Hybrid model for efficient prediction of poly(A) signals in human genomic DNA

Fahad Albalawi,a,b,1 Abderrazak Chahid,b,1 Xingang Guo,a,c Somayah Albaradei-b,1 Arturo Magana-Mora,a,c Boris R. Jankovic,a Mahmut Uludag,a Christophe Van Nestea,d Magbubah Essack,a Taous-Meriem Laleg-Kirati,b,1 Vladimir B. Bajica

a King Abdullah University of Science and Technology, Computational Bioscience Research Center, Thuwal 23955-6900, Saudi Arabia
b Taif University, Electrical Engineering, Taif 21944, Saudi Arabia
c Taif University, Electrical Engineering, Taif 21944, Saudi Arabia
d King Abdullah University of Science and Technology, Computational Bioscience Research Center, Thuwal 23955-6900, Saudi Arabia

Keywords:
Poly(A) signals
Deep learning
Machine learning
Signal processing
Computer science
Bioinformatics

ABSTRACT

Polyadenylation signals (PAS) are found in most protein-coding and some non-coding genes in eukaryotes. Their accurate recognition improves understanding gene regulation mechanisms and recognition of the 3′-end of transcribed gene regions where premature or alternate transcription ends may lead to various diseases. Although different methods and tools for in-silico prediction of genomic signals have been proposed, the correct identification of PAS in genomic DNA remains challenging due to a vast number of non-relevant hexamers identical to PAS hexamers. In this study, we developed a novel method for PAS recognition. The method is implemented in a hybrid PAS recognition model (HybPAS), which is based on deep neural networks (DNNs) and logistic regression models (LRMs). One of such models is developed for each of the 12 most frequent human PAS hexamers. DNN models appeared the best for eight PAS types (including the two most frequent PAS hexamers), while LRM appeared best for the remaining four PAS types. The new models use different combinations of signal processing-based, statistical, and sequence-based features as input. The results obtained on human genomic data show that HybPAS outperforms the well-tuned state-of-the-art Omni-PolyA models, reducing the classification error for different PAS hexamers by up to 57.35% for 10 out of 12 PAS types, with Omni-PolyA models being better for two PAS types. For the most frequent PAS types, ‘AATAAA’ and ‘ATTAAA’, HybPAS reduced the error rate by 35.14% and 34.48%, respectively. On average, HybPAS reduces the error by 30.29%. HybPAS is implemented partly in Python and in MATLAB available at https://github.com/EMANG-KAUST/PolyA_Prediction_LRM_DNN.

ARTICLE INFO

1. Introduction

Eukaryotic genomes have many important regions and signals, such as promoters, enhancers, transcription factor binding sites, translation start sites, splice sites, polyadenylation signals (PAS) and sites which define the gene regulatory landscape and demarcate gene boundaries. Many in silico prediction models have been developed based on machine learning (ML) and newer approaches based on deep learning (DL) aim at recognizing these regions and signals. For example, some ML and DL for promoter prediction models can be found in [1–8], for enhancer prediction [9–22], for transcription factor binding site prediction in [23–27], for translation initiation sites in [28–30], for poly(A) signals and sites in [29,31–38] and for splice sites in [39–42]. Currently, however, no existing computational model for predicting these signals and regions achieved satisfactory accuracy and continued interest to improve these models remains.

Signal processing approaches have been employed earlier in different bioinformatics and computational biology tasks (see [43–54]). At the time, signal processing-based methods did not receive sufficient attention for the prediction of genomic signals and regions as the other approaches generated better results. However, we are of the opinion that suitably used signal processing methods can help in developing more efficient prediction models in bioinformatics and computational biology.

In this study, we focus on the computational recognition of human PAS in genomic DNA. PAS are hexamer sequences located close to the
end of a transcript. There are numerous hexamers identical to PAS hexamers that have no links to the polyadenylation process. Different studies have developed models for the recognition of PAS signals by analyzing the sequences surrounding the PAS motifs [55–57] and achieved a moderate sensitivity and specificity. However, since the sequences surrounding the PAS motifs are poorly conserved, the accurate recognition of PAS motifs remains a very challenging task. More recent studies have used sophisticated methods to achieve better results. For instance, Omni-PolyA [58] uses a set of machine-learning algorithms in a tree-like structure (omnivariate decision tree) and a genetic algorithm for optimizing the parameters. Later, DeepGSR [29] and DeeReCt-PolyA [59] methods were proposed to further improve results and used convolutional neural network to automatically extract features from the DNA sequences.

In this study, we developed a novel method for predicting human poly(A) signals. We designed 12 different deep neural networks (DNNs) and logistic regression models (LRMs), one for predicting each of the 12 most common variants of poly(A) hexamers in the human genome. To this end, DNNs and LRM are combined in HybPAS, a hybrid PAS prediction model, based on signal processing, statistical, and compositional DNA sequence characteristics. We compared the results obtained by HybPAS to those reported by the state-of-the-art methods for PAS recognition, i.e., Omni-PolyA, DeepGSR, and DeeReCt-PolyA. From the results, we observed that HybPAS achieved an average accuracy of 91.22% and was able to improve accuracy of PAS predictions compared to performances reported in the literature, reducing the error by 30.29% compared to the published results of Omni-PolyA.

2. Results

The main contribution of our study is the development of a hybrid ML model, HybPAS, that comprises a separate prediction model be it a DNN or LRM, for each of the 12 most common PAS variants in the human genome. Input to DNN and LRM in HybPAS is a newly generated set of discriminant features that include signal processing-based, statistics-based, and position weight matrices (PWM)-based ones (see Material and Methods Section). To find the best performing classification model for each of the 12 PAS hexamers, we developed several ML/DL models (DNN, LRM, shallow Artificial Neural Network (ANN), decision tree (DT), Support Vector Machine (SVM)) and for each PAS motif we selected the model with the highest performance. In the case when more than one model had the same accuracy, as in the case for the ‘AATACA’ PAS hexamer for DNN and LRM, we selected the model that had more balanced sensitivity and specificity (DNN in this case). The models were derived by using a 5-fold cross-validation (CV) technique on the training set consisting of 30,006 PAS and 30,006 pseudo-PAS sequences (see Materials and Methods section). The performances of models that we report were determined using the independent testing set with 7510 PAS and 7510 pseudo-PAS sequences.

The testing data was balanced, as we considered the same number of PAS and pseudo-PAS sequences, allowing to present our results in terms of the error rate that is computed from the accuracy as shown in Table 1. Other statistical measures, i.e., sensitivity and specificity are also reported in Table 2 (see Material and Methods for the definitions of these measures). The results in Table 1 and Table 2 relate to the performances on the independent test set.

3. Discussion

We explored the possibility to combine different ML/DL models for more efficient PAS prediction in human genomic sequences. We found that combining two types of models, DNN and LRM, as implemented in the HybPAS model, significantly reduces the overall prediction error. We also found that a specific set of features used as input to DNN and LRM was very efficient in obtaining good prediction models. A prominent component of these features is based on signal processing. We used two signal processing-based methods, wavelet transform and Fourier transform. The choice of these transforms [53] has been motivated by their inherent properties. Discrete Wavelet Transform (DWT) has been used for different tasks in computational biology and bioinformatics [44,45]. For example, Maximal Overlap Discrete Wavelet Packet Transform (MODWPT) can effectively decompose a signal in terms of its time and frequency content enabling characterization of genomic sequences from a very different viewpoint. We also used Discrete Fourier Transform (DFT) [47,48], which may reveal hidden periodicities after transforming data from the time domain to the frequency domain space. For example, the DFT method has been used to study periodicities and repetitive elements in DNA sequences, genomes, and protein structures [59]. Because the DFT spectrum of a DNA sequence reflects the distribution of nucleotides, it does not only reveal the periodicities but also offers different views on the data in the frequency domain space. Due to the fact that the power spectrum conserves energy levels of a signal in the frequency domain according to Parseval's theorem [60], the DFT method has been employed in efficient similarity searching in sequence databases and thus has potential as an alignment-free method for predicting PAS. In addition to the signal processing-based features, statistics-based features were incorporated with the

![Table 1](image-url)

<table>
<thead>
<tr>
<th>Type</th>
<th>PAS motif</th>
<th>Testing dataset size (balanced)</th>
<th>Error rate (%) = 100-accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>LRM</td>
</tr>
<tr>
<td>Weak PAS</td>
<td>CATAAA</td>
<td>286</td>
<td>9.09</td>
</tr>
<tr>
<td></td>
<td>AATACA</td>
<td>140</td>
<td>8.57</td>
</tr>
<tr>
<td></td>
<td>AATATA</td>
<td>270</td>
<td>10.37</td>
</tr>
<tr>
<td></td>
<td>GATAAA</td>
<td>218</td>
<td>8.26</td>
</tr>
<tr>
<td></td>
<td>AGAAAG</td>
<td>104</td>
<td>14.42</td>
</tr>
<tr>
<td></td>
<td>TATAAA</td>
<td>630</td>
<td>8.10</td>
</tr>
<tr>
<td></td>
<td>AGTAAA</td>
<td>518</td>
<td>5.60</td>
</tr>
<tr>
<td></td>
<td>ACTAAA</td>
<td>124</td>
<td>14.52</td>
</tr>
<tr>
<td></td>
<td>AAAAAG</td>
<td>126</td>
<td>13.49</td>
</tr>
<tr>
<td></td>
<td>AATAGA</td>
<td>46</td>
<td>6.52</td>
</tr>
<tr>
<td>Strong PAS</td>
<td>ATTAAA</td>
<td>2796</td>
<td>9.41</td>
</tr>
<tr>
<td></td>
<td>AATAGA</td>
<td>9762</td>
<td>10.27</td>
</tr>
<tr>
<td>Weighted average</td>
<td></td>
<td></td>
<td>9.87</td>
</tr>
</tbody>
</table>

![Table 2](image-url)

<table>
<thead>
<tr>
<th>Type</th>
<th>PAS motif</th>
<th>Sensitivity</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LRM</td>
<td>DNN</td>
</tr>
<tr>
<td>PAS weak</td>
<td>CATAAA</td>
<td>88.11</td>
<td>91.61</td>
</tr>
<tr>
<td></td>
<td>AATACA</td>
<td>92.86</td>
<td>91.43</td>
</tr>
<tr>
<td></td>
<td>AATATA</td>
<td>88.89</td>
<td>92.59</td>
</tr>
<tr>
<td></td>
<td>GATAAA</td>
<td>90.83</td>
<td>90.83</td>
</tr>
<tr>
<td></td>
<td>AGAAAG</td>
<td>78.85</td>
<td>80.77</td>
</tr>
<tr>
<td></td>
<td>TATAAA</td>
<td>93.65</td>
<td>89.21</td>
</tr>
<tr>
<td></td>
<td>AGTAAA</td>
<td>93.05</td>
<td>87.64</td>
</tr>
<tr>
<td></td>
<td>ACTAAA</td>
<td>88.71</td>
<td>90.32</td>
</tr>
<tr>
<td></td>
<td>AAAAAG</td>
<td>84.13</td>
<td>90.48</td>
</tr>
<tr>
<td></td>
<td>AATAGA</td>
<td>91.30</td>
<td>90.30</td>
</tr>
<tr>
<td>PAS strong</td>
<td>ATTAAA</td>
<td>90.63</td>
<td>91.49</td>
</tr>
<tr>
<td></td>
<td>AATAGA</td>
<td>89.80</td>
<td>90.56</td>
</tr>
<tr>
<td>Weighted average</td>
<td></td>
<td>90.10</td>
<td>90.58</td>
</tr>
</tbody>
</table>
previous features to employ the stochasticity of the different sequences to discriminate between PAS and pseudo-PAS. We also generated three sets of position weight matrix PWM-based features. First, two mono-nucleotide PWMs were utilized to generate eight mono-nucleotide-based features. Second, two di-nucleotide PWMs were used to derive 32 di-nucleotide-based features. Finally, two tri-nucleotide PWMs were utilized to generate 128 tri-nucleotide-based features. A detailed description of the novel set of features is given in the Material and Methods section.

From Table 2, we observed that HybPAS achieved high sensitivity and specificity measures with more than 90% for 10 poly(A) motifs, while it fails to keep these performances for the poly(A) types AAAAG and AAGAAA. We note that DNN models appear the best for eight PAS types (including the two most frequent PAS hexamers), while LRM appeared best for the remaining four PAS types. We then asked how these results compare to the state-of-the-art methods. However, it is important to mention that in this study we used the GENCODE human release 28 and extracted 37,516 true PAS signals for the 12 considered PAS motifs, compared to the two datasets used in Omni-PolyA and DeeRecCT-PolyA, which consisted of 7370 and 18,786 PAS sequences. The comprehensive dataset in this study enable us to provide more data that may result in more robust models. Table 3 shows the reported 5-fold CV results by Omni-PolyA and DeeRecCT-PolyA tools on their dataset containing 17,786 true PAS sequences. However, we were unable to train the state-of-the-art models as the available source codes does not implement a train/test data split as the one considered in this study. From Table 3, we observed that HybPAS outperformed the well-tuned state-of-the-art model Omni-PolyA, reducing the classification error rate for 10 out of 12 different PAS hexamers by up to 57.35%, and on average across all 12 PAS hexamers the error rate is reduced by 30.29%. Similarly, HybPAS achieved smaller error rates for eight out of 12 PAS hexamers compared to DeeRecCT-PolyA, representing a relative reduction of the error rate of 8.25%. Finally, HybPAS reduced the error rate for AATAAA PAS by 29.47% compared to DeepGSR.

4. Materials and methods

4.1. Datasets

We used the human genome hg38 from GENCODE folder at EBI ftp server (ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_28/GRCh38.primary_assembly.genome.fa.gz).

Using GENCODE annotation for poly(A) (ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_humana_release_28/gencode.v28.polyAs.gff3.gz) we selected poly(A) signal annotation. Using bedtools-slop option, we retrieved regions extended 300 bp upstream and 300 bp downstream of the poly(A) hexamer. With the bedtools-getfasta option we extracted 606 bp fasta sequences from these regions. After eliminating duplicates, we obtained 37,516 presumed true functional poly (A) signal (PAS) sequences. Sequences from this set were denoted as positive.

4.1.1. Positive set (PAS sequences)

During the development of the models the average performances were considered as an estimate of the CV performance.

4.1.2. Negative set (pseudo-PAS sequences)

For the negative set, we looked for regions extended outside the region covering 1000 bp upstream and downstream of the positive poly (A) hexamer signal using bedtools-complement. There are no rules about what is best set of DNA sequences to be selected as ‘negative’ set for development of the PAS prediction models and each selection can be objected in some way. We selected sequences from this region as most of the alternative PAS are located in that region and many of the alternative PAS are pseudo PAS. Homer tool was used to find matches for the 12 most frequent human poly(A) variants. Since the number of matches was huge, sampling was used to select 37 516 pseudo-PAS sequences. Sampling was done from each chromosome proportionally to the chromosomes lengths and also to the expected frequency of the poly(A) variants. Out of these predictions, for each PAS hexamer, we selected the same number of pseudo-PAS sequences as in the positive set.

4.1.3. Training and testing sets

We selected randomly from each of the positive and negative datasets 20% of sequences for the independent test data. The testing set thus consisted of 15,020 sequences. The remaining data represented the training set and consisted of 60,012 sequences. Both datasets are balanced relative to the true PAS and pseudo-PAS sequences.

4.2. Performance measures

The performances in our study were based on accuracy, error rate, sensitivity and specificity measures. These are defined in Table 4.

To calculate the average performances based on each of the performance measures, and taking into account the different number of sequences corresponding to different PAS hexamers, we used weighted average performances defined as

$$wPM = \sum_{i=1}^{12} (PM_i \times PAS_i)/totalPAS$$

where $PM_i$ is a performance measure, $PAS_i$ is the number of specific PAS hexamer sequences, $totalPAS$ is the total number of true PAS sequences, and $wPM$ is the weighted performance measure $PM$.

During the development of the models the average performances were considered as an estimate of the CV performance.

Table 3

<table>
<thead>
<tr>
<th>Type</th>
<th>PAS motif</th>
<th>Error rate (%) = 100-Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Omni-PolyA</td>
</tr>
<tr>
<td>Weak PAS</td>
<td>CATAAA</td>
<td>14.39</td>
</tr>
<tr>
<td></td>
<td>AATACA</td>
<td>13.52</td>
</tr>
<tr>
<td></td>
<td>AATAA</td>
<td>13.41</td>
</tr>
<tr>
<td></td>
<td>GATAAA</td>
<td>8.48</td>
</tr>
<tr>
<td></td>
<td>AAGAAA</td>
<td>10.80</td>
</tr>
<tr>
<td></td>
<td>TATAAA</td>
<td>13.85</td>
</tr>
<tr>
<td></td>
<td>AGTAA</td>
<td>13.13</td>
</tr>
<tr>
<td></td>
<td>ACTAAA</td>
<td>14.49</td>
</tr>
<tr>
<td></td>
<td>AAAAG</td>
<td>4.87</td>
</tr>
<tr>
<td></td>
<td>AATTG</td>
<td>11.62</td>
</tr>
<tr>
<td>Strong PAS</td>
<td>ATTAAA</td>
<td>12.50</td>
</tr>
<tr>
<td></td>
<td>AATTG</td>
<td>14.02</td>
</tr>
<tr>
<td>Weighted average</td>
<td>12.43</td>
<td>9.57</td>
</tr>
</tbody>
</table>

The bold figures represent the best performance.

Table 4

<table>
<thead>
<tr>
<th>Statistical measure</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>$TP + TN$</td>
</tr>
<tr>
<td>Error rate</td>
<td>$\frac{TP + TN}{TP + FP + FN}$</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>$\frac{TP}{TP + FN}$</td>
</tr>
<tr>
<td>Specificity</td>
<td>$\frac{TN}{TN + FP}$</td>
</tr>
</tbody>
</table>

Methods xxx (xxxx) xxx–xxx
4.3. Generation of features for DNN and LRM

In this section, we first define the feature generation process for the three sets of features (signal processing-based, position weight matrix (PWM)-based, and statistics-based).

The DNA sequences are first transformed into numerical discrete sequences by replacing nucleotides A, C, G, and T with the numbers 1, 2, 3, and 4, respectively. Then the DFT and DWT were used to derive a set of signal processing features that represent the original DNA sequence as described later. The second type of features are statistics-based features, where we applied traditional statistical measures such as the mean, the median and the standard deviation to derive discrete sequences that represent the DNA sequences, as detailed later. Finally, the 600 nucleotide sequences surrounding the PAS and pseudo-PAS are used to extract mononucleotide PWM, dinucleotide PWM and trinucleotide PWM. Then, the PWMs are utilized to generate PWM-based features (scores).

4.4. Signal processing-based features

4.4.1. Wavelet-based features

We used MODWPT [61] to extract new features to discriminate PAS from pseudo-PAS motifs. The procedure of extracting wavelet-based features can be summarized in the following steps:

Step 1. Transform the DNA sequences to discrete signals. To extract the wavelet-based features, the following mapping was used [62]

\[
S(n) = \begin{cases} 
0.25X(n) = 'A' \\
0.5X(n) = 'C' \\
0.75X(n) = 'G' \\
1X(n) = 'T'
\end{cases}
\]

where \(X(n)\) represents the \(n^{th}\) nucleotide of the DNA sequence. In this way, DNA sequence \(X\) is encoded into discrete sequence \(S\).

Step 2. Transform the discrete sequences using MODWPT by choosing the type of orthogonal wavelet and the transformation level.

Decomposing the signal into the MODWPT bases at each level can be performed by a two-band (high-pass and low-pass) filter bank. Specifically, at the first level of the transform high-pass and low-pass filters are applied to the original discrete signals. At the second level, the same steps are repeated by filtering transformation coefficients obtained at the previous level, and so on. This process is repeated until the required level is reached.

We used Daubechies wavelets with three transformation levels as in [63]. In this case, the MODWPT produces eight different decomposition signals \(y_j(k)\) corresponding to each DNA sequence using the built-in MATLAB function modwpt.

Step 3. To extract the wavelet features, we compute total energy of the eight decomposition signals. The corresponding total energy of each decomposition signal \(E_j\) can be obtained as follows:

\[
E_j = \sum_{k=1}^{N} |y_j(k)|^2
\]

where \(j = 1, 2, \ldots, 8\) and \(N\) is the length of the decomposition signals.

The following algorithm summarizes the procedure to generate wavelet-based features.

**Algorithm 1**: Wavelet-Based Features Extraction

```
while i ≤ Ns do % Ns is the size of the dataset, i = 1 initially
    \{ Pick i-th DNA sequence Xi from the dataset \
    S_i = X_i % Convert the DNA sequence into numerical sequence \n    S → \{ % At level 3 by Daubechies wavelets \n        Get the reconstruct signals, \n        \text{Energy of reconstruct signals } y_j \} % The size of Fw is \(i! \times 1\) \n        i = i + 1 \}
    Return Fw % The output feature matrix is of size\(Ns \times 8\)

4.4.2. Frequency-based features

The feature extraction procedure using DFT can be summarized in the following three steps:

Step 1. Convert the DNA sequence into a discrete signal. For this, the numerical mapping described in Eq. (1) is applied to create the frequency-based features.

Step 2. Transform the discrete signal using DFT into its frequency domain using the following equation [64]:

\[
Y(k) = \sum_{n=1}^{N} S(n)e^{-\frac{2\pi in}{N}}
\]

where \(N\) represents the length of the PAS and pseudo-PAS sequences.

Step 3. Generate four features from the transformed signals. The first two features correspond to the dominant frequency for each signal and the corresponding magnitude at that frequency. The second two features are the mean normalized angular frequency and the median normalized angular frequency of the power spectrum of each signal. To optimize the computation time of evaluating these features, we used the built-in MATLAB functions \text{meanfreq} and \text{medfreq} to calculate the mean and median normalized angular frequencies of the power spectrum of the signal. Algorithm 2 summarizes the features extraction using DFT.

**Algorithm 2**: Frequency-Based Features Extraction

```
Input: Xi: the input DNA sequences 
Output: FF: Frequency-based features 
while i ≤ Ns do % Ns is the size of the dataset, i = 1 initially 
    \{ Pick i-th DNA sequence Xi from the dataset \n    S_i = X_i % Convert the DNA sequence into numerical sequence \n    Y = Discrete Fourier Transform (DFT) of S_i \
    Ff = Power Spectrum of Y \{i.e., \|y(k)\|\} \n    \text{Dominant frequency, corresponding magnitude, meanfreq, medfreq} of Ff \n    i = i + 1 \}
    Return FF % The output feature matrix is of size\(Ns \times 4\)

4.5. Statistics-based features

Statistical analysis is a frequently used method for DNA feature generation. We developed a number of statistical features that can help discriminate between the PAS and pseudo-PAS.

Step 1. Converting PAS and pseudo-PAS sequences into discrete sequences via the electron-ion interaction potential (EIIP) mapping [65,66]:

\[
S(n) = \begin{cases} 
0.1260X(n) = 'A' \\
0.1340X(n) = 'C' \\
0.0806X(n) = 'G' \\
0.1335X(n) = 'T'
\end{cases}
\]

Step 2. Generate features from the discrete sequence, namely: the mean value, standard deviation, median value, mean absolute
deviation, 25th percentile, 75th percentile, signal interquartile, skewness and kurtosis. These numerical values were considered as representative features for every PAS and pseudo-PAS sequence. In addition, the number of occurrences for the most discriminative mono-nucleotide and di-nucleotide within each PAS and pseudo-PAS sequence were calculated to serve as additional statistical features. Finally, three statistics-based features were computed for the PAS and pseudo-PAS discrete sequence, namely: 1) the root mean square value, 2) the root sum squared value, and 3) the ratio between the largest absolute value and the root mean square value. Algorithm 3 summarizes the generation of statistics-based features.

Algorithm 3 Statistics-Based Features Extraction

Input: X: the input DNA sequences
Output: Fs: Statistics-based features
while i ≤ Nf do % Nf is the size of dataset, i = 1 initially
  (Pick i-th DNA sequence Xi from the dataset
  Sf = Xi % Convert the DNA sequence into numerical sequence
  fS = Fs statistical features of
  fi = fi + 1
End
Return Fs % The output feature matrix is of sizeNf × 25

4.6. PWM-based features

In several studies related to poly(A) prediction [57,67], PWMs were commonly used to indicate the significance of each position along the upstream and downstream of PAS and pseudo-PAS. Two PWMs can be generated, where the first matrix is constructed from the training PAS samples and the second matrix is constructed from the training pseudo-PAS samples. Then, these two PWMs are used to generate scores for every PAS and pseudo-PAS by matching every sequence to these two PWMs. The PWMs can be generated for different nucleotide k-mers. For instance, a PWM that indicates the probability of every dinucleotide appearance along the upstream and downstream of PAS and pseudo-PAS can be constructed.

In this study, we use the PWM approach to generate a set of PWM-based features. The main idea is to decompose every sequence into binary sequences reflecting a specific pattern in this sequence. Each binary sequence describes the existence of a specific oligonucleotide: Mono-nucleotides (A, C, G, T), di-nucleotides (AA, AC, AG, AT, ...), tri-nucleotides (AAA, AAC, AAT, AAG, ...). Fig. 1 shows the different k-mer encodings.

For instance, the binary sequence of the mono-nucleotides Sa is defined as follows:

\[
S_a(n) = \begin{cases} 
1, & \text{if} n\text{th nucleotide is an} \text{A'} \\
0, & \text{elsewhere} 
\end{cases} 
\]

Similarly, the binary sequence of the tri-nucleotides STTA is defined as follows:

\[
S_{TTA}(n) = \begin{cases} 
1, & \text{if} n\text{th tri-nucleotide is} \text{TTA'} \\
0, & \text{elsewhere} 
\end{cases} 
\]

The binary mapping methodology of a nucleotides patterns of order k (i.e., k = 1 for mono-nucleotides and k = 2 for di-nucleotides) is summarized in Algorithm 4.

Algorithm 4 k-mer Pattern Extraction

Input : X: the input DNA sequences, k: k-mer order
Output : kmer: the patterns k-nucleotide
\( i \leftarrow 1 \)
\( N = \text{size}(X) \)
\( C_{ki}\text{=} \text{All possible oligonucleotides composed of k nucleotides} \)
For each oligonucleotides do
  \( C_{ki}(j) \text{= get a new oligonucleotides} \)
  If \( X[j+i+k-1] = C_{ki}(j) \text{ Then} \)
    \( kmer(i,j) \leftarrow 1 \)
  Else
    \( kmer(i,j) \leftarrow 0 \)
End
End
Return kmer

Finally, the extracted k-mer patterns were used to generate the PWM-based features. The new PWM-based feature extraction is explained in the following subsection.

4.6.1. Mono-nucleotide PWM-based features

The frequencies of the four mono-nucleotides (A, C, G, and T) are used to derive two mono-nucleotide PWMs from the training data to represent the characteristics of PAS and pseudo-PAS sequences separately. The mono-nucleotide PWMs are then utilized to compute a score (Eqs. (7)-(10)) for every mono-nucleotide binary sequence (every DNA sequence can be decomposed into four mono-binary sequences, where the first indicates the pattern of nucleotide A, the second indicates the pattern of nucleotide C, the third indicates the pattern of G, and finally the last indicates the pattern of nucleotide T). These scores indicate the likelihood of the sequence to be a true PAS and a pseudo-PAS. Following this strategy, four PAS scores and four pseudo-PAS scores are computed for each DNA sequence in the dataset and are calculated as follows: let \( S(j) \) be a DNA sequence of length \( N \), and \( P(p_j) \) be a mono-nucleotide PWM of \( N \) columns and four rows (for each mono-nucleotide). The four PAS-scores and four pseudo-PAS-scores are calculated as follows:

\[
[PAS \lor \text{pseudoPAS}]_A = \sum_{i=1}^{4} \sum_{j=1}^{N} p_j S_a(j) 
\]

\[
[PAS \lor \text{pseudoPAS}]_C = \sum_{i=1}^{4} \sum_{j=1}^{N} p_j S_c(j) 
\]

\[
[PAS \lor \text{pseudoPAS}]_G = \sum_{i=1}^{4} \sum_{j=1}^{N} p_j S_g(j) 
\]

\[
[PAS \lor \text{pseudoPAS}]_T = \sum_{i=1}^{4} \sum_{j=1}^{N} p_j S_t(j) 
\]

The total number of features extracted following this approach is eight.

4.6.2. Di-nucleotide PWM-based features

Similar to the mono-nucleotide based features, the rate of occurrences of the 16 di-nucleotide combinations (AA, AC, AG, AT, ..., AAG, ACG, ..., CAT, ...) was used to derive di-nucleotides PWMs from the training data. Two PWMs can then be generated for every di-nucleotide: PAS PWMs and pseudo-PAS PWMs. The PWMs can be generated for di-nucleotides of PAS and pseudo-PAS. Two PWMs can be generated for every di-nucleotide k-mer.

For instance, the binary sequence of the mono-nucleotides SAA is defined as follows:

\[
S_{AA}(n) = \begin{cases} 
1, & \text{if} n\text{th di-nucleotide is} \text{AA'} \\
0, & \text{elsewhere} 
\end{cases} 
\]

Similarly, the binary sequence of the tri-nucleotides STTA is defined as follows:

\[
S_{TTA}(n) = \begin{cases} 
1, & \text{if} n\text{th tri-nucleotide is} \text{TTA'} \\
0, & \text{elsewhere} 
\end{cases} 
\]

The binary mapping methodology of a nucleotides patterns of order k (i.e., k = 1 for mono-nucleotides and k = 2 for di-nucleotides) is summarized in Algorithm 4.

Algorithm 4 k-mer Pattern Extraction

Input : X: the input DNA sequences, k: k-mer order
Output : kmer: the patterns k-nucleotide
\( i \leftarrow 1 \)
\( N = \text{size}(X) \)
\( C_{ki}\text{=} \text{All possible oligonucleotides composed of k nucleotides} \)
For each oligonucleotides do
  \( C_{ki}(j) \text{= get a new oligonucleotides} \)
  If \( X[j+i+k-1] = C_{ki}(j) \text{ Then} \)
    \( kmer(i,j) \leftarrow 1 \)
  Else
    \( kmer(i,j) \leftarrow 0 \)
End
End
Return kmer

Finally, the extracted k-mer patterns were used to generate the PWM-based features. The new PWM-based feature extraction is explained in the following subsection.
be a di-nucleotide PWM of length $p = \cdots$ of length $N$.

be a DNA sequence of length $t$.

are computed as follows:

$$\sum_{i=1}^{16} \sum_{j=1}^{N-1} R_i S_{\text{mel}}(j)$$

where $d_{\text{mel}} = \{AA, AT, AC, AG, etc\}$.

The total number of features generated following this technique is 32.

4.6.3. Tri-nucleotide PWM-based features

Following similar steps to the mono-nucleotide and di-nucleotide-based feature generation, the rate of occurrences of the 64 tri-nucleotide combinations (AAA, AAC, AAG, AAT, ACA, ...) are computed to derive tri-nucleotide PWMs from the training data to represent the characteristics of both PAS and pseudo-PAS sequences separately. Then, the tri-nucleotide PWMs are utilized to calculate a score for every tri-nucleotide binary sequence (every DNA sequence can be decomposed into 64 binary sequences). Based on this approach, 64 PAS scores and 64 pseudo-PAS scores are computed for each DNA sequence in the dataset as follows: let $S(j)$ be a DNA sequence of length $N$, and $P(p_j)$ be a di-nucleotide PWM of length $N-1$ columns and 16 rows (for each di-nucleotide combination). The 16 PAS-scores and 16 pseudo-PAS-scores are computed as follows:

$$\text{PAS}_d = \sum_{i=1}^{16} \sum_{j=1}^{N-2} P_{ij} S_{\text{mel}}(j)$$

$$\text{PAS}_p = \sum_{i=1}^{16} \sum_{j=1}^{N-2} P_{ij} S_{\text{mel}}(j)$$

where $\text{mel} = \{AAA, AAT, AAC, AAG, ACC, ACG \ldots etc\}$.

The total number of features generated following this technique is 128.

4.7. Deep learning model

We used a DNN, a neural network with two hidden layers that learn more abstracted features as the signal progresses deeper through the layers. We designed a DNN model to predict if a sequence contains PAS or pseudo-PAS. Our DNN uses the same set of features as LRM (see Fig. 2). The output layer consists of two Soft-Max nodes.

The DNN models were constructed using ‘keras’ library and wrapped in Keras Classifier wrapper [68] in Python. We created a parameter search space to evaluate different configurations for DNN models using grid-search algorithm [69] including batch size, training epochs, dropout regularization rate, number of hidden layers, and size of each layer (see Table 5). When the best value for a parameter was found from the first run of the grid-search algorithm, we then evaluated more precisely the surrounding values of the parameter to get an optimized performance. Then, we selected the parameter combination that provided the best estimated performance on the training set in a 5-fold CV setting and out of the five models we selected the one with the best performance.

4.8. Other ML models

In order to find the best performing model for each of the PAS variants, we evaluated several ML models, namely: DNN (as an ANN with two or more hidden layers), LRM, DT, ANN and SVM. Each of these models was tuned by using a 5-fold CV on the training dataset. Then, we selected the best performing model for each of the PAS variants and such model was used in the HybPAS. LRM, DT, ANN, and SVM were implemented in MATLAB. All models were subjected to a model optimization process for obtaining optimized structure and parameter values. A general development framework is depicted in Fig. 3.
5. Author's contributions

FA and VBB conceived and designed the experiments. FA, AC, XG, SA and AMM performed the experiments and analyzed the data. FA, AC, XG, SA, AMM, BRJ, TMLK and VBB contributed reagents/materials/analysis tools. FA, AMM, ME, CVN, MU, TMLK and VBB wrote the paper. All authors read and approved the final manuscript.

6. Competing interests

The authors have declared that no competing interests exist.

Appendix A

Resulting parameters from grid-search for each PAS DNN model

<table>
<thead>
<tr>
<th>PAS</th>
<th>Hidden Layers</th>
<th>Node size</th>
<th>Activation</th>
<th>Kernel initializer</th>
<th>Kernel regularizer rate</th>
<th>Gaussian Noise</th>
<th>Dropout</th>
<th>Optimizers</th>
<th>Learning rate</th>
<th>Batch size</th>
<th>Epochs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATAAA</td>
<td>2</td>
<td>32</td>
<td>'relu'</td>
<td>'uniform'</td>
<td>0.001</td>
<td>0.1</td>
<td>0.1</td>
<td>'Nadam'</td>
<td>0.001</td>
<td>32</td>
<td>70</td>
</tr>
<tr>
<td>AATACA</td>
<td>2</td>
<td>32</td>
<td>'relu'</td>
<td>'uniform'</td>
<td>0.001</td>
<td>0.1</td>
<td>0.1</td>
<td>'Nadam'</td>
<td>0.001</td>
<td>32</td>
<td>70</td>
</tr>
<tr>
<td>AATATA</td>
<td>2</td>
<td>38</td>
<td>'relu'</td>
<td>'uniform'</td>
<td>0.001</td>
<td>0.1</td>
<td>0.1</td>
<td>'Nadam'</td>
<td>0.001</td>
<td>64</td>
<td>200</td>
</tr>
<tr>
<td>GATAAA</td>
<td>2</td>
<td>32</td>
<td>'relu'</td>
<td>'uniform'</td>
<td>0.001</td>
<td>0.1</td>
<td>0.1</td>
<td>'Nadam'</td>
<td>0.001</td>
<td>64</td>
<td>150</td>
</tr>
<tr>
<td>AGAAA</td>
<td>2</td>
<td>32</td>
<td>'relu'</td>
<td>'uniform'</td>
<td>0.001</td>
<td>0.1</td>
<td>0.1</td>
<td>'Nadam'</td>
<td>0.001</td>
<td>16</td>
<td>50</td>
</tr>
<tr>
<td>TATAAA</td>
<td>2</td>
<td>32</td>
<td>'relu'</td>
<td>'uniform'</td>
<td>0.001</td>
<td>0.1</td>
<td>0.1</td>
<td>'Nadam'</td>
<td>0.001</td>
<td>128</td>
<td>200</td>
</tr>
<tr>
<td>AGTAAA</td>
<td>2</td>
<td>32</td>
<td>'relu'</td>
<td>'uniform'</td>
<td>0.001</td>
<td>0.1</td>
<td>0.1</td>
<td>'Nadam'</td>
<td>0.001</td>
<td>128</td>
<td>200</td>
</tr>
<tr>
<td>ACTAAA</td>
<td>2</td>
<td>32</td>
<td>'relu'</td>
<td>'uniform'</td>
<td>0.001</td>
<td>0.1</td>
<td>0.1</td>
<td>'Nadam'</td>
<td>0.001</td>
<td>128</td>
<td>200</td>
</tr>
<tr>
<td>AAAAAA</td>
<td>2</td>
<td>32</td>
<td>'relu'</td>
<td>'uniform'</td>
<td>0.001</td>
<td>0.1</td>
<td>0.1</td>
<td>'Nadam'</td>
<td>0.001</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>ATATAA</td>
<td>2</td>
<td>32</td>
<td>'relu'</td>
<td>'uniform'</td>
<td>0.001</td>
<td>0.1</td>
<td>0.1</td>
<td>'Nadam'</td>
<td>0.001</td>
<td>32</td>
<td>30</td>
</tr>
<tr>
<td>AATAAA</td>
<td>2</td>
<td>38</td>
<td>'relu'</td>
<td>'uniform'</td>
<td>0.001</td>
<td>0.1</td>
<td>0.1</td>
<td>'Nadam'</td>
<td>0.001</td>
<td>512</td>
<td>300</td>
</tr>
</tbody>
</table>

Fig. 3. Development framework to obtain the best predictive model for an individual PAS variant.

Acknowledgments

This work has been supported by the King Abdullah University of Science and Technology (KAUST) Base Research Fund (BAS/1/1606-01-01) to VBB, (BAS/1/1627-01-01) to TMLK, and KAUST Office of Sponsored Research (OSR) under Awards No CARF – FCC/1/1976-17-01.
Appendix B
Deep learning model summary

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>dense_1 (Dense)</td>
<td>(None, 32)</td>
</tr>
<tr>
<td>gaussian_noise_1 (Gaussian noise)</td>
<td>(None, 32)</td>
</tr>
<tr>
<td>dense_2 (Dense)</td>
<td>(None, 32)</td>
</tr>
<tr>
<td>gaussian_noise_2 (Gaussian noise)</td>
<td>(None, 32)</td>
</tr>
<tr>
<td>dropout_1 (Dropout)</td>
<td>(None, 32)</td>
</tr>
<tr>
<td>dense_3 (Dense)</td>
<td>(None, 1)</td>
</tr>
</tbody>
</table>

Appendix C. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ymeth.2019.04.001.

References

resonance (FT-ICR) and orbitrap based high resolution mass spectrometry in metabolomics and lipidomics, Int J Mol Sci 17 (6) (2016) 816.


