PolyA-iEP: A Data Mining Method for the Effective Prediction of Polyadenylation Sites

George Tzanis, Ioannis Kavakiotis, and Ioannis Vlahavas
Department of Informatics, Aristotle University of Thessaloniki

Abstract

This paper presents a study on polyadenylation site prediction, which is a very important problem in bioinformatics and medicine, promising to give a lot of answers especially in cancer research. We describe a method, called PolyA-iEP, that we developed for predicting polyadenylation sites and we present a systematic study of the problem of recognizing mRNA 3΄ ends which contain a polyadenylation site using the proposed method. PolyA-iEP is a modular system consisting of two main components that both contribute substantially to the descriptive and predictive potential of the system. In specific, PolyA-iEP exploits the advantages of emerging patterns, namely high understandability and discriminating power and the strength of a distance-based scoring method that we propose. The extracted emerging patterns may span across many elements around the polyadenylation site and can provide novel and interesting biological insights. The outputs of these two components are finally combined by a classifier in a highly effective framework, which in our setup reaches 93.7% of sensitivity and 88.2% of specificity. PolyA-iEP can be parameterized and used for both descriptive and predictive analysis. We have experimented with Arabidopsis thaliana sequences for evaluating our method and we have drawn important conclusions.

Keywords: data mining, machine learning, classification, emerging pattern, bioinformatics, polyadenylation

1 Introduction

During the last decades two main scientific areas, namely biology and computer science have been characterized by major advances that have attracted the interest of all humanity. The growth of World Wide Web and the completion of Human Genome Project are two representative examples that reflect the extent of the development of these two scientific areas. However, biology and computer science have not grown separately. The need of the collaboration between biologists and computer scientists has been grown year by year as the two areas have been progressing and new scientific questions have been arising. Bioinformatics is a novel research area that has emerged as a solution to the aforementioned need. It is a very promising field that aims to provide the means to analyze and explain the vast amounts of biological data, contributing thereby to the development of other related areas like medicine.

Two relative subfields of computer science strongly related to artificial intelligence, namely data mining and machine learning, have provided biologists, as well as experts from other areas, a powerful set of tools to analyze new data types in order to extract various types of knowledge efficiently and effectively. These tools combine powerful techniques of artificial intelligence, statistics, mathematics, and database technology. This fusion of technologies aims to overcome the obstacles and constraints posed by the traditional statistical methods. A lot of interesting applications of artificial intelligence in bioinformatics is presented in (Ezziane, 2006).

In this paper we deal with polyadenylation site (or poly(A) site) prediction. Poly(A) site prediction is a challenging problem and the last years has attracted the attention of the scientific community, because the successful cure of this problem promises to provide a lot of answers in various fields of medicine, like cancer research. In many organisms, such as in Arabidopsis thaliana, which is a plant model organism, there are not many highly conserved signals or patterns around the poly(A) site and consequently the recognition of the poly(A) site is not trivial. The discrimination of mRNA 3′ ends that contain a poly(A) site from intronic or 5′ UTR sequences without a poly(A) site seems to be very difficult (mainly with intronic sequences) and the performance of the up to now proposed approaches is moderate. On the other hand, mRNA 3′ ends can be easily discriminated from coding sequences. This variability in the difficulty of discrimination has motivated our work and guided us to an effort to study this problem and define an approach that can improve prediction accuracy. Nowadays, the research in this field is focused on discovering new patterns around poly(A) site and on predicting the poly(A) site accurately. The method we propose can be used for both, pattern discovery and accurate prediction.

The prediction of poly(A) sites can be divided into two sub-problems. The first sub-problem deals with the discrimination of the sequences that contain a poly(A) site from the ones that do not and the second deals with the prediction of the position of a poly(A) site inside a sequence. The advantage of this approach is double. Firstly, a large number of irrelevant sequences are filtered out before searching for the position of a poly(A) site inside a sequence increasing notably the prediction accuracy. Secondly, a more specific method for predicting the position of a poly(A) site inside a sequence that focuses only in sequences that contain a poly(A) site leading in better models can be used. This approach can provide an increased performance against a more general method that deals concurrently with the discrimination of sequences and the prediction of poly(A) sites inside a sequence. The first sub-problem of the approach described above has not been studied yet. In this paper we focus on this sub-problem.

Our contribution is an approach that combines the concept of emerging patterns (Dong & Li, 1999) and more specifically the interesting ones with a novel distance based scoring method. Our approach maintains the high interpretability of emerging patterns and offers a high prediction performance. The extracted emerging patterns may span across many elements around the polyadenylation site and can provide novel and interesting biological insights. Our method increases significantly the performance of poly(A) site prediction and reaches 93.7% of sensitivity and 88.2% of specificity. Moreover, The method we propose can be parameterized and re-trained in order to deal with poly(A) site prediction in any organism. Beyond the proposed method we draw important conclusions on the problem of discriminating mRNA 3´ ends with poly(A) sites from other sequences without a poly(A) site.

This paper is organized as follows. Section 2 provides the necessary background knowledge. Section 3 presents a concise review of the research area that is related to the problem dealt in this study. Section 4 provides some preliminary technical terminology and section 5 is dedicated to the detailed description of our approach. The results of the experiments that were conducted in order to evaluate our method are presented in section 6 and finally, the paper is concluded in section 7.

2 Background Knowledge

Two families of molecules are responsible for the structure and functioning of every living organism, as well as for the carriage of the genetic information. These are proteins and nucleic acids, which both are linear polymers of smaller molecules (monomers). The term “sequence” is used to refer to the order of monomers in a polymer. A sequence is represented as a string of different symbols, one for each monomer. There are twenty protein monomers called amino acids and five nucleic acid monomers called nucleotides. A nucleotide is characterized by the nitrogenous base it contains: adenine (A), cytosine (C), guanine (G), thymine (T), or uracil (U). The most common nucleic acids are *deoxyribonucleic acid* (*DNA*) and *ribonucleic acid* (*RNA*). DNA may contain a combination of A, C, G, and T. In RNA, U appears instead of T.

DNA contains the genetic instructions used in the development and functioning of all known living organisms and some viruses. The processes related with DNA are described by the central dogma of molecular biology, which deals with the detailed residue-by-residue transfer of sequential information (Figure 1). It states that information cannot be transferred back from protein to either protein or nucleic acid (Crick, 1970).

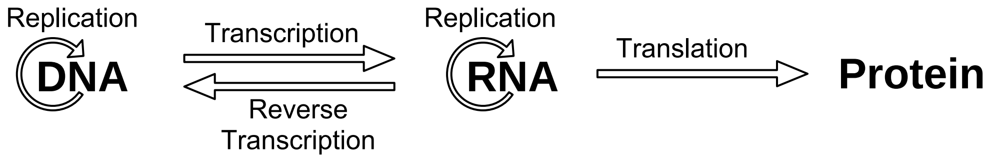


Figure 1: The Central Dogma of Molecular Biology

DNA replication, the basis for biological inheritance, is a fundamental process occurring in all living organisms to copy their DNA. *Transcription* is the process by which the information contained in a section of DNA is transferred to a newly assembled piece of *messenger RNA* (*mRNA*). In contrast, *reverse transcription* is the transfer of information from RNA to DNA (the reverse of normal transcription). This is known to occur in the case of retroviruses, such as HIV that causes acquired immunodeficiency syndrome (AIDS). *RNA replication* is the copying of one RNA to another. Many viruses replicate this way. Finally, *translation* is the production of proteins by decoding mRNA produced in transcription.

The process of *polyadenylation* occurs after transcription termination. It involves cleavage of the new transcript (mRNA), followed by template-independent addition of adenines at its newly synthesized 3′ end. The cleavage site is called *polyadenylation site* (*poly(A) site*). Polyadenylation is considered to be part of the larger process of producing mature mRNA for translation. The aim of the polyadenylation process is to protect the mRNA in order to reach intact the protein synthesis site.

The most important factors that are involved in the process of polyadenylation are the cis-regulatory elements and the trans-acting factors. The cis-regulatory elements are RNA sequences consisting of 2 to 10 nucleotides and their role is to help the trans-action factors define the poly(A) site. The most prominent cis-element is the hexamer AAUAAA or a close variant. This hexamer is located 10 – 35 nt upstream of the cleavage site (poly(A)-site) and it can be found in about 50% of human genes (Hu et al., 2005) but only in 10% of Arabidopsis genes (Loke et al., 2005). The trans-acting factors are a protein complex which also includes a specificity factor (Cleavage and Polyadenylation Specificity Factor - CPSF), an endonuclease, and Poly(A) Polymerase (PAP). The trans-acting factors are responsible for the cleavage at the appropriate site (poly(A) site) and the addition of the about 200 adenine residues (poly(A) tail) to the 3′ end (Lewin, 2004).

Nowadays, the research in this field is focused on discovering new cis-regulatory elements and on predicting the poly(A) site accurately. The accurate prediction of poly(A) site is a crucial step to define gene boundaries and get an insight in transcription termination in eukaryotes, which is a process less well understood.

3 Related Work

An early approach to the problem of poly(A) site prediction was the work of Salamov and Solovyev (1997) who developed a software called POLYAH and an algorithm for the identification of 3′-processing sites of human mRNA precursors. The algorithm was based on a linear discriminant function (LDF) trained to discriminate real poly(A) signals from the other regions of human genes possessing the AATAAA sequence which is most likely non functional. The accuracy of the method has been estimated on a set of 131 poly(A) regions and 1466 regions of human genes having the AATAAA sequence. When the threshold was set to predict 86% of poly(A) regions correctly, specificity of 51% and correlation coefficient of 0.62 had been achieved.

In 1999 Tabaska and Zhang developed polyadq, a program for detection of human polyadenylation signals. The program finds poly(A) signals using two discriminant functions: one specific for AATAAA type poly(A) sites and the other for ATTAAA type poly(A) sites. Polyadq predicts poly(A) signals with a correlation coefficient of 0.413 on whole genes and 0.512 in the last two exons of genes.

In 2000 Van Helden et al. approached the poly(A) site prediction problem with statistical methods. Other interesting approaches on this problem was the Hidden Markov Model approaches by Graber et al. (2002) and Hajarnavis et al. (2004).

In 2003 Liu et al. proposed a machine learning method to predict polyadenylation signals in human RNA sequences by analyzing features around them. The method consists of three steps: (1) Generating candidate features from the original sequence data using k-gram nucleotide patterns or amino acid patterns. (2) Selecting relevant features using an entropy-based algorithm. (3) Integrating the selected features by SVMs to build a system to recognize poly(A) sites.

Hu et al. (2005) developed a program named PROBE (Polyadenylation-Related Oligonucleotide Bidimensional Enrichment) to identify cis-elements that may play regulatory roles in mRNA polyadenylation. They found 15 cis-elements in the area of 100 nt upstream and downstream the poly(A) site. Another important conclusion of this work was that cis-elements occurring in yeast and plants also exist in human poly(A) regions. They suggested that many cis-elements are evolutionarily conserved among eukaryotes and human poly(A) sites have an additional set of cis elements that may be involved in the regulation of mRNA polyadenylation.

A year later Cheng et al. (2005) from the same lab tried to address whether those 15 cis-elements could be used to predict poly(A) sites. So they developed a program called Polya_svm which used support vector machines in order to predict poly(A) sites exploiting these 15 cis-elements. Polya_svm achieved higher sensitivity and similar specificity when compared with polyadq.

One of the most recent projects in the scientific area of polyadenylation site prediction was published in 2007 by Ji et al. Ji and his co-workers exploited the conclusions of a previous study (Loke et al., 2005) and developed a program named PASS (Poly(A) site sleuth) which used a Generalized Hidden Markov Model based algorithm in order to predict polyadenylation sites in Arabidopsis. Additionally, researchers from the same lab recently published a work in which they developed a program called Pass-Rice and predicts poly(A) sites in rice data (Shen et al., 2008).

Another approach to the poly(A) site prediction problem was made by Koh and Wong (2007). Their prediction model uses a machine learning approach which consists of four sequential steps: feature generation, feature selection, feature integration and a cascade support vector machine classifier.

The most recent publication that deals with the polyadenylation problem generally is the project of Ahmed and his co-workers (Ahmed et al., 2009). They developed a machine learning approach in order to predict polyadenylation signals in DNA sequences. More specifically they developed Support Vector Machines (SVM) models in order to predict polyadenylation signals in DNA sequences using 100 nucleotides, both upstream and downstream of this signal.

In a previous work of ours (Tzanis et al., 2008) we have presented a preliminary work dealing with poly(A) site prediction using the approach of interesting emerging patterns. The basic extensions over our past work, that are presented in this paper, include the use of wildchars in k-gram patterns, the incorporation of a distance-based scoring model, the use of a final classifier that combines the scores of the other components and a thorough experimentation. The improvement in effectiveness of our current approach is significant.

4 Preliminaries

This section provides the technical terminology that is necessary for understanding the details of our approach. The terms of frequent and emerging patters are defined, and the use of the last in classification is presented.

4.1 Frequent Itemsets

The term “frequent itemset” has been proposed in the framework of association rules mining. Association rules (Agrawal et al., 1993) have attracted the attention of the data mining research community since the early 90s, as a means of unsupervised, exploratory data analysis. The association rule mining paradigm involves searching for co-occurrences of items in transaction databases. Such a co-occurrence may imply a relationship among the items it associates. The task of mining association rules consists of two main steps. The first one includes the discovery of all the frequent itemsets contained in a transaction database. In the second step, the association rules are generated from the discovered frequent itemsets. A formal statement of the concept of frequent itemsets is presented in the following paragraph.

Let $I = \{i_1, i_2, \ldots, i_N\}$ be a finite set of binary attributes which are called *items* and D be a finite multiset of *transactions*, which is called *dataset*. Each transaction $T \in D$ is a set of items such that $T \subseteq I$. A set of items is usually called an *itemset*. The *length* or *size* of an itemset is the number of items it contains. It is said that a transaction $T \in D$ *contains* an itemset $X \subseteq I$, if $X \subseteq \mathrm{T}$. The *support* of itemset X is defined as the fraction of the transactions that contain itemset X over the total number of transactions in D:

$$supp_D(X) = \frac{\left|\{T \in D \mid T \supseteq X\}\right|}{|D|} \tag{1}$$

Given a minimum support threshold $\sigma \in (0,1]$, an itemset X is said to be *σ-frequent*, or simply *frequent* in D, if $supp_D(X) \geq \sigma$.

4.2 Emerging Patterns

Emerging patterns (Dong & Li, 1999) are itemsets whose supports increase significantly from one dataset to another.

Given two datasets D_1 and D_2, the *growth rate* of an itemset X from D_1 to D_2 is defined as (indices 1 and 2 are used instead of D_1 and D_2):

$$gr_{1\to 2}(X) = \begin{cases} 0, & \text{if } supp_1(X) = 0 \text{ and } supp_2(X) = 0 \\ \infty, & \text{if } supp_1(X) = 0 \text{ and } supp_2(X) > 0 \\ \dfrac{supp_2(X)}{supp_1(X)}, & \text{otherwise} \end{cases} \tag{2}$$

Given a minimum growth rate threshold $\rho > 1$, an itemset X is said to be *ρ-emerging pattern*, or simply *emerging pattern*, from D_1 to D_2, if $gr_{1\to 2}(X) \geq \rho$. D_1 is called *background dataset* and D_2 is called *target dataset*.

The *strength* of an emerging pattern X from D_1 to D_2 is defined as:

$$strength_{1\to 2}(X) = \begin{cases} supp_2(X), & \text{if } gr_{1\to 2}(X) = \infty \\ supp_2(X)\dfrac{gr_{1\to 2}(X)}{gr_{1\to 2}(X)+1}, & \text{otherwise} \end{cases} \tag{3}$$

Emerging patterns in contrast to other patterns or models are easily interpretable and understood. Moreover, emerging patterns, especially those with a large growth rate and strength, provide a great potential for discriminating examples of different classes. This twofold benefit of emerging patterns makes them a useful tool for exploring domains that are not well understood, providing the means for descriptive and predictive analysis as well.

4.3 Interesting Emerging Patterns

A disadvantage of emerging pattern mining is that the number of emerging patterns may be huge, especially when minimum support and minimum growth rate thresholds are set very low. Increasing the thresholds is not an ideal solution, since valuable emerging patterns may not be discovered. For example, if minimum support threshold is set high, then those emerging patterns with a low support, but with a high growth rate will be lost. Conversely, if minimum growth rate threshold is set high, then those emerging patterns with a low growth rate, but with a high support will be lost. There have been proposed some interestingness measures in order to reduce the number of mined emerging patterns without sacrificing valuable emerging patterns, or at least sacrificing as less as possible. Such an interestingness measure includes a special kind of emerging patterns, called *Chi Emerging Patterns* (Fan, 2004), which are defined as follows.

Given a background dataset D_1 and a target dataset D_2, an itemset X is called a chi emerging pattern, if all the following conditions are true:

1) $supp_2(X) \geq \sigma$, where σ is a minimum support threshold.
2) $gr_{1\to 2}(X) \geq \rho$, where ρ is a minimum growth rate threshold.
3) $\forall Y \subset X, gr_{1\to 2}(Y) < gr_{1\to 2}(X)$
4) $|X| = 1 \vee |X| > 1 \wedge (\forall Y \subset X \wedge |Y| = |X| \text{-} 1 \wedge chi(X,Y) \geq \eta)$, where η = 3.84 is a minimum chi value threshold and *chi*(X, Y) is computed using chi-squared test.

The first condition ensures that the mined emerging patterns will have at least a minimum coverage over the training dataset in order to generalize well on new instances. The second condition ensures that the mined emerging patterns will have an adequate discriminating power. The third condition is used in order to filter out those emerging patterns that have a subset with higher or equal growth rate and higher or equal support (any itemset has equal or greater support than any of its supersets). Since the subset has fewer items, there is not any reason to keep this emerging pattern. Finally, the fourth condition ensures that an emerging pattern has a significantly (95%) different support distribution in target and background datasets than the distributions of its immediate subsets.

4.4 Classification Using Emerging Patterns

Emerging patterns or interesting (e.g. chi) emerging patterns can be used in order to discriminate instances of different classes. Given two sets of instances D_+ and D_-, for example transactions that represent sequences with a poly(A) site (positive instances) and without a poly(A) site (negative sequences) respectively, two sets of emerging patterns E_+ and E_- can be mined. For mining E_+, D_- will be the background dataset and D_+ will be the target dataset. In contrast, for mining E_-, D_+ will be used as the background dataset and D_- as the target dataset. When a new instance (transaction) T has to be classified, the following scores are calculated:

$$\begin{aligned} score(T,+) &= \sum_{e \subseteq T, e \in E_+} strength_{-\to+}(e) \\ score(T,-) &= \sum_{e \subseteq T, e \in E_-} strength_{+\to-}(e) \end{aligned} \qquad (4)$$

The first score indicates if T is positive and the second if it is negative. The final decision could be made by comparing the values of the two scores and assigning the instance to the class with the higher score. However, due to the fact that the sizes of E_+ and E_- could be quite different, the scores have to be justified. In (Tzanis et al.; 2008) we have studied three alternative methods:

1) The first method was presented in (Dong et al., 1999). It calculates two base scores, $base_+$ and $base_-$ for positive and negative classes respectively. The $base_+$ score is found by calculating the positive score (using E_+) for each of the instances of the positive training set, and selecting the median of the scores to be $base_+$. Similarly *base*- is calculated, using negative training instances and E_- instead. The two scores that are calculated for a new instance are divided by the corresponding base scores and the instance is finally assigned to the class with the greatest justified score.
2) Another method (Tzanis et al., 2008) uses information entropy in order to select a threshold for the following fraction $\frac{score(T,+)}{score(T,-)}$. This fraction is calculated for all of the training (positive and negative) instances and a cut point, *entropy_threshold*, which maximizes information gain is found. A new instance is assigned to positive class if the above fraction exceeds *entropy_threshold*.
3) Finally, we have studied a combination of the above two score justification methods and

proposed another threshold for the fraction in 2. This threshold is defined as follows:

$$entropy\_base = \frac{entropy\_threshold + \frac{base_{+}}{base_{-}}}{2} \tag{5}$$

The justification method presented in 1 tends to favor the class with the smallest number of training instances, whereas the method in 2 tends to favor the class with the majority of training instances. For this reason we have proposed the score justification method in 3 that balances the previous two methods.

4.5 Classification Evaluation Metrics

The effectiveness of a classifier is evaluated according to some standard performance metrics. In this paper three metrics are used. *Sensitivity* or *TP Rate* measures the proportion of the correctly classified positive instances over the total number of positive instances:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

Specificity or *TN Rate* measures the proportion of the correctly classified negative instances over the total number of negative instances:

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{7}$$

TP (True Positives) are the positive instances classified as positives and FP (False Positives) are the negative instances classified as positives. Respectively, TN (True Negatives) are the negative instances classified as negatives and FN (False Negatives) are the positive instances classified as negatives. *Accuracy* measures the proportion of the correctly classified instances over the total number of instances. However, accuracy is skew sensitive. Thus, it may guide to misleading conclusions when the dataset is skewed (imbalanced), namely when one class has significantly more instances than the other. An alternative performance metric that is not skew sensitive is *adjusted accuracy.*

$$\text{Adjusted Accuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2} \tag{8}$$

The dataset we use in our setup is imbalanced, thus we use adjusted accuracy instead of accuracy.

5 Our Approach

In this paragraph we describe the method (PolyA-iEP) we have developed for dealing with the problem of polyadenylation site prediction in Arabidopsis thaliana mRNA sequences. Although in this study we have concentrated on a plant that poses great challenges due to low conservation of poly(A) signals, the method we propose can be re-trained and parameterized for studying different organisms. PolyA-iEP has been implemented in JAVA and consists of a number of steps that are shown in Figure 2 and presented in detail below.

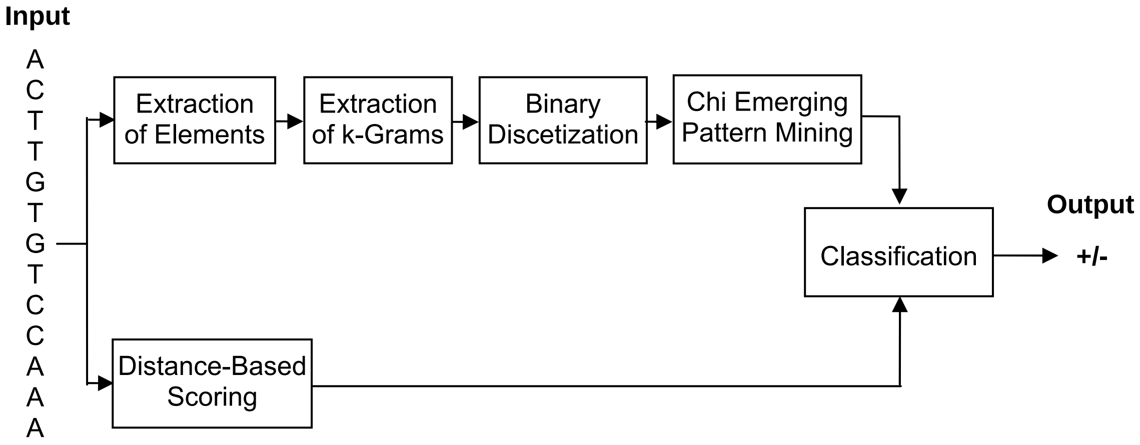


Figure 2: The architecture of PolyA-iEP

5.1 Extraction of Elements

There is a number of different elements around the poly(A) site of an mRNA 3΄ end that have been recognized in previous studies (Cheng et al., 2006). These elements are composed by different nucleotide frequencies and consequently may contain fairly different patterns. This indicates that one has to search for patterns separately in each element. However, a promising idea is to study the associations among the patterns of the different elements in order to discover possible relationships among them. This could lead to new “extended” patterns that are possibly more informative and have higher discriminating power than the single patterns found in each element separately. In our study we deal with this kind of “extended” patterns. The three basic elements located around Arabidopsis 3΄ end poly(A) sites have been proposed in previous studies (see for example Loke et al., 2005) and include the Far Upstream Element (FUE), the Near Upstream Element (NUE) and the Cleavage Element (CE). The downstream region of Arabidopsis poly(A) sites is not considered particularly important, however we have included a Near Downstream Element (NDE) in our study, for the shake of completeness. Figure 3 summarizes the elements that are used in our approach.

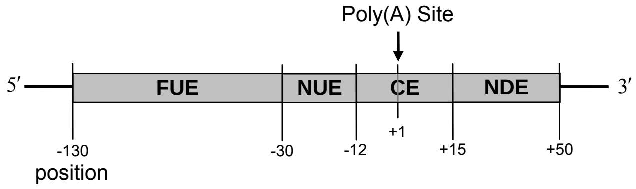


Figure 3: A model for poly(A) signals in Arabidopsis mRNA 3′ ends.

5.2 Extraction of k-Grams

Each of the sequence elements that are extracted at the first step will be represented by a vector that contains the frequencies of nucleotide patterns (*k*-grams). These patterns include all nucleotide combinations of length *k*, where in our setup $k \in \{1, 2, \ldots, 6\}$. Moreover, a number of patterns including wildcard characters (wildchars) have been utilized as an extension to original *k*-gram patterns. The length of these patterns is also of length *k*. The alphabet for generating every pattern is presented in Table 1. The last six rows of the table represent all possible wildchars that are used in our approach. For example, "AWT" is a valid pattern, which represents either AAT or ATT. So, each initial sequence after this step will be represented by a number of vectors (each element of these vectors corresponds to the frequency of one valid pattern), one for each of the specified elements (i.e. FUE, NUE, CE, NDE). The user of PolyA-iEP can specify a different *k*.

Table 1: generating patterns (based on IUPAC notions)

Alphabet Letter	**Nucleotides**	
A	A	**A**denine
C	C	**C**ytocine
G	G	**G**uanine
T	T	**T**hymine
R	A or G	pu**R**ine
Y	C or T	p**Y**rimidine
M	A or C	a**M**ino
K	G or T	**K**eto
S	C or G	**S**trong (3 H bonds)
W	A or T	**W**eak (2 H bonds)

5.3 Binary Discretization

The discretization method used in our approach is based on information entropy. For each *k*-gram pattern a cut point is sought among all pattern frequencies and the one that has the maximum information gain is finally selected. Given a set of training examples *S*, *entropy* (*E*) is defined by the following equation:

$$E(S) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{9}$$

where *c* is the number of classes and p_i is the proportion of examples in *S* that belong in class *i*. By definition, if p_i is zero, then the term $p_i \log_2(p_i)$ is set to zero.

Given an ordered set of candidate *N* cut points $T=\{t_1,\ldots,t_N\}$ for the values of an attribute *A*, that partition the set of examples in *N*+1 subsets ($S_1,\ldots,S_{N+1}$), the *information gain* (*G*) is defined by the following equation:

$$G(S; A, T) = E(S) - \sum_{i=1}^{N+1} \frac{|S_i|}{|S|} E(S_i) \tag{10}$$

where $S_i = \{s \in S_i \mid s[A] \in [t_i, t_{i+1})\}$.

In our approach we use binary discretization, so there is only one cut point. This cut point is sought among all attribute values and the one that has the maximum information gain is finally selected. The *k*-gram vectors that were previously constructed are transformed into a transaction of items. The items of the transaction are those *k*-grams that have frequency greater than the corresponding cut point, which was previously calculated. In this step the data have been transformed in a format that permits the extraction of emerging patterns.

5.4 Mining Interesting Emerging Patterns

The transactional data that have been produced in the previous step can be mined for interesting emerging patterns. For this reason we have extended the FP-Growth algorithm (Han et al., 2000) that is used for mining frequent itemsets. The extended algorithm receives as input two datasets, the background and the target dataset, and discovers all chi emerging patterns, based on the parameters specified by the user (minimum support threshold and minimum growth rate threshold). At this point it is worthwhile to repeat that the patterns that are mined by PolyA-iEP are "extended", since these patterns can include itemsets of different elements. At this step two sets of emerging patterns E_+ and E_-, are generated for the positive and the negative class respectively. In our setup we have used a dataset that contains 3 types of negative sequences (5' UTR, coding, and intronic), that present quite different nucleotide distributions. If all negatives are dealt as a whole, then the effectiveness of classification is moderate. For this reason, we have mined 4 pairs of $\mathrm{E}_+/\mathrm{E}_-$ sets of emerging patterns, one for the discrimination of positives from all negatives and three for discriminating positives from each type of negatives separately.

An example of an "extended" interesting emerging pattern, than can be mined by PolyA-iEP is the following:

{FUE_TGGA, NUE_CT, NDE_CYG}: 0.32

The above interesting emerging pattern associates the occurrence of pattern "TGGA" in the Far Upstream Element, with pattern "CT" in the Near Upstream Element, and with pattern "CYG" in the Near Downstream Element. The strength of this interesting emerging pattern is 0.32. Note the use of "Y" wildchar in the third pattern.

Scoring Using Interesting Emerging Patterns

As already described in preliminaries, emerging patterns can be used in order to discriminate instances of different classes. At this step the E_+/E_- pairs of sets of emerging patterns that were previously generated, are used for scoring an instance as being positive or negative. For this reason, pairs of scores are calculated as described by equations (4). The total number of scores that are produced in this step is 8. Two scores (one for positive and one for negative class) are assigned to each of the following discriminations: positives/all negatives, positives/5΄ UTR negatives, positives/coding negatives, and positives/intronic negatives.

5.5 Distance-Based Scoring

Distance-based scoring is independent from the previous steps. At this step the frequencies of nucleotides at each position of a sequence are calculated and a nucleotide frequency matrix is constructed for each class, as shown in Table 2. For example, nucleotide A has 0.15 frequency is position1 of the sequences used to generate the matrix presented in Table 2. Then, for each position in the sequence the rankings of the nucleotides are calculated according to their frequency at this particular position (Table 3). In our setup five nucleotide frequency ranking matrices are constructed, one for each of the following categories: positives, all negatives, 5΄ UTR negatives, coding negatives, and intronic negatives.

Table 2: An example of a nucleotide frequency matrix for sequences of length 5

nucleotide	position in sequence				
	1	2	3	4	5
A	0.15	0.08	0.28	0.10	0.10
C	0.20	0.22	0.22	0.30	0.30
G	0.40	0.33	0.25	0.40	0.30
T	0.25	0.37	0.25	0.20	0.30

Table 3: The nucleotide frequency ranking matrix that corresponds to Table 1 data

nucleotide	position in sequence				
	1	2	3	4	5
A	4	4	1	4	4
C	3	3	4	2	2
G	1	2	2.5	1	2
T	2	1	2.5	3	2

In order to calculate the distance of a sequence from a class or subclass (5΄ UTR, intronic, or coding), first, the sequence is converted into a nucleotide frequency ranking vector using the nucleotide frequency matrix of the class or subclass. Then, the distance from the unary vector is calculated and divided by the length of sequence. For example, given the ranking matrix in Table 3, the ranking vector that corresponds to the sequence "ATGGC" is <4, 1, 2.5, 1, 2>. The distance (Manhattan distance is used in our setup) of this vector from the unary vector <1, 1, 1, 1, 1> is 5.5. Dividing this distance by the length of the sequence, namely 5, the mean nucleotide distance is finally calculated to be 1.1. This is the mean nucleotide distance of the above sequence from the category to which the nucleotide frequency matrix in Table 3 belongs.

5.6 Classification

The scores calculated in previous steps are used as input to a classifier that makes the final decision and classifies the entered sequence either as positive or as negative (i.e. containing or not a poly(A) site). In particular, a total number of 13 scores are used as input to the classifier, including the eight scores calculated at the emerging patterns mining step and the five distance based scores.

Any classification algorithm that can deal with real-valued numeric attributes and binary class attributes can be utilized for building the classifier. Our study has been focused on a number of state-of-the-art classification algorithms including (neural networks, support vector machines, classification trees, and instance-based learning).

6 Experiments

In this section we describe the datasets we have used as well as the experiments we have conducted in order to evaluate our method.

6.1 Datasets

We have used four sets of Arabidopsis thaliana sequences. One of them contains 6209 positive examples, namely mRNA 3′ end sequences that contain a poly(A) site, whereas the other three contain negative examples (864 5′ UTR, 1501 coding, and 1581 intronic sequences). These data have been used in previous studies (Ji et al., 2007; Koh & Wong, 2007; Tzanis et al., 2008). The set of positive sequences will be called positive dataset and the set of all negative sequences will be called negative dataset. All sequences have a length of 400 nt. Each positive sequence has an EST-supported poly(A) site at position 301. The positive sequences underwent pair-wise global alignment against every other sequence (Koh & Wong, 2007) in order to reduce similarity among all sequences. Particularly, there are not any two sequences in the positive dataset that have more than 70% similarity. This was done for minimizing biasness due to similarity of sequences. More details about these datasets can be found in (Ji et al., 2007).

Table 4 describes the datasets used in our setup. For the purposes of experimentation we have divided the initial set of sequences into a number of datasets based on two criteria. The first criterion, which has been already mentioned above, is the intrinsic characteristics of the sequences (e.g. negative sequences are coding, 5′ UTR, or intronic sequences). This split of the data is represented by different rows in Table 4.

Table 4: Datasets

		All Sequences	**EP Mining**	**Training**	**Test**
Positive Sequences	EST Supported	**6209**	**2794**	**2173**	**1242**
Negative Sequences	5′ UTR	864	389	302	173
	Coding	1501	676	525	300
	Intronic	1581	712	553	316
	Total Negative	**3946**	**1777**	**1380**	**789**
Total Sequences (Positive + Negative)		10155	4571	3553	2031

The second criterion is the procedure that should be used in order to build and evaluate the proposed method. For this reason the sequences have also been randomly divided into three parts, one for mining interesting emerging patterns (EP Mining), one for training the classifier

(Training) and one for evaluation (Test). The percentage of sequences contained in each of the three parts was decided in a way that provides an adequate number of data for extracting emerging patterns and training the classifier. This split of the data is represented by different columns in Table 4.

6.2 Exploratory Analysis

In an effort to further investigate why the number of chi emerging patterns differs so much among the three negative sub-classes we plotted the distributions of nucleotides for the positive and each negative dataset. Figure 4 presents the distribution of each nucleotide from positions -200 to +100 with respect to the poly(A) site in Arabidopsis mRNA 3′ ends. The differences in nucleotide distributions among different elements are very clear. Figures 5, 6, and 7 depict the nucleotide distributions of intronic, 5′ UTR, and coding Arabidopsis sequences. These figures clearly depict why the discrimination between positive sequences and negative intronic sequences is very difficult, whereas the discrimination between positive and negative coding sequences is easy. Comparing figures 4 and 5, we can see that there are many similarities in nucleotide distributions of mRNA 3′ end sequences and introns. In intronic sequences, uracil is the most frequent nucleotide, followed by adenine, then guanine, and finally cytosine. This is also the case with the upstream region up to the NUE of the mRNA 3′ end. In contrast, the nucleotide distribution of coding sequences is very different than the one of mRNA 3′ ends. Finally, 5′ UTR sequences have also similar nucleotide distribution with this of introns, but they differ from mRNA 3′ ends more than introns do. That is the reason, why 5′ UTRs can be discriminated easier from mRNA 3′ ends.

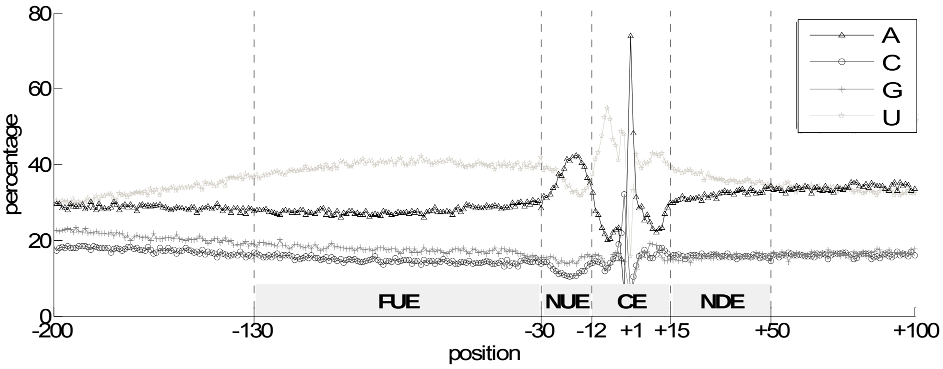


Figure 4: Nucleotide distribution (position -200, +100 around poly(A) site) in mRNA 3′ end

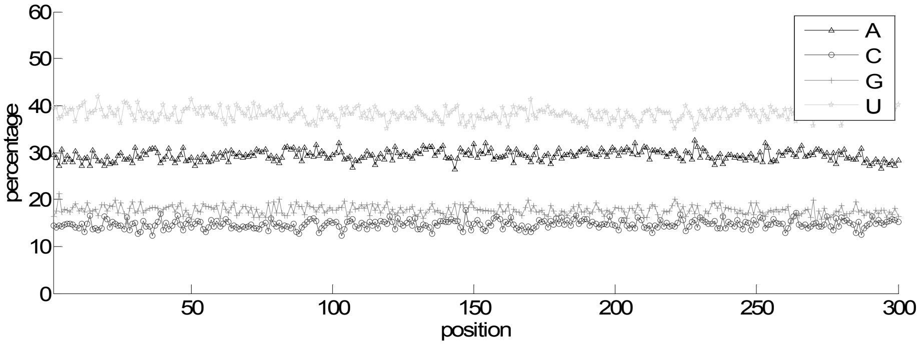


Figure 5: Nucleotide distribution in intronic sequences

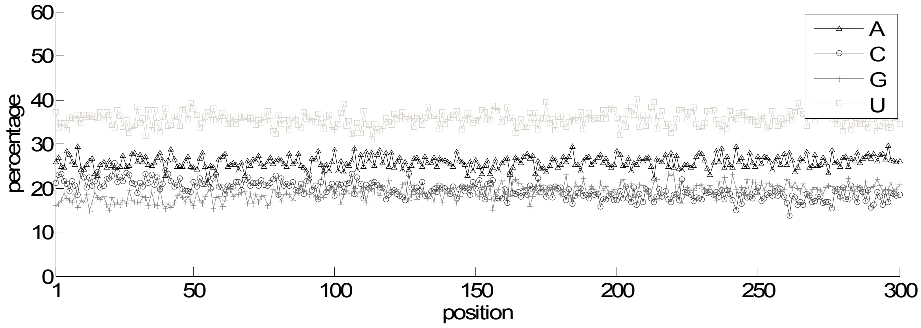


Figure 6: Nucleotide distribution in 5΄ UTR sequences

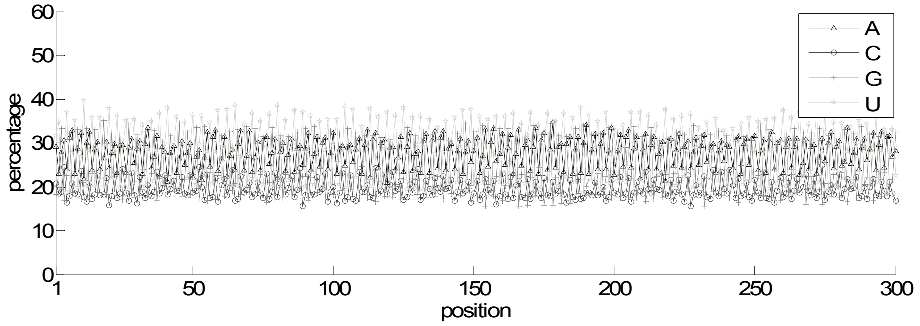
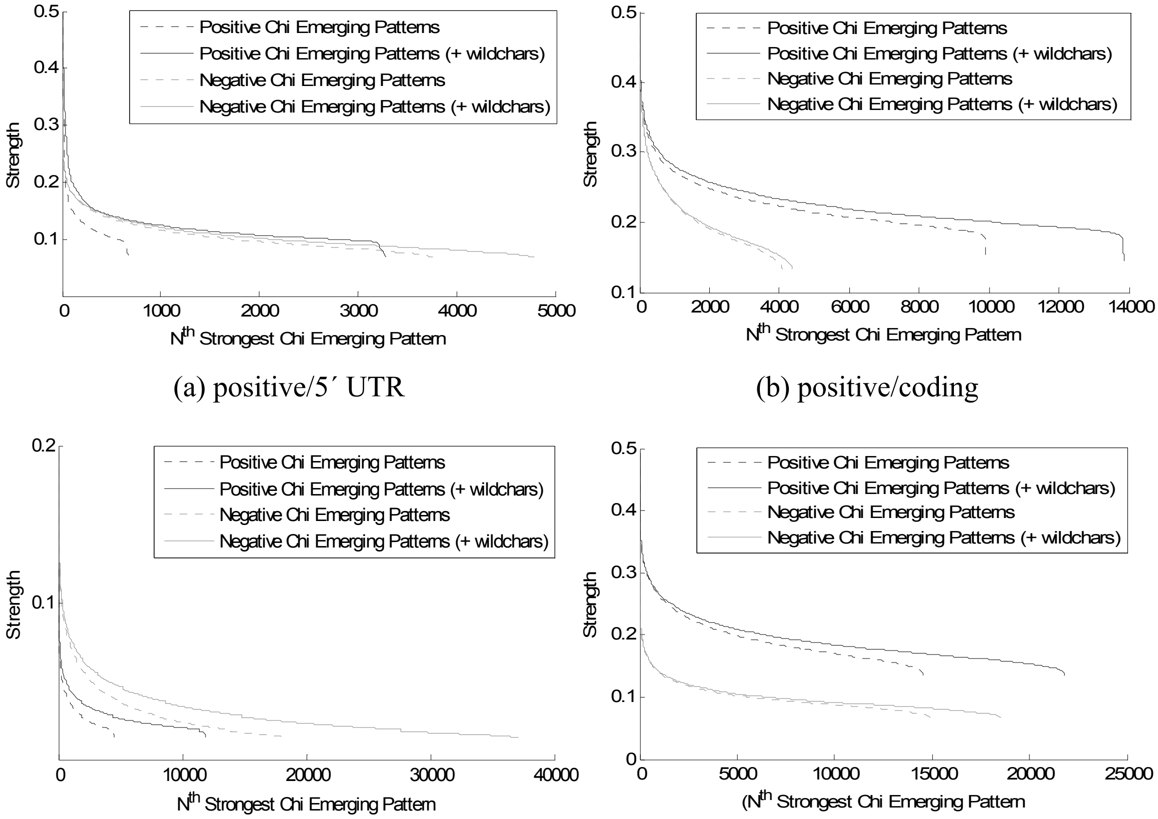


Figure 7: Nucleotide distribution in coding sequences

6.3 Contribution of Wildchars

In order to evaluate the contribution of *k*-gram patterns that include wildchars, we have compared our approach, which includes patterns with wildchars, against a baseline approach that includes only the typical (without wildchars) *k*-gram patterns. The strongest chi emerging patterns for both positives and negatives (E_+ and E_-) of each of the two approaches have been plotted according to their strength that was calculated as in equation (3). The plots of the various discriminations: Positive/5΄ UTR, positive/coding, positive/intronic, and positive/all negative are presented in Figure 8.

As shown in all plots, our method, that includes patterns with wildchars, provides stronger chi emerging patterns, when considering the first *N* patterns. This means, that if we take under consideration the *N* strongest chi emerging patterns, then our method, will provide a set of stronger chi emerging patterns. For this reason the incorporation of *k*-grams with wildchars in our approach, improves the quality of the mined chi-emerging patterns. Another interesting observation that confirms the results of our exploratory analysis presented above is that the strongest chi emerging patterns are mined for the positives/coding, whereas the less strong are mined for positives/intronic negatives.



(a) positive/5′ UTR

(b) positive/coding

(c) positive/intronic

(d) positive/all negative

Figure 8: Contribution of wildchars

6.4 Evaluation of Chi Emerging Patterns

In order to evaluate the classification performance of the sets of chi-emerging patterns we have conducted a number of experiments using various numbers of the strongest chi emerging patterns. For classifying the test instances equations (4) and (5) for calculating the scores and the threshold were used. The results are presented in Figure 9. As it is observed the accuracy increases with the number of the included chi-emerging patterns. However, the increase in accuracy is small after a critical number of the top strongest chi-emerging patterns (around 500) have been included. Once again, the positive/intronic discrimination problem appears as the more difficult, whereas the positive/coding discrimination problem appears as the easiest one. Mention that the positive/all negative discrimination has also a low performance.

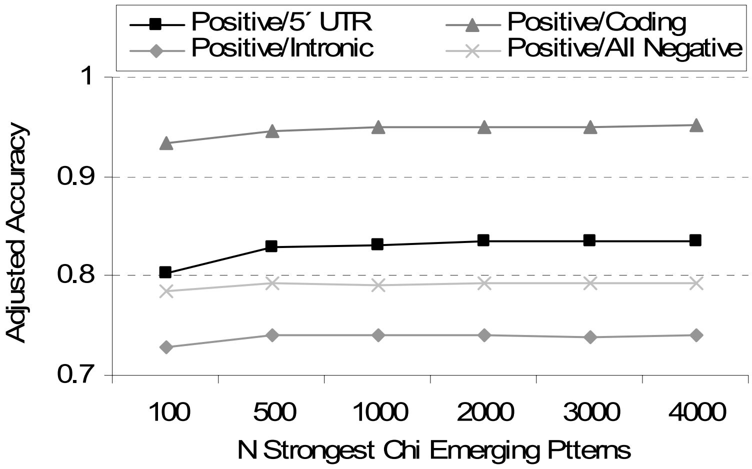


Figure 9: Emerging pattern classification performance

6.5 Evaluation of Distance-Based Scoring

Figure 10 presents the mean distances of positive class with all classes and sub-classes. In particular five nucleotide frequency ranking matrices were generated using five different training datasets (positive, 5′ UTR, coding, intronic, and all negative). Then, using the positive test dataset, the mean distances of test positives were calculated.

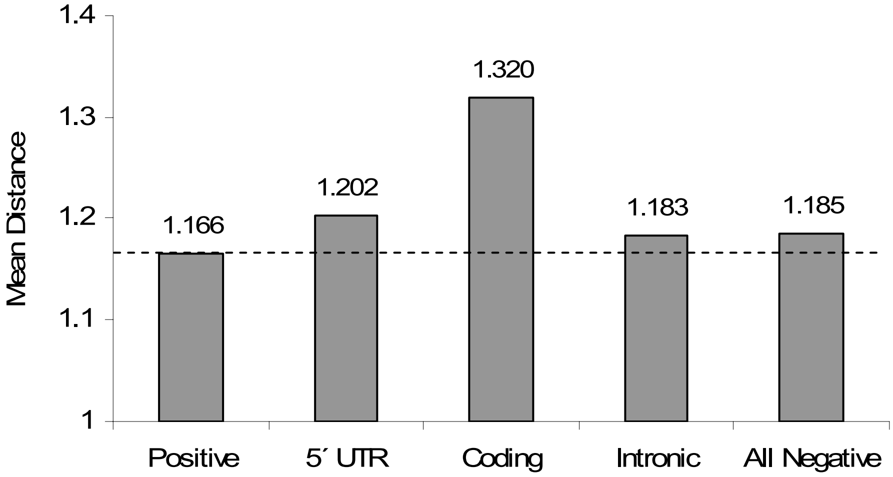


Figure 10: Mean distances of all classes (and sub-classes) from positive class

As shown in Figure 10 the class that is nearest to positives is positive. The next nearest sub-class is the intronic, whereas the more distant sub-class is the coding. These results, also, confirm our exploratory analysis. An important observation is that the distance of positives to all negatives is almost equal to the distance between positives and intronic negatives. This indicates that the discrimination between positives and all negatives is almost as difficult as the discrimination between positives and introns. This observation strengthens the thought of considering the intrinsic characteristics of each negative subclass in order to discriminate more effectively positive from negative sequences.

6.6 Evaluation of Classifier

The evaluation of the entire approach we propose appears at this section. The classifier that is built incorporates all the scores produced in the other steps of our approach and provides the final decision. We have experimented with the following classification algorithms, implemented in the Weka machine learning library (Witten & Frank, 2005):

- Neural Network without any hidden layers (NN-0). A classifier that uses backpropagation to classify instances.
- Neural Network with one hidden layer (NN-1). The same algorithm as the one for building NN-0 was used.
- Support Vector Machine (SMO-1) using a linear polynomial kernel. This is a sequential minimal optimization algorithm for training a support vector classifier (Platt, 1998). It belongs to the family of generalized linear classifiers.
- Support Vector Machine (SMO-2) using a quadratic polynomial kernel. This is also, Platt's algorithm.
- Logistic Model Tree (LMT). Classifier for building classification trees with logistic regression functions at the leaves (Landwehr et al., 2005; Sumner et al., 2005).
- C4.5. A decision tree construction algorithm (Quinlan, 1993). The classifiers were built using reduced-error pruning instead of C.4.5 pruning for improving effectiveness.
- *k*-Nearest Neighbors (*k*-NN). An instance-based classification algorithm (Aha & Kibler, 1991). The appropriate value of *k* was selected using cross-validation.

In order to evaluate the importance of combination of all components of our approach we have compared our approach with two methods. The first one does not include the distance-based scoring. Only the scores of the emerging patterns mining component are fed into the classifier. The second method does not include the chi emerging patterns scoring component, thus only the distance based scores are fed into the classifier. Finally, in order to evaluate the overall performance of our approach we have compared it with a baseline method that includes the use of a large number of features which represent the frequency of each k-gram in an instance (sequence). In particular, each instance is represented by a vector that contains the frequencies of 5460 k-gram patterns (k varies from 1 to 6). The performance of the last method using all the 5460 features for classification was very bad (no classifier with an adjusted accuracy over 0.75 was reported). For this reason, a step of feature selection was used in order to remove the irrelevant features and increase this method's performance. The correlation-based feature subset selection method (Hall, 1999) was used and 121 features were finally selected. This method finds a subset of features that is highly correlated with the class while having low intercorrelation.

Figure 11 presents the results of all methods for all classifiers in terms of adjusted accuracy. These results concern the general discrimination problem between positive/all negative. The classifiers were trained using the Training dataset and were evaluated using the Test dataset (see Table 4). As Figure 11 shows, our complete approach significantly outperforms any baseline method, with the neural network with one hidden layer being the most accurate classifier, achieving an adjusted accuracy of 0.91.

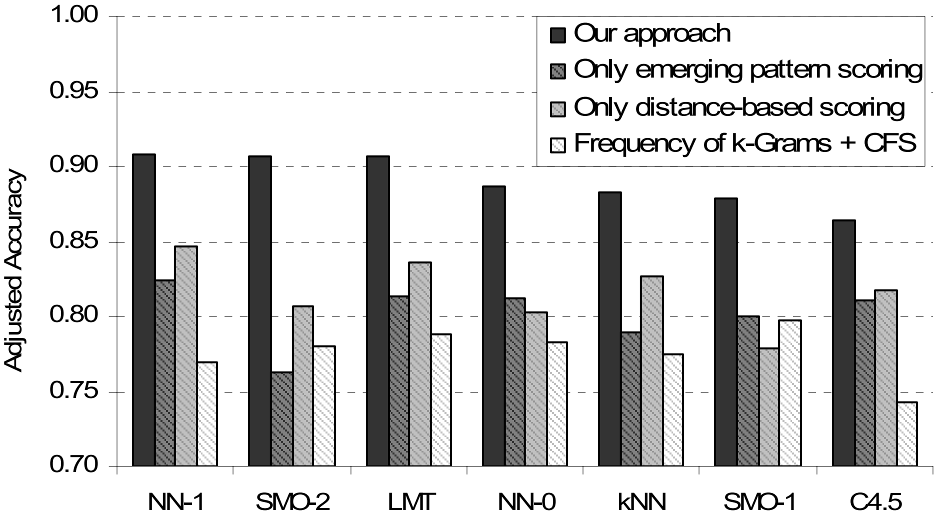


Figure 11: Performance of various classifiers for positives/all negatives discrimination

Table 5 presents the achieved sensitivity, specificity and adjusted accuracy of our method for each classifier. Another baseline method that can be used to compare our approach is the chi emerging pattern classifier. This is different from the first baseline presented above, because the scores of the chi emerging patterns are not fed into a classifier, but equations (4) and (5) for calculating scores and a threshold are used to classify the test instances. The performance of this classifier for the general discrimination problem between positive/all negative does not exceed 0.80 in terms of adjusted accuracy (see Figure 9) and in any case is fairly worst than our approach's performance.

Table 5: Performance of our approach for positives/all negatives discrimination using various classifiers

Classifier	Sensitivity	Specificity	Adjusted Accuracy
NN-1	0.937	0.882	0.910
SMO-2	0.931	0.883	0.907
LMT	0.931	0.881	0.906
NN-0	0.917	0.857	0.887
k-NN	0.919	0.847	0.883
SMO-1	0.917	0.840	0.879
C4.5	0.905	0.823	0.864

Table 6 presents a statistical comparison (95% confidence level) of all classifiers for the positive/all negative discrimination problem using our approach. The pair-wise comparison of the different schemas was based on a 10-fold cross validation run using T-Test. A plus (+) symbol in a cell of the table indicates a 95% statistical superiority of the classifier that corresponds to the row of the cell over the classifier that corresponds to the column of the cell. As it is observed three of the classifiers, NN-1, SMO-2, and LMT are all superior over the remaining algorithms (NN-0, k-NN, SMO-1, and C4.5), but none of them is superior over any of the other two. This means that these three algorithms (NN-1, SMO-2, and LMT) perform equally in comparison to each other, but better than any of the rest classifiers.

Table 6: Statistical comparison of various classifiers for positives/all negatives discrimination using our approach

	NN-1	**SMO-2**	**LMT**	**NN-0**	**k-NN**	**SMO-1**	**C4.5**
NN-1		0	0	+	+	+	+
SMO-2	0		0	+	+	+	+
LMT	0	0		+	+	+	+
NN-0	-	-	-		0	0	+
k-NN	-	-	-	0		0	+
SMO-1	-	-	-	0	0		0
C4.5	-	-	-	-	-	0	

What our method actually achieves is to increase impressively the low specificity that appears in other approaches as well as to provide a non-negligible increase in sensitivity. This is achieved because of the combination of the two basic components, the chi emerging pattern mining and the distance-based scoring. Examining carefully Figure 11, we can observe that the chi emerging pattern component is better than the distance-based component when linear classifiers are used (NN-0 and SMO-1). The reverse is observed when the non-linear classifiers are used (NN-1, SMO-2, LMT, k-NN, and C4.5). This implies that the distance-based component, that we proposed, encapsulates a non-linear dimension of the problem we deal with in this paper. The non-linearity of the problem is not represented in any way by other methods, thus the performance of these methods is moderate.

7 Conclusion

Polyadenylation site prediction is a challenging problem that attracts the interests of many researchers in the areas of medicine, biology, and bioinformatics. Nowadays, the research in this field is focused on discovering new patterns and on predicting the poly(A) site accurately. The approach we have proposed deals with these both dimensions of the problem. The difficulties on poly(A) site prediction are basically derived by the absence of highly conserved signals around the poly(A) site. In August 2009 Mayr and Bartel published their work on a study of normal and cancerous cells. Their results showed a strong correlation between 3´ UTR length and the expression of oncogenes. The important aspect is that the 3´ UTR length is determined by the position of the poly(A) site along the sequence. So it is obvious that polyadenylation is a key element in the understanding of biological processes and diseases like cancer and as a result it is going to be one of the most interesting topics in the field of bioinformatics.

In this work we studied the problem of poly(A) site prediction and proposed a method (PolyA-iEP) that can be used for both descriptive and predictive analysis. PolyA-iEP exploits emerging patterns as well as a distance-based scoring method and eventually provides a significant increase in effectiveness, which in our setup reaches 93.7% of sensitivity and 88.2% of specificity. An important benefit of our approach is that it is general, thus can be re-trained and parameterized for use with other sequences possibly from different organisms. In the future we are considering studying the use of more sophisticated classification methods like classifier ensembles in order to increase even more the effectiveness of our approach. Also, our future plans include the experimentation with mRNA sequences of other organisms.

The datasets we used and the tool we developed are available at http://mlkd.csd.auth.gr/PolyA/index.html.

References

R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases”. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1993, pp. 207-216.

D. Aha, D. Kibler. Instance-based learning algorithms. Machine Learning. 6:37-66, 1991.

F. Ahmed, M. Kumar, and G.P.S. Raghava. Prediction of polyadenylation signals in human DNA sequences using nucleotide frequencies, *In Silico Biology*, 2009, 9, pp. 135-148.

Y. Cheng, R.M. Miura, and B. Tian, “Prediction of mRNA polyadenylation sites by support vector machine”. In *Bioinformatics* 2006, 22(19), pp. 2320-2325.

F. Crick. Central Dogma of Molecular Biology. *Nature*, 1970, 227, pp. 561-563.

G. Dong and J. Li, “Efficient mining of emerging patterns: Discovering trends and differences”. In *Proceedings of ACM-SIGKDD’99*, 1999, pp. 43–52.

G. Dong, X. Zhang, L. Wong, and J. Li, “CAEP: Classification by aggregating emerging patterns”. *In Proceedings the 2nd International Conference on Discovery Science*, 1999, pp. 30–42.

Z. Ezziane, Applications of artificial intelligence in bioinformatics: A review, *Expert Systems with Applications* 30 (1), 2006, pp. 2–10.

H. Fan. *Efficient Mining of Interesting Emerging Patterns and Their Effective Use in Classification*, PhD Thesis, University of Melbourne, Australia, 2004.

J.H. Graber, G.D. McAllister, and T.F. Smith, “Probabilistic prediction of Saccharomyces cerevisiae mRNA 3'-processing sites”. In *Nucleic Acids Research* 2002, 30(8), pp. 1851-1858.

M. A. Hall. Correlation-based Feature Subset Selection for Machine Learning. *PhD Thesis*, University of Waikato, Hamilton, New Zealand, 1999.

A. Hajarnavis, I. Korf, and R. Durbin, “A probabilistic model of 30 end formation in Caenorhabditis elegans”. In *Nucleic Acids Research* 2004, 32, pp. 3392–3399.

J. Van Helden, M del Olmo, and J.E. Perez-Ortin, “Statistical analysis of yeast genomic downsream sequences reveals putative polyadenylation signals”. In *Nucleic Acids Research*, 28, 1000-1010.

J. Han, J. Pei, and Y. Yin. “Mining frequent patterns without candidate generation”. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 2000, pp. 1-12.

J Hu, C.S. Lutz, J. Wilusz, and B. Tian, “Bioinformatic identification ofcandidate cis-regulatory elements involved in human mRNA polyadenylation”. In *RNA* 2005, 11(10) pp. 1485-1493.

G. Ji, J. Zheng, Y. Shen, X. Wu, R. Jiang, Y. Lin, J. Loke, K, Davis, G. Reese, and Q. Li, “Predictive modeling of plant messenger RNA polyadenylation sites”. In *BMC Bioinformatics*, 2007, 8:43.

C.H. Koh, and L. Wong. ‘Recognition of polyadenylation sites from Arabidopsis genomic sequenses”. In *Proceedings of 18th International Conference on Genome Informatics*, pp. 73-82, 2007.

N. Landwehr, M. Hall, E. Frank. Logistic Model Trees. Machine Learning. 95(1-2):161-205, 2005.

B. Lewin, *Genes VIII*. Pearson Education Inc. 2004 pp. 721-722.

H. Liu, H. Han, J. Li, and L. Wong, “An in-silico method for prediction of polyadenylation signals in human sequences”. In *Genome Inform Ser Workshop Genome Inform* 2003, 14, pp. 84-93.

J. Loke, E.A. Stahlberg, D.G. Strenski, B.J. Haas, P.C. Wood, and Q.Q. Li, “Compilation of mRNA polyadenylation signals in Arabidopsis revealed a new signal element and potential secondary structures. In *Plant Physiol*ogy 2005, 138, pp. 1457-1468.

C. Mayr, D.P. Bartel. Widespread shortening of 3'UTRs by alternative cleavage and polyadenylation activates oncogenes in cancer cells. *Cell*, 138, pp. 673–684, 2009.

J. Platt. Machines using Sequential Minimal Optimization. In B. Schoelkopf and C. Burges and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, 1998.

R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

A. Salamov and V. Solovyev, “Recognition of 30-processing sites of human mRNA precursors”. In *Comput. Appl. Biosci*. 1997, 13, 23-28.

Y. Shen, G. Ji, B.J. Haas, X. Wu, J. Zheng, G.J. Reese, and Q.Q. Li, “Genome level analysis of rice mRNA 3’-end processing signals and alternative polyadenylation”. In *Nucleic Acids Research*, 36(9), pp. 3150-3161.

M. Sumner, E. Frank, M. Hall. Speeding up Logistic Model Tree Induction. In: 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, 675-683, 2005.

J.E. Tabaska and M.Q. Zhang, “Detection of polyadenylation signals in human DNA sequences”. In *Gene* 1999, 231, pp. 77–86.

G. Tzanis, I. Kavakiotis, and I. Vlahavas, Polyadenylation Site Prediction Using Interesting Emerging Patterns, In *Proceedings of the 8th IEEE International Conference on Bioinformatics and Bioengineering*, IEEE, Athens, Greece, 2008.

I. H. Witten and E. Frank. Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, 2005.