J. Biomedical Science and Engineering, 2016, 9, 53-57
http://www.scirp.org/journal/jbise
ISSN Online: 1937-688X
ISSN Print: 1937-6871

The Research on Identification of Gene Splice Sites by Support Vector Machine

Hongbin Li, Guangzhong He

Medical School, Xianyang Vocational and Technical College, Xianyang, China
Email: leehbin@126.com

How to cite this paper: Li, H.B. and He, G.Z. (2016) The Research on Identification of Gene Splice Sites by Support Vector Machine. *J. Biomedical Science and Engineering*, **9**, 53-57.
http://dx.doi.org/10.4236/jbise.2016.910B007

Received: August 29, 2016
Accepted: September 20, 2016
Published: September 23, 2016

Abstract

The recognition of splicing sites is a very important step in the eukaryotic DNA sequence analysis. Many scholars are working hard to improve the accuracy of identification. Our team carried out research on this issue based on support vector machine, which is one famous algorithm in data mining. The training and testing data is from the HS^{3}D dataset, and excellent accuracy rate is achieved by nucleic acid sequence orthogonal coding and RBF core function, and the cross validation experiment hints that base pattern information is mainly located within 20 nucleotides upstream and downstream splice sites.

Keywords

Splicing Sites, Recognition, Support Vector Machine

1. Introduction

Genomics is a discipline focusing on biological genomes and utilization of gene. The recognition of splicing sites is one major research direction of genomics. The eukaryotic DNA sequence is composed of a certain number of exons and introns. During transcription of one gene, exons will be kept and assembled together, while introns will be cut liking the waste. A large number of molecular biology experiments reveal that the splicing sites between exon and intron follows Chambon's rule (GT-AG) [1], that the splice site between exon to intron is two base GT, and the splice site between intron to exon is two base AG. However, the GT-AG is not a universal truth, most GT and AG sites in DNA sequences are not splicing sites of gene. With DNA sequencing of eukaryotic species genomic, the recognition of splicing sites by software is urgent and knotty task for bioinformatics scholars. So far, many algorithm based on different principle have appeared since the 90's of the last century. The well-known algorithms for the recognition of splicing sites are Brunak's Neural network [2], Henderson and Lukashin's hidden markov model [3] [4], Cai's Bayes networks [5], Zhang's discriminant analysis [6], and Sun and Zhang's support vector machine [7] [8]. Each algorithm above has its own character, and promotes the identification of gene splicing sites towards more accurate direction.

2. Materials and Methods

Support vector machine (SVM) is a machine learning method based on Vapnik statistical learning theory [9], which was developed in the 90's of last century. SVM is mainly used for classification, regression and other learning tasks. It shows advantages in solving small sample, nonlinear and high dimensional pattern recognition. SVM is the method based on inductive learning. Its principle can be divided into two steps. The first step is building set a train data set with a number of features and classification value with 1 or −1. The second step is inducing one classification decision function relative to training data above with the minimization of the empirical risk, where the classification value of 1 of the data corresponds to a class, the classification value of −1 data corresponding to another class. For SVM, the primary concern is to establish an optimal classification hyper plane with minimum generalization error. The dimension of the hyper plane corresponds to the characteristic number of the training data. The left of the super plane corresponds to the first class, and the right of the super plane corresponds to the second class. As an example with only two features, the support vector machine classification principle is shown as **Figure 1**.

Some mature SVM application systems appeared in academic circle at the beginning of this century, LIBSVM software package is currently the world's most widely popular SVM software. It can put different types of features placed under the same framework

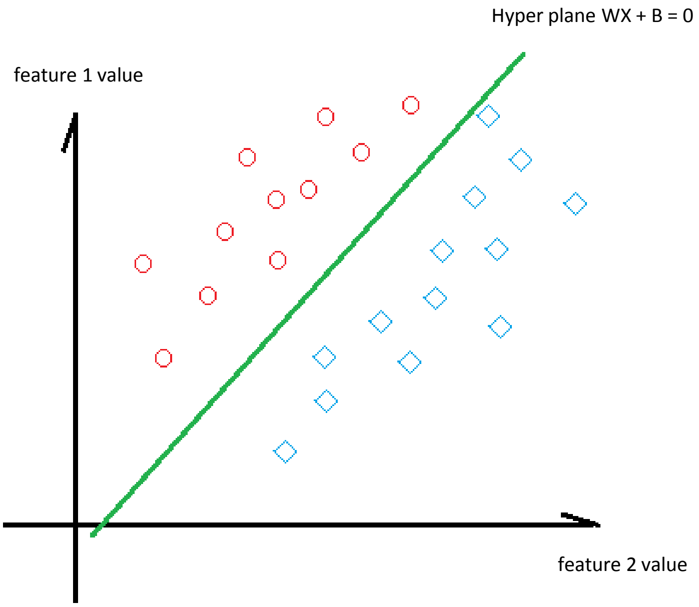
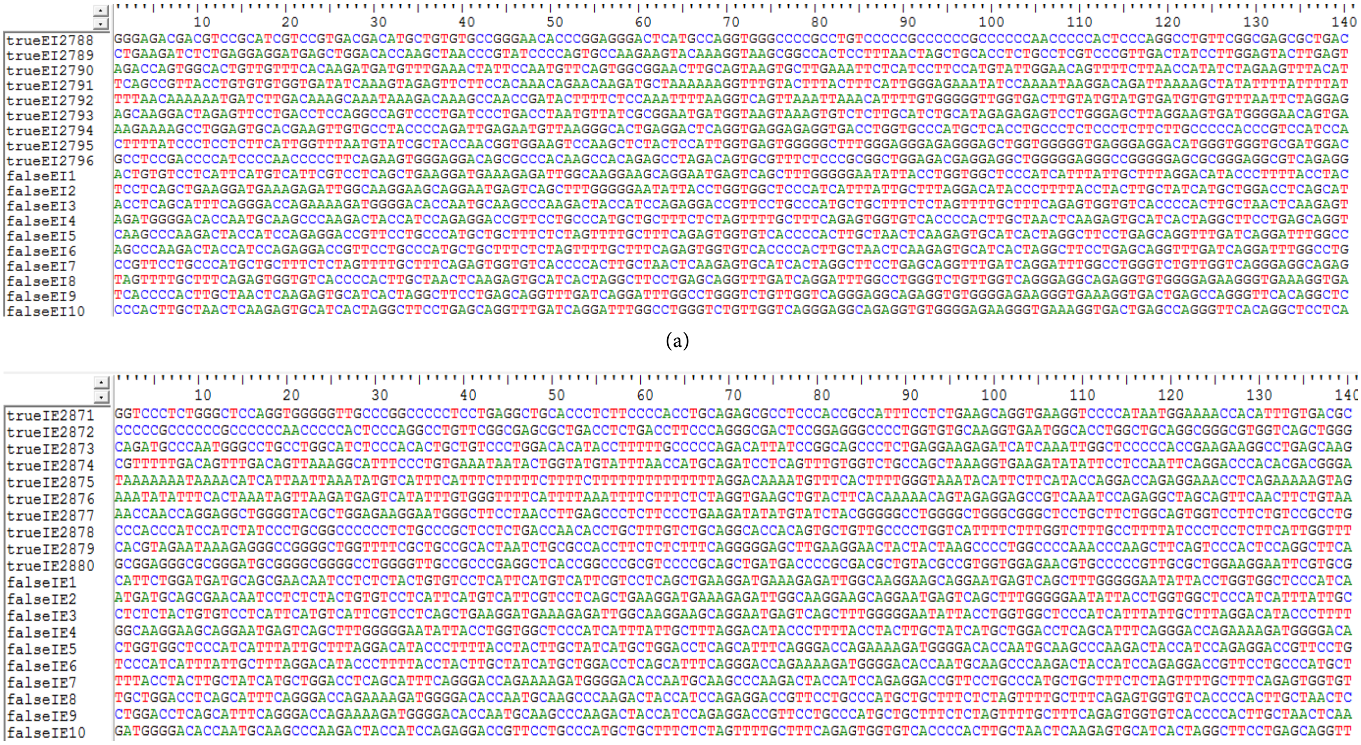


Figure 1. The principle of classification using SVM with two features.

for training and prediction, users no longer need to master the complex algorithm of SVM, and only need to provide training data in a certain format and call several SVM subroutine. The SVM training and testing data is from the HS^{3}D dataset, and more accurate SVM RBF core function is used [10]. It includes two parts, the first part of data set are sequences from exon to intron (GT, EI), and the second part of data set are sequences from intron to exon (AG, IE). Both part also include two part, one part correspond to splicing sites, and the second part correspond to false splicing sites. The sequence length of HS^{3}D is 140 nucleotides, and double-nucleotide GT or AG are located in the center of sequence. After removing the duplicate sequences and sequences with heterozygous sites in dataset, the GT training data include 2876 true IE sequences and 12004 false IE sequences, and AG training data include 2796 true EI sequences and 9998 false EI, as showed on **Figure 2**. The reason to select the sample size is that if sample size is too small, accuracy rate of prediction will decline, else if sample size is are too large, it is easy to exceed the processing ability of SVM software and make slow the speed of training and prediction. Then, pattern of nucleic acid sequence orthogonal coding was selected as the feature building for SVM [11]. Removing of characters GT or AG in every sequence of training set, the remaining sequence length is 138. The every nucleotide in this sequence can be coding with a four-dimensional vector. Base A is coded as vector (0, 0, 0, 1), base T is coded as vector (0, 0, 1, 0), base C is coded as vector (0, 1, 0, 0), and base G is coded as vector (1, 0, 0, 0). They are pairwise orthogonal,



```
                     10        20        30        40        50        60        70        80        90        100       110       120       130       140
trueEI2788  GGGAGACGACGTCCGCATCGTCCGTGACGACATGCTGTGTGCCGGGAACACCCGGAGGGACTCATGCCAGGTGGGCCCCGCCTGTCCCCGCCCCCCGCCCCCCAACCCCCACTCCCAGGCCTGTTCGGCGAGCGCTGAC
trueEI2789  CTGAAGATCTCTGAGGAGGATGAGCTGGACACCAAGCTAACCCGTATCCCCAGTGCCAAGAAGTACAAAGGTAAGCGGCCACTCCTTTAACTAGCTGCACCTCTGCCTCGTCCCGTTGACTATCCTTGGAGTACTTGAGT
trueEI2790  AGACCAGTGGCACTGTTGTTTCACAAGATGATGTTTGAAACTATTCCAATGTTCAGTGGCGGAACTTGCAGTAAGTGCTTGAAATTCTCATCCTTCCATGTATTGGAACAGTTTTCTTAACCATATCTAGAAGTTTACAT
trueEI2791  TCAGCCGTTACCTGTGTGTGGTGATATCAAAGTAGAGTTCTTCCACAAACAGAACAAGATGCTAAAAAAGGTTTGTACTTTACTTTCATTGGGAGAAATATCCAAAATAAGGACAGATTAAAAGCTATATTTTATTTTAT
trueEI2792  TTTAACAAAAAATGATCTTGACAAAGCAAATAAAGACAAAGCCAACCGATACTTTTCTCCAAATTTTAAGGTCAGTTAAATTAAACATTTTGTGGGGGTTGGTGACTTGTATGTATGTGATGTGTGTTTAATTCTAGGAG
trueEI2793  AGCAAGGACTAGAGTTCCTGACCTCCAGGCCAGTCCCTGATCCCTGACCTAATGTTATCGCGGAATGATGGTAAGTAAAGTGTCTCTTGCATCTGCATAGAGAGAGTCCTGGGAGCTTAGGAAGTGATGGGGAACAGTGA
trueEI2794  AAGAAAAGCCTGGAGTGCACGAAGTTGTGCCTACCCCAGATTGAGAATGTTAAGGGCACTGAGGACTCAGGTGAGGAGAGGTGACCTGGTGCCCATGCTCACCTGCCCTCTCCCTCTTCTTGCCCCCACCCGTCCATCCA
trueEI2795  CTTTTATCCCTCCTCTTCATTGGTTTAATGTATCGCTACCAACGGTGGAAGTCCAAGCTCTACTCCATTGGTGAGTGGGGGCTTTGGGAGGGAGAGGGAGCTGGTGGGGGTGAGGGAGGACATGGGTGGGTGCGATGGAC
trueEI2796  GCCTCCGACCCCATCCCCAACCCCCTTCAGAAGTGGGAGGACAGCGCCCACAAGCCACAGAGCCTAGACAGTGCGTTTCTCCCGCGGCTGGAGACGAGGAGGCTGGGGGAGGGCCGGGGGAGCGCGGGAGGCGTCAGAGG
falseEI1    ACTGTGTCCTCATTCATGTCATTCGTCCTCAGCTGAAGGATGAAAGAGATTGGCAAGGAAGCAGGAATGAGTCAGCTTTGGGGGAATATTACCTGGTGGCTCCCATCATTTATTGCTTTAGGACATACCCTTTTACCTAC
falseEI2    TCCTCAGCTGAAGGATGAAAGAGATTGGCAAGGAAGCAGGAATGAGTCAGCTTTGGGGGAATATTACCTGGTGGCTCCCATCATTTATTGCTTTAGGACATACCCTTTTACCTACTTGCTATCATGCTGGACCTCAGCAT
falseEI3    ACCTCAGCATTTCAGGGACCAGAAAAGATGGGGACACCAATGCAAGCCCAAGACTACCATCCAGAGGACCGTTCCTGCCCATGCTGCTTTCTCTAGTTTTGCTTTCAGAGTGGTGTCACCCCACTTGCTAACTCAAGAGT
falseEI4    AGATGGGGACACCAATGCAAGCCCAAGACTACCATCCAGAGGACCGTTCCTGCCCATGCTGCTTTCTCTAGTTTTGCTTTCAGAGTGGTGTCACCCCACTTGCTAACTCAAGAGTGCATCACTAGGCTTCCTGAGCAGGT
falseEI5    CAAGCCCAAGACTACCATCCAGAGGACCGTTCCTGCCCATGCTGCTTTCTCTAGTTTTGCTTTCAGAGTGGTGTCACCCCACTTGCTAACTCAAGAGTGCATCACTAGGCTTCCTGAGCAGGTTTGATCAGGATTTGGCC
falseEI6    AGCCCAAGACTACCATCCAGAGGACCGTTCCTGCCCATGCTGCTTTCTCTAGTTTTGCTTTCAGAGTGGTGTCACCCCACTTGCTAACTCAAGAGTGCATCACTAGGCTTCCTGAGCAGGTTTGATCAGGATTTGGCCTG
falseEI7    CCGTTCCTGCCCATGCTGCTTTCTCTAGTTTTGCTTTCAGAGTGGTGTCACCCCACTTGCTAACTCAAGAGTGCATCACTAGGCTTCCTGAGCAGGTTTGATCAGGATTTGGCCTGGGTCTGTTGGTCAGGGAGGCAGAG
falseEI8    TAGTTTTGCTTTCAGAGTGGTGTCACCCCACTTGCTAACTCAAGAGTGCATCACTAGGCTTCCTGAGCAGGTTTGATCAGGATTTGGCCTGGGTCTGTTGGTCAGGGAGGCAGAGGTGTGGGGAGAAGGGTGAAAGGTGA
falseEI9    TCACCCCACTTGCTAACTCAAGAGTGCATCACTAGGCTTCCTGAGCAGGTTTGATCAGGATTTGGCCTGGGTCTGTTGGTCAGGGAGGCAGAGGTGTGGGGAGAAGGGTGAAAGGTGACTGAGCCAGGGTTCACAGGCTC
falseEI10   CCCACTTGCTAACTCAAGAGTGCATCACTAGGCTTCCTGAGCAGGTTTGATCAGGATTTGGCCTGGGTCTGTTGGTCAGGGAGGCAGAGGTGTGGGGAGAAGGGTGAAAGGTGACTGAGCCAGGGTTCACAGGCTCCTCA
```

(a)

```
                     10        20        30        40        50        60        70        80        90        100       110       120       130       140
trueIE2871  GGTCCCTCTGGGCTCCAGGTGGGGGTTGCCCGGCCCCCTCCTGAGGCTGCACCCTCTTCCCCACCTGCAGAGCGCCTCCCACCGCCATTTCCTCTGAAGCAGGTGAAGGTCCCCATAATGGAAAACCACATTTGTGACGC
trueIE2872  CCCCCGCCCCCCGCCCCCCAACCCCCACTCCCAGGCCTGTTCGGCGAGCGCTGACCTCTGACCTTCCCAGGGCGACTCCGGAGGGCCCCTGGTGTGCAAGGTGAATGGCACCTGGCTGCAGGCGGGCGTGGTCAGCTGGG
trueIE2873  CAGATGCCCAATGGGCCTGCCTGGCATCTCCCACACTGCTGTCCCTGGACACATACCTTTTTGCCCCCAGACATTATCCGGCAGCCCTCTGAGGAAGAGATCATCAAATTGGCTCCCCCACCGAAGAAGGCCTGAGCAAG
trueIE2874  CGTTTTTGACAGTTTGACAGTTAAAGGCATTTCCCTGTGAAATAATACTGGTATGTATTTAACCATGCAGATCCTCAGTTTGTGGTCTGCCAGCTAAAGGTGAAGATATATTCCTCCAATTCAGGACCCACACGACGGGA
trueIE2875  TAAAAAAATAAAACATCATTAATTAAATATGTCATTTCATTTCTTTTTCTTTTCTTTTTTTTTTTTTTAGGACAAAATGTTTCACTTTTGGGTAAATACATTCTTCATACCAGGACCAGAGGAAACCTCAGAAAAAGTAG
trueIE2876  AAATATATTTCACTAAATAGTTAAGATGAGTCATATTTGTGGGTTTTCATTTTAAATTTTCTTTCTCTAGGTGAAGCTGTACTTCACAAAAACAGTAGAGGAGCCGTCAAATCCAGAGGCTAGCAGTTCAACTTCTGTAA
trueIE2877  AACCAACCAGGAGGCTGGGGTACGCTGGAGAAGGAATGGGCTTCCTAACCTTGAGCCCTCTTCCCTGAAGATATATGTATCTACGGGGGCCTGGGGCTGGGCGGGCTCCTGCTTCTGGCAGTGGTCCTTCTGTCCGCCTG
trueIE2878  CCCACCCATCCATCTATCCCTGCGGCCCCCCTCTGCCCGCTCCTCTGACCAACACCTGCTTTGTCTGCAGGCACCACAGTGCTGTTGCCCCTGGTCATTTTCTTTGGTCTTTGCCTTTTATCCCTCCTCTTCATTGGTTT
trueIE2879  CACGTAGAATAAAGAGGGCCGGGGCTGGTTTTCGCTGCCGCACTAATCTGCGCCACCTTCTCTCTTTCAGGGGGAGCTTGAAGGAACTACTACTAAGCCCCTGGCCCCAAACCCAAGCTTCAGTCCCACTCCAGGCTTCA
trueIE2880  GCGGAGGGCGCGGGATGCGGGGCGGGGCCTGGGGTTGCCGCCCGAGGCTCACCGGCCCGCGTCCCCGCAGCTGATGACCCCGCGACGCTGTACGCCGTGGTGGAGAACGTGCCCCCGTTGCGCTGGAAGGAATTCGTGCG
falseIE1    CATTCTGGATGATGCAGCGAACAATCCTCTCTACTGTGTCCTCATTCATGTCATTCGTCCTCAGCTGAAGGATGAAAGAGATTGGCAAGGAAGCAGGAATGAGTCAGCTTTGGGGGAATATTACCTGGTGGCTCCCATCA
falseIE2    ATGATGCAGCGAACAATCCTCTCTACTGTGTCCTCATTCATGTCATTCGTCCTCAGCTGAAGGATGAAAGAGATTGGCAAGGAAGCAGGAATGAGTCAGCTTTGGGGGAATATTACCTGGTGGCTCCCATCATTTATTGC
falseIE3    CTCTCTACTGTGTCCTCATTCATGTCATTCGTCCTCAGCTGAAGGATGAAAGAGATTGGCAAGGAAGCAGGAATGAGTCAGCTTTGGGGGAATATTACCTGGTGGCTCCCATCATTTATTGCTTTAGGACATACCCTTTT
falseIE4    GGCAAGGAAGCAGGAATGAGTCAGCTTTGGGGGAATATTACCTGGTGGCTCCCATCATTTATTGCTTTAGGACATACCCTTTTACCTACTTGCTATCATGCTGGACCTCAGCATTTCAGGGACCAGAAAAGATGGGGACA
falseIE5    CTGGTGGCTCCCATCATTTATTGCTTTAGGACATACCCTTTTACCTACTTGCTATCATGCTGGACCTCAGCATTTCAGGGACCAGAAAAGATGGGGACACCAATGCAAGCCCAAGACTACCATCCAGAGGACCGTTCCTG
falseIE6    TCCCATCATTTATTGCTTTAGGACATACCCTTTTACCTACTTGCTATCATGCTGGACCTCAGCATTTCAGGGACCAGAAAAGATGGGGACACCAATGCAAGCCCAAGACTACCATCCAGAGGACCGTTCCTGCCCATGCT
falseIE7    TTTACCTACTTGCTATCATGCTGGACCTCAGCATTTCAGGGACCAGAAAAGATGGGGACACCAATGCAAGCCCAAGACTACCATCCAGAGGACCGTTCCTGCCCATGCTGCTTTCTCTAGTTTTGCTTTCAGAGTGGTGT
falseIE8    TGCTGGACCTCAGCATTTCAGGGACCAGAAAAGATGGGGACACCAATGCAAGCCCAAGACTACCATCCAGAGGACCGTTCCTGCCCATGCTGCTTTCTCTAGTTTTGCTTTCAGAGTGGTGTCACCCCACTTGCTAACTC
falseIE9    CTGGACCTCAGCATTTCAGGGACCAGAAAAGATGGGGACACCAATGCAAGCCCAAGACTACCATCCAGAGGACCGTTCCTGCCCATGCTGCTTTCTCTAGTTTTGCTTTCAGAGTGGTGTCACCCCACTTGCTAACTCAA
falseIE10   GATGGGGACACCAATGCAAGCCCAAGACTACCATCCAGAGGACCGTTCCTGCCCATGCTGCTTTCTCTAGTTTTGCTTTCAGAGTGGTGTCACCCCACTTGCTAACTCAAGAGTGCATCACTAGGCTTCCTGAGCAGGTT
```

(b)

Figure 2. The EI and IE dataset for SVM training. (a) EI dataset including true EI and false EI; (b) IE dataset including true IE and false IE.

so the whole sequence can be converted into 552-dimensional vector, and every vector has only two optional value (1 or 0). In view of SVM is a statistical learning algorithm, this study uses 10-fold cross validation to evaluate the performance of the algorithm. All data sets were randomly and equally divided into ten parts, nine for training and one for testing. When each round ends, the sensitivity, specificity, accuracy and correlation coefficient will be recorded, and set the mean value of ten times as the estimation of the algorithm performance.

3. Results and Conclusion

In order to evaluate the performance of the algorithm, the nucleotide sequences near GT or AG sites but not including GT or AG sites in training dataset were intercepted, which includes whole length (138), 100 nucleotides (50 nucleotides before and after splice site GT or AG), 60 nucleotides (30 nucleotides before and after splice site GT or AG), 40 nucleotides (20 nucleotides before and after splice site GT or AG), and 20 nucleotides (10 nucleotides before and after splice site GT or AG). The experimental results are shown as **Table 1** and **Table 2**. From the datum in table, we can find that the performance of the algorithm is excellent, and the accuracy of EI or IE is higher than 90%. From the datum in table, we can also find that the accuracy scores of five methods are almost, which hints that base pattern information is mainly located within 20 nucleotides upstream and downstream splice sites.

Table 1. The 10-fold cross validation to EI true and false training sample.

Method	Cross validation item			
	Sensitivity	Specificity	Accuracy	Correlation coefficient
138 nucleotides	0.8325	0.9660	0.9368	0.8122
100 nucleotides	0.8443	0.9651	0.9387	0.8189
60 nucleotides	0.8443	0.9628	0.9369	0.8140
40 nucleotides	0.8493	0.9595	0.9354	0.8106
20 nucleotides	0.8547	0.9593	0.9365	0.8140

Table 2. The 10-fold cross validation to IE true and false training sample.

Method	Cross validation item			
	Sensitivity	Specificity	Accuracy	Correlation coefficient
138 nucleotides	0.7062	0.9673	0.9168	0.7207
100 nucleotides	0.7076	0.9648	0.9151	0.7152
60 nucleotides	0.7264	0.9576	0.9128	0.7117
40 nucleotides	0.7226	0.9556	0.9105	0.7042
20 nucleotides	0.6618	0.9493	0.8936	0.6443

Through the cross validation experiment, it is clear that method of SVM is applicable for the recognition of splicing sites, and base pattern information is mainly located within 20 nucleotides upstream and downstream splice sites.

Acknowledgements

This study was supported by Scientific Research Foundation of Shaanxi Provincial Office of Health, P.R. China (No. 2016SF-343).

Conflict and Interest

We declare that there is not any conflict and interest from this study.

References

[1] Lu, W., Wainwright, G., Webster, S.G., Rees, H.H. and Turner, P.C. (2000) Clustering of Mandibular Organ-Inhibiting Hormone and Moult-Inhibiting Hormone Genes in The crab, Cancer Pagurus, and Implications for Regulation of Expression. *Gene*, **253**, 197-207. http://dx.doi.org/10.1016/S0378-1119(00)00282-1

[2] Brunak, S., Engelbrecht, J. and Knudsen, S. (1990) Neural Network Detects Errors in the Assignment of mRNA Splice Sites. *Nucleic Acids Research*, **18**, 4797-4801. http://dx.doi.org/10.1093/nar/18.16.4797

[3] Henderson, J., Salzberg, S. and Fasman, K.H. (1997) Finding Genes in DNA with a Hidden Markov Model. *Journal of Computational Biology*: *A Journal of Computational Molecular Cell Biology*, **4**, 127-141. http://dx.doi.org/10.1089/cmb.1997.4.127

[4] Lukashin, A.V. and Borodovsky, M. (1998) GeneMark.hmm: New Solutions for Gene Finding. *Nucleic Acids Research*, **26**, 1107-1115. http://dx.doi.org/10.1093/nar/26.4.1107

[5] Cai, D., Delcher, A., Kao, B. and Kasif, S. (2000) Modeling Splice Sites with Bayes Networks. *Bioinformatics*, **16**, 152-158. http://dx.doi.org/10.1093/bioinformatics/16.2.152

[6] Zhang, L. and Luo, L. (2003) aLL: Splice Site Prediction with QUADRATIC Discriminant Analysis Using Diversity Measure. *Nucleic Acids Research*, **31**, 6214-6220. http://dx.doi.org/10.1093/nar/gkg805

[7] Sun, Y.F., Fan, X.D. and Li, Y.D. (2003) Identifying Splicing Sites in Eukaryotic RNA: Support Vector Machine Approach. *Computers in Biology & Medicine*, **33**, 17-29. http://dx.doi.org/10.1016/S0010-4825(02)00057-4

[8] Zhang, X.H., Heller, K.A., Hefter, I., Leslie, C.S. and Chasin, LA. (2003) Sequence Information for the Splicing of Human Pre-mRNA Identified by Support Vector Machine Classification. *Genome Research*, **13**.

[9] Vapnik, V.N. (1998) Statistical Learning Theory. Adaptive and Learning Systems for Signal Processing, Communications, and Control. *Signal Processing*, *Communications*, *and Control*.

[10] Pollastro, P. and Rampone, S. (2002) HS3D: Homosapiens Splice Site Data Set.

[11] Damaševičius, R. (2008) Optimization of SVM Parameters for Promoter Recognition in DNA Sequences. 20*th EURO Mini Conference* "*Continuous Optimization and Knowledge-Based Technologies*" *Eur OPT*-2008, 99-104.