Polyadenylation site prediction using PolyA-iEP method

Ioannis Kavakiotis, George Tzanis, Ioannis Vlahavas

Affiliations: Department of Informatics, Aristotle University of Thessaloniki, Greece

Corresponding Author: Ioannis Kavakiotis

Email: ikavak@csd.auth.gr

Running Head: PolyA – iEP Method

Summary

This chapter presents a method called PolyA-iEP that has been developed for the prediction of polyadenylation sites. More precisely PolyA-iEP is a method that recognizes mRNA 3'ends which contain polyadenylation sites. It is a modular system which consists of two main components. The first exploits the advantages of emerging patters and the second is a distance-based scoring method. The outputs of the two components are finally combined by a classifier. The final results reach very high scores of sensitivity and specificity.

Key Words: Data Mining, Machine Learning, Classification, Emerging Patterns, Bioinformatics, Polyadenylation.

1. Introduction

PolyA-iEP [1] is a method that addresses the problem of discriminating sequences that contain polyadenylation sites from the ones that do not. The discrimination of mRNA 3' ends that contain polyadenylation sites from intronic or 5' UTR sequences without polyadenylation site seems to be very difficult and the performance of the existing methods is moderate. PolyA-iEP has emerged from a previously published method called PolyA-EP [2], although the new one is more robust and sophisticated.

PolyA-iEP is a modular system which consists of two main components. The first component exploits the concept and the advantages of the *frequent itemsets*. The term frequent itemset has been proposed in the framework of association rule mining. *Association rule mining* is a popular field of data mining which has been proposed by Ragesh Agrawal [3]. It is a research field that aims to discover interesting relations between variables in large databases. This field was initially introduced in the framework of market basket analysis but currently it is applied in many application areas including bioinformatics with outstanding results.

More precisely, the first component uses the concept of *interesting emerging patterns* [4]. *Emerging patterns* are defined as itemsets whose supports increase significantly from one dataset to another. A significant drawback of this method is that the number of emerging patters that can occur may be huge. An approach to overcome this drawback is to use a measure of interestingness in order to reduce the number of the mined patterns to those that carry the most information. In our method we used as interestingness measure the chi-test and therefore the emerging patterns are called chi emerging patterns [5]. The formal definition of frequent patterns, emerging patterns and chi-emerging patterns are going to be presented in detail in the following section.

The second component is completely independent from the first. It is a distance-based scoring for the sequences. In order to calculate the distance, the method uses the Manhattan distance. The equation that calculates the Manhattan distance is also presented in the following section.

Every component calculates some scores. The first component calculates eight scores and the second one calculates five scores. The total of thirteen scores is used as input to a classifier, which decides whether a sequence contains a polyadenylation site or not. In that step any classifier that handles real-valued attributes can be used. Some of the state of the art machine learning algorithms that have been used are *Support Vector Machines*, *Neural Networks* and *Classification Trees*.

2. Materials

In this section we are going to present the definitions and the equations of every method that is used by PolyA-iEP.

2.1 Datasets

The datasets that our method can handle are divided in two major categories, namely the positive and the negative datasets. The positive dataset are mRNA 3` end sequences. The negative examples are a combination of 5` UTR, coding and Intronic sequences. The datasets are provided to the method in text files which contain the sequences. Each sequence has length of 400 nucleotides. In the positive sequences the polyadenylation site is found at the 301st position.

2.2 Frequent Patterns and Association Rules

Let $I = \{i_1, i_2, ..., i_N\}$ be a finite set of binary attributes called *items* and $D = \{t_1, t_2, ..., t_N\}$ be a finite multiset of transactions, which is called the *database*. Each *transaction*

t_i contains a subset of items chosen from *I* and has a unique transaction ID. A set of items is reffered to as an *itemset*. If an itemset contains *k* items, it is called a *k*-itemset. The number *k* is called *size* or *length* of the itemset. The itemset that does not contain any items is called an empty itemset. A transaction $T \in D$ is said to contain an itemset $X \subseteq I$, if $X \subseteq T$.

An *Association Rule* is an implication of the form $X \Rightarrow Y$ where $X \subset I$, $Y \subset I$ and $X \cap Y = \varnothing$. The itemset *X* is called *antecedent* or *Left-Hand-Side* (LHS) of the rule and the itemset *Y* is called *consequent* or *Right-Hand-Side* (RHS) of the rule.

There are many measures that have been proposed in order to evaluate a rule's interestingness. The most popular are *support* and *confidence*. They respectively reflect the usefulness and certainty of discovered rules. More specifically, support determines how often a rule is applicable to a given dataset, whereas confidence determines how frequently items in *Y* appear in transactions that contain *X*. The *support* of a rule $X \Rightarrow Y$ is equal to the support of the itemset $X \cup Y$ and is defined as the fraction of transactions in the database which contain the itemset. The support of an itemset *X* is calculated as presented in the following equation:

$$support_D(X) = \frac{\left|\{T \in D \mid X \subseteq T\}\right|}{|D|}$$

The *confidence* of the rule $X \Rightarrow Y$ is defined as the fraction of transactions in database that contains $X \cup Y$ over the number of transactions that contain only *X*. In other words, confidence is equal to the fraction of the support of $X \cup Y$ in *D*, over the support of *X* in *D*. The equation that defines confidence is presented below:

$$confidence_D(X \Rightarrow Y) = \frac{supp_D(X \cup Y)}{supp_D(X)}$$

2.3 Emerging Patterns

Emerging patterns 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):

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

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:

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

2.4 Interesting Emerging Patterns

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) $|X| = 1 \vee |X| > 1 \wedge (\forall Y \subset X \wedge |Y| = |X| - 1 \wedge chi(X,Y) \geq \eta)$, Where η = 3.84 is a minimum chi value threshold and *chi*(*X*, *Y*) is computed using chi-squared test.

More information about the used method can be found in [1].

2.5 Manhattan Distance

The *Manhattan distance* between two items is the sum of the differences of their corresponding components

$$d(x, y) = \sum (x_i – y_i)$$

2.6 Machine Learning and Classification

Supervised learning is probably the most common type of machine learning problems. In general, it is the task in which a function is generated in order to map inputs to desired outputs. The function, which is called classifier if the output is a discrete value, is inferred through analyzing training data. The problem of inferring this function is called classification or prediction. The training data which are given to the machine learning algorithm are in most cases a pair of a vector and the category to which the example belongs. In the field of machine learning the category which is assigned to each training data is called label. The classification process is presented in Figure 1.

3. Methods

In this section we are going to present in detail every step of the PolyA-iEP method. Figure 2 presents the architecture of our method. The upper side presents the first component which is related to Chi Emerging Patterns. The lowest side presents the distance-based scoring. It is clear that the results from the two components are used as inputs to a classification algorithm which will decide whether the sequence is a positive or negative example, i.e contains or not a polyadenylation site.

3.1 Extraction of Elements

3.1.1 Input

Nucleotide sequences in the form that is presented in paragraph 2.1.

3.1.2 Processing

Previous studies have shown that the region near the polyadenylation site can be divided in four elements. These elements contain different nucleotide frequencies and so they must contain different patterns which our method is intended to mine. The elements, which are presented in Figure 3, are FUE (Far Upstream Element), NUE (Near Upstream Element), CE (Cleavage Element) and NDE (Near Downstream Element).

The application is fully customizable. The user can choose the boundaries of the elements. The main reason behind the consideration of the elements is to search for extended patterns in the sequences, i.e. patterns in different elements that occur simultaneously. An example of a simple pattern could be {ATTA}. An example of an extended pattern would be {FUE_ATCT, NUE_AAA, CE_TT, NDE_AAAG}. This pattern can be interpreted as following: Simultaneously appear the ATCT in the FUE element, the AAA in the NUE element, the TT in the CE element and the AAAG in the downstream element. Extended patterns can definitely be more informative than the simple patterns, which can lead in better distinction of the positives and negative examples and the increasing of the overall accuracy of the method.

3.1.3 Output

Each sequence divided in the four elements.

3.2 Extraction of k-grams

3.2.1 Input

Each element of the sequence has been produced in the previous step (see 3.1.3).

3.2.2 Processing

In this step each sequence is going to be presented by a number of vectors one for each element, i.e. FUE, NUE, CE and NDE. These vectors will contain the frequencies of each valid nucleotide pattern. The user can specify the maximum length of the pattern. The patterns can contain every combination of the four nucleotides. Moreover, we have included some wildchars which represent the presence of one or another nucleotide, based on the IUPAC notions. These wildchars are: R (A or G – puRine), Y (C or T - pYrimidine), M (A or C – aMino), K (G or T – Keto), S (C or G – Strong, 3 H bonds), W (A or T – Weak, 2 H bonds). For instance, if CCT and CTT are valid patterns then also the valid pattern CYT occurs.

3.2.3 Output

Vectors which contain the frequencies of each valid nucleotide pattern found in the sequences.

3.3 Binary Discretization

3.3.1 Input

The vectors with the frequencies of each valid nucleotide pattern that where calculated in the previous step.

3.3.2 Processing

Information entropy was used as the discretization method in PolyA-iEP. All possible cut points are checked for each *k*-gram pattern among all pattern frequencies. The cut point that has the maximum information gain is finally selected. Given a set of training examples *S*, entropy (*E*) is defined by the following equation:

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

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:

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

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

As mentioned below PolyA-iEP uses binary discretization and a single cut point which maximizes information gain is sought among all attribute values.

3.3.3 Output

The *k*-gram vectors that were extracted as described in section 3.2 are transformed into a transaction of items. The items included in the transaction are those *k*-grams that have frequency greater than the corresponding cut point, which was previously calculated. In this step the data are transformed in a format that permits the extraction of emerging patterns.

3.4 Mining Interesting Emerging Patterns

3.4.1 Input

The transactional data that have been produced in the previous step are used in this step for mining interesting emerging patterns.

3.4.2 Processing

For mining the interesting emerging patterns FP-Growth, a frequent itemsets mining algorithm [6] has been modified accordingly. The modified algorithm receives as input two datasets, the background and the target dataset, and discovers all chi-emerging patterns, based on the user-specified parameters (i.e. minimum support threshold and minimum growth rate threshold). For this reason, two sets of emerging patterns E_+ and E_-, are generated for the positive and the negative class respectively. As already mentioned a dataset that contains 3 types of negative sequences (5' UTR, coding, and intronic), has been used in the proposed setup. These negative sequences express quite different nucleotide distributions. If all negatives were dealt as a whole only, then the effectiveness of classification would be moderate. So, PolyA-iEP mines four pairs of E_+/E_- sets of emerging patterns, one for discriminating positives from all negatives as a whole 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_AGT, NUE_CT}: 0.25. This interesting emerging pattern associates the appearance of pattern "AGT" in the Far Upstream Element, with pattern "CT" in the Near Upstream Element. The strength of this interesting emerging pattern is 0.25

3.4.3 Output

The four E_+/E_- pairs of sets of emerging patterns.

3.5 Distance-Based Scoring

3.5.1 Input

Nucleotide sequences in the form that is presented in paragraph 2.1.

3.5.2 Processing

The distance-based scoring of PolyA-iEP is independent from the previous steps. This step includes the calculation of the frequencies of nucleotides at each position of a sequence and the construction of a nucleotide frequency matrix for each class, as shown in Table 2. For example, nucleotide A has 0.14 frequency is position 1 of the sequences used to generate the matrix presented in Table 1. Then, for each position in the sequence the rankings of the nucleotides are calculated according to their frequency at this particular position (Table 2). 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.

3.5.3 Output

The five nucleotide frequency ranking matrices that are constructed, for each of the following categories: positives, all negatives, 5΄ UTR negatives, coding negatives, and intronic negatives.

3.6 Classification

3.6.1 Input

Any unlabeled sequence that has been processed through the steps 3.1-3.3 and is represented in transactional format.

3.6.2 Processing

As mentioned before the two PolyA-iEP components are used in order to produce thirteen scores. The first component produces eight scores and the second one five scores. These scores represent the attributes of each sequence from the initial dataset. These scores are given as inputs to any classifier that can handle real-valued numeric attributes and decides whether the sequence contains or not the polyadenylation site. Some of the state of the art machine learning algorithms that have been used are *Support Vector Machines*, *Neural Networks* and *Classification Trees*.

From the first PolyA-iEP component the mined E_+/E_- pairs of sets of emerging patterns (see 3.4.3) are used for scoring an instance as being positive or negative. For this reason, pairs of scores for an instance *T* in transaction format are calculated as described by the following equations.

$$score(T,+) = \sum_{e \subseteq T, e \in E_+} strength_{- \to +}(e)$$

$$score(T,-) = \sum_{e \subseteq T, e \in E_-} strength_{+ \to -}(e)$$

The first score indicates if *T* is positive and the second if it is negative. The final classification could be made by comparing the values of the two scores and assigning the instance to the class with the highest score. The total number of scores that are produced in this step are 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.

From the second PolyA-iEP component the distance of a sequence from a class and subclass (5´ UTR, intronic, or coding) is calculated. For this reason, the sequence is converted into a nucleotide frequency ranking vector using the nucleotide frequency matrix of the class or subclass (see section 3.5). Then, the distance from the unary vector is calculated and divided by the length of sequence. For example, given the ranking matrix in Table 2, 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 2 belongs. Five distances-scores are finally calculated, one for each of the following categories: positives, all negatives, 5´ UTR negatives, coding negatives, and intronic negatives.

3.6.3. Output

The final classification of the input sequences as containing or non-containing a PolyA site.

4. Notes

4.1 Machine Learning Algorithms

We have mentioned in paragraph 3.6 that the last step of the PolyA-iEP method is the building of a classifier in order to classify unknown sequences. For this purpose, the WEKA machine learning library [7] has been used. WEKA provides many algorithms implemented in a very efficient way that can handle real valued attributes. The state of the art machine learning algorithms that have been used are referred to in paragraph 3.6. The implementation of the Support Vector Machines can be found under the tab classify to the path *classifiers/ functions/ SMO*. The implementation of Neural Networks can be found also under the tab classify to the path *classifiers/ fuctions/ multilayerPerceptrons.* Lastly, in weka there is an implementation of a classification tree algorithm called C4.5. The implementation of this algotithm can be found under the tab classify to the path *classifiers/ trees/ J48.*

References

[1] Tzanis G., Kavakiotis I., Vlahavas I. (2011) PolyA-iEP: A Data Mining Method for the Effective Prediction of Polyadenylation Sites. *Expert Syst Appl*, Elsevier, 38(10): 12398-12408

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

[3] Agrawal R., Imielinski T., & Swami A. (1993) Mining association rules between sets of items in large databases. *In Proceedings of the ACM SIGMOD conference on management of data* (pp. 207–216).

[4] Dong G., & Li J. (1999) Efficient mining of emerging patterns: Discovering trends and differences. *In Proceedings of ACM-SIGKDD'99* (pp. 43–52).

[5] Fan H. (2004) Efficient mining of interesting emerging patterns and their effective use in classification. PhD thesis, University of Melbourne, Australia.

[6] Han J., Pei J., & Yin. (2000). Mining frequent patterns without candidate generation. *In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 1-12.

[7] Hall M., Frank E., Holmes G., Pfahringer B., Peter R., & Witten I. H. (2009) The weka data mining software: An update. SIGKDD Explorations, 11(1).

CAPTIONS

Figure 1: Training and prediction processes.

Figure 2: PolyA-iEP architecture.

Figure 3: The four sequence elements used in PolyA-iEP.

Table 1: An example of a nucleotide frequency matrix for sequences of length 5.

nucleotide	**position in sequence**				
	1	2	3	4	5
A	0.14	0.06	0.30	0.11	0.16
C	0.21	0.21	0.18	0.29	0.28
G	0.35	0.36	0.26	0.40	0.28
T	0.30	0.37	0.26	0.20	0.28

Table 2: 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