WAM: A Weight Array Model for Prediction of Eukaryotic Genetic Splice Sites

1st Ziwen Zhao
College of Life Science and Technology
Huazhong University of Science and Technology
Wuhan, China
justn582@gmail.com

***Abstract*—Splice sites are a vitally important gene sequence pattern amongst the functional sites inside a eukaryotic gene. However, splice site prediction does not come easy thanks to the extreme complexity of human genome. In this paper, an optimized frequency-based method to predict eukaryotic genetic splice site patterns with the Weight Array Model (WAM) is proposed, which is a feasible lightweight computational approach for gene functional site finding, in order to deal with splice site predictions. We prove its accuracy and high efficiency by comparison studies on the renowned Kulp & Reese human genome dataset, during which we achieve excellent results on several different metrics. The source code is available on GitHub and can be obtained from https://github.com/Newiz430/SplicePredictor.**

***Index Terms*—splice site, weight array, statistics, frequency-based, probability matrix**

I. Introduction

GENE finding by computational methodologies, the foundation for all further investigation in functional genomics, has attracted considerable research attention since the 20th century [1]. With the thriving of functional genomics after the completion of Human Genome Project (HGP), functions of the elements of eukaryotic gene sequences were beginning to surface. Researchers came to realize that DNA sequences, other than genes, contain a huge amount of information, most of which is correlated with the structural features of nucleic acids and in general determines the DNA - protein, or DNA - RNA interactions. Such information is mostly provided by a variety of functional sites (i.e. sequence motif). The splice sites are a vitally important eukaryotic gene sequence pattern amongst all these sites. As terminal points of RNA splicing, splice sites label the junction of transcript exons and introns, assisting biologists in identifying and positioning the coding sequence within a gene. Splicing, itself, also influences the structure and function of genes, which makes genes more "modular", allowing new combinations of exons to be created during evolution. Furthermore, new exons can be inserted into old introns, creating new proteins without disrupting the function of the old gene [2]. Hence the discovery and prediction of splice sites are of great significance for genetic selective expression research.

A splice site locates in the edge of an intron, including a donor site (5' end of the intron) and an acceptor site (3' end of the intron). As a typical sequence motif, the donor site includes an almost invariant sequence GU at the 5' end of

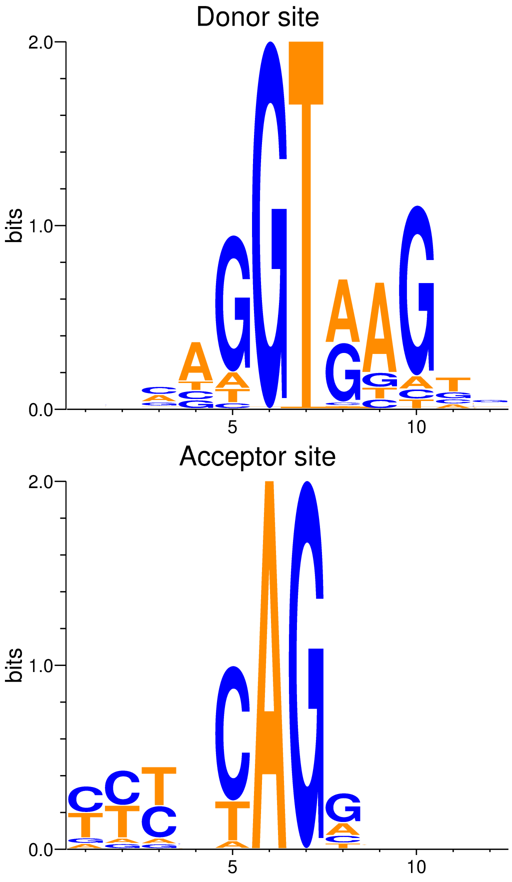

Fig. 1. Base distribution represented as sequence logos [3] for eukaryotic gene splice sites (5 upstream sites, 2 conservative sites & 5 downstream sites): **top**, donor site; **bottom**, acceptor site. The most conserved sites is revealed within logo pictures: for donor sites, GT (6, 7); acceptor sites, AG (6, 7). Despite these two, distribution of adjacent sites (5, 8, 9, 10 for the donor, 5 for the acceptor) appear to have some consistency, too. The polypyrimidine tract can also be observed upstream of the acceptor site (1, 2, 3). The information content at a certain point R_i is given by $R_i = \log_2 4 - (H_i + e_n)$ where $H_i = -\sum_{i=1}^{4} P_i(\log_2 P_i)$ is the Shannon entropy [4], using bit as the basic unit. Higher the bits, higher the relative frequency of that base [5].

the intron, within a larger, less highly conserved region. The splice acceptor site at the 3' end terminates the intron with an almost invariant AG sequence [6]. Some sections of the intron foretell the positions of these two sites. For example, a fragment of sequence upstream of the acceptor consisting of cytosines and thymines, which is called a polypyrimidine tract [7]. Fig. 1 shows the base distributions adjacent of the splice donor sites and acceptor sites.

As a matter of fact, accurate prediction does not come easy thanks to the extreme complexity of human genome. On one hand, the number and length of exons and introns in a eukaryotic gene exhibit great uncertainty. One eukaryotic gene contains 5.48 exons with 30 - 36 bps long on average. While the longest exon in the human genome is 11555 bp long, several exons have been found to be only 2 bp long [8]. On the other, the existence of alternate splicing make it harder to predict [6]. In this paper, we apply a Bayesian method for gene functional site finding to predict eukaryotic gene splice sites, and prove its feasibility.

A. Related Work

Several typical computational methods that attempt to predict eukaryotic splice sites from unknown gene sequences (i.e. *ab initio* prediction) have been proposed previously.

Frequency-based methods count the nucleotide frequencies of each site via multiple sequence alignment, etc. and work out the log-odds ratio to compare and find conservative sections in the alignment results. Rodger, et al. (1983) [9] proposed a computational model using a weight matrix to represent each type of recognition sequence. A weight matrix is a two dimensional array of values that represent the score for finding each of the possible sequence characters at each position in the target signal. The Weight Matrix Model (WMM) now becomes deprecated owing to its poor accuracy and its independence assumption, that is, WMM only takes point-wise base distribution into consideration, regardless of the potential dependence between adjacent points which is more conformable to the realistic situations.

Bayesian methods are ones that consider long range dependency among nucleotide sequences. Chen, et al. (2005) [10] proposed a method using dependency graphs and its expanded form - Bayesian Network to fully capture the intrinsic interdependency between base positions in a splice site. The entire model, including the DAG structures and conditional probabilities, is learned *ab initio* from a training set. However, with astronomical computation complexity and model training difficulty, the performance of Bayesian models is not in keeping with them.

Supervised learning methods learn a model from existing training set which is able to identify the most effective pattern automatically. Duan, et al. (2008) [11] developed the support vector machine (SVM) for position-specific residue preference feature prediction which determines the second structure of double helices. Ryen et al. (2008) [12] introduced the artificial neural network (ANN) in this area and trained the model with backpropagation, which can make predictions without prior knowledge of any sensor signals. Accurate and efficient learning approaches they are, supervised learning methods are heavily dependent on the mass and quality of training sets. Models may not be improved and a computational resource waste may happen when an unbalanced dataset or one with too many noises is provided. For SVMs, kernel function selection is a tricky problem, and neural networks acquire a suitable framework and initial hyperparameters.

B. Contributions

In this work, we propose an optimized frequency-based method to predict splice site patterns with the weight array model. At bottom, a weight array method continues to extract splice signals, count the frequencies of nucleotides and fill the matrices, identical with WMM. What we can use to distinguish WAM from WMM is that WAM takes into account the correspondence between current position and an adjoining position, which we certify conducive to promote accuracy of splice site prediction. Our contributions are listed as follows.

- We implemented the weight array model by Python using the given KR set and estimated its performance on the BG set, referring to the existing experiment by Zhang et al. (1993) [13].
- We did a comparison study between WMM and WAM model to prove the superiority of our model.
- We applied our model on the prediction of both donor splice sites and acceptor splice sites.

II. Methods

Our method is illustrated in Fig. 2, which mainly contains two parts. The statistics of base pair distribution is carried out at the “training” step, and sequences of the testing set are scored by probability matrices at the “predicting” step. See below for more detailed presentation.

A. Hypothesis

WAM is suitable for predicting work only when the two assumptions below stand. Firstly, we assume conservation around the functional splice sites through the entire experiment. Additionally, we consider the intrinsic interdependency only exists between adjacent sites. As for data, we assume that everything about the splice site pattern remained unknown (including the obligatory sites GT / AG) until we dug them out, in order to guarantee the generality of our model, since our aim is to make it possible to branch out to other unidentified functional sites.

B. Data Extraction

Dash et al. (2001) [14] found in predicting splice sites by a Bayesian network that it achieves better performance when both of the upstream and downstream feature lengths are greater than 15. With a view to simplifying model and decreasing the computation, we choose 5 upstream sites and 7 downstream sites of intron / exon junctions to form 12 nt long signal sequences from the primary training set. We abandon sequences containing ambiguous bases, whose correspondence with the splice sites we consider inapparent. The training set

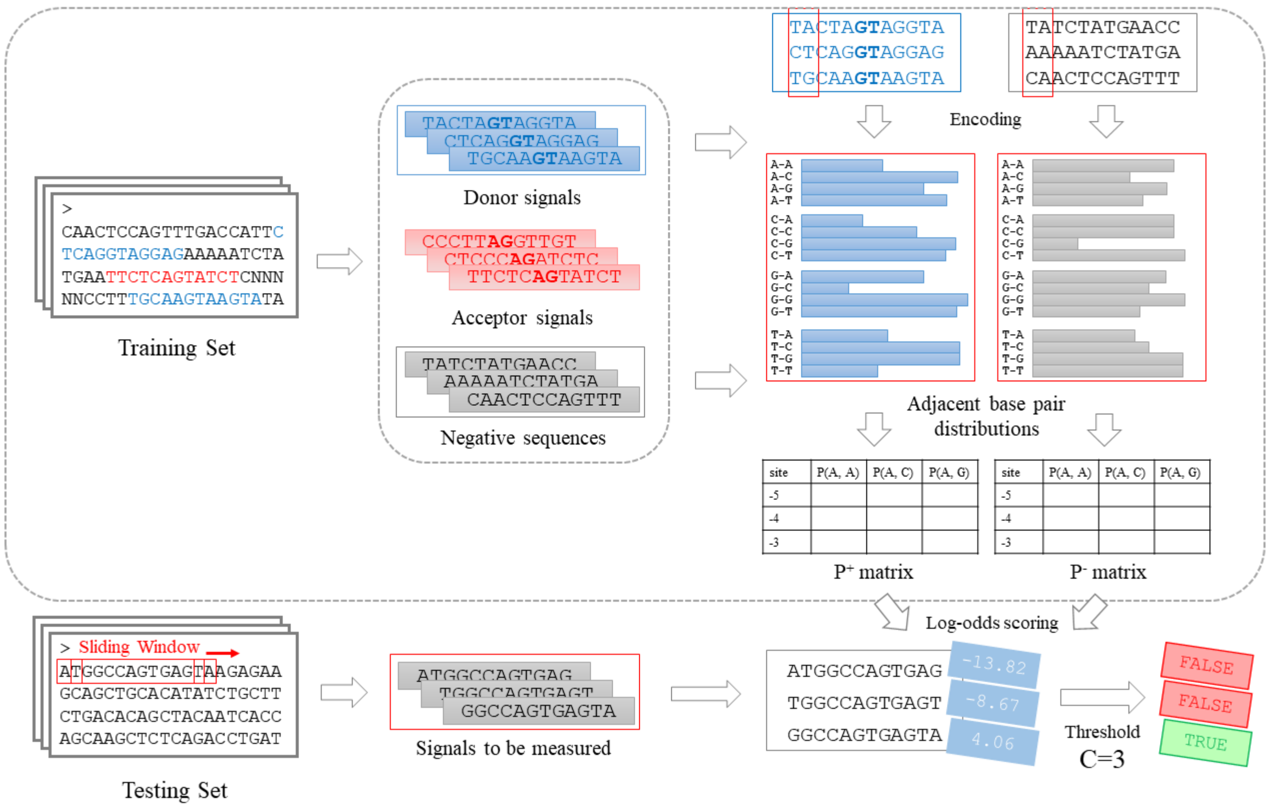

Fig. 2. Overall architecture of the WAM splice predictor. We use the training set to create two matrices by these steps: extracting positive site signals & randomly choosing negative signals, encoding & counting the bases for each position, calculating distribution probabilities, filling the matrices and saving for future predictions. We use P matrices to predict unknown signals by these steps: extracting testing sequences of same length by window-sliding, scoring the sequences position-wise with obtained probabilities, calculating the binary log-odds scores, and comparing them with a given threshold to make final judgment.

provides 2,381 donor signals and 2,381 acceptor signals. As for negative samples, Chen, et al. [10] used pseudo splice sites as false data, extracted by searching for negative sample sequences with $P_{+1}P_{+2}$ = GU / AG whereas, according to the splice site hypothesis above, We randomly selected about 5,000 sites in sections which do not intersect with all donor and acceptor sites, and combined with positive ones to get actual training dataset, the positive-negative ratio of which is about 1:2. Additionally, we export sequences with the same length by window sliding from the primal testing set and build the actual testing set in a positive-negative ratio of 1:20.

C. Construction of Probability Matrices

For convenient computing, we encode the extracted sequences. We use 0, 1, 2, and 3 to represent bases A, C, G, T which indicates the position in the P matrices of each base. For positive samples with the length of λ, we create a λ * 4 * 4 probability matrix P^+, in which each position (N_j, N_{j+1}) of the ith 4 * 4 submatrix stores the conditional probability $P^+[i, N_j, N_{j+1}]$ which denotes the probability of N_j at the current position j if the nucleotide at position $j+1$ is N_{j+1}. Each probability is calculated by counting the frequencies of every situations $\text{freq}(i, N_j, N_{j+1})$ in the positive samples and expressed as

$$P^+[i, N_j, N_{j+1}] = \frac{\text{freq}(i, N_j, N_{j+1})}{\sum\limits_{N_{j+1}} \text{freq}(i-1, N_j, N_{j+1})} \tag{1}$$

with

$$i = 2, 3, \cdots, \lambda, \quad j = 1, 2, 3, 4.$$

The conditional distribution probabilities of negative samples $P^-[i, N_j, N_{j+1}]$ are defined likewise:

$$P^-[i, N_j, N_{j+1}] = \frac{\text{freq}(i, N_j, N_{j+1})}{\sum\limits_{N_{j+1}} \text{freq}(i-1, N_j, N_{j+1})}. \tag{2}$$

Then we put the base distribution probabilities of the first position $P^+[1, N]$ into 1D-arrays P_0^+ and P_0^- :

$$P_0^+[1, N_j] = \frac{\text{freq}(1, N_j)}{\sum\limits_{N_j} \text{freq}(1, N_j)} \tag{3}$$

$$P_0^-[1, N_j] = \frac{\text{freq}(1, N_j)}{\sum\limits_{N_j} \text{freq}(1, N_j)} \tag{4}$$

The base distribution of positive samples nicely dovetails with the currently accepted splice site pattern, as shown in Fig. 3, which further proves the validity of our proposed method: the conditional probability matrices are capable of reflecting special functional signals in nucleotide sequences.

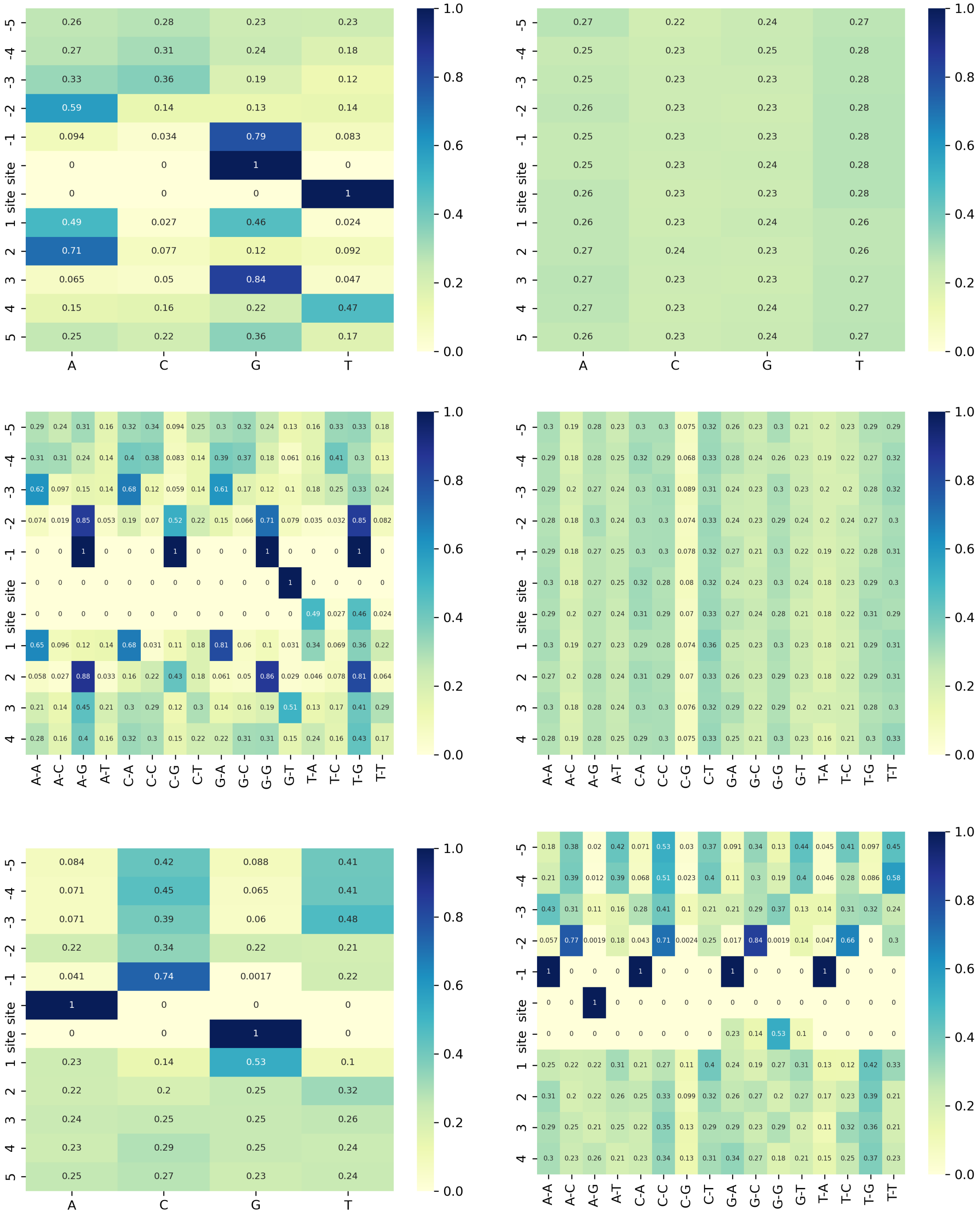

Fig. 3. Base distribution represented as heatmaps for donor splice sites. **Top**: single base distribution of positive donor & negative samples. Aside from two conservative sites GT (labeled with "site"), Positions (-2, -1, +1, +2, +3) shows additional conservation of adjacent sites of a splice site. For example, it is attractive that the odds reach 84% of position +3 being a guanine. **Center**: adjacent base distribution of positive donor & negative samples which shows the correspondence between bases more clearly compared with the single base heatmaps. **Bottom**: base distribution of positive acceptor samples. The polypyrimidine tract can be observed (-5, -4, -3) likewise, with the sum of $P(C)$ and $P(T)$ exceeds 80%, which is part of the powerful acceptor signal.

D. Prediction

We apply the P matrices above to make judgment of splice sites in unknown sequences. To be specific, we set a sliding window to extract sequences of every available position of the testing set, and score them with a scoring function $S(X)$. For a binary classifier, the scoring formula is a log-odds ratio as:

$$S(X) = \ln\frac{P^+(X)}{P^-(X)} = \ln\frac{P_0^+[1,N_1]}{P_0^-[1,N_1]} + \sum_{i=2}^{\lambda}\ln\frac{P^+[i,N_{i-1},N_i]}{P^-[i,N_{i-1},N_i]} \quad (5)$$

Since there is zeros in the probability matrices, we set $P = 10^{-6}$ to avoid division-by-zero error which, in the meantime, guarantees a higher penalty for a sequence including a $P^+(X) = 0$ site, and vice versa. By this way, we score all sequence txts of the testing set and get the score distributions of donor and acceptor predicting models, as is shown in Fig. 4. The distributions are conducive to the selection of thresholds at the following steps.

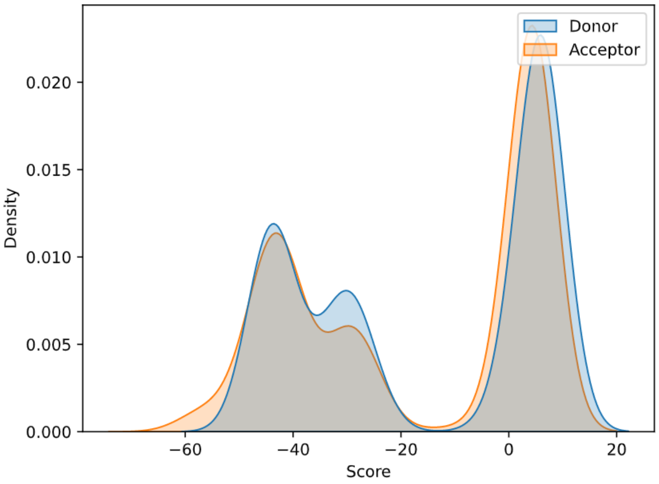

Fig. 4. Density of sequence scores predicted separately by donor (blue) and acceptor (coral) WAMs. Compared to the histograms Rodger [9] provided earlier, there are a lot of low scores in our testing consequence. This is caused by the sliding selection of testing sequences without filtering pseudo ones with the conservative sites on purpose. It can also be observed that low and high scores are widely separated, hence can be easily distinguished. Comparing donor and acceptor scores latitudinally, we can see difference between the sequence pattern of donor and acceptor sites, but it is small enough to allow the acceptor prediction without changing the framework of our approach.

Transformation from scores to predicting results needs the comparison. We can filter the true positive sites we need from batches of scores by taking different thresholds. It is necessary to exercise caution in selecting the threshold C. A large C will exclude potential positive sites, while a small C misclassifies negative sites as positive. Hence an appropriate threshold is a tradeoff based on the specificity and sensitivity of a model. For the threshold optima selection, see IV for details.

III. Experiments

A. Data

We conduct our experiment on the eukaryotic gene sequence dataset Kulp & Reese [15] and Burset & Guigo [16]. Human genome sequence dataset Kulp & Reese (KR set) is used as training set which contains 462 sequence text files, each records the name, length, CDS terminal points and the segment. 2,381 donor sites and 2,381 acceptor sites are extracted from the KR set. Vertebrate genome dataset Burset & Guigo (BG set) is used as testing set which contains 570 sequence text files with a similar format, except for a lack of the sequence length.

The KR and BG set is open access and you can get the entire dataset at https://www.fruitfly.org/sequence/human-datasets.html and http://www1.imim.es/databases/genomics96/.

B. Metrics

Our model accuracy measures are given by (6) – (12):

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \quad (6)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \quad (7)$$

$$\text{FPR} = \frac{\text{FP}}{\text{TN}+\text{FP}} \quad (8)$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}} \quad (9)$$

$$\text{F1-Score} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \quad (10)$$

where TP, FP, TN, FN are metrics of the confusion matrix [17]. Precision-Recall curves and ROC curves [18][19] are plotted to make the performance of our model more intuitive. We also calculate areas under the curves by:

$$\text{AP} = \int_0^1 P(R)\mathrm{d}R = \sum_{i=1}^{n} P_i\Delta R_i \quad (11)$$

$$\text{AUC} = \int_0^1 T(F)\mathrm{d}F = \sum_{i=1}^{n} T_i\Delta F_i \quad (12)$$

where AP summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight [20]. AUC is equal to the probability that the model will rank a randomly chosen positive sample higher than a randomly chosen negative one (assuming that “positive” ranks higher than “negative”), where F denotes false positive rate and T denotes true positive rate [19].

C. Implementation

We encapsulate the WAM model in the class `Wam` which is derived from the base class `Base` in `./Model/basemodel.py`. Additionally, intermediate class `Ssm` is created for the shared parts of WMM and WAM. Sequences are extracted by `./Utils/extract.py` and saved temporarily in an `Sequence` object. All of the statistical graphs involved in this paper are drawn by the scripts in `./Utils` using Matplotlib [21], Seaborn [22] & Weblogo, and saved in `./Pics`. We evaluate the model using Scikit-Learn for confusion matrices, precision-recall pairs and FPR-TPR pairs [23]. These tools saved considerable time for model training and prediction. Models can be easily saved or loaded by methods `save_model()` and `load_model()`.

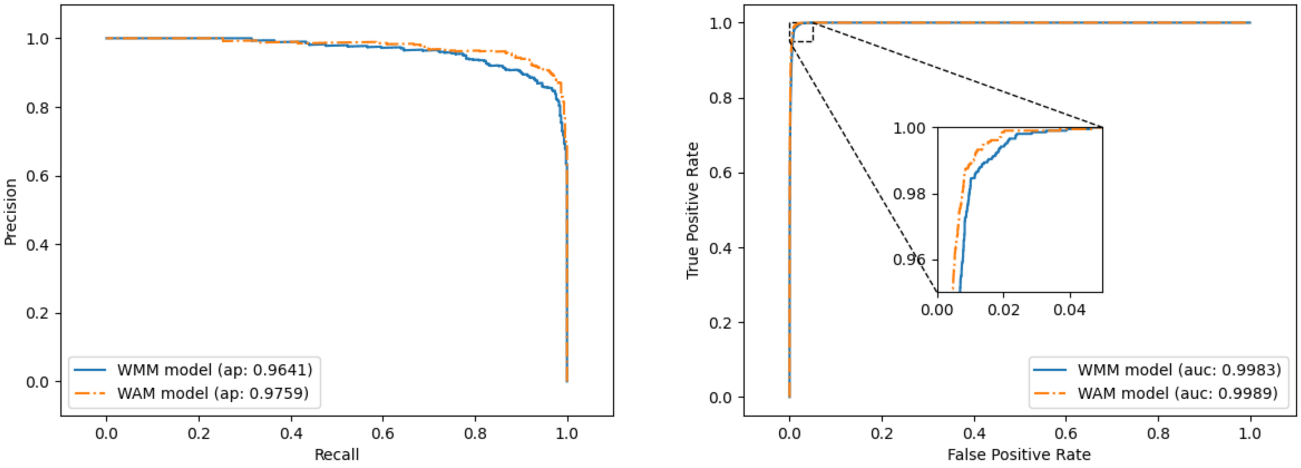

Fig. 5. Measuring WMM & WAM models on part of the BG set. **Left**: Precision-Recall curves plotted by using a bunch of thresholds. Average Precision is marked at the figure legend, which represents areas under the P-R curves. Both two models performed well on the given training data with impressive predicting precisions, aside from which WAM model has a much higher AP (0.9759) than WMM (0.9641), which means that WAM predicts fewer false positive results under the circumstance of the correct prediction of the same amount of positive sites. **Right**: ROC curves with AUC marked at the figure legend, which represents areas under the ROC. The results conforms to P-R curves' as expected, although it's harder to tell the difference between two models.

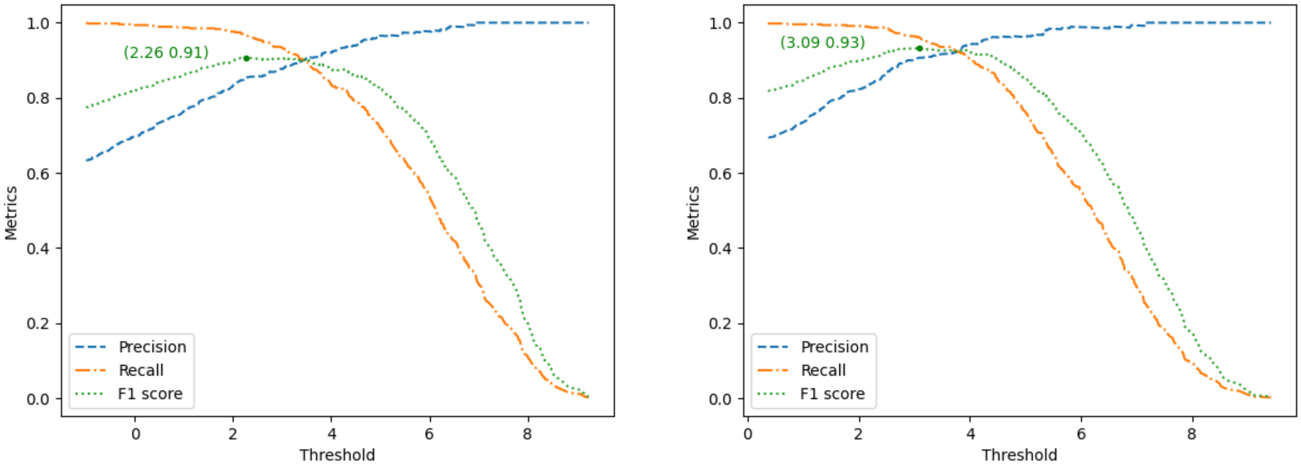

Fig. 6. Searching for the best thresholds for donor prediction. F1-score (10) is a balanced metric between precision and recall which expresses best performance of models. We have found the maximum F1 points and values for both models: **left**, WMM, (2.26, 0.91) ; **right**, WAM, (3.09, 0.93).

All the components have their corresponding interface methods provided in the aforementioned classes. For the code implementation details of WAM (also WMM), see `./Model/wam.py`.

Training & Predicting process is operated by Ubuntu 18.04.5 LTS on 16 CPU cores with 2 threads each. The source code is available on GitHub and can be obtained from https://github.com/Newiz430/SplicePredictor.

IV. Results

A. Donor Site Prediction

Comparison studies. We compared the performance of our WAM model with the conventional WMM model by Rodger, et al. [9] We evaluated two models under a set of different thresholds and the results are shown in Fig. 5. These results validate that WAM outperforms WMM in pattern identification, as We also found the thresholds with the highest F1 score which indicates the best predicting result, as is shown in Fig. 6.

We used the training set of the same size and predicting procedure for both two models, and performed them on the same set of testing data with the best threshold Fig. 6 pointing out. In Table I, we present the stats of accuracy for both models. It indicates that WAM improves the overall signal predicting effect within an approximate predicting time. WAM shows a +2.27% improvement on F1-score from 0.9061 up to 0.9267.

B. Acceptor Site Prediction

We did the same experiment on acceptor sites. Consequences are displayed in Fig. 7 – 8 and Table II.

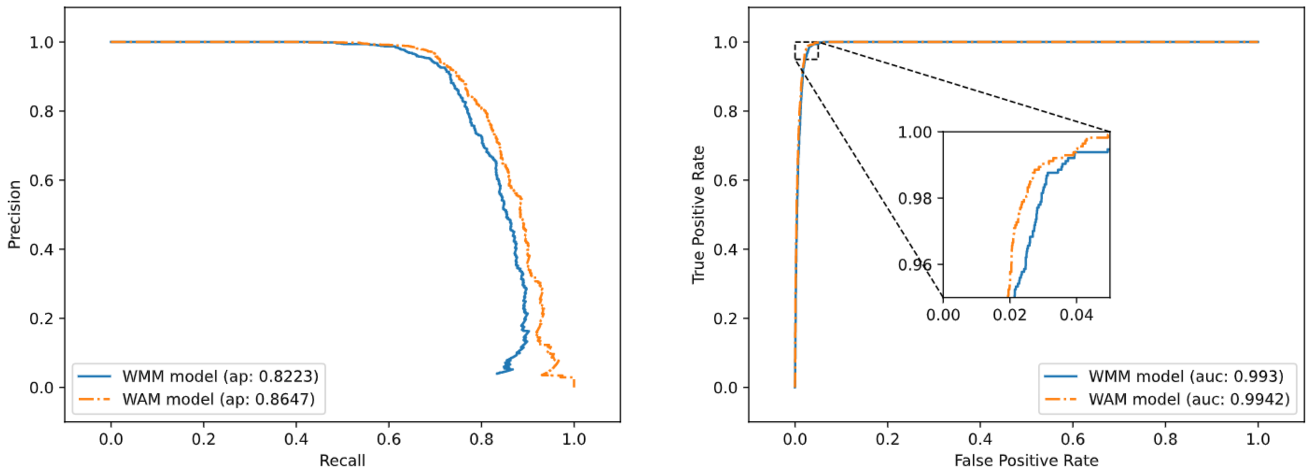

Fig. 7. Precision-Recall curves (**left**) and ROC curves (**right**) for acceptor signal. WAM Model used for acceptor seems to have a lower predicting ability relative to the one for donor. This can be explained by our hypothesis that adjacent dependencies for acceptor influences the accuracy slighter than donor's.

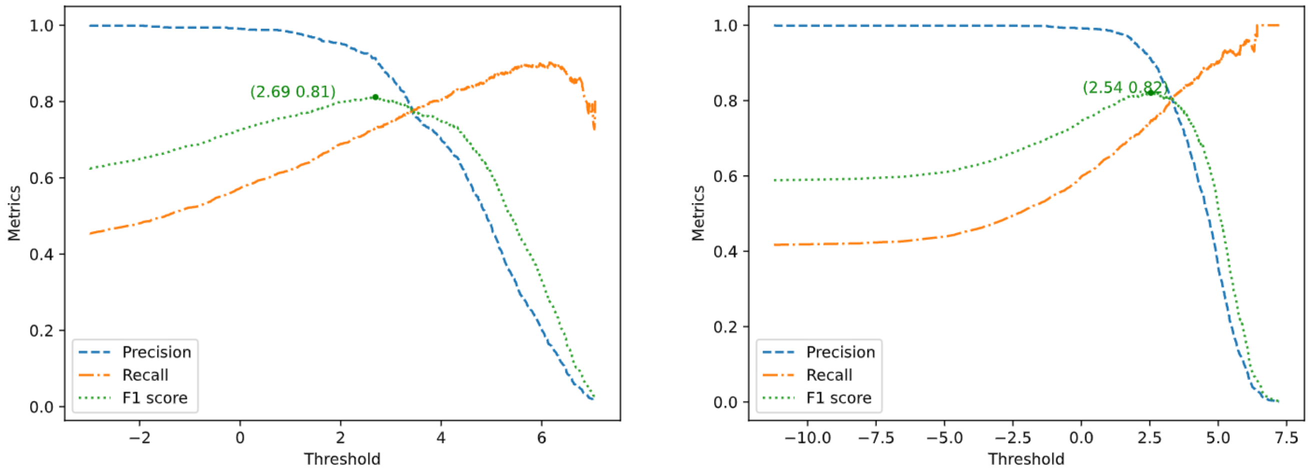

Fig. 8. Searching for the best thresholds for acceptor prediction. We have found the maximum F1 points and values for both models: **left**, WMM, (2.69, 0.81) ; **right**, WAM, (2.54, 0.82).

TABLE I
PERFORMANCE OF WAM AGAINST WMM ON DONOR SITES[1]

Method	Precision	Recall (TPR)	FPR	F1-score	Run time(s)
WMM (threshold = 2.26)	0.8503	**0.9697**	0.0085	0.9061	**1.6614**
WAM (threshold = 3.09)	**0.9006**	0.9543	**0.0053**	**0.9267**	1.7940

[1] The argmax thresholds are assigned to these models to get the best metrics. Run time represents the seconds cost in the predicting step. Same for tables below.

TABLE II
PERFORMANCE OF WAM AGAINST WMM ON ACCEPTOR SITES

Method	Precision	Recall (TPR)	FPR	F1-score	Run time(s)
WMM (threshold = 2.69)	0.7385	**0.9038**	0.0160	0.8128	**1.3336**
WAM (threshold = 2.54)	**0.7554**	**0.9038**	**0.0146**	**0.8229**	1.4286

For acceptor, WAM shows an +1.24% improvement on F1-score from 0.8128 up to 0.8229. In a nutshell, data certifies that our model is available for high precision predictions, which neither costs much time nor quantities of computing resources.

V. Discussion

Overall, we formulate and re-implement an application of WAM model by training it on the Kulp & Reese dataset. We compare its performance against the conventional WMM, and successfully prove its superiority on the accuracy of predicting donor & acceptor splice sites.

As a matter of fact, there are still some blemishes in our methods which need to be taken serious consideration. We only sampled a fraction of data for matrix construction thus our model may not attain its best performance with the given training set. We ignored the odds of base indels in signal sequences. We omitted unambiguous bases at the beginning of our work which is likely to be part of the splice site patterns. What's more, we only tried one single feature selection tactic limited by the deadline (our model is actually designed for feature sequences of different lengths as input).

In the future, we will explore deeper into the frequency-based nucleotide pattern finding methods with better generality, efficiency and practicality, aside from addressing the issues above.

Acknowledgment

This work was supported by Prof. Zhou from College of Life Science & Technology, Huazhong University of Science and Technology, and Wuhan National Laboratory for Opto-electronics for providing computing resources. Also acknowledge our classmates for helpful suggestions & corrections.

References

[1] Chris Burge and Samuel Karlin. Prediction of complete gene structures in human genomic dna. *Journal of molecular biology*, 268(1):78–94, 1997.

[2] Suzanne Clancy et al. Rna splicing: introns, exons and spliceosome. *Nature Education*, 1(1):31, 2008.

[3] Thomas D Schneider and R Michael Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic acids research*, 18(20):6097–6100, 1990.

[4] Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[5] Thomas D Schneider, Gary D Stormo, Larry Gold, and Andrzej Ehrenfeucht. Information content of binding sites on nucleotide sequences. *Journal of molecular biology*, 188(3):415–431, 1986.

[6] Douglas L Black. Mechanisms of alternative pre-messenger rna splicing. *Annual review of biochemistry*, 72(1):291–336, 2003.

[7] Harvey Lodish, Arnold Berk, Chris A Kaiser, Chris Kaiser, Monty Krieger, Matthew P Scott, Anthony Bretscher, Hidde Ploegh, Paul Matsudaira, et al. *Molecular cell biology*. Macmillan, 2008.

[8] Meena Kishore Sakharkar, Vincent TK Chow, and Pandjassarame Kangueane. Distributions of exons and introns in the human genome. *In silico biology*, 4(4):387–393, 2004.

[9] Rodger Staden. Computer methods to locate signals in nucleic acid sequences. 1984.

[10] Te-Ming Chen, Chung-Chin Lu, and Wen-Hsiung Li. Prediction of splice sites with dependency graphs and their expanded bayesian networks. *Bioinformatics*, 21(4):471–482, 2005.

[11] Mojie Duan, Min Huang, Chuang Ma, Lun Li, and Yanhong Zhou. Position-specific residue preference features around the ends of helices and strands and a novel strategy for the prediction of secondary structures. *Protein science*, 17(9):1505–1512, 2008.

[12] Tom Ryen, Trygve Eftes, Thomas Kjosmoen, Peter Ruoff, et al. Splice site prediction using artificial neural networks. In *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 102–113. Springer, 2008.

[13] MO Zhang and TG Marr. A weight array method for splicing signal analysis. *Bioinformatics*, 9(5):499–509, 1993.

[14] Denver Dash and Vanathi Gopalakrishnan. Modeling dna splice regions by learning bayesian networks. 2001.

[15] Martin G Reese, Frank H Eeckman, David Kulp, and David Haussler. Improved splice site detection in genie. *Journal of computational biology*, 4(3):311–323, 1997.

[16] Moises Burset and Roderic Guigo. Evaluation of gene structure prediction programs. *genomics*, 34(3):353–367, 1996.

[17] Stephen V Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, 62(1):77–89, 1997.

[18] David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[19] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[20] Mu Zhu. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2(30):6, 2004.

[21] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[22] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[23] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.