EnsembleSplice: Ensemble Deep Learning for Splice Site Prediction

Trevor Martin\textsuperscript{1}, Oluwatosin Oluwadare\textsuperscript{2}
\textsuperscript{1}Department of Computer Science, Oberlin College
\textsuperscript{2}Department of Computer Science, University of Colorado, Colorado Springs
tmartin2@oberlin.edu, ooluwada@uccs.edu

Abstract

Identifying splice site (SS) regions is an important step in genomic DNA sequencing pipelines for biomedical and pharmaceutical research. Within this research purview, efficient and accurate SS detection is highly desirable, and a variety of computational models have been developed towards this end. In particular, neural network (NN) architectures have recently been shown to outperform classical machine learning (ML) approaches for the task of SS prediction. Despite these advances, there is still considerable potential for improvement, especially in terms of model accuracy and inter-species generalizability. Bearing these issues in mind, EnsembleSplice is a deep learning (DL) model that incorporates the hitherto unseen method of ensemble learning for splice site prediction. When evaluated on genomic DNA datasets for the species \textit{Homo sapiens} and \textit{Arabidopsis thaliana}, EnsembleSplice outperformed existing state-of-the-art SS detection models, attaining average accuracies of 96.02\% for donor SS and 94.59\% for acceptor SS.

Introduction

Organismal genomes are studied primarily through genome annotation, which involves classifying genomic elements based on their function or location (Abril and Castel-lano Hereza 2019). This annotation is typically performed at the nucleotide-level to determine the locations of key genetic elements in DNA sequences, at the protein-level to evaluate proteomic function, or at the process-level to study the mechanisms underlying gene interaction (de Sa et al. 2018).

Genes responsible for protein coding are composed of alternating nucleotide regions called introns, which are the non-protein coding regions, and exons, which are the protein coding regions. During DNA transcription in eukaryotic cells, introns are cut out by spliceosomes and exons are combined together; this general process is called RNA splicing, and is critical for the creation of mature mRNA from pre-mRNA and for protein synthesis (Pohl et al. 2013). The dinucleotides AG and GT are often present in the 3’ intron boundary, or donor splice site (DoSS) region, and the 5’ intron boundary, or acceptor splice site (AcSS) region, respectively, and are biological markers involved in RNA splicing (Pertea, Lin, and Salzberg 2001) (see Figure 1). Nucleotide-level annotation was designed to accurately detect the location of these splice sites, which can be used to identify genes in eukaryotic genomes; a variety of other computational approaches have also been developed for this purpose.

Figure 1: Illustration detailing the process of splicing.

EnsembleSplice is one such computational method, and is a deep learning pipeline that employs ensemble learning for splice site prediction. Ensemble learning methods have been shown to enhance classification results, and have, in recent years, been successfully applied within the field of bioinformatics (Sagi and Rokach 2018; Cao et al. 2020).

We contribute the following to research on splice site prediction:

• We develop EnsembleSplice, a DL architecture that learns from an ensemble of convolutional neural network (CNN) and dense neural network (DNN) architectures to achieve state-of-the-art performance at predicting splice sites.

• We evaluate the performance of EnsembleSplice across three datasets and two organisms.

• We create a usage tutorial, detail all architectural design choices, and, for reproducibility, make the code available at https://github.com/tmartin2/EnsembleSplice.
Methodology

Datasets
Each dataset used in this paper consists of both confirmed true (positive) DoSS/AcSS and confirmed false (negative) AcSS/DoSS. Evaluation of classification performance is separated by splice site type, which means that one model is trained to distinguish between false/true DoSS regions and another is trained to distinguish between false/true AcSS regions. It is important to note that EnsembleSplice is tested on both imbalanced datasets (HS^3D) and balanced ones (Homo sapiens and Arabidopsis thaliana). See Table 1.

**HS^3D.** The Homo Sapiens Splice Sites Dataset (HS^3D) consists of human genomic DNA introns and exons extracted from the Primate Division of GenBank Rel.123 (Pollastro and Rampone 2002). There are 2,796 confirmed positive DoSS regions, 2,880 confirmed positive AcSS regions, 271,937 confirmed negative DoSS regions, and 329,374 confirmed negative AcSS regions. This paper randomly selects 10,000 false DoSS regions and 10,000 false AcSS regions from the 271,937 and 329,374 available in the dataset, respectively; the Python code `random.seed(123454)` is used to shuffle the entire HS^3D dataset before the false DoSS and false AcSS subsets are selected. The nucleotide consensus AG for AcSS regions occurs at positions 69 and 70, and the nucleotide consensus GT for DoSS regions occurs at positions 71 and 72. The HS^3D dataset can be accessed at [http://www.sci.unisannio.it/docenti/rampone/](http://www.sci.unisannio.it/docenti/rampone/).

**Homo sapiens and Arabidopsis thaliana.** The Homo sapiens and Arabidopsis thaliana datasets consist of splice site regions selected from annotated genomic DNA sequences for Homo sapiens and Arabidopsis thaliana in Ensembl (Zerbino et al. 2018). The peripheral nucleotide sequences padding each AcSS or DoSS were determined via Bedtools (Albaradei et al. 2020) [Quinlan and Hall 2010]. Each splice site region in the datasets is 602 nucleotides long; each DoSS region has consensus GT at positions 301 and 302, and each AcSS has consensus AG also at positions 301 and 302. In the Homo sapiens dataset, there are 250, 400 confirmed positive and negative DoSS regions, and 248, 150 confirmed positive and negative AcSS regions. The Arabidopsis thaliana dataset includes 110, 314 confirmed positive and negative DoSS regions, and 112, 336 confirmed positive and negative AcSS regions. The confirmed negative AcSS and DoSS regions were selected from chromosomes 21, 2, 2L, 1, and I. Again, this paper randomly selects 8000 true/false DoSS regions and 8000 true/false AcSS regions from both datasets. As with the HS^3D dataset, the Python code `random.seed(123454)` is used for shuffling the Homo sapiens and Arabidopsis thaliana datasets before the DoSS and AcSS subsets are selected. The Homo sapiens and Arabidopsis thaliana datasets can be accessed at [https://github.com/SomayahAlbaradei/Splice_Deep](https://github.com/SomayahAlbaradei/Splice_Deep).

**EnsembleSplice Pipeline**
We now propose EnsembleSplice, a DL architecture that consists of an ensemble of three CNN and three DNN sub-models, for the task of splice site detection. See Figure 2 for the full architecture.

The sub-models in EnsembleSplice generate predictions for whether genomic DNA input sequences are positive DoSS or negative DoSS, or if the AcSS model is being used, for whether the sequences are positive AcSS or negative AcSS. These binary predictions are then aggregated (stacked) into a new dataset, where each sub-model’s predictions become a column vector, and this dataset is then fed into a Logistic Regression classifier, which produces the final predictions for the inputted sequences.

Consider a family
\[ D = \{S_0, S_1, \ldots, S_n\} \]
of nucleotide splice site regions. We have the ordered set
\[ S_i = \{x_1, x_2, \ldots, x_{|S_i|}\} \]
where \(S_i\) is the \(i\)-th nucleotide splice site region, and \(x_j \in X = \{A, C, G, T\}, 0 \leq j \leq |S_i|\).

For all \(0 \leq i \leq n\), \(S_i\) is encoded as a \(|S_i| \times |X|\) binary matrix through one-hot encoding. These encoded sequences are fed to the three CNNs and three DNNs.

Each CNN sub-model in EnsembleSplice is composed of three convolutional layers and a dropout layer. The convolutional layers automatically extract local and global features from the AcSS or DoSS input sequences. In particular, these layers form complex representations of the sequences, and are the components of the CNN that allow it to accurately discriminate between the true/false acceptor/donor sites. The first layer has 72 convolutional filters and a kernel size of 5, the second layer has 144 convolutional filters and a kernel size of 7, and the third layer has 168 convolutional filters and a kernel size of 7. Each convolutional layer employs the ReLU activation function as its final component; this removes noisy or otherwise irrelevant features, thus improving feature selection (Hahnloser et al. 2000) [Krizhevsky and Hinton 2010]. Additionally, each convolutional layer has a stride size of 1. Next, a dropout layer

<table>
<thead>
<tr>
<th>Splice Site</th>
<th>Dataset</th>
<th>Sequence Count</th>
<th>Pos:Neg Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceptor (AcSS)</td>
<td>HS^3D</td>
<td>2,880 (true), 10,000 (false)</td>
<td>1:3,472</td>
</tr>
<tr>
<td></td>
<td>Homo Sapiens</td>
<td>8000 (true), 8000 (false)</td>
<td>1:1</td>
</tr>
<tr>
<td></td>
<td>Arabidopsis thaliana</td>
<td>8000 (true), 8000 (false)</td>
<td>1:1</td>
</tr>
<tr>
<td>Donor (DoSS)</td>
<td>HS^3D</td>
<td>2,796 (true), 10,000 (false)</td>
<td>1:3,577</td>
</tr>
<tr>
<td></td>
<td>Homo Sapiens</td>
<td>8000 (true), 8000 (false)</td>
<td>1:1</td>
</tr>
<tr>
<td></td>
<td>Arabidopsis thaliana</td>
<td>8000 (true), 8000 (false)</td>
<td>1:1</td>
</tr>
</tbody>
</table>
prunes a percentage (20%) of each network’s total convolutional nodes, which limits the co-dependencies each node in the network has on other nodes in the network, subsequently reducing model overfitting (Srivastava et al. 2014). Lastly, the output is fed through a Softmax activation function, which produces, for each given input sequence, a probability of that sequence being a true/false acceptor/donor site. The ADAM optimizer with an inverse time decay learning rate schedule is used during model compilation (Kingma and Ba 2014). This architecture is consistent across the CNN sub-models, and is used for both AcSS prediction and DoSS prediction. The CNN architecture parameters were selected using hyperparameter tuning. For the hyperparameter tuning, see Table 2.

The DNN sub-models in EnsembleSplice consist of two fully-connected dense layers, followed by a dropout layer, another fully-connected dense layer, and another dropout layer. The first two fully-connected dense layers have 704 and 224 nodes, respectively, and both use a kernel regularizer with an L2 regularization penalty of 0.025. The third fully-connected dense layer has 512 nodes, but uses no regularization penalties. The first dropout layer prunes 10% of the DNN’s nodes and the second dropout layer prunes 15%. Each fully-connected layer incorporates the ReLU activation function. Identical to the CNN sub-models, the output layer is a Softmax activation function and model compilation for the DNN sub-models is completed via the ADAM optimizer with an inverse time decay learning rate schedule. All DNN sub-models use this architecture for both AcSS prediction and DoSS prediction, and the parameters for this architecture were also selected using hyperparameter tuning.

EnsembleSplice is implemented via the TensorFlow/Keras framework (Abadi et al. 2016; Chollet and others 2018). For all experiments conducted, 30 was the maximal number of epochs used for training. The early model stopping callback, which ceases training if the model’s validation loss does not improve for a selected number of epochs, was used during training and validation, which occurred in Google Colaboratory (https://colab.research.google.com/) and made use of Graphical Processing Unit (GPU) hardware. Cross validation was used for initial CNN and DNN sub-model architecture testing.

One-hot Encoding
Genomic DNA splice site regions are composed of four nucleotides - A (Adenine), C (Cytosine), G (Guanine), or T (Thymine). Given the input constraints of DL architectures, these nucleotides are encoded numerically, as opposed to categorically. Each nucleotide corresponds to a row in a 4 x 4 identity matrix. The encoding scheme utilized in this paper is that A corresponds to [1, 0, 0, 0], C corresponds to [0, 1, 0, 0], G corresponds to [0, 0, 1, 0], and T corresponds to [0, 0, 0, 1]. Since each splice site region consists of some N nucleotides, the final numerical representation for each splice site region is a N x 4 matrix, where each row is a one-hot encoded nucleotide that occurs at the same index as it did in the splice site region’s original representation.

<table>
<thead>
<tr>
<th>CNN Parameters</th>
<th>Search Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv. Layer 1 Filters</td>
<td>{8, 16, . . . , 72, . . . , 400}</td>
</tr>
<tr>
<td>Conv. Layer 1 Kernel Size</td>
<td>[1, 3, 5, 7, 9]</td>
</tr>
<tr>
<td>Conv. Layer 2 Filters</td>
<td>{8, 16, . . . , 144, . . . , 400}</td>
</tr>
<tr>
<td>Conv. Layer 2 Kernel Size</td>
<td>[1, 3, 5, 7, 9]</td>
</tr>
<tr>
<td>Conv. Layer 3 Filters</td>
<td>{8, 16, . . . , 168, . . . , 400}</td>
</tr>
<tr>
<td>Conv. Layer 3 Kernel Size</td>
<td>[1, 3, 5, 7, 9]</td>
</tr>
<tr>
<td>Dropout Layer</td>
<td>( {\frac{1}{20}, \frac{2}{20}, \frac{3}{20}, \frac{4}{20}, \frac{5}{20}} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DNN Parameters</th>
<th>Search Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense Units 1</td>
<td>{32, 64, . . . , 704}</td>
</tr>
<tr>
<td>Dense Kernel Reg. 1</td>
<td>( {\frac{1}{1000}, \frac{1}{2000}, \frac{1}{3000}, \frac{1}{4000}, \frac{1}{5000}} )</td>
</tr>
<tr>
<td>Dense Units 2</td>
<td>{32, 64, . . . , 224, . . . , 704}</td>
</tr>
<tr>
<td>Dense Kernel Reg. 2</td>
<td>( {\frac{1}{1000}, \frac{1}{2000}, \frac{1}{3000}, \frac{1}{4000}, \frac{1}{5000}} )</td>
</tr>
<tr>
<td>Dropout Layer 1</td>
<td>( {\frac{1}{20}, \frac{20}{20}, \frac{10}{20}} )</td>
</tr>
<tr>
<td>Dense Units 3</td>
<td>{32, 64, . . . , 512, . . . , 704}</td>
</tr>
<tr>
<td>Dropout Layer 2</td>
<td>( {\frac{1}{20}, \frac{2}{20}, \frac{3}{20}, \frac{4}{20}, \frac{5}{20}} )</td>
</tr>
</tbody>
</table>

Cross Validation, Training, and Testing
The HS3D, Homo sapiens, Arabidopsis thaliana dataset subsets were each split into a training (80% of the data) and a testing (20% of the data) subset. Cross-validation has been demonstrated to be an effective tool for model selection, and as such, 10-fold cross validation was used for evaluating alternative EnsembleSplice sub-model architectures (Shao 1993). For each dataset, the training subset of that dataset was partitioned into 10 approximately equal sized subsets. Every subset was used at some point to evaluate the performance of EnsembleSplice; for each of the 10 runs, training occurred on 9 subsets, and testing occurred on the last subset. The cross-validation performance for a particular dataset was the average performance over the 10 folds. Once the sub-model architectures for EnsembleSplice were chosen, EnsembleSplice was trained on the full training subset of each dataset, and then tested once on the testing subset of the respective dataset. No additional training or validation occurred following the final test run; the results reported for EnsembleSplice in this paper are the performances from this final test run.

Experiments and Results
Evaluation Metrics
To measure the performance of EnsembleSplice, and to compare EnsembleSplice with with other splice site detection models, the counts of correctly identified true AcSS or DoSS (true positive, “TP”), correctly identified false AcSS or DoSS (true negative, “TN”), incorrectly identified true AcSS or DoSS (false positive, “FP”), and incorrectly identified false AcSS or DoSS (false negative, “FN”) are used. See Table 3.
Table 3: Confusion Matrix for Binary Classification Tasks

<table>
<thead>
<tr>
<th>Actual Class</th>
<th>Predicted Class</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Positive</td>
<td>True Positive (TP)</td>
<td>False Negative (FN)</td>
</tr>
<tr>
<td>Class Negative</td>
<td>False Positive (FP)</td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>

From these metrics, additional metrics common to classification tasks can be used for evaluation: Accuracy (Acc) - the fraction of AcSS or DoSS correctly identified, Precision (Pre) - the fraction of positive classifications for AcSS or DoSS that were positive, Sensitivity (Sn) - the fraction of positive AcSS or DoSS that were classified as positive, Specificity (Sp) - the fraction of negative AcSS or DoSS that were classified as negative, Matthew’s correlation coefficient (Mcc) - the correlation between true/false AcSS and DoSS and the classifications for them generated by the model, and $F_1$ score - the harmonic means of the fraction of positive classifications for AcSS or DoSS that were positive and the fraction of positive AcSS or DoSS that were correctly identified. Lastly, the error rate measures how often the classifier misclassified the data. The equations for these metrics are in Table 4.

Table 4: Evaluation Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acc</td>
<td>$\frac{TP+TN}{TP+FN+TN+FP}$</td>
</tr>
<tr>
<td>Sp</td>
<td>$\frac{TN}{TN+FP}$</td>
</tr>
<tr>
<td>Sn</td>
<td>$\frac{TP}{TP+FP}$</td>
</tr>
<tr>
<td>$F_1$</td>
<td>$\frac{2\times TP}{2\times TP+FP+FN}$</td>
</tr>
<tr>
<td>Pre</td>
<td>$\frac{TP}{TP+FP}$</td>
</tr>
<tr>
<td>Mcc</td>
<td>$\frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$</td>
</tr>
<tr>
<td>Error rate</td>
<td>$1 - \text{Accuracy}$</td>
</tr>
</tbody>
</table>

Model Benchmarking

The state-of-the-art models iss-CNN and SpliceRover were used in a comparison with EnsembleSplice (Zuallaert et al. 2018; Tayara, Tahir, and Chong 2019).

iss-CNN was trained on a subset of HS$^3$D data, and consists of a convolutional layer with 16 filters, a kernel size of 7, and stride size of 3, a dropout layer that prunes 30% of the nodes, and a fully-connected dense layer that uses the Sigmoid activation function. The testing was conducted on iss-CNN’s public web server, which can be found at [http://nsclbio.jbnu.ac.kr/tools/iSS-CNN/](http://nsclbio.jbnu.ac.kr/tools/iSS-CNN/) and the classification threshold used for predicting AcSS or DoSS was 0.5. The HS$^3$D testing subset, which was also used to evaluate EnsembleSplice, was used for benchmarking iss-CNN. See Figure 2.

Figure 2: iss-CNN Webserver

SpliceRover was trained on human genomic DNA data and Arabidopsis thaliana genomic DNA data, and is another CNN. Its architecture consists of a convolutional layer with filters equal in number to the AcSS or DoSS length, a max-pooling layer, and a series of convolutional and max-pooling layers. A fully-connected dense layer follows the convolutional layers, and the output is lastly fed through a Softmax activation function. To benchmark SpliceRover, their publicly available web server was used; a cut of 0.5 was again utilized, as this is what EnsembleSplice uses. The web server can be found at the following link: [http://bioit2.irc.ugent.be/rover/splicerover](http://bioit2.irc.ugent.be/rover/splicerover). See Figure 4.

Figure 4: SpliceRover Webserver

SpliceRover – Detect splice sites in your sequence

Figure 3: The iss-CNN public webserver.

The benchmarked results, along with EnsembleSplice’s results, can be found in Table 5.

Figure 5: The SpliceRover public webserver.

The benchmarked results, along with EnsembleSplice’s results, can be found in Table 5.
Table 5: Model Performances

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Splice Site</th>
<th>Model</th>
<th>Sp</th>
<th>Sn</th>
<th>Pre</th>
<th>Err</th>
<th>Acc</th>
<th>Mcc</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>HS3D</td>
<td>Acceptor</td>
<td>issCNN</td>
<td>89.20</td>
<td>91.84</td>
<td>83.05</td>
<td>9.83</td>
<td>90.16</td>
<td>79.53</td>
<td>87.22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EnsembleSplice</td>
<td>97.75</td>
<td>92.36</td>
<td>92.20</td>
<td>3.45</td>
<td>96.55</td>
<td>90.07</td>
<td>92.29</td>
</tr>
<tr>
<td></td>
<td>Donor</td>
<td>issCNN</td>
<td>94.50</td>
<td>94.99</td>
<td>90.61</td>
<td>5.32</td>
<td>94.68</td>
<td>88.61</td>
<td>92.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EnsembleSplice</td>
<td>98.25</td>
<td>96.96</td>
<td>93.93</td>
<td>2.03</td>
<td>97.97</td>
<td>94.14</td>
<td>95.42</td>
</tr>
<tr>
<td>Arabidopsis thaliana</td>
<td>Acceptor</td>
<td>SpliceRover</td>
<td>88.31</td>
<td>89.25</td>
<td>88.42</td>
<td>11.22</td>
<td>88.78</td>
<td>77.57</td>
<td>88.83</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EnsembleSplice</td>
<td>93.88</td>
<td>93.44</td>
<td>93.85</td>
<td>6.34</td>
<td>93.66</td>
<td>87.31</td>
<td>93.64</td>
</tr>
<tr>
<td></td>
<td>Donor</td>
<td>SpliceRover</td>
<td>86.88</td>
<td>87.13</td>
<td>86.91</td>
<td>13.00</td>
<td>87.00</td>
<td>74.00</td>
<td>87.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EnsembleSplice</td>
<td>94.06</td>
<td>94.81</td>
<td>94.11</td>
<td>5.56</td>
<td>94.44</td>
<td>88.88</td>
<td>94.46</td>
</tr>
<tr>
<td>Homo Sapiens</td>
<td>Acceptor</td>
<td>SpliceRover</td>
<td>88.25</td>
<td>93.44</td>
<td>88.83</td>
<td>9.16</td>
<td>90.84</td>
<td>81.80</td>
<td>91.08</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EnsembleSplice</td>
<td>93.19</td>
<td>93.94</td>
<td>93.24</td>
<td>6.44</td>
<td>93.56</td>
<td>87.13</td>
<td>93.59</td>
</tr>
<tr>
<td></td>
<td>Donor</td>
<td>SpliceRover</td>
<td>85.44</td>
<td>91.13</td>
<td>86.22</td>
<td>11.72</td>
<td>88.28</td>
<td>76.69</td>
<td>88.61</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EnsembleSplice</td>
<td>95.31</td>
<td>96.00</td>
<td>95.34</td>
<td>4.34</td>
<td>95.66</td>
<td>91.31</td>
<td>95.67</td>
</tr>
</tbody>
</table>

Figure 6: Donor site detection accuracies for each method tested.

Figure 7: Acceptor site detection accuracies for each method tested.
The accuracies for each DoSS model benchmarked can be found in Figure 6, and the accuracies for each AcSS model benchmarked can be found in Figure 7.

Conclusion and Future Work

From the results, it can be observed that on all metrics employed, EnsembleSplice performed better than either iss-CNN or SpliceRover, two state-of-the-art methods that exist for splice site prediction and that use DL architectures. For future work, we consider evaluating model robustness; this would consist of testing a model trained on genomic DNA from one species on another species genomic DNA. In this case, that would mean testing the EnsembleSplice models trained on Homo sapiens data on the Arabidopsis thaliana data, and seeing how well the performance generalizes across species.

Acknowledgement

The work reported in this paper is supported by the National Science Foundation under Grant No. 2050919. Any opinions, findings and conclusions or recommendations expressed in this work are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References


