An Evolutionary Algorithm Approach for Feature Generation from Sequence Data and Its Application to DNA Splice Site Prediction

Uday Kamath, Jack Compton, Rezarta Islamaj-Doğan, Kenneth A. De Jong, and Amarda Shehu

Abstract—Associating functional information with biological sequences remains a challenge for machine learning methods. The performance of these methods often depends on deriving predictive features from the sequences sought to be classified. Feature generation is a difficult problem, as the connection between the sequence features and the sought property is not known a priori. It is often the task of domain experts or exhaustive feature enumeration techniques to generate a few features whose predictive power is then tested in the context of classification. This paper proposes an evolutionary algorithm to effectively explore a large feature space and generate predictive features from sequence data. The effectiveness of the algorithm is demonstrated on an important component of the gene-finding problem, DNA splice site prediction. This application is chosen due to the complexity of the features needed to obtain high classification accuracy and precision. Our results test the effectiveness of the obtained features in the context of classification by Support Vector Machines and show significant improvement in accuracy and precision over state-of-the-art approaches.

Index Terms—Evolutionary computation, genetic programming, feature extraction and construction, classifier design and evaluation, data mining, DNA splice sites.

1 Introduction

Predicting information such as protein crystallizability, enzymatic activity, subcellular localization, and DNA splice sites continues to spur research in machine learning [36], [51], [34], [15], [16], [20], [18], [19], [17]. Inferring that a biological sequence exhibits a certain property is difficult when no a priori information is available on what gives rise to the sought property. Sequence-based classification aims to discover signals or features hidden in the sequence data that correlate with the sought property and discriminate between sequences that contain the property and those that do not.

Sequence-derived features can be global or local. For instance, biological insight that certain biophysical properties allow proteins to operate in certain cellular environments resulted in the discovery of amino-acid composition as a global feature strongly correlated with subcellular localization [12]. Biological insight can also reveal local features like the sequence motifs documented in the PROSITE database [9] that correlate well with protein domains, families, folds, and functional sites [41].

Insight from biological experts in a particular problem domain is difficult to translate into meaningful features when a combination of local and global features are needed. Many problems call for complex features [51], [34], [15], [16], [20], [18], [19], [17]. For instance, work in [15] shows that different types of features are needed to obtain high accuracy and precision in DNA splice site prediction. In absence of biological insight to guide feature generation and faced with the intractability of enumeration on a large number of features, both the number of feature types considered and the complexity of designed features are limited. Reduction techniques, such as Information Gain, Chi-Square, Mutual Information [31], and KL-distance [24], are additionally employed to further reduce the size of the feature set [16].

It is important to propose feature generation methods that are not limited by biological insight, the considered types of features, or the ability to enumerate features. The dilemma, of course, is that, by expanding the scope and complexity of the feature generation process, one is invariably confronted with an NP-hard problem [40].

A variety of general purpose search techniques are effective for NP-hard problems. In this paper, we explore the use of evolutionary algorithms (EAs) to search a large and complex feature space. The goal is to obtain features from sequence data that can significantly improve the classification accuracy of a Support Vector Machine (SVM). The proposed approach is evaluated on the difficult problem of DNA splice site prediction.

The basic idea is as follows: For most problem domains, there is some information on the basic building blocks of

- *U. Kamath is with the Department of Computer Science, George Mason University, 20929 Ivymount Terrace, Ashburn, VA 20147. E-mail: ukamath@gmu.edu.*
- *J. Compton is with the Barquin International, 5990 Founders Hill ct. #303, Alexandria, VA 22310. E-mail: jack.compton@gmail.com.*
- *R. Islamaj-Doğan is with the National Center for Biotechnology Information (NCBI), National Library of Medicine (NLM) at the National Institute of Health (NIH), 8600 Rockville Pike, Building 38A, 10N03D, Bethesda, MD 20894. E-mail: Rezarta.Islamaj@nih.gov.*
- *K.A. De Jong and A. Shehu are with the Department of Computer Science, George Mason University, 4400 University Dr., MSN 4A5, Fairfax, VA 22030. E-mail: {kdejong, amarda}@gmu.edu.*

Manuscript received 15 Dec. 2010; revised 1 Mar. 2012; accepted 25 Mar. 2012; published online 13 Apr. 2012.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-2010-12-0286. Digital Object Identifier no. 10.1109/TCBB.2012.53.

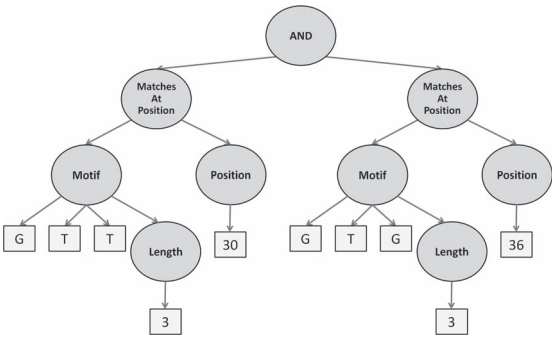


Fig. 1. The tree represents the feature: "GTT in position 30 AND GTG in position 36."

effective features. With DNA sequences, obvious blocks are k-mers (sequences of k nucleotides) and positional information. The goal is to construct effective classification features that are expressed as boolean combinations of basic building blocks. For example, one might specify the basic feature set to include all k-mers, $k \in \{3, \ldots, 6\}$, and positional information in the form of an integer in the range $[20, 50]$. A complex feature, like the one illustrated in Fig. 1, can then be constructed and evaluated on a given classification task.

What remains then is to describe how one can explore this large, open-ended feature space of complex compositions of a large set of simple primitives. Our approach uses a unique combination of evolutionary computation techniques. We use Genetic Programming (GP) techniques to evolve the kinds of structures illustrated in Fig. 1. Using an efficient fitness function, we identify a set of candidate features (a hall of fame) to be used as input to a standard SVM classification procedure.

We refer to this approach as FG-EA for Feature Generation with an Evolutionary Algorithm. The power of FG-EA is demonstrated on the DNA splice site prediction problem. This problem has been shown to require complex features [15]. Our results evaluate the effectiveness of the top features reported by FG-EA in the context of SVM classification. Given that FG-EA is a novel feature generation method for sequence-based classification, our primary direct comparisons are with state-of-the-art feature-based classification methods for DNA splice sites. The comparisons show that FG-EA features significantly improve the classification performance. Given that the problem of DNA splice site prediction is the focus of many other non feature-based methods due to its central role to gene finding, we also conduct direct comparisons with state-of-the-art kernel-based methods and achieve comparable performance.

The rest of this paper is organized as follows: In Section 1.1, we introduce the DNA splice site prediction problem and summarize relevant machine learning work. Section 1.2 summarizes related work in EAs and their applications on biological sequences. FG-EA is described in Section 2. Results obtained by the application of FG-EA on the DNA splice site prediction problem are presented in Section 3. A discussion of these results and analysis of the top features and their biological relevance follows in Section 4. The paper concludes in Section 5.

1.1 The DNA Splice Site Prediction Problem

Transcription of a eukaryotic DNA sequence into messenger RNA (mRNA) occurs only after enzymes splice away noncoding regions (introns) from the precursor (pre-mRNA) sequence to leave only coding regions (exons). For this reason, prediction of splice sites is a fundamental component of the gene-finding problem [10]. An acceptor splice site marks the start of an exon; a donor splice site marks the end. The sites have different consensus sequences. AG is a consensus dinucleotide among canonical acceptor splice sites, whereas GT is a consensus among canonical donor splice sites.

Splice site prediction is a difficult problem. AG and GT cannot be used as features due to their abundance in nonsplice site sequences. Nucleotide composition and coding and noncoding length and composition also do not make for discriminating features [36]. Early approaches employing positional probabilities fared poorly [47].

Recent state-of-the-art methods in splice site prediction are kernel based or feature based. Kernel-based methods like the ones in [45], [50], [39], [46] achieve some of the best performance in recognition of splice sites in a diverse list of species. Though not the primary focus of this paper, our experiments in Section 3 compare the classification performance that FG-EA features confer to an SVM to the performance reported in [46] by the weighted degree kernel (WD) and weighted degree kernel with shifts (WDS).

Feature-based methods focus on identifying discriminating features. The feature generation algorithm (FGA) in [15] is one of the most successful feature-based classification methods for splice site prediction. FGA expands upon the list of features of an earlier hallmark method, GeneSplicer [36], which included only position-specific nucleotides and upstream/downstream 3-mers.

FGA conducts a systematic search over features of different types. All k-mers ($2 \leq k \leq 6$) are considered due to their broad efficacy in feature-based classification [30], [34], [18], [19]. Region-specific (upstream/downstream) and positional compositional features (upstream or downstream) are also enumerated due to evidence that they are useful for finding signals in DNA stream data [21]. FGA systematically generates such features, even considering combinations through conjunction. Due to the large ensuing feature space, the features are regularly reduced to a top 5,000 before being expanded to include more features. The result is a high prediction performance and features that encode important biological signals [15], [16]. Success is attributed to the different types of features enumerated.

The FG-EA algorithm we propose here generalizes the process of feature generation from sequence data by not limiting the types of features considered. Such considerations, while restrictive for enumeration-based algorithms, can be handled well by evolutionary-based search methods. Indeed, FG-EA obtains features that afford an SVM classifier an average precision about 4 percent higher than FGA on cross-validation training data and 7 percent higher on test data. These features result in a classification performance that is similar to WD and WDS on cross-validation training data. In order to place the description of our novel FG-EA algorithm in context, we dedicate the next section to a brief summary of EAs and their demonstrated

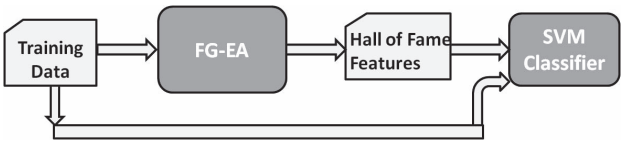


Fig. 2. The diagram shows the main steps we employ to predict DNA splice sites. The top features obtained after the exploration of the feature space with FG-EA allow transforming input sequences into feature vectors on which an SVM classifier can then operate.

ability to explore feature spaces. Our FG-EA algorithm is detailed next.

1.2 Related EA Work

EAs mimic biological evolution to evolve a population of candidate solutions toward the true solutions of a difficult optimization or search problem [6]. Their ability to explore exponentially large feature spaces makes them appealing methods for feature generation in addition to the classic enumeration and branch-and-bound techniques. The superiority of EAs was recognized early [43]. Since then, many studies have demonstrated the advantages of EAs for feature generation in different domains [2], [27], [38], [35], [14], [29], [20], [18], [17].

Recent applications of EAs to obtain predictive features from sequence data have shown success in diverse bioinformatics problems. Some of our recent work has shown significant improvements in classification accuracies when genetic algorithms (GA) replace k-mer feature enumeration techniques in predicting DNA hypersensitive and splice sites [18], [19], [17]. Work on predicting enzymatic activity in proteins additionally shows the power of EAs in feature generation [20].

Unlike standard GAs in which individual are fixed-length strings of symbols, an individual in GP is a variable-length tree composed of functions and variables. The functions are internal nodes also referred to as nonterminals, whereas the variables are the leaves also known as terminals. Originally introduced to evolve computer programs and complex functions [44], [4], [42], [25], GP algorithms allow evolving S-expressions that can be represented as parse trees [6].

Since their introduction, GP algorithms have seen an increase in their usage in diverse problems in bioinformatics [49], [33], [52], [5], [37], [20]. Abundant applications can be found in bioinformatics on quantitative structure-activity analysis in drug design, cancer classification from gene expression data, classification of genetically modified organisms, and classification of cognitive states from fMRI data [49], [33], [52], [5], [37], [32], [13], [28], [8]. This paper provides a new example of how GP techniques can be employed to generate predictive features from sequence data.

2 Methods

Our overall approach is shown in Fig. 2. The FG-EA algorithm generates complex features represented internally as GP trees and evaluates them on splice site training data using a surrogate fitness function. The top features are incrementally obtained via a "hall of fame" mechanism. The features in the hall of fame transform input sequence data into feature vectors. An SVM operating over the feature vectors finally allows evaluating the accuracy of the resulting classifier.

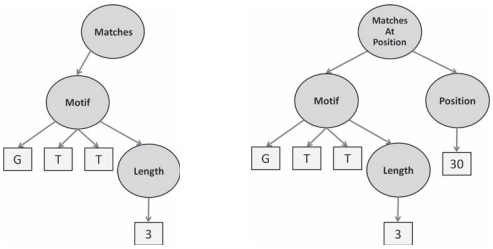


Fig. 3. Compositional (left) and positional (right) are just some of the features that can be constructed by FG-EA.

It is worth noting that this approach is generally applicable to sequence-based classification problems other than DNA splice site prediction. The training data and the application of the fitness function to evaluate features on the training data are the only components tied to the specific problem at hand.

2.1 The FG-EA Algorithm

The key element in the process illustrated in Fig. 2 is our FG-EA algorithm. FG-EA uses a standard GP algorithm to explore a large, complex space of potentially useful features. Features are represented as standard GP trees, and a population of features is evolved over time using standard GP mechanisms of mutation and crossover. Since constructing SVM classifiers is a computationally intensive process, FG-EA uses a surrogate fitness function to estimate the usefulness of the GP-generated features. A hall of fame mechanism incrementally collects the best estimated features for subsequent use with an SVM. A description of the main steps in our FG-EA follows (details are provided in the appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2012.53).

2.1.1 Feature Representation

One of the novel components of FG-EA is its effective representation of complex features without explicitly listing the feature types considered. GP individuals in a population can be complex constructs represented as parse trees [25]. Each internal node in a parse tree is a function, and its child subtrees form arguments to that function. In FG-EA, the leaves of a parse tree, also referred to as terminals, are either characters from the DNA alphabet {A, C, G, T} or integers corresponding to positions or motif (k-mer) length. The internal nodes are the operators `Length`, `Position`, `Motif`, `Matches`, `MatchesAtPosition`, `AND`, `OR`, and `NOT`.

Basic compositional features. The `Matches` operator allows constructing simple compositional features. An example is provided in Fig. 3. The nucleotides that make up the motif serve as leaves. The evaluation involves obtaining the occurrence of the motif in a given sequence. Since work in [15], [16] shows that no longer than 6-mers are useful for splice sites, we similarly limit motif length between 2 and 6.

Positional features. The `MatchesAtPosition` operator allows constructing simple positional features. An example is provided in Fig. 3. The positional features correspond to local features often employed in classification of biological

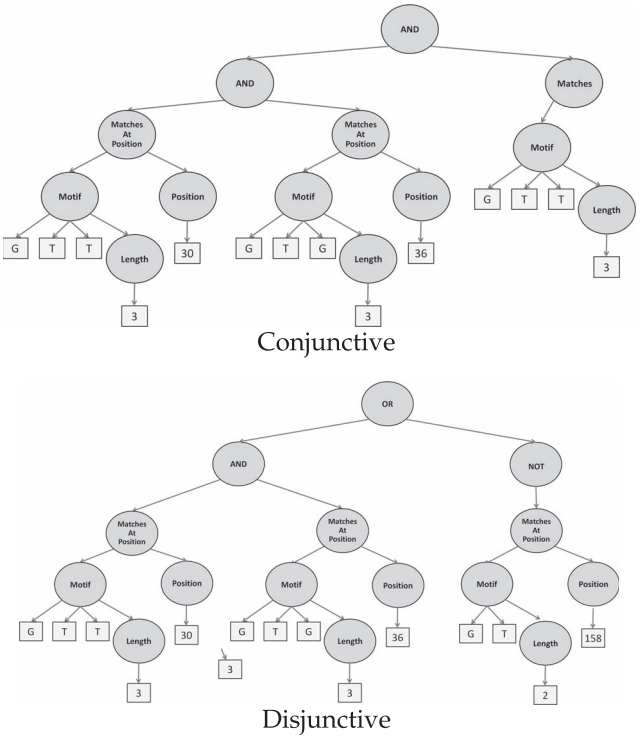


Fig. 4. The trees are examples of complex conjunctive and disjunctive features constructed by FG-EA.

sequences. In these features, the goal is to find a specific motif at a specific position in the sequence. It is important to note that the `Matches` and `MatchesAtPosition` operators are limited to operate directly over motifs and positions. The rest of the operators, `AND`, `OR`, and `NOT`, can only apply directly over one another, `Matches`, and `MatchesAtPosition`.

Correlational features. The parse tree representation allows constructing not only compositional and positional features, but also the region-specific compositional features shown to be important for DNA splice site prediction [15]. The `AND` operator allows specifying correlational features. An example is provided in Fig. 1. The region-specific compositional features employed in [15] are a subset of correlational features. Since the position ranges of the downstream and upstream regions are different, an AND operator over two positional features essentially results in region-specific compositional features.

Conjunctive and disjunctive features. FG-EA samples a much richer set of features than just correlational features. The operators `AND`, `OR`, and `NOT` allow constructing diverse conjunctive and disjunctive features. Examples of such complex features are shown in Fig. 4. For instance, the disjunctive feature shown specifies either finding two specific motifs in specific positions in the upstream region or not finding a specific motif in a specific position in the downstream region of a sequence. Repeated applications of `AND`, `OR`, and `NOT` can result in more complex features.

Ephemeral constants. The terminal elements are also referred to as ephemeral constants or ERCs (the squares in the features shown in Fig. 4). There are two ERC types in the parse trees FG-EA constructs, character ERCs (ERC-char) and integer ERCs (ERC-int). Table 1 lists all the ERCs, nonterminals and their arguments, return-types, and constraints.

TABLE 1
Table Shows the Nonterminals and Terminals Employed

Name	Args	Return Type	Constraints
`AND`	2 non-terminals	Boolean	
`OR`	2 non-terminals	Boolean	
`NOT`	2 non-terminals	Boolean	
`Matches`	Motif	Boolean	
`MatchesAtPosition`	Motif, Position	Boolean	
`Motif`	ERC-chars	Motif	
`Position`	ERC-int	Integer	$\{1, \ldots, 162\}$
`Length`	ERC-int	Integer	$\{2, \ldots, 6\}$
ERC-char		Character	$\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$
ERC-int		Integer	

2.1.2 Generating Features

Generation 0 consists of N randomly generated features using the well-known *ramped half-and-half* generative method [25] described in the appendix, available in the online supplemental material. Subsequent generations are evolved using standard GP selection, crossover, and mutation mechanisms. The process of evolving features continues for a fixed number of generations. The size of the population in a generation is not kept constant. An ever-decreasing population model is employed (see the appendix, available in the online supplemental material, for more details).

2.1.3 Fitness Function

The fitness function is key to achieving an efficient and effective EA search heuristic. Ideally, a "wrapper" approach would be employed [22], where a feature subset is fed for evaluation to a machine learning process. Accuracies obtained through sound empirical methodologies like k-fold validation would then be translated into fitness values. However, the wrapper approach is infeasible for large feature sets, large training sets, and scenarios like EAs where it needs to be employed multiple times. The "filter" approach is more practical [22]. Essentially, a simpler "surrogate" fitness function is designed to evaluate features in each generation. The subset of the fittest features after the EA terminates are then fed to the classifier for a more rigorous validation.

FG-EA employs the filter approach. Generated features are associated fitness values with a heuristic fitness function. A good fitness function is both simple and correlates well with the true objective function of the optimization problem at hand. Since the goal in feature-based classification is to improve precision while managing the discriminating power of features, we formulate the fitness function: $\mathrm{Fitness}(f) = \frac{C_{+,f}}{C_+} * \mathrm{IG}(f)$.

In this equation, f refers to a feature, $C_{+,f}$ is the number of positive (splice site) training sequences that contain the feature f, and C_+ is the total number of positive training sequences. Through the ratio $\frac{C_{+,f}}{C_+}$, the fitness function tracks only the occurrence of a feature in positive sequences, as negative sequences may not have any common features or signals. Moreover, the ratio $\frac{C_{+,f}}{C_+}$ is weighted by the information gain (`IG`) afforded by the feature f (see the appendix, available in the online supplemental material, for more details).

2.1.4 Hall of Fame

The ℓ fittest individuals of a generation are added to a hall of fame, which keeps the fittest individuals of each generation. Maintaining a hall of fame guarantees that fit individuals will not be lost or changed. We employ it for two reasons. First, the hall of fame serves as an external memory of the best individuals and allows maintaining diversity in the solution space. Second, the hall of fame represents the solution space at the end of a generational run and guarantees optimal performance [7].

2.2 Post FG-EA Feature Selection

The set of features in the hall of fame can be further narrowed through Recursive Feature Elimination (RFE) [53], [54], [16]. The main idea in RFE is to start with a large feature set and gradually reduce this set by removing the least successful features (according to some metric) until a stopping criterion is met. We employ RFE in the context of SVM classification, as in [11], using precision as the metric by which to determine whether a feature can be removed. We employ RFE in order to estimate the impact of feature set sizes on the precision and accuracy of the classification, as detailed in Section 3, and directly compare with existing work.

2.3 Support Vector Machines as Classifier

The FG-EA obtained feature allow transforming input sequences into feature vectors. The feature vectors associated with training sequences are employed to train an SVM classifier and estimate the discriminating power afforded by the top FG-EA features. Describing an SVM in great detail is not the focus of this paper, and we direct the reader to [48] for a detailed presentation. A brief description of SVMs is given in the appendix, available in the online supplemental material.

3 Materials

3.1 Data Sets

The experiments described in this paper show the efficacy of the features obtained through FG-EA in the context of classification and annotation by an SVM. We compare the classification performance to two different groups of state-of-the-art methods in splice site prediction, feature-based, and kernel-based. Our comparison with the feature-based methods FGA [15] and GeneSplicer [36] employs sequences extracted from the 2005 NCBI RefSeq collection of human pre-mRNA sequences (www.ncbi.nlm.nih.gov/). Our comparison with the kernel-based methods WD and WDS [46] employs sequences extracted from the worm data set (http://www.wormbase.org). Annotation of splice sites by FG-EA is carried out on a few selected human pre-mRNA sequences.

The 5,057 human pre-mRNA sequences in the NCBI RefSeq collection are annotated with exon start (acceptor) and end (donor) positions. The annotations are used to extract 51,008 positive (containing splice sites) and 200,000 negative sequences as in [15], [16]. Acceptor and donor splice site sequences (25,504 acceptor and 25,504 donor) consist of 162 nucleotides each, 80 nucleotides upstream of the annotated AG or GT dinucleotide, respectively, and 80 downstream ($80{+}AG/GT{+}80$). Negative sequences are 162 nucleotides long and centered around randomly selected AG/GT dinucleotides not annotated as splice sites. The significant difference in size between the negative and positive training data sets makes it harder for a classifier to obtain a high number of positive matches at random [15], [16].

This data set is employed to train an SVM and evaluate the top FG-EA features through classification in comparison with FGA and GeneSplicer. The FG-EA features are further validated on a testing data set, the B2hum 1,115 human pre-mRNA sequences employed to train GeneSplicer [36]. To show the applicability of FG-EA in annotation, five pre-mRNA sequences selected from the B2hum set are annotated with splice site information.

The worm data set is extracted from the worm genome and prepared in [46]. The genome is aligned through blat with all known cDNA sequences available at http://www.wormbase.org and all known EST sequences in [1]. A splicing graph representation built over the clustered alignments reveals a list of acceptor and donor splice sites. Using this list, 64,844 donor and 64,838 acceptor splice site sequences are extracted. Each sequence is 142 nucleotides long ($60{+}\mathrm{AG/GT}{+}80$) and centered around splice sites. Negative training sequences are also 142 nucleotides long and centered around nonsplice sites in intronic regions. In keeping with the worm data set employed in [46], 1,777,912 of these sequences are centered around nonsplice site AG dinucleotides, and 2,846,598 sequences are centered around nonsplice site GT dinucleotides.

3.2 Overview of Conducted Experiments

We first analyze the distribution of fitness values over generations to show that FG-EA converges fast to a high fitness value. The rest of the experiments evaluate FG-EA features in the context of SVM classification and lastly show the applicability of these features for the purpose of annotation. The annotation experiments employ the SVM trained on the human splice site training data set to annotate five pre-mRNA sequences selected from the B2hum testing data set. The classification experiments show results separately for acceptor and donor data in order to obtain a more detailed picture of performance and directly compare to other methods. Two sets of classification experiments are conducted, one that allows to compare the performance of FG-EA to FGA and GeneSplicer, and another that allows comparison with the WD and WDS methods.

The first set of classification experiments conduct a three-fold cross validation on the human data set described above. The SVM is trained over 2/3 of the data and tested on the remaining 1/3. This process is repeated three times to obtain an average performance. The entire experiment is repeated with 30 different sets of hall of fame features obtained from 30 different independent runs of FG-EA. Deviations in the measurements are insignificant, demonstrating that FG-EA reliably generates effective features. The obtained cross-validation results are compared with those of FGA and GeneSplicer. Finally, employing the feature set that yields the highest precision over the training

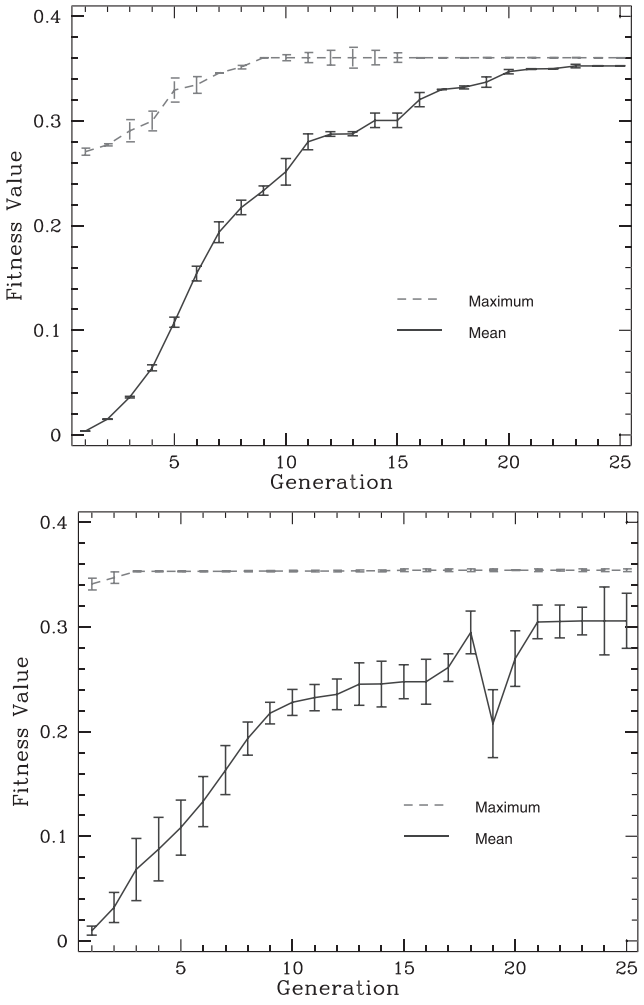


Fig. 5. Mean and maximum fitness values per generation (top: acceptor, bottom: donor) are averaged over 30 independent GP runs. Error bars are standard deviations.

data set, the trained SVM is applied to classify the B2hum testing data set.

The second set of classification experiments compare the performance of FG-EA over the worm data set to that of the kernel-based WD and WDS methods in [46] in the context of five-fold cross validation. Employing the entire worm data set for feature generation is infeasible, particularly when considering that we employ 30 independent runs of FG-EA and SVM evaluation of resulting features to obtain a measure of performance deviations due to stochasticity in FG-EA. For this reason, we sample smaller subsets from the overall worm data set, as detailed below.

First, to showcase the ability of FG-EA to train even on smaller data sets with similar or better performance, 40,000 sequences are sampled from the worm data set. Second, FG-EA performance is measured on larger data sets, where kernel-based methods have an advantage [46]. Ten different subsets of 360,000 sequences are randomly sampled from the worm data set without replacement, and the average performance is compared to WD and WDS. All sampled sets maintain the same ratio of acceptor/donor and positive/negative sequences as the entire worm data set. We note that parameters such as cost factor, kernel shift parameter, and degree were extensively evaluated in order to obtain the best performance out of WD and WDS on the sampled data sets. The values of these parameters can be found on our website (http://www.cs.gmu.edu/~ashehu/?q=OurTools).

3.2.1 Performance Measurements

We measure performance in terms of 11-point average precision (11ptAVG), false positive rate (FPR), area under receiver-operating-characteristic curve (auROC), and area under precision-recall curve (auPRC). An SVM labels and orders data from most to least confident. Given a confidence threshold, only the data above that threshold can be considered correctly labeled. For any recall ratio, precision can be calculated at the threshold which achieves that recall ratio (the reader is directed to [31] for a definition of recall and precision.). The 11ptAVG is the average of precisions calculated at 11 recall values $\{0\%, 10\%, \ldots, 100\%\}$. In addition to 11ptAVG, (PRCs) are employed to show the ability of FG-EA to discriminate true splice sites from other sequences. FPR is also computed for recall values by varying the confidence threshold to employ FPR-recall curves and show that FG-EA makes very few mistakes.

3.3 Evaluation of Fitness Quality and Convergence

Our implementation of FG-EA employs 25 generations. The distribution of fitness values of the features sampled by each generation can be visualized in terms of two statistics, the mean and maximum. To obtain a measure of deviations due to stochasticity in FG-EA, these two statistics can be tracked over 30 independent runs of FG-EA. Fig. 5 shows the average and standard deviation (over the 30 runs) of the mean and maximum fitness values per generation. The evaluation of features over acceptor and donor sequences is presented separately.

Fig. 5 shows convergence of fitness values around generation 20. Moreover, around this generation, the mean fitness value approaches the fitness of the best individual in the population. The steady state reached after generation 20 further validates the employment of the ever-decreasing population model in FG-EA. The model facilitates high exploration and diversity in the beginning while focusing FG-EA toward more exploitation with further generations.

3.4 Performance on Human Training Data Set

3.4.1 Precision versus Recall on Training Data Set

Fig. 6 plots and compares precision values corresponding to 11 recall points among GeneSplicer, FGA, and FG-EA. Precision values of FG-EA are averaged over 30 runs. Standard deviations are shown as error bars. Fig. 6 shows significant differences between FG-EA, GeneSplicer, and FGA in all precision values calculated at the 11 recall points. The break-even points on the PRCs for acceptor data are 54.9, 67.8, and 91.3 percent for GeneSplicer, FGA, and FG-EA, respectively. The break-even points for donor data are 58.7, 66.7, and 91.2 percent for GeneSplicer, FGA, and FG-EA, respectively. FG-EA shows significant improvements of 23.5 and 24.5 percent in the break even values for acceptor and donor splice sites, respectively. Table 2, which summarizes the PRCs by comparing 11ptAVG values, shows similar results. FG-EA outperforms GeneSplicer and FGA with

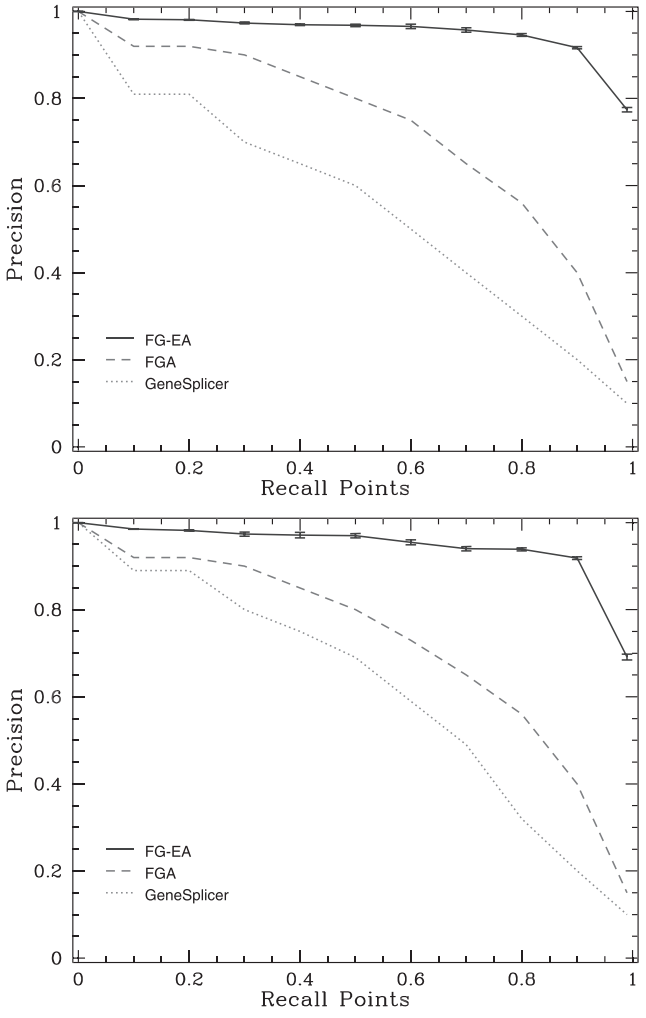


Fig. 6. Precision values are plotted over recall points (top: acceptor, bottom: donor). Values are averages over 30 FG-EA runs. Error bars are standard deviations.

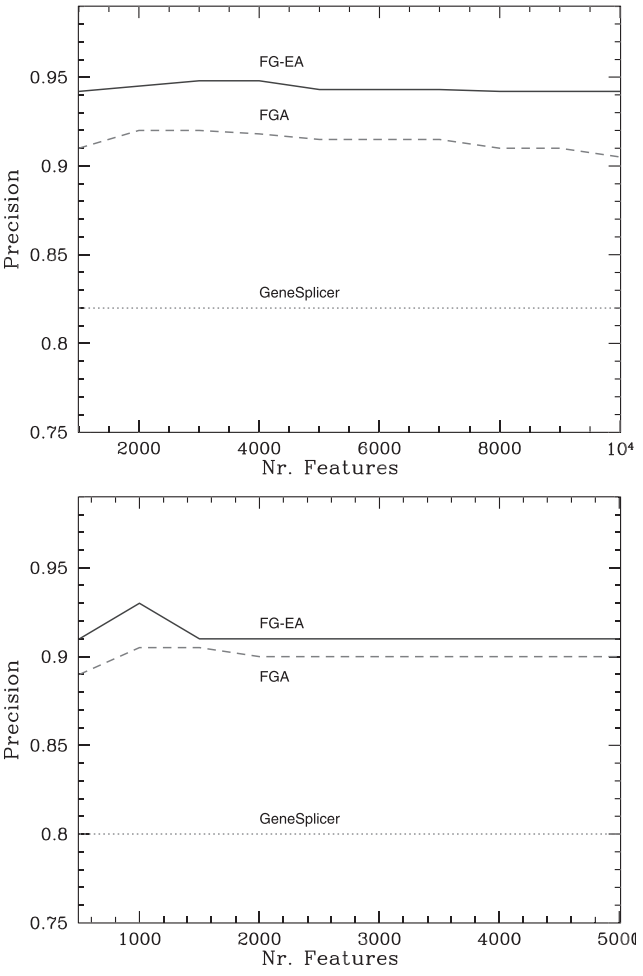


Fig. 7. Precisions are plotted over recall points (top: acceptor, bottom: donor) for RFE feature subsets. No RFE analysis is reported in GeneSplicer.

11ptAVG values of 94.89 and 93.69 percent for acceptor and donor data, respectively. Paired t-test shows the 11ptAVG values are statistically significant ($\alpha = 0.005$).

3.4.2 Precision on Training Data Over Reduced Feature Sets

We conduct the following experiment to show that the high classification performance that the FG-EA features confer to an SVM does not come from the sheer number of features. Given 5,000 hall of fame features, we employ RFE to repeatedly remove 500 least relevant features until 500 top features remain. In order to compare directly with the RFE analysis in [15], we expand our hall of fame to 10,000 features and remove 1,000 features at a time when evaluating on acceptor training data. Fig. 7, which plots the precisions obtained on the decreasing feature sets, shows that FG-EA confers higher precision than FGA and GeneSplicer at each feature subset. This suggests that FG-EA features are of high quality.

TABLE 2
Comparison of 11ptAVG Data

	GeneSplicer	FGA	FG-EA (μ, σ)
Acceptor	81.89	92.08	94.89, 0.35
Donor	80.1	89.08	93.86, 0.57

The average and standard deviation in column 4 are obtained over 30 FG-EA runs.

3.5 Performance on B2Hum Testing Data Set

We analyze the performance over the B2hum testing data set. AUC, the area under the receiver operating characteristic (ROC) curve for FG-EA over acceptor sequences is 99.41 percent compared to 99.37 and 98.71 percent for FGA and GeneSplicer, respectively. The FG-EA AUC score over donor sequences is 99.39 percent compared to 99.25 and 98.58 percent for FGA and GeneSplicer, respectively.

3.5.1 Precision versus Recall on Testing Data Set

PRCs are shown in Fig. 8. The break-even points on the curves for acceptor data are 55.2, 67.9, and 77.7 percent for GeneSplicer, FGA, and FG-EA, respectively. The break-even points for donor data are 58.53, 67.2, and 78.11 percent for GeneSplicer, FGA, and FG-EA, respectively. FG-EA shows significant improvements of 23.5 and 24.5 percent in the break-even values for acceptor and donor splice sites, respectively. FG-EA shows improvements of 9.8 and 10.9 percent in the break-even values for acceptor and donor splice sites, respectively. These results show that all

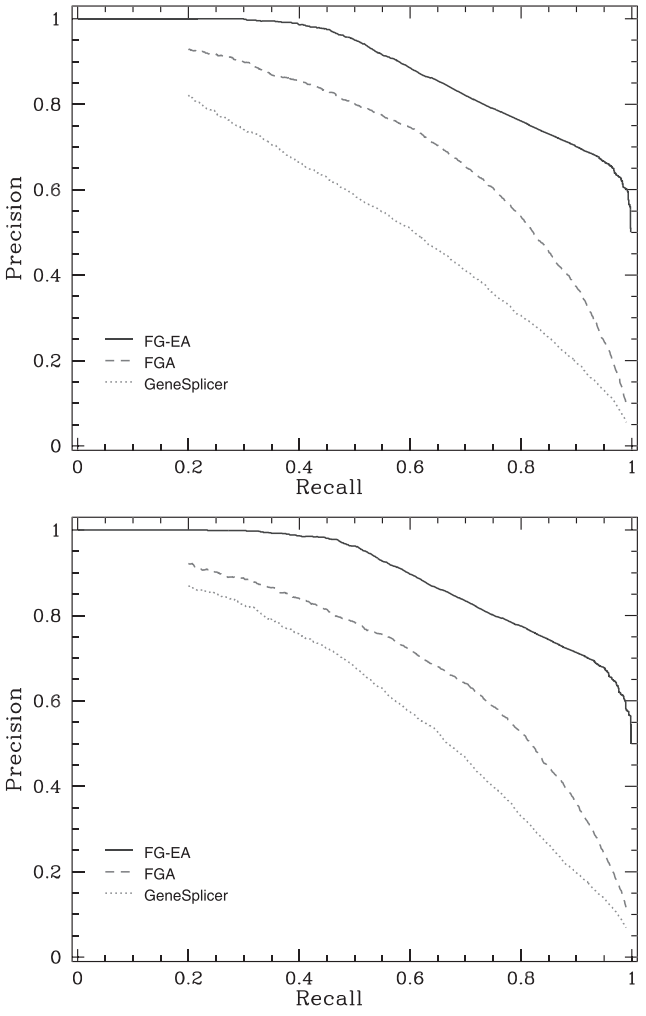


Fig. 8. Precision over recall (top: acceptor, bottom: donor) are plotted for the B2hum testing data set.

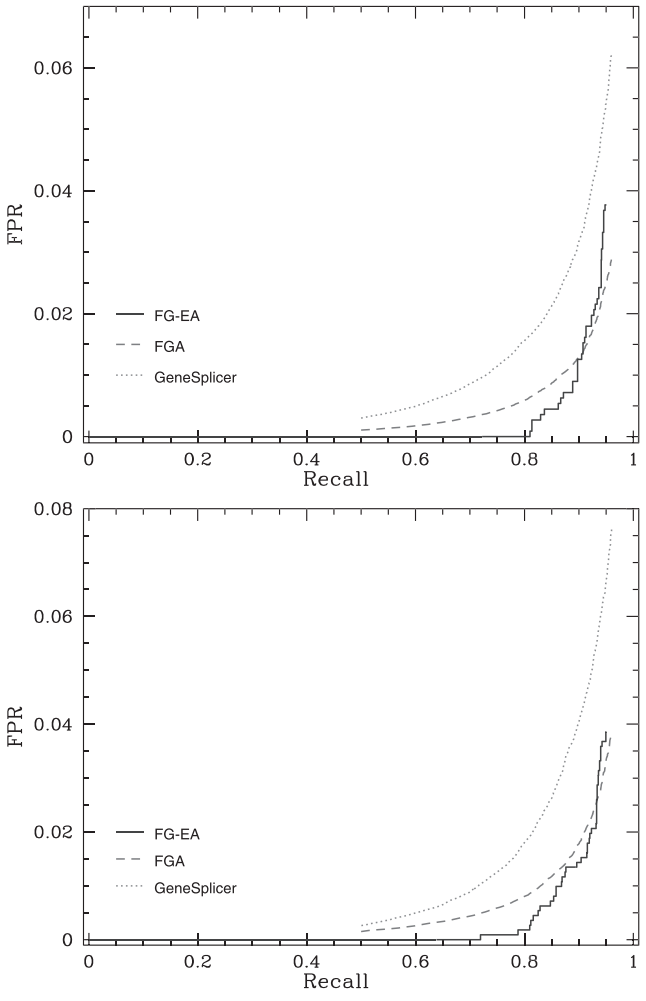


Fig. 9. FPR over recall (top: acceptor, bottom: donor) are plotted for the B2hum testing data set.

three methods achieve lower precision on the testing data compared to the results on the training data. On both training and testing data, FG-EA achieves higher precision.

3.5.2 FPR versus Recall on Testing Data Set

Fig. 9 compares the FPR versus recall curves among FG-EA, FGA, and GeneSplicer. At 95 percent sensitivity, FG-EA performs similar to FGA with an FPR of 3.7 percent over FGA's FPR of 3.3 percent. Both FPR values are significantly better than the 6.2 percent achieved by GeneSplicer. Having low FPR at high recall is particularly important when classifying testing data where the negative sequences significantly outnumber positive sequences.

3.6 Performance on Worm Training Data Set

Fig. 10 compares FG-EA to WD and WDS in [46] on 40,000 randomly sampled sequences from the worm data set in terms of PRCs obtained after the five-fold cross validation (acceptor and donor results are shown separately). The break-even points on the curves for acceptor data are 81.37, 86.89, and 91.1 percent for WD, WDS, and FG-EA methods, respectively. The break-even points for donor data are 86.2, 86.4, and 90.34 percent for the three methods, respectively. FG-EA shows improvements of 4.21 and 3.94 percent in the break-even values obtained over the acceptor and donor data, respectively. These results make the case that similar or slightly better results are obtained with FG-EA on small data sets. There is a slight degradation in accuracy, which we attribute to the bias toward precision in our fitness function.

These results make the case that FG-EA performs very well even when trained over small-size data sets. This is also demonstrated in Table 3, which summarizes the performance through measurements of auROC and auPRC values. We note that the results shown for FG-EA are averaged over 30 independent runs over the same subset in order to properly take into account the stochasticity of FG-EA.

The average performance of FG-EA on 10 different subsets of 360K randomly sampled sequences from the worm data set is compared to that of WD and WDS in [46] over these subsets. This performance is summarized in Table 4 in terms of auROC and auPRC values. The shown data make the case that the performance of FG-EA over the larger sampled data sets is comparable to that of WD and WDS. While accuracy is slightly lower due to the bias toward precision in our fitness function, the obtained precision is slightly higher in FG-EA.

3.7 Annotation Performance on B2Hum Data Set

Five pre-mRNA sequences are randomly selected from the B2Hum testing data set for annotation. A window of

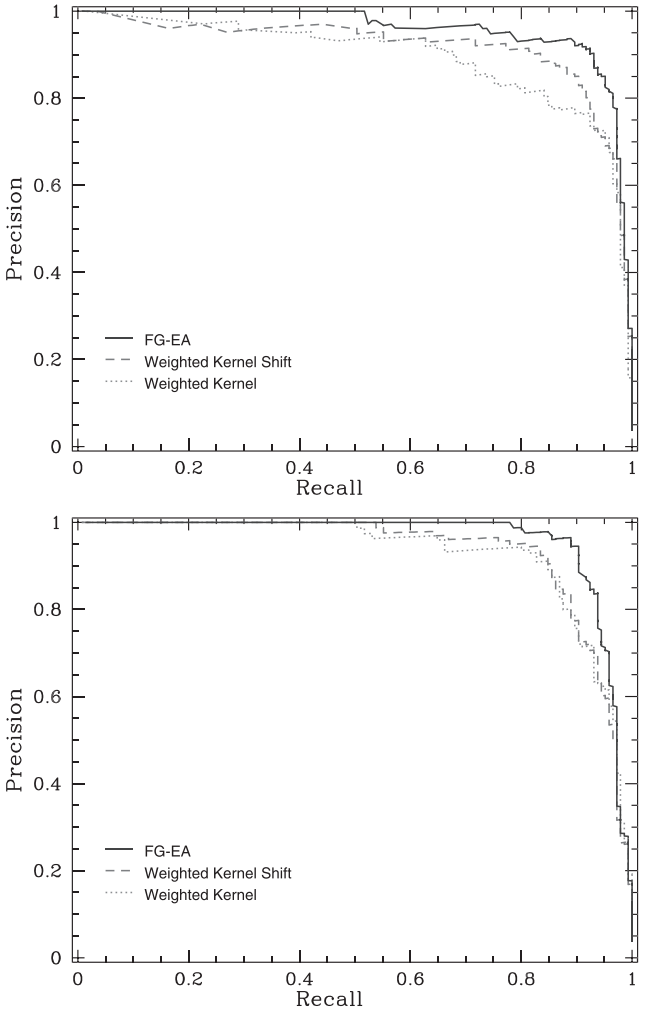


Fig. 10. Precision over recall (top: acceptor, bottom: donor) are plotted for the 40K subset sampled from the worm training data set.

TABLE 3
Comparison of auROC and auPRC Values on 40K Sequences Sampled from the Worm Data Set

	Acceptor				Donor			
	auROC		auPRC		auROC		auPRC	
	μ	σ	μ	σ	μ	σ	μ	σ
WD	99.2	0.3	86.7	1.2	99.1	0.2	87.1	0.3
WDS	99.3	0.2	89.1	0.8	99.1	0.1	88.6	0.2
FG-EA	98.7	0.2	97.1	0.7	98.8	0.4	96.7	0.8

Reported standard deviations are a result of the five-fold cross validation. Additional deviation for FG-EA results from 30 independent FG-EA runs on the same data set.

162 nucleotides is scanned with overlap of 161 nucleotides over each pre-mRNA sequence to obtain shorter sequences for classification. The SVM trained over the human splice site data set is then employed to classify each of the shorter sequences. The results of the classification are employed to annotate the pre-mRNA sequences with splice site information.

For brevity, annotation results are graphically shown on only one pre-mRNA sequence in Fig. 11 (the rest of the results can be viewed on our website (http://www.cs.gmu.edu/~ashehu/?q=OurTools)). Fig. 11 plots the SVM prediction scores for each of the windows. The high prediction scores (above 0.6) agree well with the known exon locations, also shown in the plot. The results shown in Fig. 11 further support

TABLE 4
Comparison of auROC and auPRC Values on 10 Different Sets of $360K$ Sequences Sampled from the Worm Data Sets

	Acceptor				Donor			
	auROC		auPRC		auROC		auPRC	
	μ	σ	μ	σ	μ	σ	μ	σ
WD	99.7	0.3	93.9	0.3	99.6	0.2	93.8	0.1
WDS	99.8	0.2	94.2	0.4	99.5	0.1	94.1	0.1
FG-EA	98.8	0.2	96.1	0.3	98.6	0.3	96.2	0.4

Reported standard deviations are a result of the five-fold cross validation and the different sets. Additional deviation for FG-EA results from the 30 independent FG-EA runs over a data set.

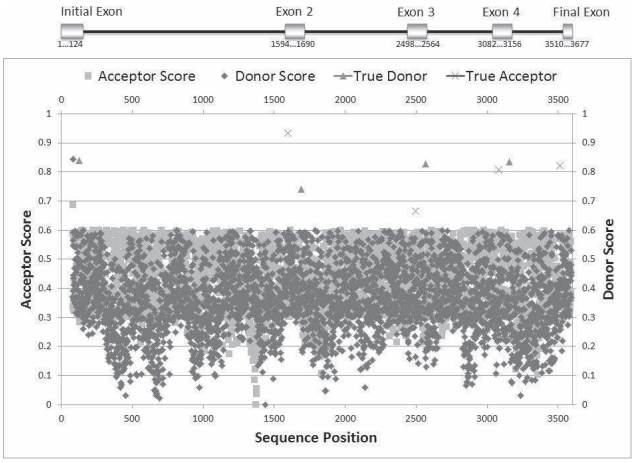


Fig. 11. Acceptor and donor prediction scores are shown on the left and right axes, respectively. The selected sequence is AB012229. High prediction scores agree well with the known exon end positions as shown on the plots.

TABLE 5
IG Sums of Subsets of Features Evaluated over Acceptor (Top) and Donor Data (Bottom)

Acceptor	Nr.	IG
Compositional	600	2.53
Positional, Correlational, Regional	2451	10.34
Conjunctive and Disjunctive	1949	21.78
Donor	**Nr.**	**IG**
Compositional	738	4.22
Positional, Correlational, Regional	2791	11.43
Conjunctive and Disjunctive	1471	36.23

the prediction power of our method and the general applicability of FG-EA for the purpose of annotation.

4 Discussion

It is interesting to analyze the type distribution of the top features obtained by FG-EA and measure the contribution of each type. We divide the hall of fame features in three types or subsets. One subset consists of all compositional features. The second subset consists of all region-specific compositional, positional, and correlational features. The third and final subset contains all remaining features and consists of conjunctive and disjunctive features. Table 5 breaks down the distribution of features into these three subsets.

The contribution of each feature subset to the performance detailed above is estimated by associating an IG value to each subset. The IG value of a subset sums the IG values of the features in a subset, assuming naive Bayes

```
(OR
 (MP (Motif (5) AGGCG) @ 84)
  (OR
   (MP (Motif (3) TCG) @ 47)
    (OR
     (OR
      (OR
       (OR
        (MP (Motif (2) GT) @ 85)
         (OR
          (MP (Motif (3) AGC) @ 79)
          (MP (Motif (2) AG)  @ 84) ) )
        (MP (Motif (3) GAG) @ 83) )
       (MP (Motif (2) TC) @ 47) )
      (MP (Motif (3) AGG) @ 84) ) ) )
                    (a)
(OR
 (AND
  (Match (Motif (3) AGC)
  (MP (Motif (2) GT) @ 85)
 )
 (OR
  (OR
   (MP (Motif (2) GT) @ 85)
   (MP (Motif (2) AG) @ 84) )
  (AND
   (MP (Motif (2) CC) @ 65)
   (MP (Motif (6) TAACCG) @ 151) ) ) )
                    (b)
```

Fig. 12. Generated feature examples.

independence. The distribution of IG values is also shown in Table 5. Evaluation of the features on acceptor and donor data is kept separate. Table 5 clearly shows that the largest increase in IG is attributed to the complex conjunctive and disjunctive features. This result further justifies the employment of GP in exploring complex and vast feature spaces. The improvements in classification of splice site sequences over FGA and GeneSplicer suggest that the complex conjunctive and disjunctive features are important to detection of splice sites.

A closer inspection of the hall of fame reveals the fittest features are complex rule sets. For instance, one of the fittest features on acceptor data is the pure disjunctive feature shown in Fig. 12a. This composite feature combines eight positional subfeatures with motif lengths from 2 to 5. The feature specifies these motifs to be found at various interesting locations. Note that the operator MatchesAtPosition is abbreviated as MP here. Other fit disjunctive features combine correlational and positional subfeatures. For instance, the feature shown in Fig. 12b is the result of correlational and positional features in downstream and upstream regions combined during evolution in FG-EA.

The fittest FG-EA features contain useful biological signals reported around splice sites [36], [15], [16]. Known signals in a typical pre-mRNA include the branch site, the pyrimidine-rich region, splice site consensus signals, and exonic splicing enhancers.

The mammalian branch-site signal is degenerate and shows low levels of purifying selection [23]. To identify such signals, we search for compositional features of six nucleotides 40 to 20 nucleotides upstream of the acceptor splice site. Our hall of fame contains such compositional features over motifs CTGACC, CCTGAC, CTTTT, etc. Similar features are also reported in [16]. FG-EA features additionally capture the acceptor splice site pyrimidine tract interval. Well-known positional tetramers, such as CTGA, CTTT, CTAA, and TTTT in this interval are present in the hall of fame and have high fitness values when evaluated over the acceptor training data.

Studies have suggested a potential role for the GGG and GGGG motifs in splicing [26]. The role of these motifs is validated by our FG-EA. The hall of fame contains compositional and positional features over these motifs. These features have high fitness values when evaluated over donor training data. Additionally, many A/C-rich motifs, such as CACACA, GCCCAA, CATTCA, CCTACA, can be found among FG-EA fittest features. Such motifs, originally described in [3] and additionally discovered in [16], have not been extensively characterized.

The IG analysis in Table 5 shows that additional, complex disjunctive, and conjunctive features play a significant role in discriminating splice sites from nonsplice sites. These features, some of them listed above, display complex biological signals that may be of interest to biologists for further characterization. For this reason, we have made the entire list of features in the hall of fame available on our website, http://www.cs.gmu.edu/~ashehu/?q=OurTools.

We note that Fig. 12 also illustrates the presence of redundant terms in the features. The FG-EA method does not concern itself regarding removal of redundancy during its evolutionary-based search, since redundancy does not affect classification accuracy. Some postprocessing can be conducted over the hall of fame features to improve their readability and analysis.

5 Conclusions

We have presented an evolutionary algorithm, FG-EA, which employs GP to automate the process of feature generation for feature-based classification. Detailed analysis of the discriminative power of the FG-EA features shows that FG-EA outperforms state-of-the-art feature generation methods in splice site classification. FG-EA reveals the significant role of novel complex conjunctive and disjunctive features. The abundance of disjunctive features shows that complex features are essentially rule sets that combine many small interesting rules in one complex feature. The combination of many small rules is known as the "Pitt-Approach" and has shown success in rule classification in various domains [44].

The proposed FG-EA algorithm can easily be employed in other prediction problems on biological sequences. Similar to our previous work on kernel GP evolution [17], further extensions of FG-EA can combine the evolution of features with evolution of SVM kernels for greater classification accuracy. Additionally, we plan on employing regular expressions to further combine and reduce the bloat in the expressions and so improve readability and performance.

The noted increases in time and memory during the SVM classification of large data sets (detailed in our website) can be addressed in the future by using distributed evaluations and sampling techniques. Additional future work can consider

incorporating shift-based positional comparisons in the features to further increase prediction power. On the other hand, kernel-based methods can also benefit by incorporating regional-, correlational-, conjunctive-, and disjunctive-based calculations when comparing two sequences.

ACKNOWLEDGMENTS

The authors would like to thank Sean Luke for useful discussions on GP, Gunnar Rätsch, Söeren Sonnenburg, and Sebastian Schultheiss for their support in running some of the kernel-based methods used for comparison in this work, and Chih-Jen Lin for discussions and help on running LibLinear on large data sets.

REFERENCES

[1] M.S. Boguski, T.M. Lowe, and C.M. Tolstoshev, "dbest-Database for 'Expressed Sequence Tags'," *Nature Genetics,* vol. 4, no. 4, pp. 332-333, 1993.

[2] F.A. Brill, D.E. Brown, and W.N. Martin, "Fast Genetic Selection of Features for Neural Networks," *IEEE Trans. Neural Networks,* vol. 3, no. 2, pp. 324-328, Mar. 1992.

[3] L.R. Coulter, M.A. Landree, and T.A. Cooper, "Identification of a New Class of Exonic Splicing Enhancers by in Vivo Selection," *Molecular Cellular Biology,* vol. 17, no. 4, pp. 2143-2150, 1997.

[4] N.L. Cramer, "A Representation for the Adaptive Generation of Simple Sequential Programs," *Proc. Int'l Conf. Genetics Algorithms and the Applications,* pp. 183-187, 1985.

[5] R.A. Davis, A.J. Chariton, S. Oehlschlager, and J.C. Wilson, "Novel Feature Selection Method for Genetic Programming Using Metabolomic ^{1}H NMR Data," *Chemometrics and Intelligent Laboratory Systems,* vol. 81, no. 1, pp. 50-59, 2005.

[6] K.A. De Jong, *Evolutionary Computation: A Unified Approach.* MIT Press, 2001.

[7] C.D. Dosin and R.K. Belew, "New Methods of Competitive Coevolution," *Evolutionary Computation,* vol. 5, no. 1, pp. 1-29, 1997.

[8] J.A. Driscoll, B. Worzel, and D. MacLean, "Classification of Gene Expression Data with Genetic Programming," *Genetic Programming: Theory and Practice,* R.L. Riolo and B. Worzel, eds., Kluwer, pp. 25-42, 2003.

[9] L. Falquet, M. Pagni, P. Bucher, N. Hulo, C.J.A. Sigrist, K. Hofmann, and A. Bairoch, "The PROSITE Database, Its Status in 2002," *Nucleic Acids Research,* vol. 30, no. 1, pp. 235-238, 2002.

[10] R. Guigo, P. Filcek, J. Abril, A. Reymond, J. Lagarde, F. Denoeud, S. Antonarakis, M. Ashburner, V. Bajic, E. Birney, R. Castelo, E. Eyras, C. Ucla, T. Gingeras, J. Harrow, T. Hubbard, S. Lewis, and M. Reese, "Egasp: The Human ENCODE Genome Annotation Assessment Project," *Genome Biology,* vol. 7, no. S2, pp. 1-31, 2006.

[11] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene Selection for Cancer Classification Using Support Vector Machines," *Machine Learning,* vol. 46, pp. 389-422, 2002.

[12] T. Habib, C. Zhang, J.Y. Yang, M.Q. Yang, and Y. Deng, "Supervised Learning Method for the Prediction of Subcellular Localization of Proteins Using Amino Acid and Amino Acid Pair Composition," *BMC Genomics,* vol. 9, no. Suppl 1, pp. S1-16, 2008.

[13] J.-H. Hong and S.-B. Cho, "Lymphoma Cancer Classification Using Genetic Programming," *Proc. Seventh European Conf. Genetic Programming (EuroGP),* pp. 78-88, 2004.

[14] J. Huang, Y. Cai, and X. Xu, "A Hubrid Genetic Algorithm for Feature Selection Wrapper Based on Mutual Information," *J. Pattern Recognition Letters,* vol. 28, pp. 1825-1844, 2007.

[15] R. Islamaj-Dogan, L. Getoor, and W.J. Wilbur, "A Feature Generation Algorithm with Applications to Biological Sequence Classification," *Computational Methods of Feature Selection,* H. Liu and H. Motoda, eds., pp. 355-376, Chapman and Hall, 2007.

[16] R. Islamaj-Dogan, L. Getoor, W.J. Wilbur, and S.M. Mount, "Features Generated for Computational Splice-site Prediction Correspond to Functional Elements," *BMC Bioinformatics,* vol. 8, pp. 410-416, 2007.

[17] U. Kamath, A. Shehu, and K.A. De Jong, "Feature and Kernel Evolution for Recognition of Hypersensitive Sites in DNA Sequences," *Proc. Int'l Conf. Bio-Inspired Models of Network, Information, and Computing Systems (BIONETICS),* LNICST, vol. 87, pp. 213-238, Springer, 2010.

[18] U. Kamath, K.A. De Jong, and A. Shehu, "Selecting Predictive Features for Recognition of Hypersensitive Sites of Regulatory Genomic Sequences with an Evolutionary Algorithm," *Proc. 12th Ann. Conf. Genetic and Evolutionary Computation,* pp. 179-186, 2010.

[19] U. Kamath, A. Shehu, and K.A. De Jong, "Using Evolutionary Computation to Improve SVM Classification," *Proc. IEEE World Conf. Evolutionary Computation,* 2010.

[20] A. Kernytsky and B. Rost, "Using Genetic Algorithms to Select Most Predictive Protein Features," *Proteins: Structure Function Bioinfomatics,* vol. 75, no. 1, pp. 75-88, 2009.

[21] W. Kim and W.J. Wilbur, "DNA Splice Site Detection: A Comparison of Specific and General Methods," *Proc. Assoc. Moving Image Archivists Symp,* pp. 390-394, 2002.

[22] R. Kohavi and G.H. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence,* vol. 97, nos. 1/2, pp. 273-324, 1997.

[23] G. Kol, G. Lev-Maor, and G. Ast, "Human-Mouse Comparative Analysis Reveals that Branch-Site Plasticity Contributes to Splicing Regulation," *Human Molecular Genetics,* vol. 14, no. 11, pp. 1559-1568, 2005.

[24] D. Koller and M. Sahami, "Toward Optimal Feature Selection," *Proc. Int'l Conf. Machine Learning,* pp. 284-292, 1996.

[25] J. Koza, *On the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

[26] J. Královicová and I. Vorechovsky, "Position-Dependent Repression and Promotion of DQB1 Intron 3 Splicing by GGGG Motifs," *J. Immunology,* vol. 176, no. 4, pp. 2381-2388, 2006.

[27] L.I. Kuncheva and L.C. Jain, "Nearest Neighbor Classifier: Simultaneous Editing and Feature Selection," *Pattern Recognition Letters,* vol. 20, nos. 11-13, pp. 1149-1156, 1999.

[28] W. Langdon and B. Buxton, "Genetic Programming for Mining DNA Chip Data from Cancer Patients," *Genetic Programming and Evolvable Machines,* vol. 5, no. 3, pp. 251-257, 2004.

[29] R. Leardi, R. Boggia, and M. Terrile, "Genetic Algorithms as a Strategy for Feature Selection," *J. Chemometrics,* vol. 6, no. 5, pp. 267-281, 2005.

[30] N.W. Leslie CS, E. Eskin, "The Spectrum Kernel: A String Kernel for SVM Protein Classification," *Proc. Pacific Symp. Biocomputing,* vol. 7, pp. 564-575, 2002.

[31] T.M. Mitchell, *Machine Learning,* first ed. Mc-Graw Hill Companies, Inc., 1997.

[32] J.H. Moore, J.S. Parker, N.J. Olsen, and T.M. Aune, "Symbolic Discriminant Analysis of Microarray Data in Autoimmune Disease," *Genetic Epidemiology,* vol. 23, no. 1, pp. 57-69, 2002.

[33] D.P. Muni, N.R. Pal, and J. Das, "Genetic Programming for Simultaneous Feature Selection and Classifier Design," *Ann. Rev. Genomics and Human Genetics,* vol. 36, no. 1, pp. 106-117, 2006.

[34] W.S. Noble, S. Kuehn, R. Thurman, M. Yu, and J.A. Stamatoyannopoulos, "Predicting the in Vivo Signature of Human Gene Regulatory Sequences," *Bioinformatics,* vol. 21, no. Suppl 1, pp. 338-343, 2005.

[35] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid Genetic Algorithms for Feature Selection," *IEEE Trans. Pattern Analysis and Machine Learning,* vol. 26, no. 11, pp. 1424-1437, Nov. 2004.

[36] M. Pertea, X. Lin, and S.L. Salzberg, "Genesplicer: A New Computational Method for Splice Site Prediction," *Nucleic Acids Research,* vol. 29, no. 5, pp. 1185-1190, 2001.

[37] R. Ramirez and M. Puiggros, "A Genetic Programming Approach to Feature Selection and Classification of Instantaneous Cognitive States," *Proc. European Workshop Evolutionary Computation in Image Analysis and Signal Processing (EvoWorkshop),* LNCS, vol. 4448, pp. 311-319, Springer, 2007.

[38] M.L. Raymer, W.F. Punch, E.D. Goodman, L.A. Kuhn, and A.K. Jain, "Dimensionality Reduction Using Genetic Algorithms," *IEEE Trans. Evolutionary Computing,* vol. 4, no. 2, pp. 164-171, July 2000.

[39] M.L. Raymer, W.F. Punch, E.D. Goodman, L.A. Kuhn, and A.K. Jain, *Accurate Splice Site Detection for Caenorhabditis Elegans,* pp. 277-298. MIT Press, 2004.

[40] R. Riviere, D. Barth, J. Cohen, and A. Denise, "Shuffling Biological Sequences with Motif Constraints," *J. Discrete Algorithms,* vol. 6, no. 2, pp. 192-204, 2007.

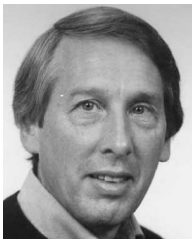
[41] L. Salwinski and D. Eisenberg, "Motif-Based Fold Assignment," *Protein Science,* vol. 10, no. 12, pp. 2460-2469, 2008.

[42] J. Schmidhuber, "Evolutionary Principles in Self-Referential Learning," PhD thesis, Technical Univ. Munich, 1987.
[43] W. Siedlecki and J. Sklansky, "A Note on Genetic Algorithms for Large-Scale Feature Selection," *Pattern Recognition Letters,* vol. 10, no. 5, pp. 335-347, 1989.
[44] S.F. Smith, A Learning System Based on Genetic Adaptive Algorithms," PhD thesis, Univ. of Pittsburgh, 1980.
[45] S. Sonnenburg, G. Rätsch, A. Jagota, and K. Müller, "New Methods for Splice-Site Recognition," *Proc Int'l Conf. Artificial Neural Networks,* pp. 329-336, 2002.
[46] S. Sonnenburg, G. Schweikert, P. Philips, J. Behr, and G. Rätsch, "Accurate Splice Site Prediction Using Support Vector Machines," *BMC Bioinformatics,* vol. 8, no. S10, article S7, 2007.
[47] R. Staden, "Methods to Locate Signals in Nucleic Acid Sequences," *Nucleic Acids Research,* vol. 12, no. 1, pp. 505-519, 1984.
[48] V.N. Vapnik, *Statistical Learning Theory.* Wiley & Sons, 1998.
[49] V. Venkatraman, A.R. Dalby, and Z.R. Yang, "Evaluation of Mutual Information and Genetic Programming for Feature Selection in QSAR," *J. Chemical Information and Computer Sciences,* vol. 44, no. 5, pp. 1686-1692, 2004.
[50] G. Yamamura and O. Gotoh, "Detection of the Splicing Sites with Kernel Method Approaches Dealing with Nucleotide Doublets," *Genome Informatics,* vol. 14, pp. 426-427, 2003.
[51] G. Yeo, "Maximum Entropy Modeling of Short Sequence Motifs with Applications to RNA Splicing Signals," *J. Computational Biology,* vol. 11, no. 2, pp. 377-394, 2004.
[52] J. Yu, J. Yu, A.A. Almal, S.M. Dhanasekaran, G.D., W.P. Worzel, and A.M. Chinnaaiyan, "Feature Selection and Molecular Classification of Cancer Using Genetic Programming," *Neoplasia,* vol. 9, no. 4, pp. 292-303, 2007.
[53] T. Zhang and F.J. Oles, "Text Categorization Based on Regularized Linear Classification Methods," *Information Retrieval,* vol. 4, no. 1, pp. 5-31, 2000.
[54] X.H. Zhang, K.A. Heller, I. Hefter, C.S. Leslie, and L.A. Chasin, "Sequence Information for the Splicing of Human Pre-mRNA Identified by Support Vector Machine Classification," *Genome Research,* vol. 13, no. 12, pp. 2637-2650, 2003.

Uday Kamath received the BS degree in electrical electronics from Bombay University in 1996 and the master's degree in computer science from the University of North Carolina at Charlotte in 1999. He is currently working toward the PhD degree in computer science at George Mason University. He also works as a technical architect in the Analytics and Detection group at Norkom Technologies concentrating on using machine learning, evolutionary software, and statistical modeling techniques in various fraud detection domains. His research interests include the applications of evolutionary computation methods to finance and to computational biology and bioinformatics. He is a member of the IEEE and the ACM.

Jack Compton received the BS degree in computer science from George Mason University in 2010. He is currently working as a software developer at Barquin International in Washington D.C. He was with George Mason University when the work presented in this paper was conducted. His research interests include the application of machine learning methods on bioinformatics problems.

Rezarta Islamaj-Doğan received the PhD degree in computer science from the University of Maryland at College Park, where she was a member of the LINQS Machine Learning Research Group. She is a research fellow in the Computational Biology Branch at the National Center for Biotechnology Information (NCBI/NLM/NIH). Her research interests encompass machine learning and data mining approaches for identifying useful information in biomedical databases. Her work on improving biomedical information retrieval focuses on understanding user needs and their search habits in PubMed. She is also interested in discovering and building domain appropriate features in order to model the biomedical information for accurate classification and prediction.



Kenneth A. De Jong is a professor of computer science and an associate director of the Krasnow Institute at George Mason University. His research interests include evolutionary computation, adaptive systems and machine learning. He is an active member of the evolutionary computation research community with a variety of papers, PhD students, and presentations in this area. He is also responsible for many of the workshops and conferences on evolutionary algorithms. He is the founding editor-in-chief of the journal *Evolutionary Computation* (MIT Press), and a member of the board of ACM SIGEVO. He is the recipient of an IEEE Pioneer award in the field of evolutionary computation and a lifetime achievement award from the Evolutionary Programming Society. He is a member of the IEEE.

Amarda Shehu received the PhD degree in computer science from Rice University in Houston, Texas, in 2008, where she was also an NIH fellow of the Nanobiology Training Program of the Gulf Coast Consortia. She is an assistant professor in the Department of Computer Science at George Mason University. She holds affiliated appointments in the Departments of Bioinformatics and Computational Biology and Bioengineering at George Mason University. Her research focuses on sequence- and structure-central problems in computational biology. Her work encompasses robotics-inspired probabilistic search frameworks for protein biophysics and evolutionary algorithms and machine learning for sequence analysis and design. She is a member of the IEEE and the ACM.