

Prediction of Plant mRNA Polyadenylation Sites

Xiaohui Wu, Guoli Ji, and Qingshun Quinn Li

Abstract

Messenger RNA polyadenylation is one of the essential processing steps during eukaryotic gene expression. The site of polyadenylation [poly(A) site] marks the end of a transcript, which is also the end of a gene in most cases. A computation program that is able to recognize poly(A) sites would not only be useful for genome annotation in finding genes ends, but also for predicting alternative poly(A) sites. PASS [Poly(A) Site Sleuth] and PAC [Poly(A) site Classifier] were developed to predict poly(A) sites in plants. PASS was built based on the Generalized Hidden Markov Model (GHMM), which consists of four functional modules: input model, poly(A) site recognition module, graphic process module, and output module. PAC is a classification model, integrating several features that define the poly(A) sites including *K*-gram pattern, Z-curve, position-specific scoring matrix, and first-order inhomogeneous Markov sub-model. PAC can be used to predict poly(A) sites from species whose polyadenylation profile is unknown. The result of PASS and PAC is an output of a few files with one of them containing the score or probability of being a poly(A) site for each position of a given sequence. While the models were built mostly based on poly(A) profile data from *Arabidopsis*, it is also functional in other higher plants since their profiles are quite similar.

Key words Classification based modeling, Polyadenylation, Predictive modeling, GHMM, PASS, PAC

1 Introduction

The location of a terminal nucleotide of the 3'-untranslated region (3'-UTR), which is exposed after an endonuclease cleavage, is known as a poly(A) site. A poly(A) site marks the end of the transcribed mature mRNA, and, as such, it can be used to find and annotate the end of a gene in most cases. The location of a poly(A) site for a gene is mostly predetermined by the so-called polyadenylation signals. Traditionally, poly(A) sites are identified by examining the expressed sequence tags (ESTs) which are reverse-transcribed from mature mRNA. Since the poly(A) tail is added post-transcriptionally, alignment of ESTs to their respective genomic sequences will reveal the location of poly(A) sites. Indeed, there are a number of datasets with collections of poly(A) sites [1–5]. Further analyses of these

datasets have elucidated the poly(A) signals that determine the poly(A) site locations at the genome level. Such information about poly(A) signals, particularly those from Arabidopsis [3] and rice [5], is the foundation of this work which involves the construction of predictive models for a systematic prediction of plant poly(A) sites.

The complexity of poly(A) sites is demonstrated by the fact that poly(A) sites can be located in a short region of the 3'-UTR [3, 5, 6]. Furthermore, plant poly(A) signals possess little conservation. These properties, coupled with a limited knowledge of numerical prediction and its application to plant polyadenylation, make it hard to predict exact poly(A) sites using computational methods. PASS [7] and PAC [8] were developed to predict poly(A) sites in plants, which were based on generalized hidden Markov model (GHMM) and a classification-based prediction model, respectively. PASS consists of four functional modules: input module, poly(A) site recognition module, graphic process module, and output module. In the input module, sequences or sequences with PASS predicted scores can be entered by universal FASTA format or manually. The input system will also examine the authenticity of the sequences in that warning will be given when undesirable characters other than A, T, G, C, or U (the five nucleotides found in DNA and/or RNA) is entered. As the core of the algorithm, the function of the poly(A) site recognition module is to implement the forward-backward algorithm of the GHMM-based poly(A) site recognition model, and produces a probability score that each of the nucleotides to be a potential poly(A) site. In the graphic processing module, given the output value (or scores) from the site recognition module, PASS is designed to show the numerical scores and other results in a graphic display. Moreover, users may choose their desire color and line styles etc. The graphs can be saved as pictures in .bmp or .wmf formats. In the output module, the poly(A) site recognition results are saved in a format that allows traceability. Thus, the results as a whole (including scores, Sensitivity, and Specificity) are saved for future reference.

Another poly(A) site prediction tool, PAC, was designed to extract poly(A) sites from known dataset (with poly(A) specified) and use this information to predict poly(A) sites from unknown sequences. This is powerful when dealing with a new species that is substantially different (or do not know if there is different) from Arabidopsis or rice models. It constitutes three major steps. In the first step, PAC takes a set of nucleotide sequences as input and imports the parameter file for the classification model and the features. In the second step, a classification algorithm is performed to train and test the model. In the final step, the output files are generated. The output of PAC is a few text files with one of them containing the probabilities of each sequence to determine the potential poly(A) sites and the others as intermediate files.

While originally designed to identify the poly(A) sites in Arabidopsis, PASS or PAC may be adapted for species other than Arabidopsis if the species in question has sufficient number of poly(A) sites to train the model parameters. PASS and PAC had been successfully applied to identify the poly(A) sites in Arabidopsis [7, 8], rice [4] and green alga *Chlamydomonas reinhardtii* [9]. In this chapter, we use Arabidopsis data set as an example to illustrate how to use PASS and PAC to identify poly(A) sites.

2 Materials

2.1 Equipment

Computer Hardware: MS Windows XP/Vista/7 or later.

To use PAC, please make sure that Weka and Java VM are installed in your system. You can download a self-extracting executable version of Weka that includes Java VM 5.0 from <http://prdownloads.sourceforge.net/weka/weka-3-5-6jre.exe>. Other versions of Weka can be downloaded from <http://www.cs.waikato.ac.nz/ml/weka/>.

2.2 Data

The 8 K dataset or other data sets can be downloaded from the following links: <http://www.users.miamioh.edu/liq/links.html>; or <http://www.polyA.org>.

2.3 Setting Up

Download PASS or PAC packages from the following links: <http://www.users.miamioh.edu/liq/links.html>; or <http://www.polyA.org>.

2.4 Install PASS or PAC on MS Windows

Simply unzip the PASS or PAC package to any directory.

3 Methods

3.1 Using PASS to Predict Poly(A) sites

3.1.1 Generalized Hidden Markov Model Used in PASS

PASS was developed based on the GHMM. The topological structure was one of the most important factors in designing a GHMM model. PASS employed a GHMM model that recognized the signals from left to right, and only allowed the recognition of signals from the current stage to the next stage in one direction. Based on the analysis of the model of plant poly(A) signals [3], the sequences can be classified into five regions (Fig. 1a). The poly(A) signals are distributed in these regions with some spacing between the two signal elements. Based on this, a background stage was added between the two signal stages. To simplify the model, the length of every signal was fixed but the length of background was variable. It was also possible that two signals were next to each other and thus the length of the background may be 0. The final model was designed in such a way that all calculations began on the first stage and ended at the last stage (Fig. 1b).

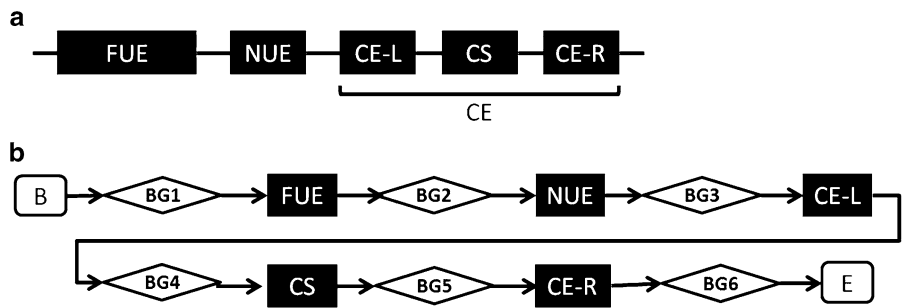


Fig. 1 The structure of plant mRNA polyadenylation signals, the order of the GHMM, and a flowchart of PASS. (a) A working model. (b) The order of GHMM. The *arrowheads* indicate the probability of changing of states (all probabilities were set to be 1). The *rectangles* represent regions with fixed length while the *braces* indicate regions with variable length. *FUE* far upstream element, *NUE* near upstream element, *CE* cleavage element, *CE-L*, *CE-R* cleavage element left or right to the poly(A) site, *CS* cleavage site, also known as poly(A) site, *B* beginning of the scan, *Bg* background sequences between cis-elements, *E* end of the scan. Note that because YA is not found in all sequences, other dinucleotide combinations are also considered in GHMM

In this model, some basic parameters were set as follows: the number of stages was 11 (Fig. 1b, from Bg1 through Bg6); the array of signals in every stage was set to be {A, T, C, G}. Stages in odd numbers were the background stage with variable length, and stages in even numbers were signal stages with fixed length. Because the model begins with the first stage and ends at the last stage, the initial state distribution was set in $\pi = \{1, 0, \dots, 0\}$. Every stage (*i*) can be only transferred to the *i* + 1 stage. Each stage was assigned a few parameters including the signal nucleotide composition, signal pattern length, etc.

A sliding 180 nt-wide window was applied to calculate the outputs of scores for the sequences. For every nucleotide, PASS deduced a score in all windows that contained this nucleotide. The window slides along the entire sequence, combining values of forward-backward variables to generate a score value for each nucleotide in that window.

3.1.2 Interface of PASS

The interface of PASS mainly consists of a title bar, a menu bar, a graph display area, a sequence display area, and an analysis operation area, as shown in Fig. 2. Furthermore, PASS possesses the function of language switching (between English and Chinese), which is very convenient for people using these two languages. All the terms and concepts in captions of PASS rest on the knowledge about bioinformatics and the experience from biology researchers.

3.1.3 Import One or More Sequences from File

Choose the menu “File → Import sequence only”. The format of the sequences is FASTA, and PASS omits blank lines. If you already have the output file of PASS and want to display the result graphically, you may choose “Import Sequences with Score From File”, which is almost the same as import sequences without scores from file.

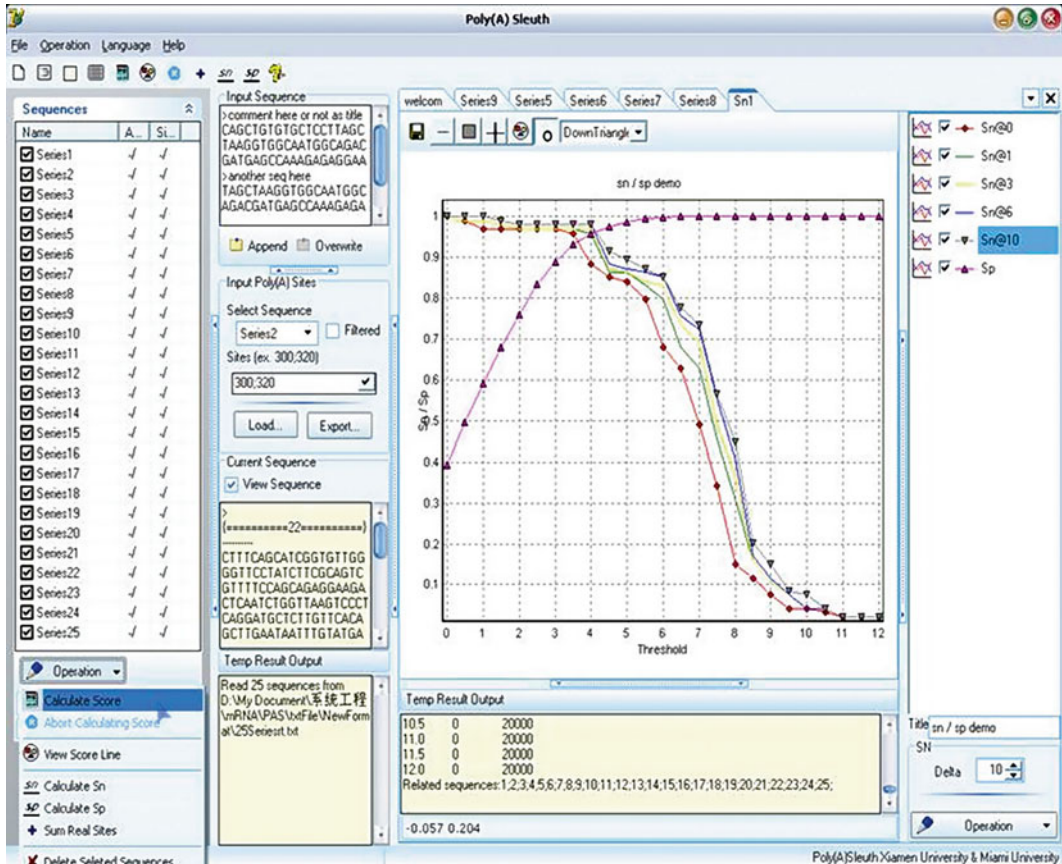


Fig. 2 Screenshot of the interface of PASS

3.1.4 Calculate Scores

Choose the menu “Operation→Calculate Score”. During the calculation, the marquee status bar will show “calculating seriesX”, which means seriesX is being calculated. When a sequence has been calculated, the corresponding place of column [A..] will be marked as “√”. To display the sequence graphically, choose “Operation→View Score Line” or double click a sequence.

3.1.5 Evaluation of the Results from the Model

Before evaluation, you need to give known poly(A) sites in the sequences. To input the known poly(A) sites, you need to switch to the [Input Poly(A) Sites] Panel. You may load the poly(A) sites from a file with the numbers indicating the positions of the sites or manually input the known sites. Then you can choose “Operation→Evaluate(Sn,Sp)” to calculate the sensitivity (Sn) and specificity (Sp) (see **Note 1**).

3.1.6 Related Operations

1. *Calculate Score*: calculate score of the selected sequences.
2. *Abort Calculate Score*: abort Calculating Score of the selected sequences. This action will abort the calculation of all the selected sequences.

3. *View Score Line*: view the score line of the selected sequence. Double click the sequence can have the same effect.
4. *Calculate Sn*: calculate Sn and Sp using the selected sequences.
5. *Calculate Sp*: calculate Sp separately. You can specify your own random sequences.
6. *Calculate Sp of Selected Sequences*: calculate Sp of selected Sequences. You must select some calculated sequences first.
7. *Sum Real Sites*: calculate the total count of the sites of the selected sequences. The result will be shown in [Temp result Output] Panel (*see Note 2*).

3.1.7 An Example to Predict Poly(A) Sites

1. Input Sequences
method1:[File] → [Import] → [Import sequences only].
method2:manually input sequences from [Input Sequences] panel.
2. Calculate Score
Choose “[Operation] → [Calculate Score]” (*see Note 3*).
3. View Score Line
Choose “[Operation] → [View Score Line]” or double click a sequence.
4. Input Poly(A) Sites
Use “[Input Poly(A) Sites]” Panel. You can load sites from a file or manually input the sites.
5. Calculate Sn and Sp
Choose “[Operation] → [Evaluate(Sn,Sp)]” to calculate Sn and Sp.

3.2 Using PAC to Predict Poly(A) Sites

3.2.1 Classification- Based Model Used in PAC

When using a classification algorithm for predicting poly(A) sites, the nucleotide sequence needs to be converted into numeric format. Consequently, the features of poly(A) signals around the cleavage sites were extracted based on the profile of nucleotide sequence distribution around poly(A) sites. In PAC, five feature representation methods were adopted to describe the makeup of nucleotide sequences. These methods were chosen to confirm whether each one could generate unique features from different training datasets. Finally, the numerical vector was used as the input of the classification algorithm. The distribution of features in different areas of a window sequence is shown in Fig. 3. There are several kinds of feature representation methods in PAC, including *K*-gram nucleotide sequence pattern, Z-Curve, the position-specific scoring matrix based score, the probability based on first order inhomogeneous Markov sub-model, and the weight of NUE Signals.

To find the feature group that is most effective for poly(A) site prediction, the selection algorithm involves feature evaluation and searching algorithms. Six commonly used feature evaluation methods

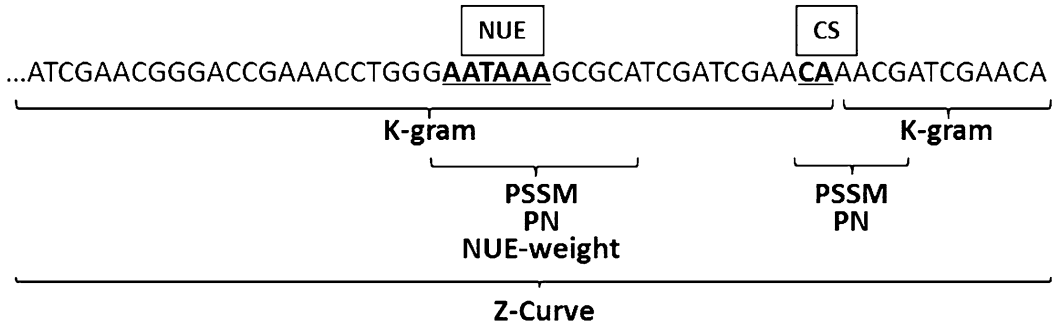


Fig. 3 Different methods were used to extract features from different regions of the poly(A) signals. *NUE* near upstream element, *CS* cleavage site, *PSSM* position-specific scoring matrix, *PN* probability based on first-order inhomogeneous Markov sub-model

were adopted to select the relatively optimal subset of features. These methods include ChiSquaredAttributeEval, GainRatioAttributeEval, InfoGainAttributeEval, OneRAttributeEval, ReliefAttributeEval, and SymmetricalUncertAttributeEval, and four searching algorithms, including ExhaustiveSearch, GeneticSearch, GreedyStepwise, and RandomSearch, in an open source software called WEKA3.5 [10].

The classification-based poly(A) site recognition model (PAC) is based on two steps: training and testing. For the training classification model, the window sequences in the training dataset were converted into numeric form using the final feature-space. Then, the numeric formatted sequences were classified using one of the classification algorithms to build a training model. Different classification algorithms were used to build different training models. In the testing step, the training model was used for predicting the poly(A) sites from the test dataset. As each given sequence passed through the test model, each position of the sequence was predicted as a poly(A) site, or not.

Long sequence analysis involved a pre-processing step in which a given length, $n - 162 + 1$ (where n is the length of the sequence) of sub-sequences, was first produced according to the window size before entering into the model. Then, the final output of the model could be shown as the prediction results of the sequences. Both the training and test processes permit inputting sequences longer than 162 nt; however, the long sequences were manually cut into 162 nt window sequences as inputs for the training process.

3.2.2 Interface of PAC

The interface of PAC mainly consists of a title bar, a menu bar, several panels for sequence input, Weka settings, Feature settings, and sequence output.

3.2.3 Introduction About the Related Files

1. Ftr File(.ftr) The file integrating all settings of wanted features.
2. Arff File(.arff) The file for WEKA transformed from the data matrix.
3. Model File(.model) The output file of training model needed in the test model.
4. Matrix File(.mtx) The file of a data matrix.
5. Pr File(.pr) The output file of WEKA.
6. Pred File(.pred) The file transformed from the pr File.

3.2.4 Sequence Input

Two ways are allowed for sequence input. One way is to input the “True/False sample” files, which inputs the short sequences with or without poly(A) sites, respectively. Another way is to input the “long sequences/sites” files, which specifies the long sequences with multiple sites and the sites file.

3.2.5 Weka Settings

Click the “weka.jar” button to specify the file path of weka.jar. If weka is installed in your windows, PAC will get the file path automatically. If the checkbox “-p distribution” is checked, the pr file will contain the probability of the classification, otherwise the confusion matrix.

3.2.6 Output Settings

When the checkbox “Continuous Output” is checked, the Ftrs/Arff/Model files will be generated if the Ftrs/Arff/Model files are specified. When it is unchecked, only the final file is generated.

When the checkbox “Optimal Output” is checked, PAC will not calculate the matrix but use the arff file specified to generate other output files (e.g., arff, model file). If it is unchecked, PAC will not use the specified files but calculate all related files in the whole process.

3.2.7 Features Settings

Set the parameters for the five features integrated in PAC, including *K*-gram, Markov sub-model, PSSM, Z-curve, and Weight of NUE.

3.2.8 An Example to Predict Poly(A) Sites

1. Model training (Fig. 4): Input the sequences for training and import the .ftr file to specify the feature parameters. Then click “output [.model]” button to output model files.
2. Model testing (Fig. 5): Input the sequences for testing and import the .ftr file to specify the feature parameters. Specify the model files from **step 1**. Then click “output [.pred]” button to output the prediction files.

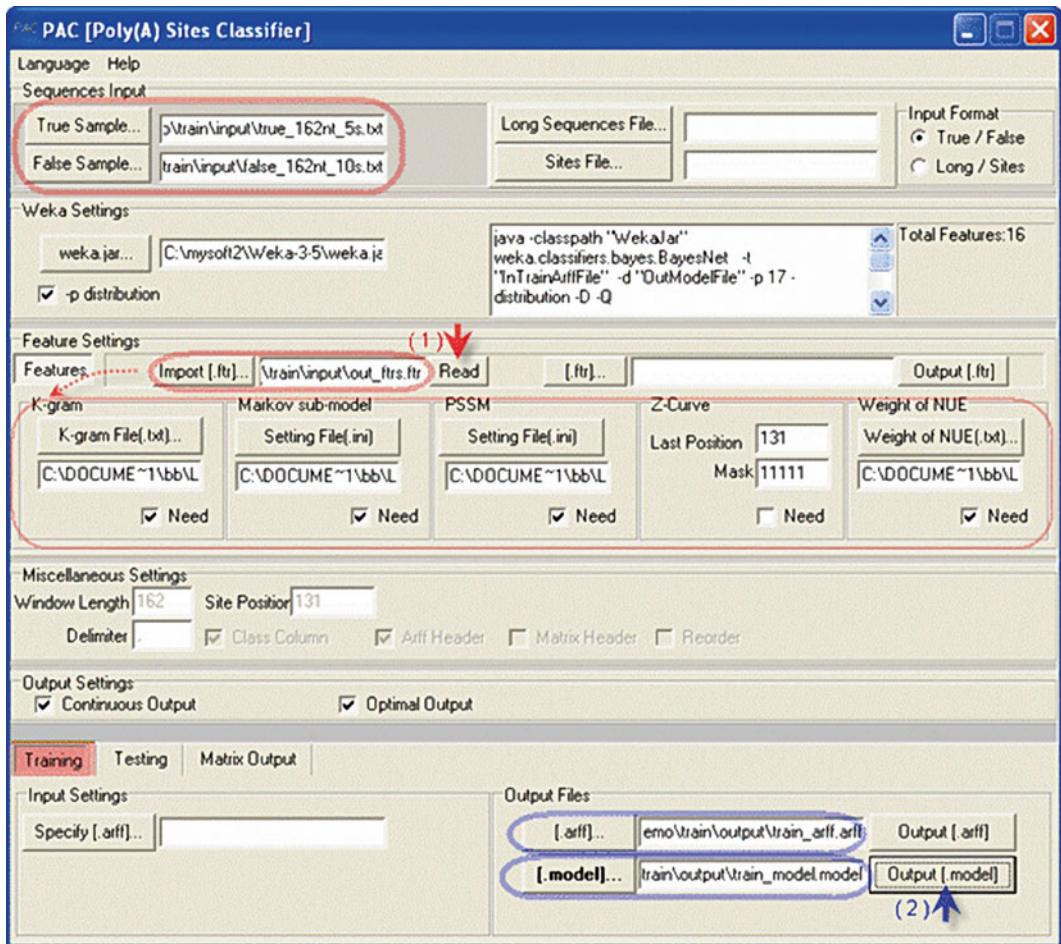


Fig. 4 Screenshot of the training step of PAC

4 Notes

1. You can calculate Sp without any specified sequence file or poly(A) sites. This operation requires a relatively long sequence and it is already calculated (i.e., the specified file must have both the sequence and scores). If there is any sequence in selection whose background color is in grey, it will be used in calculation of Sp and a new line will be created. Otherwise, a blank page is created, then you can specify a file to calculate Sp. Choose "Operation → Calculate Sp" to calculate Sp.
2. Except the action of view score line and calculate Sp, other operations are only in effect when some sequences are selected. That is, before these actions, you need to selected sequences from [sequences] panel.

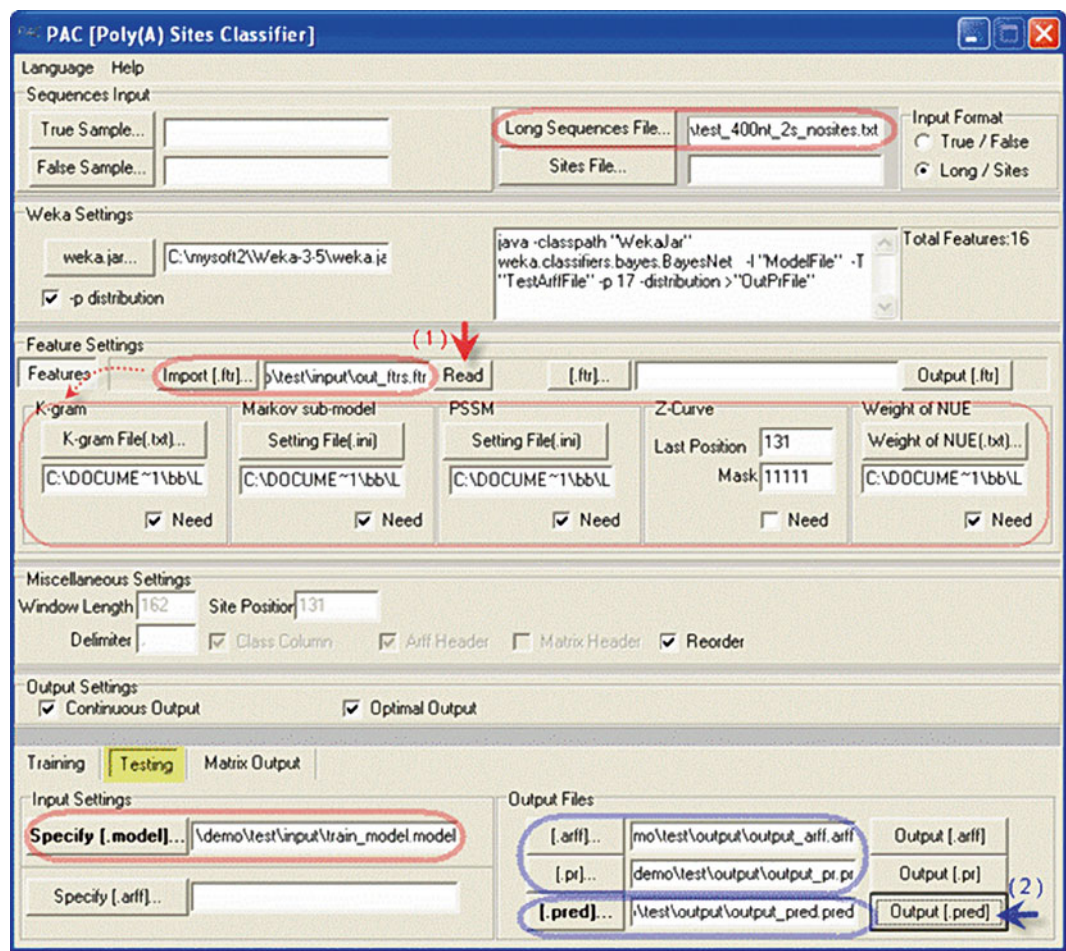


Fig. 5 Screenshot of the testing step of PAC

3. In the calculation, the marquee status bar will show “calculating seriesX”, which means seriesX is being calculated. When a sequence has been calculated, the corresponding place of column [A..] will be marked as “√”.

Acknowledgement

Funding supports for this work were from the National Natural Science Foundation of China (Nos. 61174161 and 61304141), the Natural Science Foundation of Fujian Province of China (No. 2012J01154), the specialized Research Fund for the Doctoral Program of Higher Education of China (Nos. 20130121130004 and 20120121120038), and the Fundamental Research Funds for the Central Universities in China (Xiamen University: No. 2013121025), Xiamen Shuangbai Talent Plan (to QQL), and US National Science Foundation (grant nos. IOS-0817829 and IOS-1353354 to QQL).

References

1. Graber JH, Cantor CR, Mohr SC, Smith TF (1999) In silico detection of control signals: mRNA 3' end-processing sequences in diverse species. *Proc Natl Acad Sci U S A* 96(24): 14055–14060
2. Graber JH, Cantor CR, Mohr SC, Smith TF (1999) Genomic detection of new yeast pre-mRNA 3'-end-processing signals. *Nucleic Acids Res* 27(3):888–894
3. Loke JC, Stahlberg EA, Strenski DG, Haas BJ, Wood PC, Li QQ (2005) Compilation of mRNA polyadenylation signals in Arabidopsis revealed a new signal element and potential secondary structures. *Plant Physiol* 138(3):1457–1468
4. Shen Y, Ji G, Haas BJ, Wu X, Zheng J, Reese GJ, Li QQ (2008) Genome level analysis of rice mRNA 3'-end processing signals and alternative polyadenylation. *Nucleic Acids Res* 36(9):3150–3161
5. Shen Y, Liu Y, Liu L, Liang C, Li QQ (2008) Unique features of nuclear mRNA poly(A) signals and alternative polyadenylation in *Chlamydomonas reinhardtii*. *Genetics* 179(1): 167–176
6. Li QQ, Hunt AG (1997) The polyadenylation of RNA in plants. *Plant Physiol* 115:321–325
7. Ji G, Zheng J, Shen Y, Wu X, Jiang R, Lin Y, Loke JC, Davis KM, Reese GJ, Li QQ (2007) Predictive modeling of plant messenger RNA polyadenylation sites. *BMC Bioinform* 8(43):43
8. Ji G, Wu X, Shen Y, Huang J, Li QQ (2010) A classification-based prediction model of messenger RNA polyadenylation sites. *J Theor Biol* 265(3):287–296. doi:[10.1016/j.jtbi.2010.05.015](https://doi.org/10.1016/j.jtbi.2010.05.015)
9. Ji G, Wu X, Li Q, Zheng J (2010) Messenger RNA polyadenylation site recognition in green alga *Chlamydomonas Reinhardtii*. *Lect Notes Comput Sci* 6063:17–26. doi:[10.1007/978-3-642-13278-0_3](https://doi.org/10.1007/978-3-642-13278-0_3)
10. Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques. Elsevier, San Francisco, CA

Polyadenylation in Plants

Methods and Protocols

Hunt, A.G.; Li, Q.Q. (Eds.)

2015, IX, 225 p. 34 illus., 23 illus. in color., Hardcover

ISBN: 978-1-4939-2174-4

A product of Humana Press