Prediction of acceptor sites in DNA Splicing

Mridul Garg[1,] Viveka Gorla[2]

[1]Department of Industrial Engineering, [2] Department of Industrial Engineering

[*]Equally Contributed

Abstract

Motivation: One of the important steps in transfer of biological sequential information is gene splicing. Several challenges hinder study of gene splicing, such as lack of labeled data which can be used to identify splice sites. Like many other applications in biological problems, machine learning has been used to identify these splice sites. The lack of labeled data, however, makes it a difficult task to build supervised classifiers which can accurately predict these splice sites. We have aimed to address this problem by integrating gene splice site domain knowledge with machine learning techniques.

Data availability: Data set used is available at http://cbio.mskcc.org/public/raetschlab/user/cwidmer/

Contact: mridulgarg11@tamu.edu, viveka-93@tamu.edu

Keywords: Gene splice site, machine learning, supervised classifiers

1. Introduction

The central dogma of molecular biology lists three general transfers of biological sequential information- DNA replications, transcription and translation. One more important step between transcription and translation is gene splicing. Gene splicing refers to the editing of pre-mRNA to RNA. During the splicing process a part of mRNA, called intron, is lost and remaining two ends, called exon, join together. The site of mRNA where this exon or intron are present is called splice site. The exon site at the beginning of an intron is called a donor splice site and at the end of intron is called the acceptor. However, identifying these splice sites is a relatively tougher task for supervised classifiers, because only 1% or less of AG dimers in a DNA sequence make up the acceptor splice sites. Past research [1] into this topic addresses this problem by incorporating domain knowledge from other species for which there is abundance of labeled data. We have, however, attempted to predict the splice sites only by using the labeled data for the same species.

Our approach includes more features for the prediction problem than past work, and also uses more sophisticated machine learning techniques.

2. Method

Splice site prediction is a classification problem, with the input being a DNA sequence and output being a binary label indicating whether the AG dimer is a splice site or not. Since a DNA sequence cannot be used by a machine learning algorithm for learning, we first generate splice site features from this DNA sequence. The high dimensionality of this feature set poses computational problems, and hence feature selection methods are first applied to make the feature matrix sparse and only use the most relevant features. A sampling technique is used to generate synthetic samples, which leads to equal proportion of positive and negative samples. Another approach to deal with class imbalance is by adjusting the threshold for classification. Logistic Regression, Random Forrest classifiers and One-Class SVM models are then built for determining the labels for the test samples. Area under the Precision-Recall Curve (auPR curve) is used as the assessment criteria. The following sections describe these steps in detail-

2.1 Techniques used

2.1.1. Logistic Regression

Logistic regression is a statistical method for analyzing a data set in which there are one or more independent varia-

bles that determine an outcome where the outcome is qualitative. It predicts the odds of an observation being in a particular class. Like other classifiers, logistic regression makes use of one or more predictor variables that may be either continuous or categorical. Unlike linear regression, which predicts a continuous variable, logistic predicts a categorical variable and thus has bounds. To be in that bounds, logistic takes logarithm of ratio of odds of an observation, which will be non negative.
The logistic function can be written as:

$$g(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(w_0+w_1x_1+w_2x_2+w_3x_3)}}$$

2.1.2 Random Forest

Random forest is an ensemble machine learning model and is regarded as one of the most powerful classification algorithms. The basics of Random forest is not depending on any single model but to analyze from numerous models at a time. Random forest fits n (user defined) number of independent decision trees to the given data and uses the result of majority trees for classification. Decision Trees is a nonparametric supervised machine learning algorithm which predicts the target variable by learning simple decision rules inferred from the features of the data. Building multiple decision trees helps capture the non-linearity in the data.

2.1.3 One class SVM

SVM (Support Vector Machines) are supervised learning models which separate data points of different classes with a separating hyperplane having as wide a margin as possible. One class SVM [2], also known as novelty or outlier detection, only uses the data from one of the classes, the minority class (or the event to be detected). This method is especially useful for data with unbalanced classes (which the given data set has). One class SVMs build a spherical boundary around the target class to separate the classes. The boundary is built such that it accepts maximum number of target classes as well as has low probability of accepting the outliers.

2.1.4. SMOTE

SMOTE [3] (Synthetic Minority Over-Sampling Technique) is an over sampling technique for overcoming the lack of data in the minority class. The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. SMOTE essentially creates new data points in the minority class neighborhood using convex combination of k-nearest neighbors.

3. Data set

The data set used for training and validation purposes is the same data set that was used in previous related studies. [1]. This is the genome data of species *C.Elegans.* The data set contains 25000 instances of DNA sequences of length 141 and binary response indicating the acceptor splice site at the 61st position (site of AG dimer). Only 1% of the instances are positive cases '1' (i.e., acceptor splice site). This makes the data highly imbalanced and very difficult for predictive models to capture them.

4. Feature generation

The feature generation framework discussed in [4] is mainly used for feature extraction from a DNA string. Three types of features are generated to capture various aspects from the data.

1.*Compositional features*: A k-mer is a string of k-characters. As each combination is a different feature, there are 4^k features for a given k. Under each feature the frequency of a given k-mer in the string is noted. Following the previous research works we selected the k values as one and three.
2. *Region Specific Features*: Splice-site sequences characteristically have a coding region and a non-coding region. The region left (upstream) to the acceptor splice site is called a non-coding region and the region on the right (downstream) is called the coding region. It can be expected that the compositional features of coding and non-coding region have affect on the splice site. Since the compositional features are split into two, there are 2*4^k features for a given k.
3. *Positional Features*: These are the most common features used in any DNA sequence analysis. Positional features capture the correlation between different nucleotides and their relative positions. This can be done in form of orthogonal encoding. That is every position of DNA, a vector of 4 levels is generated and the presence of a particular level is indicated by "1" while absence by "0".

5. Feature selection

A total of 68 compositional features (created for k=1,3), 136 Region specific features (created for k=1,3) and 564 positional features were generated in the previous step. This makes the dimensionality of the feature space really high and some techniques to introduce sparsity in the feature space are required. There are three general approaches for feature selection- filtering, embedded methods and wrapper approaches. We have used an example of each of them during the feature selection for our problem. These are discussed in detail in the next section-

5.1. LASSO

Lasso is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces. The algorithm iteratively adds the predictor with the best assessment metrics until all the factors are exhausted. The best features are selected by tuning the regularization parameter lambda. Usually the minimum value of lambda is selected which corresponds to the number of features after which the model accuracy does not increase. The following graph shows the value of AUC as a function of log of lambda as the number of features are iteratively added. The value of lambda is chosen in between the vertical lines shown. This is an example of wrapper approach.

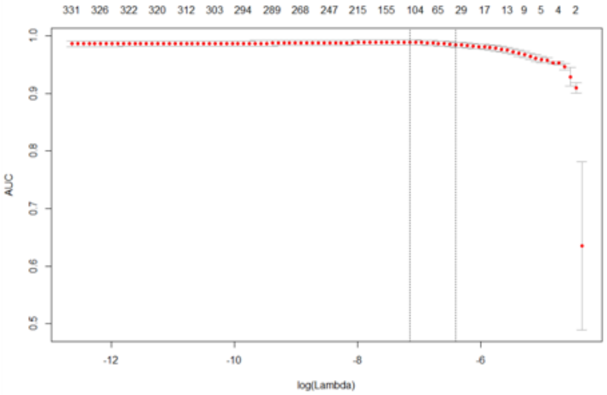

Fig 1: AUC value as a function on regularization parameter lambda

5.2 Embedded method

For feature selection in Random Forest [5], we use a two-step process involving filtering and embedded feature selection approaches. The first step is filtering out the least important variables based on their variable importance. The variables having a scaled importance (on a scale of 0-1) below 0.20 are filtered out. This is the initial step for dimensionality reduction. After preliminary selection, we use embedded feature selection approach to select the most important features. Embedded feature selection is a machine learning algorithm that returns a model based on limited number of features. Embedded methods use specific structure of the machine learning algorithm to select the most important features. It assesses it's results by cross validation, and calculating the out-of-bag error. Following the above mentioned steps, we have reduced the dimensionality of our feature space from 760 to 99 (for the training data). The selected variables are then used to build optimized random forest models, using sampling techniques and by adjusting the probability threshold for classification.

6. Results and Discussions

Tables shows the auPRC values for different techniques used. The reason for using auPRC as opposed to the usual auROC is as follows- due to the class imbalance in the data, setting the output for all the samples to be the majority class will give auROC value of about 98%. This makes auROC not a true measure of model performance. Instead auPRC is analyzed which is area under Precision-Recall Curve. Recall is defined as the fraction of minority class that is actually predicted to be in minority class. Precision can be explained as fraction of predicted minority class that are actually minority class.
The Table I a, b show the auPRC values for *C.Elegans* and *C.Remanei* (source and target species respectively in [1]) with different classifiers.

Notations of the models:
L1 R is the L1 logistic regression model. *L1 RS* is the L1 logistic regression model built on sampled data using *SMOTE. RF* is the Random forest model on original data with embedded variable selection. RF *S* is the Random forest model on sampled data (SMOTE) with embedded variable selection. One-Class *SVM* is a single class support vector model built on minority class.

Table I
a. *C.Elegans*

Classifiers	auPRC value (%)
L1_R	81.14
L1_RS	80.11
RF	63
RF_S	
One class SVM	64.59

b. *C.Remanei*

Classifiers	auPRC value (%)
L1_R	72.24
L1_RS	72.11
RF	63
RF_S	
One class SVM	62.79

Although the non-linear RF model is expected to give better results than a linear logistic model, it does worse than L1 models. This is a bit surprising, but not impossible. We reckon this is probably due to the over-fitting of RF model. The general solution to over-fitting is to reduce the number of features. Through the embedded approach we used more than 90 features are selected as important. More screening of these features may yield better results.

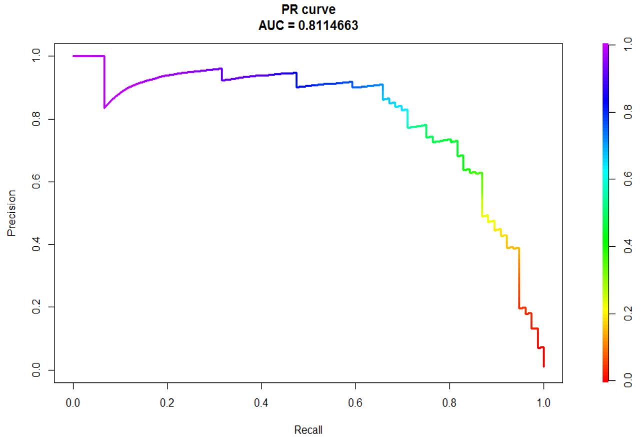

Fig.2. The PR curve of L1_S Model

One class SVM is an algorithm useful in situations where there is highly imbalanced data. Therefore, the results were expected to be better than than L1. But the practical results are reversed. This is due to the very limited instances of target class. There are only 250 instances of target class while features of data are 764, due to which we think the performance of one class SVM is very poor.

To make a overall comparison of our model to the existing models, the LR DA model for target species *C.Remanei* from [1] performs much better with a auPRC of 82.61% than the L1 model (auPRC:72.24%). But the LR DA model is trained on 100000 instances of source data and 40000 instances of target data to achieve their result, while L1 model trains on only 20000 instances of the target species alone but uses a powerful machine learning algorithm. This shows the potential to reduce the computational load by almost 80% through careful training of sophisticated algorithms.

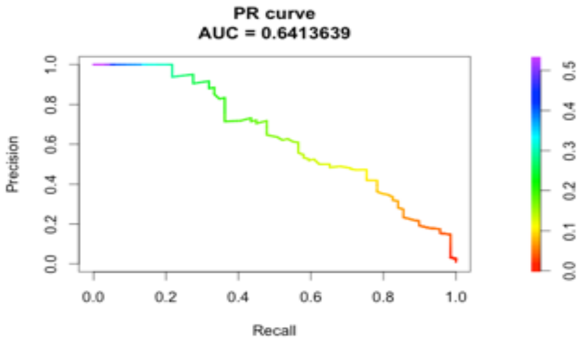

Fig 3: The PR Curve for RF Model

7. Conclusions

The results obtained in this paper are only a starting point, and provide good support to our hypothesis that using instances only from the target species to train machine learning models can help predict the unlabeled data. Our results were not better than previous researchers who worked on the same topic, and it can probably be attributed to the fact that we did not have the computational means to handle such large volumes of data. The main points to take away from this project are that supervised machine learning algorithms can be used to identify DNA acceptor sites, and have an important role to play in genome annotation. Working with limited computational power shows us that simple linear models like logistic regression give better results than some more advanced models like Random Forests and One-Class SVM because of the failure to optimize such models. Our contribution to previous research is that using a better set of generated features, and optimized non-linear classifiers give comparable results to domain adapted linear classifiers.

We would like to acknowledge Dr. Nicolae Herndon for his swift response to our questions, and Dr. Yang Shen for his guidance through the course of this project.

References

[1] Herndon, N., & Caragea, D. (2016). A Study of Domain Adaptation Classifiers Derived from Logistic Regression for the Task of Splice Site Prediction.
[2] Rana, D. One Class SVM Vs SVM Classification.
[3] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 321-357.
[4] Islamaj, R., Getoor, L., & Wilbur, W. J. (2006). A feature generation algorithm for sequences with application to splice-site prediction. In Knowledge Discovery in Databases: PKDD 2006 (pp. 553-560). Springer Berlin Heidelberg.
[5] Robin Genuer, Jean-Michel Poggi, Christine Tuleau-Malot. Variable selection using Random Forests. Pattern Recognition Letters, Elsevier, 2010, 31 (14), pp.2225-2236.