A Feature Generation Algorithm for Sequences with Application to Splice-Site Prediction

Rezarta Islamaj[1], Lise Getoor[1], and W. John Wilbur[2]

[1] Computer Science Department, University of Maryland, College Park, MD 20742
[2] National Center for Biotechnology Information, NLM, NIH, Bethesda, MD 20894
{rezarta, getoor}@cs.umd.edu, wilbur@ncbi.nlm.nih.gov

Abstract. In this paper we present a new approach to feature selection for sequence data. We identify general feature categories and give construction algorithms for each of them. We show how they can be integrated in a system that tightly couples feature construction and feature selection. This integrated process, which we refer to as *feature generation*, allows us to systematically search a large space of potential features. We demonstrate the effectiveness of our approach for an important component of the gene finding problem, splice-site prediction. We show that predictive models built using our feature generation algorithm achieve a significant improvement in accuracy over existing, state-of-the-art approaches.

1 Introduction

Many real-world data mining problems involve data best represented as sequences. Sequence data comes in many forms including: 1) human communication such as speech, handwriting and language, 2) time sequences and sensor readings such as stock market prices, temperature readings and web-click streams and 3) biological sequences such as DNA, RNA and protein. Sequence data in all domains contains useful 'signals', features, that enable the correct construction of classification algorithms.

Extracting and interpreting the features is known to be a hard problem. In many cases a brute force approach is taken, in which the sequence classification models are provided with a huge number of features in the hope that the important features are not overlooked. The large number of features introduces a dimensionality problem which has several disadvantages. First, enumerating all possible features is impractical. Second, many features are irrelevant to the classification task and have an adverse effect on accuracy. And third, knowledge discovery becomes difficult because of the large number of parameters involved.

The focus of this paper is on a scalable method for feature generation for sequences. We present an algorithm that explores the space of possible features and identifies the most useful ones. Our focused **feature generation algorithm (FGA)** integrates feature construction and feature selection in a systematic way. Our method is scalable because it incrementally generates more complex features from currently selected ones.

We validate our method on the task of splice-site prediction for pre-mRNA sequences. Splice sites are the locations in the DNA sequence, which are boundaries for protein coding and non-coding regions. Accurate location of splice sites is an important component in the gene finding problems. It is a particularly difficult problem since the sequence characteristics, i.e. pre-mRNA sequence length, coding sequence length, number of exons and their lengths, and interrupting intron sequence lengths do not follow any known pattern, making it hard to locate the genes.

We demonstrate the effectiveness of our approach by comparing it with a state-of-the-art method, GeneSplicer. Our predictive models show significant improvement in accuracy. Our final feature set, achieves a 6.3% improvement in the 11-point average precision when compared to GeneSplicer. At the 95% sensitivity level, our method yields a 10% improvement in specificity. Our contribution is two-fold. First, we give a general feature generation framework appropriate for any sequence data problem. Second, we provide new results and identify a set of features for splice-site prediction that should be of great interest to the gene-finding community.

2 Related Work

Feature selection techniques have been studied extensively in text categorization[1–5]. Recently they have begun receiving more attention for applications to biological data. A good introduction for filtering methods in the prediction of translation initiation sites is given in [6]. Various feature selection techniques for prediction of splice sites have been studied in [7–9]. And in [10], SpliceMachine is described with compelling results. In addition, there is a significant amount of work on splice-site prediction. One of the most well-known approaches is GeneSplicer proposed by Pertea et al [11].

3 Data Description

We validate our methods on a dataset which contains $4,000$ RefSeq[3] pre-mRNA sequences. In a pre-mRNA sequence, a human gene is a protein coding sequence which is characteristically interrupted by non-coding regions, called introns. The coding regions are referred to as exons. The acceptor splice site marks the start of an exon and the donor splice site marks the end of an exon. All the pre-mRNA sequences in our dataset follow the AG consensus for acceptors and GT consensus for donors.

We focus on the prediction of acceptor splice sites which is considered to be a harder problem. Following the GeneSplicer format, we mark the splice site and take a subsequence consisting of 80 nucleotides upstream from the site and 80 nucleotides downstream. We construct negative examples by choosing random AG-pair locations that are not acceptor sites and selecting subsequences as we do for the true acceptor sites. Our data contains 20,996 positive instances and 200,000 negative instances.

4 Feature Generation

The feature types that we consider capture compositional and positional properties of sequences. These apply to any sequence data defined over some fixed alphabet. For each feature type we describe an incremental feature construction procedure. The feature construction starts with an initial set of features and produces an expanded set of features. Incrementally, it produces richer, more complex features for each iteration.

[3] http://www.ncbi.nlm.nih.gov/RefSeq/

4.1 Feature Types and Construction Procedures

Compositional features A *general k-mer* is a string of k-characters. This feature type is useful for capturing information like coding potential and composition in the sequence.
Construction Method. Given an initial set of k-mer features, this construction method expands them to a set of $(k+1)$-mers by appending the letters of the alphabet to each k-mer feature.

Region-specific compositional features Splice-site sequences characteristically have a coding region and a non-coding region. For the acceptor splice-site sequences, the region of the sequence on the left of the splice-site position (upstream) is the non-coding region, and the region of the sequence from the splice-site position to the end of sequence (downstream) is the coding region. These regions may exhibit distinct compositional properties. In order to capture these differences we use *region-specific k-mers*.
Construction Method. The construction procedure of upstream and downstream k-mer features is the same as the general k-mer method, with the addition of region indicator.

Positional features Position-specific nucleotides are the most common features used for finding signals in the DNA stream data [12]. These features capture the correlation between different nucleotides and their relative positions. The *position specific k-mers* capture the correlations between k-adjacent nucleotides. At each position i in the sequence these features represent the substring appearing at positions $i, i+1, .., i+k$.
Construction Method. This construction method starts with an initial set of position-specific k-mer features and extends them to a set of position-specific $(k+1)$-mers by appending the letters of the alphabet to each position-specific k-mer feature.

Conjunctive positional features To capture the correlations between different nucleotides in nonconsecutive positions in the sequence, we propose *conjunctive position-specific features*. We construct these complex features from conjunctions of basic position-specific features. The dimensionality of this kind of feature is inherently high.
Construction Method. Given an initial set of k-conjuncts, this construction method selects from the set of basic position-specific features to add another conjunct in an unconstrained position, therefore constructing the set of $(k+1)$-conjuncts.

4.2 Feature Selection

Feature selection methods reduce the set of features by keeping only the useful features for the task at hand. The problem of selecting useful features has been the focus of extensive research and many approaches have been proposed [1–5].

In our experiments we consider several feature selection methods to reduce the size of our feature sets. We use several filter approaches including *Information Gain (IG), Chi-Square (CHI), Mutual Information (MI)*, *KL-distance (KL)* for initial pruning of feature types sets during the generation stage. Due to space limitations, in the experiments section, we present the combination that produced the best results. In our data, we found that mutual information performed best for selecting compositional features, chi-squared for positional features and information gain for conjunctive features. At the final collection step, we combine this with an embedded method based on recursive feature elimination [9] used in our final feature collection stage. The weights w_i of the decision boundary of a linear SVM can be used as feature weights to derive feature ranking. We use the C-Modified Least Squares (CMLS) classifier [13] and refer to this method as W-CMLS. We recursively train the classifier and remove low scoring features.

4.3 Feature Generation Algorithm (FGA)

The traditional feature selection approaches consider a single brute force selection over a large set of all features of all different types. By categorizing the features into different feature types we can apply appropriate construction and selection methods suitable to the different types. Thus we can extract relevant features from each feature type set more efficiently than if a singe selection method had been applied to the whole set. We use the following algorithm:

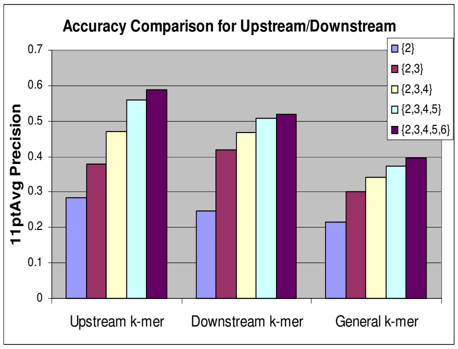
- *Feature Generation.* The first stage generates feature sets for each feature type. For each defined feature type, we tightly couple the corresponding feature construction step with a specified feature selection step. We iterate through these steps to generate richer and more complex features. During each iteration, we eliminate features that are assigned a low selection score by the feature selection method.
- *Feature Collection and Selection.* In the next stage, we collect the features of different types and apply another selection step.
- *Classification.* The last stage of our algorithm builds a classifier over the final set of features.

For this type of problem it is not unusual to spend a lot of computational resources, especially in the training phase. While feature generation remains a computationally intensive process, the organization of the generation process according to the different types allows us to search a much larger space efficiently. For the time complexity of the classification algorithm, we use CMLS which is very efficient. In addition, this feature generation approach has other advantages such as the flexibility to adapt with respect to the feature type and the possibility to incorporate the module in a generic learning algorithm.

5 Experimental Results for Splice-Site Prediction

We conducted a wide range of experiments using a variety of classifiers, and here we present a summary of them. We present results for the classifier that consistently gave the best results, CMLS.

We use the *11-point average* (11ptAvg) [14] to evaluate the performance of our algorithm. For any recall ratio, we calculate the precision at the threshold which achieves that recall ratio and compute the average precision. The 11ptAvg is the average of precisions estimated at recall values 0%, 10%, 20%, ., 100%. The ability of our algorithm to discriminate true acceptor site sequences from normal sequences is evaluated also using Receiver Operating Characteristic (ROC) curve analysis Another performance



(a)

No. features	IG	MI	Chi	KL
9000	87.44	87.39	87.41	87.44
8000	87.40	86.79	87.35	87.33
7000	87.14	85.24	87.15	87.19
6000	86.51	82.92	86.51	86.68
5000	85.93	80.33	85.94	85.89
4000	83.69	75.46	83.57	83.87
3000	82.71	66.63	82.50	82.97
2000	76.64	67.49	80.28	68.65

(b)

Fig. 1: (a)Comparison between different feature type sets performances, upstream k-mers, downstream k-mers, and general k-mers shown for different k (b)11ptAvg precision results for FGA varying the feature set size of position-specific collection of k-mers through different feature selection methods

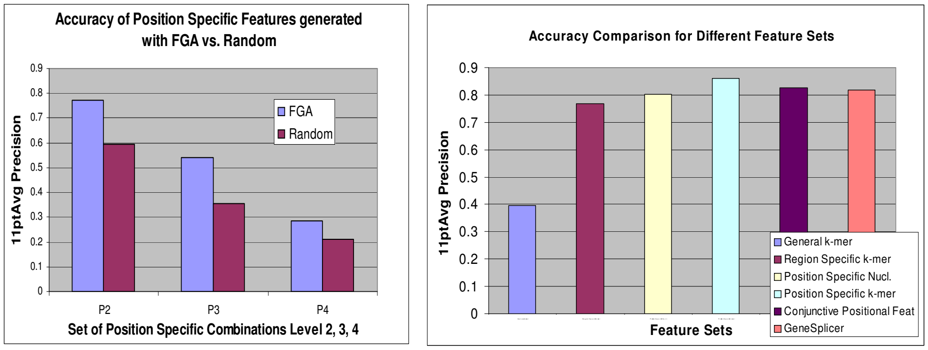
measure commonly used for biological data is the *false positive rate(FPr)* defined as $FPr = \left(\frac{FP}{FP+TN}\right)$ where FP, and TN are the number of false positives and true negatives respectively. FPr can be computed for all recall values by varying the decision threshold of the classifier. We also present results using this measure. In all our experiments, the results reported use three-fold cross-validation.

5.1 Accuracy Results of FGA

We begin with a brief evaluation of the effectiveness of the different feature types used in isolation.

Compositional features and region-specific compositional features We examine each k-mer feature set independently for each value of k from 2 to 6. Figure 1(a) shows the accuracy results for the region-specific k-mers and the general k-mer feature sets as we collect them after each iteration. In our experiments, *MI* selection method worked best for compositional features. We notice that k-mer features carry more information when they are associated with a specific region (upstream or downstream) and this is shown by the significant increase in their 11ptAvg precisions.

Positional features Next, we examine each *position-specific k-mer* feature set independently. We explore k-values from 1 to 6. The prediction results for this feature type (data not shown) after each generation step gradually increase until level 3, then gradually drop. This can be explained with the exponential increase in the number of features after each level. In Figure 1(b), we use feature selection to have a mix of position-specific k-mers for k values from 1 to 3. This table shows results of repeated selection for *IG, MI, CHI* and *KL* feature selection methods. Of these, *CHI* retains the highest precision among the four methods. Our paired-t tests for statistical significance reveal that these values although similar are statistically significant.



(a)

(b)

Fig. 2: a) 11ptAvg results for the position specific feature sets generated with FGA algorithm vs randomly generated features. b) Performance results of the FGA method for different feature types as well as the GeneSplicer program

Conjunctive positional features Finally, we examine conjunctive positional features. The number of these features grows exponentially and it is clearly very cumbersome to test for relevance more than 40 million unique combinations of triple conjuncts. We explore sets of 2 to 4 conjuncts denoted as $(P2, P3, P4)$. We use the *IG* selection method to select the top scoring $1,000$ features and repeat the generation on the selected set to get the next level. In Figure 2(a), we show the performances of the conjunctive feature sets. For comparison, we introduce a baseline method, which is the average of 10 trials of randomly picking $1,000$ conjunctive features from each level.

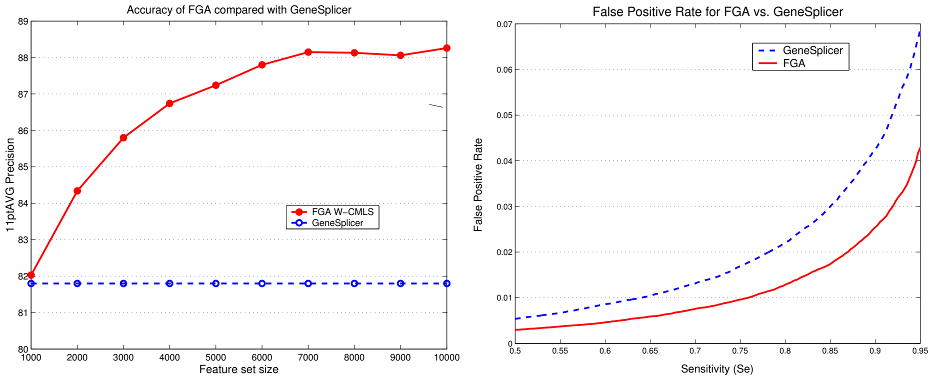
Summary Next, we compare collections of different levels of the feature sets of different types. The results are summarized in Figure 2(b).

Compositional features and region-specific compositional features The first two bars show the results for the best $2,000$ k-mer features for k ranging from 2 to 6. General k-mers result in an 11ptAvg of only 39.84%, while the result of the combined upstream and downstream k-mer features is 77.18%.

Position-specific k-mers The third bar shows the results for position specific nucleotides and the next bar shows $5,000$ position-specific k-mer features selected using the *CHI* selection method for k ranging from 1 to 3. The 11ptAvg precision is 85.94%.

Conjunctive positional features The next bar shows the results for a collection of $3,000$ conjunctive positional features for k ranging from 1 to 4 selected using *IG*. The 11ptAVG precision that this collection set gives is 82.67%. These results clearly show that using complex position-specific features is beneficial. Interestingly, these features typically are not considered by existing splice-site prediction algorithms.

Figure 2(b) also shows the performance of GeneSplicer on the same dataset. We see that even in isolation, our positional features and our conjunctive positional features perform better than GeneSplicer. These results are also statistically significant.



(a) (b)

Fig. 3: (a) 11ptAvg precision results for FGA varying the feature set size, compared to GeneSplicer (b) The false positive rate results for FGA varying the sensitivity threshold, compared to GeneSplicer

Results using the full feature-type collection In the following set of experiments, we show the results after we collect all the features that we have generated. We run our CMLS classification algorithm with a total feature set of size $10,000$ containing general k-mers, upstream/downstream k-mers, position-specific k-mers and conjunctive position-specific features. We achieve an 11ptAvg precision performance of 88.20%. This compares quite favorably with one of the leading programs in splice-site prediction, GeneSplicer, which yields an accuracy of 81.89% on the same dataset. The precision results at all individual recall points (data not shown) are consistently higher than those of GeneSplicer. In Figure 3(a) we explore more aggressive feature selection options and see that smaller feature sets of even $2,000$ also outperform GeneSplicer. In these experiments it is the more expensive W-CMLS selection method that we use in order to select a smaller working feature set.

We present the false positive rates for various recall values in Figure 3(b). Our feature generation algorithm, with its rich set of features, consistently performs better than GeneSplicer. Our false positive rates are favorably lower at all recall values. At a 95% sensitivity rate the FPr decreased from 6.2 to 4.3%. This significant reduction in false positive predictions can have a great impact when splice-site prediction is incorporated into a gene-finding program. It should also be noted that there is no significant difference in the running time of FGA compared to GeneSplicer. FGA performs a linear search (in terms of sequence length) along the given sequence in search for high scoring sites.

6 Conclusions

We presented a general feature generation framework which integrates feature construction and feature selection in a flexible manner. We showed how this method can be used to build accurate sequence classifiers. We presented experimental results for the problem of splice-site prediction. We were able to search over an extremely large space of feature sets effectively, and we were able to identify the most useful set of features of each type. By using this mix of feature types, and searching over combinations of them, we were able to build a classifier which achieves an accuracy improvement of 6.3% over an existing state-of-the-art splice-site prediction algorithm. The specificity values are consistently higher for all sensitivity thresholds and the false positive rate has favorably decreased. In future work, we plan to apply our feature generation algorithm to more complex feature types and other sequence prediction tasks, such as translation start site prediction.

7 Acknowledgments

This research was supported in part by an appointment to the National Center of Biotechnology Information Scientific Visitors Program sponsored by the National Library of Medicine and administered by the Oak Ridge Institute for Science and Education.

References

1. Kohavi, R., John, G.: The wrapper approach. In: *Feature Extraction, Construction and Selection : A Data Mining Perspective*, Liu,H.,Motoda,H.,eds. Kluwer Academic Publishers (1998)
2. Koller, D., Sahami, M.: *Toward optimal feature selection*. In: ICML. (1996) 284–292
3. Yang, Y., Pedersen, J.: *A comparative study on feature selection in text categorization*. In: ICML. (1997)
4. Yu, L., Liu, H.: *Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution*. In: ICML. (2003)
5. Blum, A., Langley, P.: *Selection of relevant features and examples in machine learning*. Artificial Intelligence (1997)
6. Liu, H., Wong, L.: *Data mining tools for biological sequences*. Journal of Bioinformatics and Computational Biology (2003)
7. Degroeve, S., Baets, B., de Peer, Y.V., Rouze, P.: *Feature subset selection for splice site prediction*. In: ECCB. (2002) 75–83
8. Yeo, G., Burge, C.: *Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals*. In: RECOMB. (2003)
9. Zhang, X., Heller, K., Hefter, I., Leslie, C., Chasin, L.: *Sequence information for the splicing of human pre-mRNA identified by support vector machine classification*. Genome Research **13** (2003) 2637–2650
10. Degroeve, S., Saeys, Y., Baets, B.D., Rouz, P., de Peer, Y.V.: *SpliceMachine: predicting splice sites from high-dimensional local context representations*. Bioinformatics **21** (2005) 1332–1338
11. Pertea, M., Lin, X., Salzberg, S.: *GeneSplicer: a new computational method for splice site prediction*. Nucleic Acids Research **29** (2001) 1185–1190
12. Zhang, M.: *Statistical features of human exons and their flanking regions*. Human Molecular Genetics **7** (1998) 919–932
13. Zhang, T., Oles, F.: *Text categorization based on regularized linear classification methods*. Information Retrieval **4** (2001) 5–31
14. Witten, I., Moffat, A., Bell, T., eds.: *Managing Gigabytes*. 2 edn. Van Nostrand Reinhold (1999)