Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 23 (2013) 36 – 43

Procedia Computer Science

4th International Conference on Computational Systems-Biology and Bioinformatics, CSBio2013

Classification of eukaryotic splice-junction genetic sequences using averaged one-dependence estimators with subsumption resolution

Zaw Zaw Htike[a,]*, Shoon Lei Win[b]

[a]*Department of Electrical and Computer Engineering, Faculty of Engineering, IIUM, P.O. Box 10, 50728 Kuala Lumpur, Malaysia*
[b]*Department of Biotechnology Engineering, Faculty of Engineering, IIUM, P.O. Box 10, 50728 Kuala Lumpur, Malaysia*

Abstract

DNA is the building block of life, which contains encoded genetic instructions for building living organisms. Because of the fact that proteins are constructed in accordance with the genetic instructions encoded in DNAs, errors in RNA synthesis and translation into proteins can cause genetic disorders. Therefore, understanding and recognizing genetic sequences is one step towards the treatment of these genetic disorders. Since the discovery of DNA, there has been a growing interest in the problem of genetic sequence recognition, motivated by its enormous potential to cure a wide range of genetic disorders. The completion of the human genome project in the last decade has generated a strong demand in computational analysis techniques in order to fully exploit the acquired human genome database. This paper describes a state-of-the-art machine learning based approach called averaged one-dependence estimators with subsumption resolution to tackle the problem of recognizing an important class of genetic sequences known as eukaryotic splice junctions. To lower the computational complexity and to increase the generalization capability of the system, we employ a genetic algorithm to select relevant nucleotides that are directly responsible for splice-junction recognition. We carried out experiments on a dataset extracted from the biological literature. This proposed system has achieved an accuracy of 96.68% in classifying splice-junction genetic sequences. The experimental results demonstrate the efficacy of our framework and encourage us to apply the framework on other types of genetic sequences.

Keywords: Genetic sequence classification; eukaryotic splice-junction classification; AODEsr

* Corresponding author
E-mail address: zaw@ieee.org

doi:10.1016/j.procs.2013.10.006

1. Introduction

The nucleus of a human cell contains 46 chromosomes, each of which comprises a single linear molecule of deoxyribonucleic acid (DNA), which is intimately complexed with proteins in the form of chromatin[1]. DNA is the building block of life, which contains encoded genetic instructions for living organisms. Figure 1 depicts the process of RNA synthesis and protein translation. A DNA is transcribed to become a precursor mRNA, which is then spliced to become an mRNA, which is in turn translated to become a protein. Because of the fact that proteins are constructed in accordance with the genetic instructions encoded in DNAs, errors in RNA synthesis and translation into proteins can cause genetic disorders. Therefore, understanding and recognizing genetic sequences is one step towards the treatment of these genetic disorders[2-4]. Since the discovery of the DNA, there has been a growing interest in the problem of genetic sequence recognition, motivated by its enormous potential to cure a wide range of genetic diseases. The completion of the human genome project in the last decade has generated a strong demand in computational analysis techniques in order to fully exploit the acquired human genome database.

In this paper, we tackle the problem of recognizing an important class of genetic sequences known as *eukaryotic splice junctions*. Splice junctions are the points on a DNA sequence at which redundant DNA segments are removed during the process of protein creation[5]. As shown in Figure 1, a precursor mRNA contains exons and introns. During splicing, introns get spliced out while extrons get reunited[6-8]. This paper presents a classification system that can recognize splice junctions. In other words, for any given sequence of nucleotides, the system predicts whether the sequence belongs to an exon-intron (EI) boundary, intron-exon (IE) boundary, or neither of them. When split genes were discovered, the nucleotide sequences at the boundaries between introns and extrons were initially thought to be haphazard. However, soon after the discovery of split genes, researchers have started noticing patterns in the boundaries[9]. Therefore, pattern recognition and machine learning techniques can be utilized to recognize splice junctions. There have been some attempts to classify splice junctions using machine learning techniques. For example, Noordewier *et al.*[10] proposed a hybrid symbolic and connectionist approach called knowledge-based artificial neural network (KBANN) to recognize splice junctions. Essentially, KBANN utilizes a prior knowledge based hierarchically-structured rules and pattern recognition-based heuristics to recognize patterns. Chan *et al.*[11] developed a dependency graph model to fully capture the intrinsic interdependency between base positions in a splice site. Rätsch *et al.*[12]. came up with a novel SVM kernel to predict genetic sequences. Most of the existing approaches require prior knowledge bases. This paper describes a state-of-the-art machine learning based approach called averaged one-dependence estimators with subsumption resolution to tackle the problem of recognizing eukaryotic splice junctions without any prior knowledge.

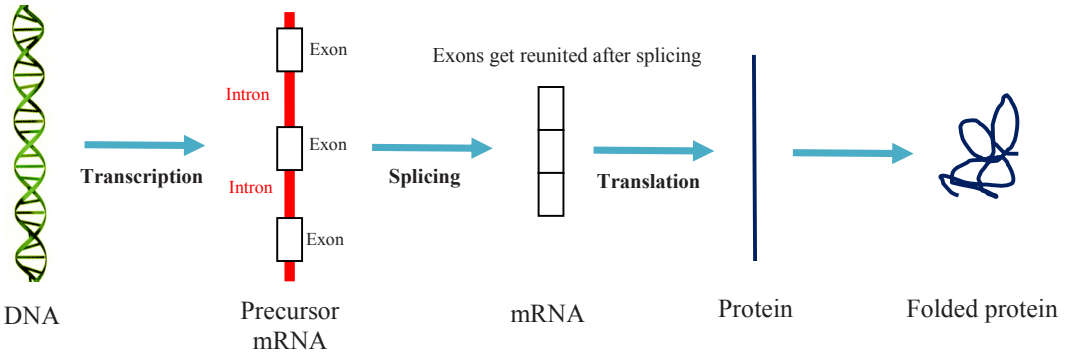

Fig. 1. RNA synthesis and translation into protein.

2. Eukaryotic splice-junction genetic sequence recognition

The goal of eukaryotic splice-junction genetic sequence classification is to predict, given a sequence of DNA, whether the sequence belongs to an intron/exon (IE) boundary, exon/intron (EI) boundary, or neither of them (N). We propose a two-layered framework which consists of nucleotide selection and junction classification as shown in Figure 2. The complexity of any machine learning classifier depends upon the dimensionality of the input data[13]. There is also a phenomenon known as the 'curse of dimensionality' that arises with high dimensional input data[14]. In the case of eukaryotic splice-junction genetic sequence classification, not all the nucleotides in a genetic sequence might be responsible for discriminating between IE and EI. Therefore, we employ a nucleotide selection process to select relevant nucleotides from a given genetic sequence in an unsupervised manner. Section 2.1 describes the process of nucleotide selection. After selecting relevant nucleotides, we perform splice-junction classification using the averaged one-dependence estimators with subsumption resolution (AODEsr). Section 2.2 describes the process of classification.

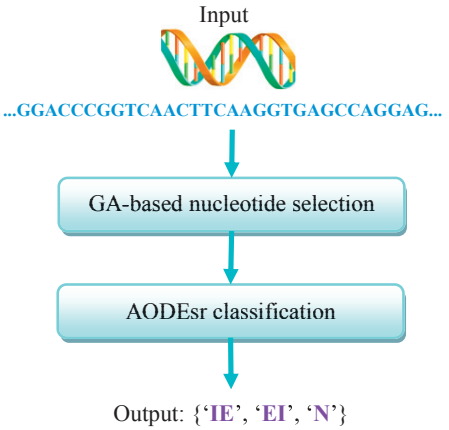

Fig. 2. High-level flow diagram of eukaryotic splice-junction genetic sequence classification framework.

2.1. Genetic algorithm-based nucleotide selection

The complexity of any machine learning classifier depends upon the dimensionality of the input data[13]. Generally, the lower the complexity of a classifier, the more robust it is. Moreover, classifiers with low complexity have less variance, which means that they vary less depending on the particulars of a sample, including noise, outliers, etc. [13]. In the case of splice-junction genetic sequence classification, not all the nucleotides in a genetic sequence might be responsible for discriminating between IE and EI. Therefore, we need to have a nucleotide selection method that chooses a subset of relevant nucleotides that can discriminate between IE and IE, while pruning the rest of the nucleotides in the input genetic sequence.

We are interested in finding the best subset of the set of nucleotides that can sufficiently discriminate splice junctions. Ideally, we have to choose the best subset that contains the least number of nucleotides that most contribute to the classification accuracy, while discarding the rest of the nucleotides. There are 2^n possible subsets that can arise from an n-nucleotide long genetic sequence. In essence, we have to choose the best subset out of 2^n possible subsets. Because performing an exhaustive sequential search over all possible subsets is computationally

expensive, we need to employ heuristics to find a reasonably good subset that can sufficiently discriminate splice junctions. There are generally two common techniques: forward selection and backward selection[13]. In forward selection, we start with an empty subset and add a nucleotide (that increases the classification accuracy the most) in each iteration until any further addition of a nucleotide does not increase the classification accuracy. In backward selection, we start with the full set of nucleotides and remove a nucleotide (that increases the classification accuracy the most) in each iteration until any further removal of a nucleotide does not increase the classification accuracy. There are also other types of heuristics such as scatter search[15], variable neighborhood search[16], and correlation-based feature selection[17].

We employ a nucleotide subset selection technique based on the genetic algorithm (GA) because the GA can work with multiple local optima and noisy or stochastic objective function unlike other search techniques. Figure 3 illustrates the operation of the GA-based nucleotide subset selection. A subset of nucleotides is encoded as an *n*-bit binary chromosome. A group of random chromosomes form an initial population. A fitness function then evaluates each chromosome in the population and produces a fitness score which represents an approximate accuracy that the corresponding subset of nucleotides would produce in actual splice-junction classification. Certain chromosomes in the population are chosen for reproduction in accordance with their fitness scores. Crossover and mutation operations are then performed for certain chromosomes in the population in accordance with the pre-defined cross-over probability and mutation probability values. Offspring are then generated and the whole procedure is repeated over and over again. Over thousands of generations, reasonably good subsets of nucleotides are left in the population. The chromosome in the final population with the largest fitness score is chosen as the final winner. This chromosome is decoded to obtain the final subset of nucleotides. The selected nucleotides are carried forward to classification.

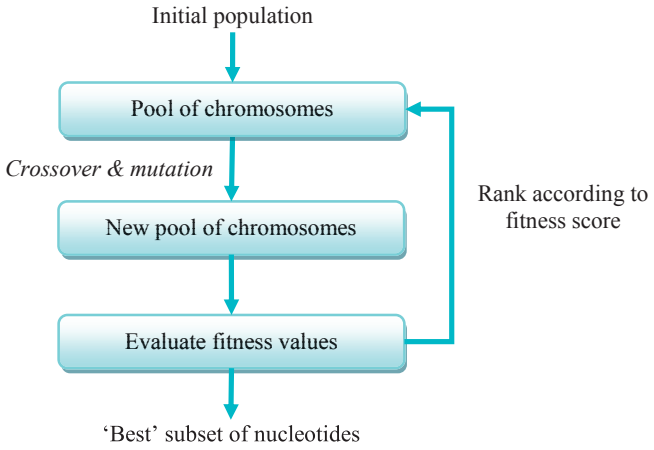

Fig. 3. Genetic algorithm-based nucleotide selection.

2.2. Classification

Naive Bayes (NB), which is fundamentally built on the strong independence assumption, is a very popular classifier in machine learning due to its simplicity, efficiency and efficacy[18-21]. There have been numerous applications of NB and variants thereof. The conventional NB algorithm uses the following formula for classification[22]:

$$Output = \underset{y}{\operatorname{argmax}} \left(P(y \mid x_1, \cdots, x_n) \right) \tag{1}$$

NB performs fairly accurate classification. The only limitation to its classification accuracy is the accuracy of the process of estimation of the base conditional probabilities. One clear drawback is its strong independence assumption which assumes that attributes are independent of each other in a dataset. In the field of genetic sequence classification, NB assumes that nucleotides are independent of each other in a genetic sequence despite the fact that there are apparent dependencies among individual nucleotides. Because of this fundamental limitation of NB, researchers have proposed various techniques such as one-dependence estimators (ODEs)[23] and super parent one-dependence estimators (SPODEs)[24] to ease the attribute independence assumption. In fact, these approaches alleviate the independence assumption at the expense of computational complexity and a new set of assumptions. Webb[18] proposed a semi-naive approach called averaged one-dependence estimators (AODEs) in order to weaken the attribute independence assumption by averaging all of a constrained class of classifiers without introduction of new assumptions. The AODE has been shown to outperform other Bayesian classifiers with substantially improved computational efficiency[18]. The AODE essentially achieves very high classification accuracy by averaging several semi-naive Bayes models that have slightly weaker independence assumptions than a pure NB. The AODE algorithm is effective, efficient and offers highly accurate classification. The AODE algorithm uses the following formula for classification[22]:

$$Output = \underset{y}{\operatorname{argmax}} \left(\sum_{i:\, 1 \le i \le n \,\wedge F(x_i) \ge m} P(y, x_i) \prod_{j=1}^{n} P\left(x_j \mid y,\, x_i\right) \right) \tag{2}$$

Semi-naive Bayesian classifiers attempt to preserve the numerous strengths of NB while reducing error by relaxing the attribute independence assumption[22]. Backwards sequential elimination (BSE) is a wrapper technique for attribute elimination that has proved to be effective at this task. Zheng *et al*.[22] proposed a new approach called *lazy estimation* (LE), which eliminated highly related attribute values at classification time without the computational overheads that are intrinsic in classic wrapper techniques. Their experimental results show that LE significantly reduces bias and error without excessive computational overheads. In the context of the AODE algorithm, LE has a significant advantage over BSE in both computational efficiency and error. This novel derivative of the AODE is called the averaged one-dependence estimators with subsumption resolution (AODEsr). In essence, the AODEsr enhances the AODE with a subsumption resolution by detecting specializations among attribute values at classification time and by eliminating the generalization attribute value[22]. Because the AODEsr has a very weak independence assumption, it is very suitable for classification of genetic sequences. Therefore, we employ an AODEsr classifier to recognize genetic sequences.

3. Experiments

We tested our proposed system using a dataset extracted from the biological literature[10]. The dataset contains 3190 samples, where roughly 25% of the samples belong to IE, 25% of the samples belong to EI and the rest of the samples are labeled as N and they belong to neither IE nor EI. Each sample contains a 60 nucleotide-long DNA sequence and a label of the category (IE, EI, or N) to which the sample belongs. The samples were obtained by taking the documented "split" genes from all primate gene entries in Genbank release 64.1 that were described as complete[10]. This procedure resulted in 751 examples of IE and 745 examples of EI. Negative samples were derived from similarly-sized windows, which did not cross an intron/exon boundary, sampled at random from these sequences[10]. We perform the GA-based nucleotide selection process on this dataset. We used the following parameters for the GA algorithm: crossover probability = 0.6, maximum number of generations = 20,000, mutation probability = 0.033, and population size = 20. The following 22 nucleotides came out as winners and were then selected as discriminating nucleotides: 6^{th}, 9^{th}, 12^{th}, 14^{th}, 16^{th}, 17^{th}, 18^{th}, 19^{th}, 20^{th} , 21^{st}, 23^{rd}, 24^{th}, 25^{th}, 28^{th}, 29^{th}, 30^{th}, 31^{st}, 32^{nd}, 33^{rd}, 34^{th}, 35^{th}, and 55^{th} (counting from left to right).

We carried out a 10-fold stratified cross validation where the dataset was partitioned into 10 equal-sized sub-datasets. Out of the 10 sub-datasets, a single sub-dataset was retained as the validation data for testing the model, and the remaining 9 sub-datasets were used as training data. The whole cross-validation process was then repeated 9 more times such that each of the 10 sub-datasets got used exactly once as the validation data. The results were then averaged over all the 10 trials. We used a critical value of 10, frequency limit of 3, an M-estimate weight value of 0.5 for the AODEsr model.

Table 1 lists the summary of the 10-fold stratified cross-validation results. The system correctly classified a total of 3084 instances out of 3190 instances with an accuracy rate of 96.68% and an error rate of 3.32%. Kappa coefficient, which measures inter-rater agreement of predicted values with the true values over all the trials of the 10-fold cross-validation, was found to be 0.9462. It means that the individual predictions are quite consistent in multiple trials and that the proposed system is robust. MAE and RMSE were found to be 0.137 and 0.137 respectively, which were small. RAE and RRSE were found to be significantly large. However, the RAE and RRSE metrics are not very meaningful in the task of classification. Table 2 displays the detailed results by output class. One thing interesting to note is that both true positive (TP) rate and false positive (FP) rate for EI are higher than those for IE. This implies that the system produces more ‘EI’ output values than ‘IE’ output values. This is confirmed by a lower precision score for EI.

Table 1. 10-fold cross-validation results summary.

Metric	Value	
Correctly classified instances	3084	(96.6771 %)
Incorrectly classified instances	106	(3.3229 %)
Kappa coefficient	0.9462	
Mean absolute error (MAE)	0.0354	
Root mean squared error (RMSE)	0.137	
Relative absolute error (RAE)	8.6279 %	
Root relative squared error (RRSE)	30.2526 %	
Total number of instances	3190	

Table 2. Detailed results by output class.

Class	TP Rate	FP Rate	Precision	Recall	F-Score	ROC Area
EI	0.971	0.017	0.947	0.971	0.959	0.995
IE	0.958	0.016	0.951	0.958	0.955	0.994
N	0.969	0.017	0.984	0.969	0.976	0.994

Table 3 compares the accuracy of the proposed system with other machine learning models. The proposed system, an AODEsr classifier with a GA-based nucleotide selection process produced an average error rate of 3.32%. To find out the significance of the GA-based nucleotide selection process, we used an AODEsr classifier without a nucleotide selection process to predict all the samples from the same dataset. The AODEsr classifier alone produced an error rate of 3.47%. It implies that the GA-based nucleotide selection process does improve the overall accuracy of the system, albeit not rather significantly. It also implies that 22 out of 60 nucleotides are enough to discriminate splice junctions. The accuracy rate of the proposed eukaryotic splice-junction classification system using the AODEsr classifier with the GA-based nucleotide selection process seems to be higher than those of other classification systems.

Table 3. Accuracy benchmark.

Technique	Avg. Error Rate (%)
AODEsr with GA-based nucleotide selection	3.32
AODEsr without GA-based nucleotide selection	3.47
JRIP	5.61
J48 tree	5.64
SMO	6.58
KBANN	6.88
BACKPROP	7.26
PEBLS	7.53
ID3	11.1
COBWEB	12.1
PERCEPTRON	12.6
k-NN	17.95
Decision table	21.19

4. Conclusion

Understanding and recognizing genetic sequences is one step towards the treatment of a wide range of genetic disorders. We have presented a machine learning based approach to recognize nucleotide sequences. Conventional naïve Bayes classifiers cannot accurately recognize nucleotide sequences because of their unrealistic assumption that forbids dependencies among individual nucleotides. We employ a state-of-the-art machine learning approach called the averaged-on dependence estimator with subsumption resolution (AODEsr) to tackle the problem of recognizing an important class of genetic sequences known as eukaryotic splice junctions. Given a sequence of nucleotides, the system predicts whether the sequence is part of an extron-intron junction (EI), intron-extron (IE) junction, or neither one of them (N). To lower the computational complexity and to increase the generalization

capability of the system, we employ a genetic algorithm to select relevant nucleotides that are directly responsible for splice-junction recognition. We found 22 nucleotides that were responsible for splice-junction recognition. We have carried out experiments on a dataset extracted from the biological literature. This proposed system has achieved an accuracy of 96.68% in classifying splice-junction genetic sequences. The accuracy rate of the proposed system was found to be higher than those of other machine learning classifiers. The experimental results demonstrate the efficacy of our framework. As future work, we would like to extend this framework to recognize other types of genetic sequences, DNA sequence motifs, and structural motifs in proteins.

References

1. Colledge NR, Walker BR, and RalstonSH. *Davidson's Principles and Practice of Medicine*. 21st ed. 2010: Churchill Livingstone.
2. Tae, H-J et al. *A novel splice site mutation of the arginine vasopressin–neurophysin II gene identified in a kindred with autosomal dominant familial neurohypophyseal diabetes insipidus.* Molecular Genetics and Metabolism, 2005. **86**(1–2): p. 307-313.
3. Sadusky T, Newman AJ, and Dibb NJ. *Exon Junction Sequences as Cryptic Splice Sites: Implications for Intron Origin.* Current Biology, 2004. **14**(6): p. 505-509.
4. Kay PH and Ziman MR. *Alternate Pax7 paired box transcripts which include a trinucleotide or a hexanucleotide are generated by use of alternate 3′ intronic splice sites which are not utilized in the ancestral homologue.* Gene, 1999. **230**(1): p. 55-60.
5. Towell GG, Craven M, and Shavlik JW. *Constructive Induction in Knowledge-Based Neural Networks*. in *8th International Workshop on Machine Learning*. 1991.
6. Wang L et al. *Observations on novel splice junctions from RNA sequencing data.* Biochemical and Biophysical Research Communications, 2011. **409**(2): p. 299-303.
7. Rekha TS and Mitra CK. *Comparative Analysis of Splice Site Regions by Information Content.* Genomics, Proteomics & Bioinformatics, 2006. **4**(4): p. 230-237.
8. Bruno C et al. *A Splice Junction Mutation in the αMGene of Phosphorylase Kinase in a Patient with Myopathy.* Biochemical and Biophysical Research Communications, 1998. **249**(3): p. 648-651.
9. Mount SM. *A catalogue of splice junction sequences.* Nucleic Acids Research, 1982. **10**(2): p. 459-472.
10. Noordewier MO, Towell GG, and Shavlik JW. *Training Knowledge-Based Neural Networks to Recognize Genes in DNA Sequences.* Advances in Neural Information Processing Systems, 1991. **3**.
11. Chen T-M, Lu C-C, and Li W-H. *Prediction of splice sites with dependency graphs and their expanded bayesian networks.* Bioinformatics, 2005. **21**(4): p. 471-482.
12. Rätsch G, Sonnenburg S, and Schäfer C. *Learning Interpretable SVMs for Biological Sequence Classification.* MC Bioinformatics 2006. **7**.
13. Alpaydin E. *Introduction to Machine Learning*. 2nd ed. 2010: The MIT Press.
14. Bishop CM. *Pattern Recognition and Machine Learning*. 2007: Springer.
15. García López F et al. *Solving feature subset selection problem by a Parallel Scatter Search.* European Journal of Operational Research, 2006. **169**(2): p. 477-489.
16. García-Torres M et al. *Solving Feature Subset Selection Problem by a Hybrid Metaheuristic*, in *First International Workshop on Hybrid Metaheuristics*. 2004. p. 59–68.
17. Hall MA. *Correlation-based Feature Selection for Machine Learning*, in *Computer Science*. 1999, University of Waikato: Hamilton, NewZealand.
18. Webb GI, Boughton JR, and Wang Z. *Not So Naive Bayes: Aggregating One-Dependence Estimators.* Machine Learning, 2005. **58**(1): p. 5-24.
19. Hand D and Yu K. *Idiot's Bayes---Not So Stupid After All?* International Statistical Review, 2001. **69**(3): p. 385-398.
20. Domingos P and Pazzani M. *On the Optimality of the Simple Bayesian Classifier under Zero-One Loss.* Mach. Learn., 1997. **29**(2-3): p. 103-130.
21. Rish I. *An empirical study of the naive Bayes classifier*. in *IJCAI-01 workshop on "Empirical Methods in AI"*.
22. Zheng F and Webb GI. *Efficient lazy elimination for averaged one-dependence estimators*, in *Proceedings of the 23rd international conference on Machine learning*. 2006, ACM: Pittsburgh, Pennsylvania. p. 1113-1120.
23. Sahami M. *Learning Limited Dependence Bayesian Classifiers*. in *Second International Conference on Knowledge Discovery and Data Mining*. 1996: AAAI Press.
24. Yang Y et al. *Ensemble Selection for SuperParent-One-Dependence Estimators*, in *AI 2005: Advances in Artificial Intelligence*, S. Zhang and R. Jarvis, Editors. 2005, Springer Berlin Heidelberg. p. 102-112.