complexity profiles of prokaryotic genomic sequences: A fast algorithm for calculating linguistic complexity. *Bioinformatics*, 18(5):679–688, 2002.