Poly(A) signals (PAS) Detection based on optimal nucleotides mapping

Junjie Yue
University of Illinois at Urbana-Champaign
junjie4@illinois.edu

Abstract

Poly(A) signals (PAS) are frequently found in eukaryotic genes and they are associated with the end of transcribed region. The correct recognition of poly(A) signals (PAS) helps to clarify 3-end genomic boundaries of a transcribed DNA region, gene regulation mechanisms and identification of transcripts containing premature termination codons. This work follows the spirt of [1], in which feature extraction strategies are developed for the PAS detection. It is obvious that the signal processing-based features are highly dependent on the numerical mapping of the DNA nucleotides. However, they only used some existing mappings proposed for other purpose, which may not generate suitable features for classification task. Hence, in this work, an extension is performed, which aims to determine the optimal numerical mapping for this specific task. For predictions, Linear Regression model was developed to recognize the most common poly(A) motifs in human DNA. In order to improve the predict result, the optimization problem was formulated by taking the accuracy of the prediction as objective function, the range of variables as constraints.

1. Introduction

Roughly speaking, when DNA is transcribed to RNA, a string of adenine (A) nucleotides, referred to as the polyadenylation tail or the poly(A) tail, is added to the 3-end of the primary RNA transcript. Such a process is called polyadenylation, which is a step to protect RNA stability, nuclear export and translation. A poly(A) tail is downstream of a signaling site that consists of 6 nt, which in human cells most commonly is AATAAA. This signaling site is known as a poly(A) signal and the corresponding 6 nt subsequences in DNA are called poly(A) motifs [2].

The poly(A) signal prediction problem has been studied for decades. There are two versions of the problem: predicting poly(A) signals in mRNA sequences and predicting poly(A) motifs in DNA sequences. Intuitively, the former version is much simpler than the latter one because once the mRNA sequence is given, the poly(A) tail can be identified relatively easily and we need only to search for the poly(A) signal(s) in a sliding window. In DNA, however, the presence of introns in eukaryotes and the absence of poly(A) tails make the recognition much more challenging.

The genomic sequences can be represented mathematically by character strings of symbols from 4 alphabet sequences consisting of the letters A, T, G and C, which represent each one of the nucleotide bases. So we can properly map the nucleotides into one or more numerical sequences, then digital signal processing (DSP) provides a set of novel and useful tools to generate features of the DNA sequences. In [1], a hybrid classification model was developed to predict the poly(A) signals, in which the authors extract features from human genomic DNA and perform the classification task. In their work, they transfer DNA sequences into numerical signals, then propose some signal-based features (e.g., Frequency-based, Wavelet-based features, etc.) as the input of the machine learning models (linear regression model and deep neural network). However, the numerical mappings used in their work are some existing mappings proposed for other purposes, which may not generate suitable features for the classification task. In this work, an optimization problem is formulated to determine the optimal numerical mapping for this specific task.

In terms of single processing-based features, I reproduce the wavelet-based features as mentioned in [1]. The discrete wavelet packet transform (DWPT) was chosen as signal processing tool to generate features for DNA sequences. The major advantage of the WT is that it reveals both time and frequency information about the signal, and thus, can be used effectively for decomposing a function in terms of its time and frequency contents. To illustrate this concept, consider for instance, a signal whose first half consists of a step and the other half is a straight line (DC). Although the FT will reveal in the signal spectrum the presence of high frequencies due to the steps edge, it cannot provide information about the point in time at which the edge has occurred. The WT, on the other hand, conveys

both frequency and time information. Furthermore, the WT has its energy concentrated in time, and is therefore, more suited for the analysis of transient, time-varying signals. More details about feature generation will be discussed in the next section

2. Method

In this section, the dataset information will be introduced first, followed by feature generation. Then the classification model–linear classification will be introduced in the last subsection.

2.1. Datasets

The PAS dataset and pseudo PAS dataset are given with the same number of samples. Specifically, there are 11520 PAS samples and 11520 pseudo PAS samples. Every sample contains 606 nucleotides. The dataset is in FASTA format, which is a text-based format for representing DNA sequences. Nucleotides are represented using single-letter codes.

2.2. Wavelet Bases Feature Extraction

This work reproduce the wavelet-based features of PolyA signal via the discrete wavelet packet transform (DWPT). The wavelet packet analysis is a generalization of wavelet decomposition that offers a richer range of possibilities for signal analysis. This inspires the use of the DWPT to represent the properties of the original DNA sequences [3]. Specifically, the input sequence is passed through high-pass filter and low-pass filter. The DWPT is then applied to each block. For each transformed block, the approximation coefficients along with a few high-frequency coefficients are extracted, as shown in Fig 1.

The procedure of extracting DWPT features are shown as follows:

Step1: Transfer the DNA sequences into discrete signals. To extract the wavelet based features, we define four variables corresponding to four nucleotides respectively:

$$y[i] = \begin{cases} x_A, & x[i] = A \\ x_C, & x[i] = C \\ x_G, & x[i] = G \\ x_T, & x[i] = T \end{cases}$$

where $x[i]$ is the i-th nucleotide of the DNA sequence. In this way, one DNA sequence **x** is changed to a number sequence **y**.

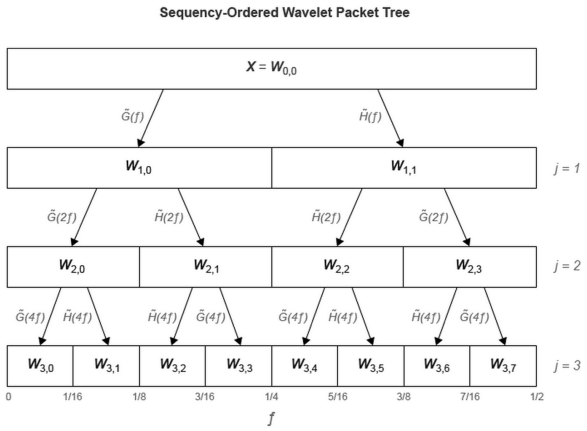

Figure 1. Wavelet Packet Decomposition.

For example:

... AATCGTAATG ...

will be converted to:

$$... \; x_A \; x_A \; x_T \; x_C \; x_G \; x_T \; x_A \; x_A \; x_T \; x_G \; ...$$

Step2: Define the orthogonal wavelets used in the DWPT, specify the transform level. In this work, the Daubechies wavelets were chosen with transform level equal to three.

Step3: Produces two sets of coefficients: approximation coefficients (scaling coefficients), and detail coefficients (wavelet coefficients) based on the chosen wavelet basis.

Step4: The approximation coefficients are split into two parts by using the same algorithm and are replaced by approximation coefficients and detail coefficients, and so on. This decomposition process is repeated until the required level is reached [4]. Then eight reconstruct signals will be generated.

Step5: Compute the energy of the reconstruct signals as features.

Algorithm 1 summarizes the approach followed to obtain the wavelet-based features.

2.3. Linear Classification

In this work, the goal is to predict a binary-valued target (i.e. PAS or pseudo-PAS), which is suitable to develop a linear regression model. Basically to say, linear classification is an optimization problem, Fig 2 shows the basic idea of linear classification, the idea is to find a hyperplane that predict the samples correctly.

Algorithm 1 Wavelet Features Extraction

input DNA sequences
output Wavelet based features
while $i \leq N$ **do**
    ▷ N is the size of dataset, i = 1 initially
    Pick i-th DNA sequence x_i from the dataset
    $y_i \leftarrow x_i$
    Compute the wavelet packet decomposition of y_i
    Reconstruct signals
    $f_w \leftarrow$ Energy of reconstruct signals
    $i \leftarrow i$ +1
return f_w ▷ The output features are of size 8

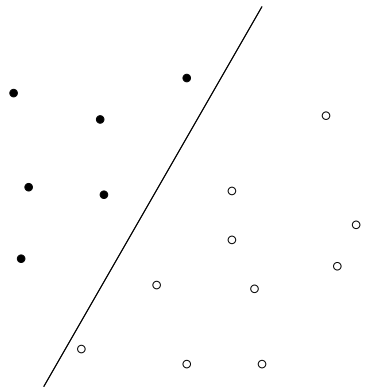

Figure 2. Linear classification.

As discussed above, for each sample, we generate eight features. Associated with each sample is a binary-valued target, the thing we are trying to predict. A binary target takes two possible values, called classes, and which are typically referred to as positive and negative. (E.g. the positive class be PAS and the negative class might be pseudo PSA.) Data cases belonging to these classes are called positive examples and negative examples, respectively. The training set consists of a set of N pairs $(f_w(i), t(i))$, where $f_w(i)$ is the input and $t(i)$ is the binary-valued target, or label. N is the size of dataset.

In this work, 80 % of the data was used to train the classifier, the remaining data was used to test the performance of the model.

In the training stage, the linear classifier model was generated based on the training dataset, then we use the testing dataset to test the performance of the model. The way classifiers work is simple: they compute a linear function of the inputs, and determine whether or not the value is larger than threshold. Therefore, the prediction p can be computed as follows:

$$\mathbf{z} = \mathbf{w}^T f_w + b \tag{1}$$

$$p = \begin{cases} 1, & z \geq 0; \\ 0, & z < 0 \end{cases}$$

where **w** was computed in the training setup.

In the testing stage, there are some exits indicators to analysis the model performance, such as sensitivity, specificity and accuracy etc.

2.4. Optimization Problem Formulation

This section will focus on the optimization problem in terms of mapping of the nucleotides. As discussion above, if we choose different mappings, the extracted features will be different in general. We may have different predict result. This inspires to formulate an optimization problem to find the optimal mapping. The optimization problem was formulated as following:

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & f(x) \\ \text{subject to} \quad & 0 \leq x_i \leq 2,\ i = A, C, G, T. \end{aligned}$$

where $f(x)$ is the accuracy of the test result.

Because the cost function is non-convex and non-differentiable. The Matlab built-in function *pattern search* is applied here to solve the optimization problem. More details about the optimization setup will be discussed in the next section.

2.5. Procedure of Optimization

The objective function in this problem is complicated, it contains transfer the DNA sequences into discrete signals, extract the features from the discrete signals, split the dataset into training set and testing set, training the prediction model, test the performance. Algorithm 2 shows one iteration of this procedure.

Algorithm 2 One Iteration of Optimization Problem

input Dataset
output Accuracy of the test result
Transfer the DNA sequences into discrete sequences based on current mapping
Generate features by Algorithm 1
Training Set $\leftarrow$ 80 % of the dataset
Testing Set $\leftarrow$ 20 % of the dataset
Train the linear classification model by training set
Compute the model performance by testing set
return Accuracy, specificity, sensitivity

From the above procedure, it can be seen that it will take time to just finish one iteration. Hence in order to make this process more efficient, a rough grid search is needed before

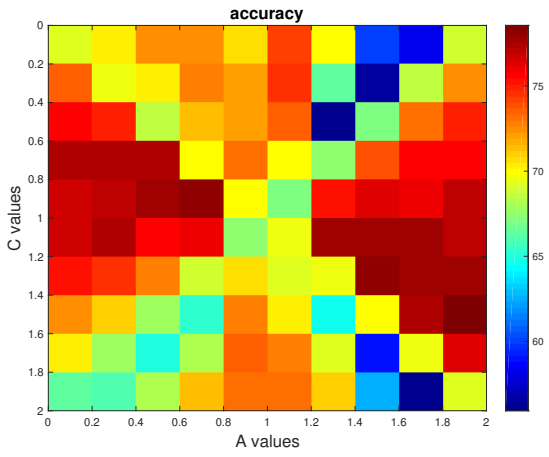

Figure 3. Result by global search.

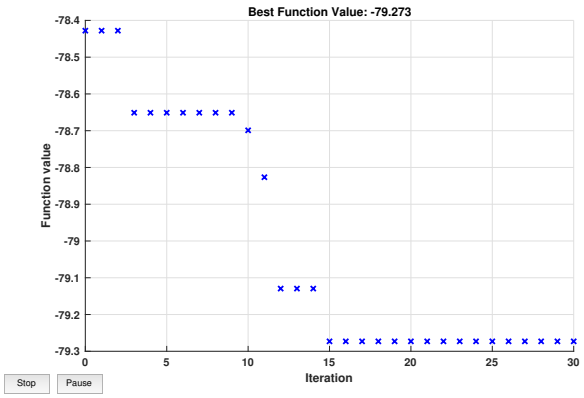

Figure 4. Result by pattern search.

performing the optimization stage. Specifically, we can find all the possible combination of four variables with proper step in the constraints. Then compare the results to find the best one. Use this as initial point to find more accuracy solution. In this work, the step is chosen as 0.1. In addition, in order to visualize the result and reduce the dimension of the variable, assuming that $x_A + x_T = 2$ and $x_C + x_G = 2$, which are valid in some exist mapping. The result will be shown in the next section.

3. Result and Disscussions

As discussed above, a grid search was done before pattern search as shown in Fig 3. It is obvious that the mapping of the nucleotides will influence the result significantly. We may have accuracy around 50 % with some mappings. Based a proper choice, we can increase the accuracy up to 75 %. From the figure, we can choose the following mappings as initial points for the next step:

$$y[i] = \begin{cases} 1.9, & x[i] = A \\ 1.5, & x[i] = C \\ 0.5, & x[i] = G \\ 0.1, & x[i] = T \end{cases}$$

Based on the above initial point, we apply the pattern search, we can get the following optimal mapping with 79.273 % accuracy. In this step, we relax the constraint on the summation of two nucleotides.

$$y[i] = \begin{cases} 1.8687, & x[i] = A \\ 1.5, & x[i] = C \\ 0.9531, & x[i] = G \\ 0.1313, & x[i] = T \end{cases}$$

Fig 4 shows the result of each iteration. We increase the result form 78 % to 79.273 %.

4. Conclusion

In this work, the wavelet-based features were generated to predict the PAS. By wavelet features alone we can achieve the accuracy close to 80 %, which is an encouraging result because the accuracy by the existing research is around 89 % by combing a lot other features [1]. From the result of this work, it also shows that the mapping of nucleotides plays an important role. It is necessary to find the optimal mapping instead of using the exist mapping from the other research topic.

References

[1] Albalawi, Fahad, et al. *Hybrid model for efficient prediction of poly (A) signals in human genomic DNA*. Methods 166 (2019): 31-39.

[2] Magana-Mora, A., Kalkatawi, M., Bajic, V. B. *OmniPolyA: a method and tool for accurate recognition of Poly (A) signals in human genomic DNA*. BMC genomics, 18(1), 620, 2017.

[3] Percival, D. B., A. T. Walden *Wavelet Methods for Time Series Analysis*. (UK) Cambridge, UK: Cambridge University Press, 2000.

[4] Y. Liu. *Wavelet feature extraction for high dimensional microarray data*. (UK) Neurocomputing, 72 (4-6), pp. 985-990, 1999