Recognizing DNA splice sites with the frequent pattern mining technique

Nittaya Kerdprasop and Kittisak Kerdprasop

***Abstract*—**The completion of Human Genome Project in 2001 yields the entirety of human genetic information, or genome. A genome is organized in chromosomes and composed of thousands of genes, which are the heredity units of traits such as hair color and blood type. Genes in complex organisms such as primates and humans are composed of regions that code for protein, called exons, and non-coding regions, called introns. During the transcription from the DNA template for later translating into amino acid chain of protein structure, introns are to be removed and exons are then joined to form a continuous messenger-RNA strand. Splice sites are the junctions or borders between introns and exons. Accurate detection of splice sites from the fragments of DNA sequence is important to the success of gene prediction. Due to huge amount of genetic information in most genomes, computational techniques are essential for the interpretation and recognition of specific genetic sequences. In this paper, we propose a splice site prediction technique based on frequent pattern analysis. We apply association mining to each splice junction types, that is, exon/intron, intron/exon, and none of the two types. The frequent DNA patterns are then combined and prioritized with respect to their annotated confidence and support values. The final result of our method is a set of cascaded rules to be used for gene prediction. From the experimental results, our method can make a high recall prediction comparative to other classification-based methods. We also demonstrate computational improvement via a concurrency technique. Running time reduction is considerably observable.

***Keywords*—**Gene expression, splice site prediction, DNA sequence, frequent pattern analysis.

I. Introduction

PROTEINS perform essential functions in most living organisms. The diverse functions of proteins include providing rigidity and mass in bones and tissues, forming enzymes for biological catalysts, and acting as special substances such as hemoglobin and insulin. Proteins are macromolecules comprised of the polymers of 20 different types of amino acids [18]. The amino acid composition of proteins is in turn determined by the DNA sequence, or a gene. Deoxyribonucleic acid (DNA) is the carrier of information to build proteins, and the ribonucleic acid (RNA) is involved in the biosynthesis of proteins. Relationship between DNA, RNA, and protein is called the central dogma of molecular biology [3] and can be illustrated in Fig. 1.

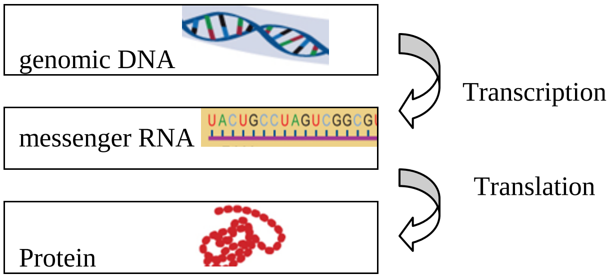


Fig. 1 The central dogma of molecular biology

The basic constituents of DNA and RNA are nucleotides that are made of three chemical components: phosphate group, ribose sugar, and nitrogenous bases. RNA normally forms a single strand of nucleotides with a nitrogenous base as either adenine (A), guanine (G), cytosine (C), or uracil (U). DNA, on the contrary, is composed of the deoxy form of ribose and the four nitrogenous bases adenine, guanine, cytosine, and thymine (T). A single DNA strand is not stable; therefore, it normally appears in antiparallel double helix structure with the complementary pair A-T and C-G.

The base sequence, or gene, encodes genetic information to generate proteins. This information in DNA is transferred to the messenger RNA (or mRNA) during the process called *transcription*. The genetic flow followed by the *translation* process to build proteins from mRNA.

In prokaryotes (one-cell organisms with no nucleus, hence genetic materials disperse in the cell), genetic information is encoded continuously on a DNA strand. But in eukaryotes (organisms with one or more cells that have nucleus to encompass genetic materials), regions that code for proteins are interrupted by the non-coding regions. The non-coding regions in DNA are called *introns* (intervening sequences), whereas the coding regions are called *exons* (expresses sequences) [11]. During the transcription process of most eukaryotic genes, the primary RNA transcript needs additional modification step called *splicing*.

The splicing process involves the removal of introns by spliceosome and joining exons together to make one long continuous mRNA strand (Fig. 2). The spliced mRNA is then exported out of the cell nucleus and passed to the ribosome for translation to different kinds of amino acids. Ribosome translates code for protein synthesis by reading three consecutive bases, called *codon*. A specific codon encoded for the starting point of protein coding region is AUG (or ATG in DNA strand). During the translation process, tRNA associates each codon with a unique amino acid to form a chain of polypeptide. The translation process stops when ribosome encounters one of the three stop codons: UAA, UAG, or UGA.

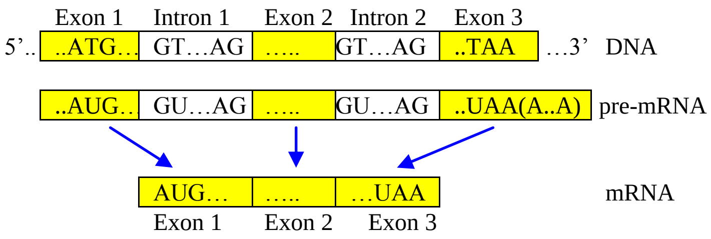


Fig. 2 Removal of introns and joining of exons in the splicing process

Predicting genes in eukaryotes is a hard problem because of the splicing mechanism to exile the introns and joining the exons. Each exon can be as short as a few nucleotides, whereas the introns are often very long. The length and the number of introns in eukaryotic genes vary widely among species and among genes within the same species. Introns are known to have a minimal length of about 60 base pairs to accommodate the splicing signals [11]. There are different kinds of introns, but the one involved in the protein-coding process is spliceosomal introns.

Nearly all spliceosomal introns conform to the GT-AG rule in which the first two nucleotides at the 5' end begin with GT and end with AG at the 3' end. Since splicing occurs after a DNA template is transcribed to the RNA strand, the beginning of the spliceosomal introns is GU, not GT.

The borders of introns are called splice sites. The GT (or GU) end is called the donor splice site and the AG end is called the acceptor splice site. In some rare cases, the GU-AG rule does not hold and introns are spliced out with different splice site sequences. Moreover, from a primary transcript (pre-mRNA) many different mRNA sequences can be generated due to alternative splicing (such as exons can be extended or skipped, or introns can be retained). Different splicing pathways can thus lead to different protein products, given the same mRNA. Therefore, splice site prediction is an important computational problem to the recognition of protein structure from the known genomic DNA sequence.

This paper focuses on the algorithmic approach to the recognition of splice sites in DNA sequences. This classification problem aims at recognizing the extron/intron junction (a donor), the intron/exon junction (an acceptor), or none of the junction sites. Many previous work tackle this problem with classification techniques such as neural network [4], [17], [19], a tree-based C4.5 [20], Bayes method [8], support vector machine [5], [12], [22], or even the ensemble methods [12], [13], [16]. We, on the contrary, employ a different approach to the splice site prediction problem. Our prediction technique is based on frequent pattern analysis and the experimental results show a predictor model with a high recall performance.

The rest of this paper is organized as follows. Section 2 states the splice site recognition problem and reviews related work regarding this problem. Section 3 explains our prediction method that employs association mining technique. Section 4 shows experimental design and results. Section 5 concludes the paper.

II. DNA Splice Site Recognition Problem and Related Work

The splice site recognition problem can be formulated as the following. Given some part of unclassified genomic DNA sequences, decide whether this is an intron/exon border, an exon/intron border, or none of the two splice sites. To develop an accurate prediction model, a machine learning technique is usually applied. The learning task is that given sequences of genomic DNA with known splice junction labeled as either an intron/exon, an exon/intron, or none, the learning objective is to find a classification rule that can successfully predict the region of uncharacterized genomic DNA sequence.

Splice site prediction can be considered as a subproblem of gene prediction that aims at correctly recognizing gene from the given fragment of DNA sequence. The task of splice site prediction is to recognize the actual boundaries of the protein-coding regions in the DNA sequence. There are many computational techniques applied to tackle this problem. The direct method [6], [7], [10], [21] is to analyze the sequence to capture gene profile and identify specific features that that can accurately predict the splice junctions. Researchers from the machine learning community prefer to attack this problem via a single or multiple classification learning algorithms [12] , [13], [16].

In the early years of computational molecular biology, common learning methods used to predict gene position and structure are neural network and hidden Markov model [2], [17], [19]. These techniques are still in use by current researchers, for instance, Mubark and colleagues [15] applied the two techniques to predict the structure of hemoglobin (which is a protein component of red blood cells) from the given DNA sequence. Cristea and his team [4] also applied neural network on the reduced feature dataset to detect mutations in the genomic sequences.

Different classification schemes from the machine learning area are soon realized to be capable of producing a high accurate predictor. These classification techniques include the support vector machines [5], [12], [22], C4.5 [20], and Bayesian method [8].

Our approach to solve the splice site recognition problem is different from those appeared in the literature in that our predictor is built from the association analysis technique [1], not the classification ones. The advantage of the proposed technique is that the prediction model can contain nucleotides at arbitrary position, not necessarily be the contiguous base sequences.

III. Proposed Method to Splice Site Recognition and Prediction

At the initial stage of our proposed method (named assoDNA), the training dataset with a mixture of exon/intron, intron/exon, and none of the two DNA sequence splice sites is separated into three subsets according to splice junction types. Each data subset is then processed through the same steps of frequent patterns and association analysis. The conceptual model and flow diagram of our method are depicted in Figs. 3 and 4, respectively.

Once the three data subsets are processed through the frequent pattern analysis method, the three sets of learning results (displayed as prediction rules) are finally combined and prioritized according to the confidence and support values. The proposed assoDNA method can be explained as follows.

Step 1: Initialization phase

Split the training dataset into three subsets according to the class value. Thus, we will get data of class exon/intron, data of class intron/exon, and data of class none.

Step 2: Generation of frequent patterns

Each data subset is processed through the following steps:

```
2.1 Set the given minimum support as minSup
2.2 Initialize R ( a set of frequent patterns) to be
    empty, $R = \varnothing$
2.3 Build a candidate pattern P of length K
        $P = \wedge_K (L_i = B_j)$
        where K starts from 1, $i \in \{-30, .., +30\}$, and
            $B_j \in \{A,C,T,G,D,N,S,R\}$
2.4 Select a pattern P with support $\geq$ minSup to
    contain in a set S
2.5 Set $R = R \cup S$
2.6 If $S = \varnothing$, then continue to step 3
        else increment K and go back to step 2.3
```

Step 3: Confidence computation

```
3.1 Compute confidence value of every pattern P in R,
    and annotate confidence value to every pattern
3.2 Sort P in descending order according to confidence
    value, for a tie then descending sort with respect to
    a support value
```

Step 4: Rule generation

```
4.1 Set the given minimum confidence as minConf
4.2 Generate association rules from every pattern P in
    R that has confidence $\geq$ minConf
```

Step 5: Building predictor model

Combine rules from the process of every data subset and sort according their confidence and support values

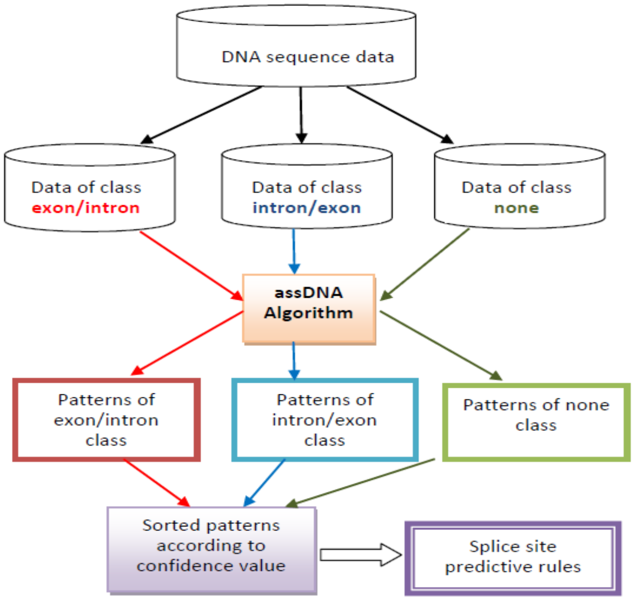

DNA sequence data
Data of class exon/intron
Data of class intron/exon
Data of class none
assDNA Algorithm
Patterns of exon/intron class
Patterns of intron/exon class
Patterns of none class
Sorted patterns according to confidence value
Splice site predictive rules

Fig. 3 A conceptual model of splice site recognition

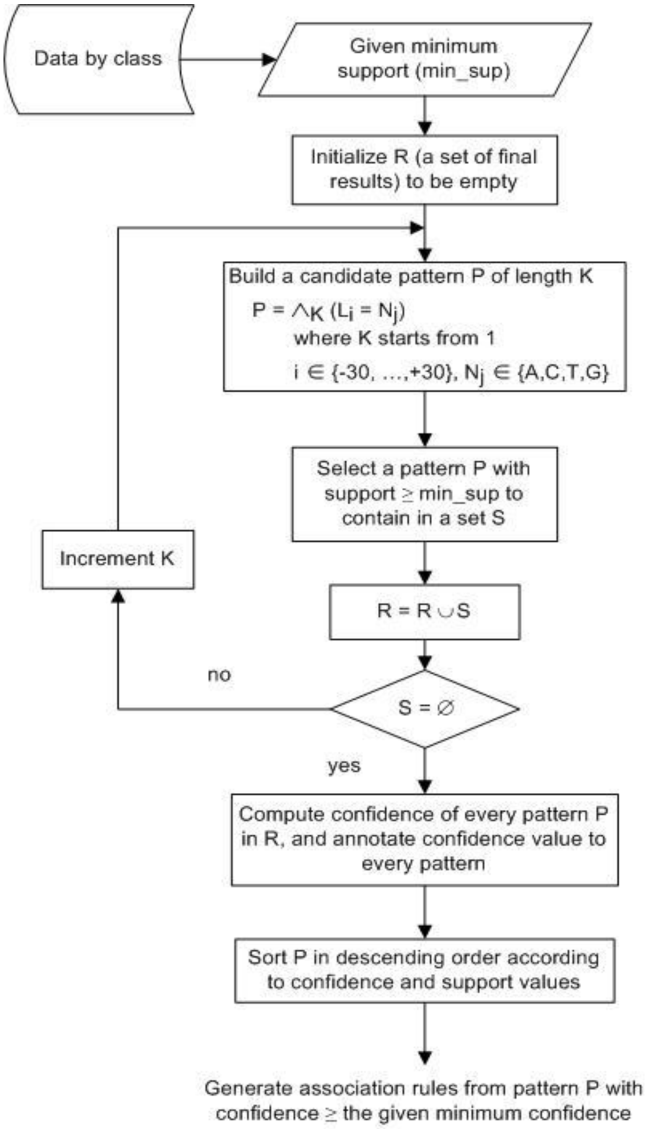

Data by class
Given minimum support (min_sup)
Initialize R (a set of final results) to be empty
Build a candidate pattern P of length K
P = ∧K (Li = Nj)
where K starts from 1
i ∈ {-30, …,+30}, Nj ∈ {A,C,T,G}
Select a pattern P with support ≥ min_sup to contain in a set S
Increment K
R = R ∪ S
no
S = ∅
yes
Compute confidence of every pattern P in R, and annotate confidence value to every pattern
Sort P in descending order according to confidence and support values
Generate association rules from pattern P with confidence ≥ the given minimum confidence

Fig. 4 A flow diagram illustrating the assoDNA method

Confidence and support are basic metrics [1] used in association analysis to delimit the number of association rules and size of the search space, respectively. The two metrics can be defined through the following example. Given the association pattern that nucleotide base B1 appears at location L1 and base B2 appears at location L2, then the prediction result is that this DNA sequence splice site is in category C1. This prediction based on association pattern can be written as the following classification rule:

$$\textit{IF } (L1{=}B1 \wedge L2{=}B2) \textit{ THEN } (\textit{splice-site}{=}C1) \qquad (1)$$

Note that for this specific problem spilce site category, C, can either be the exon/intron junction, intron/exon junction, or none. Nucleotide symbol, N, can be one of the eight symbols: A, C, T, G, D, N, S, R, where the last four symbols represent ambiguity. D means nucleotide base can be either A, or G, or T. N is ambiguity among A or C or G or T. S is ambiguity among C or G, and R is ambiguity among A or G. The base location, L, is in the range -30 to +30.

The confidence of rule (1) is computed as:

$$\frac{\textit{support}(L1{=}B1 \wedge L2{=}B2 \wedge \textit{splice-site}{=}C1)}{\textit{support}(L1{=}B1 \wedge L2{=}B2)} \qquad (2)$$

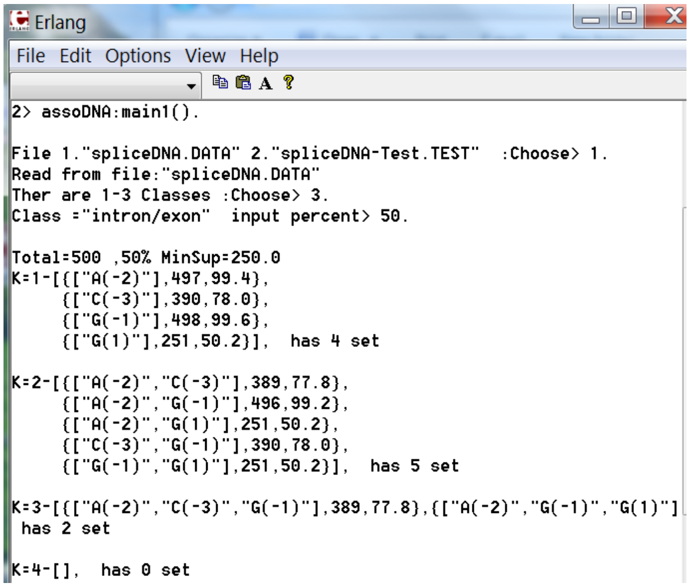
The value of support(L1=B1 ∧ L2=B2 ∧ splice-site=C1) is the number of times that the three events (B1 appears at location L1, B2 appears at location L2, and splice-site junction is of type C1) occur together within the same DNA sequence in the training dataset. The support value of (L1=B1 ∧ L2=B2) is thus the frequency that both events co-occur in the training dataset.

We implement the assoDNA method with the Erlang functional language. Some major functions are displayed in Fig. 5.

```
main1() ->
        {AllInput,FNo,ThisClass} = input(),
        DB = myToSet(AllInput),
        Total = length(AllInput),
        {_,Per} = io:read(" input percent> "),
        {FNo, ThisClass, DB, Per} .


apriori(DB, Items, Min) ->
        C1=[ {from_list([X]), findSup(from_list([X]), DB) }
                              || X <- Items ],
        L1=[{FS,Sup} || {FS,Sup} <- C1,Sup>=Min] ,
        LkPrint=[ {to_list(FS), Sup,Sup/length(DB)*100}
                              || {FS,Sup} <- L1] ,
        K = 2,
        LS = [FS || {FS,_} <- L1],
        aprioriLoopPar(L1, DB, LS, K, Min) .
```

Fig. 5 Some main functions of the assoDNA program



```
2> assoDNA:main1().

File 1."spliceDNA.DATA" 2."spliceDNA-Test.TEST"  :Choose> 1.
Read from file:"spliceDNA.DATA"
Ther are 1-3 Classes :Choose> 3.
Class ="intron/exon"  input percent> 50.

Total=500 ,50% MinSup=250.0
K=1-[{["A(-2)"],497,99.4},
     {["C(-3)"],390,78.0},
     {["G(-1)"],498,99.6},
     {["G(1)"],251,50.2}],  has 4 set

K=2-[{["A(-2)","C(-3)"],389,77.8},
     {["A(-2)","G(-1)"],496,99.2},
     {["A(-2)","G(1)"],251,50.2},
     {["C(-3)","G(-1)"],390,78.0},
     {["G(-1)","G(1)"],251,50.2}],  has 5 set

K=3-[{["A(-2)","C(-3)","G(-1)"],389,77.8},{["A(-2)","G(-1)","G(1)"]
 has 2 set

K=4-[],  has 0 set
```

Fig. 6 Screenshot of the assoDNA program

The screenshot in Fig. 6 shows the total of eleven frequent patterns found by the assoDNA program on an intron/exon group when setting minimum support to be 50%. Each pattern is annotated with its occurrence frequency and percentage of support value. The pattern format is implemented as “base(location)”, for example, A(-2) means the base nucleotide A appears at location -2.

IV. Experimental Design and Results

A. Dataset Preparation

The dataset used in this work primate splice-junction gene sequences available at the UCI repository of machine learning databases [14]. This dataset are taken from GenBank 64.1 containing 3,190 DNA sequences. Each sequence is a window of 60 DNA base pairs starting at position -30 and ending at position +30 corresponding to the splice site location. The splice junction can be either a junction between intron and exon (labeled in the dataset as intron/exon), a junction between exon and intron (labeled as exon/intron), or no junction at all (labeled as none).

From the total of 3,190 DNA sequences, we split the data into two sets: training set (with 2,000 DNA sequences) and test set (with 1,190 DNA sequences). We try to conserve class distribution of both data subsets close to the original data as much as possible. Class distribution of training and test sets are summarized in Table 1. From the original 767 DNA sequences of class exon/intron, we split them by taking the first 500 DNA sequences to be in the training set and the remaining 267 sequences to be in the test set. The intron/exon and none classes of training and tests sets are prepared in the same manner.

Table 1. Class distribution in the training and test sets

Classes	Training data	Test data
exon/intron	500 (25%)	267 (22.4%)
intron/exon	500 (25%)	268 (22.5%)
none	1,000 (50%)	655 (55.1%)
Total	2,000	1,190

B. Experimentation

We prepare a separate test set in order to make a fair comparison between a prediction model obtained from our method (assoDNA) and models from other learning techniques. Other machine learning methods [9] used in this experimentation include the C4.5 algorithm, naïve Bayes technique, instance based method with 10 nearest neighbors (1 and 5 neighbors yields worse performance), and support vector machine (algorithm SMO). We also tested the neural network algorithm on the training dataset, but it ran out of memory space. Class prediction identified by the model is compared against the real class of each DNA sequence in the test set.

The outcome of the prediction can be either correct, or incorrect. For the incorrect cases, a more specific analysis of incorrectness is described in a form of confusion matrix as shown in Table 2.

Table 2. Confusion matrix of the three-class prediction model

Real class	Class prediction made by the model		
	exon/intron	intron/exon	none
exon/intron	a	b	c
intron/exon	d	e	f
none	g	h	i

The variable ‘a’ means the number of DNA sequences with exon/intron splice sites that are actually predicted by the model as exon/intron. Variable ‘b’ is the number real exon/intron splice sites that are incorrectly predicted as intron/exon, whereas ‘c’ is the number of real exon/intron junctions that are incorrectly predicted as none. Other variables (‘d’ through ‘i’) can be interpreted in the same manner.

To evaluate the performance of the prediction model, we use the four measurement metrics: accuracy, precision, recall, and F-measure. The four measures are defined as:

$$\text{Accuracy} = \frac{(a + e + i)}{(a + b + c + d + e + f + g + h + i)} \tag{3}$$

Precision (or specificity)

$$\text{for class exon/intron} = \frac{a}{(a + d + g)} \tag{4}$$

$$\text{for class intron/exon} = \frac{e}{(b + e + h)} \tag{5}$$

$$\text{for class none} = \frac{i}{(c + f + i)} \tag{6}$$

Recall (or sensitivity)

$$\text{for class exon/intron} = \frac{a}{(a + b + c)} \tag{7}$$

$$\text{for class intron/exon} = \frac{e}{(d + e + f)} \tag{8}$$

$$\text{for class none} = \frac{i}{(g + h + i)} \tag{9}$$

$$\text{F-measure (by class)} = \frac{(2\times \text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{10}$$

Experimental results regarding the performance comparison of our method against other classification-based learning methods are given by class in Tables 3-5. It can be seen from the results that our method is superior in its recall capability, especially for the class exon/intron. The proposed method is as good as others in its precision power.

Table 3. Prediction performance on recognizing exon/intron splice site

Method	Performance measures			
	Accuracy	Precision	Recall	F-measure
assoDNA	0.961	0.911	**0.993**	0.950
C4.5	0.936	0.895	0.963	0.928
Naïve Bayes	0.966	**0.944**	0.951	0.948
Instance based (IB10)	0.801	0.665	0.959	0.785
Support vector machine	0.920	0.878	0.914	0.895

Table 4. Prediction performance on recognizing intron/exon splice site

Method	Performance measures			
	Accuracy	Precision	Recall	F-measure
assoDNA	0.961	0.932	**0.978**	0.954
C4.5	0.936	0.884	0.914	0.899
Naïve Bayes	0.966	**0.942**	0.970	0.956
Instance based (IB10)	0.801	0.707	0.955	0.813
Support vector machine	0.920	0.869	0.914	0.891

Table 5. Prediction performance on recognizing none of the splice site

Method	Performance measures			
	Accuracy	Precision	Recall	F-measure
assoDNA	0.961	**0.998**	0.942	0.969
C4.5	0.936	0.978	0.934	0.956
Naïve Bayes	0.966	0.984	**0.969**	0.977
Instance based (IB10)	0.801	0.995	0.673	0.803
Support vector machine	0.920	0.962	0.925	0.943

Comparative results of different gene finding schemes in terms of precision and recall rates on recognizing DNA splice sites can graphically displayed in Figs. 6 and 7, respectively. It can be noticed that our method (assoDNA) produce high recall results on recognizing exon/intron and intron/exon borders. For the class of none of the two borders, our method has lower recall rate than naïve Bayes.

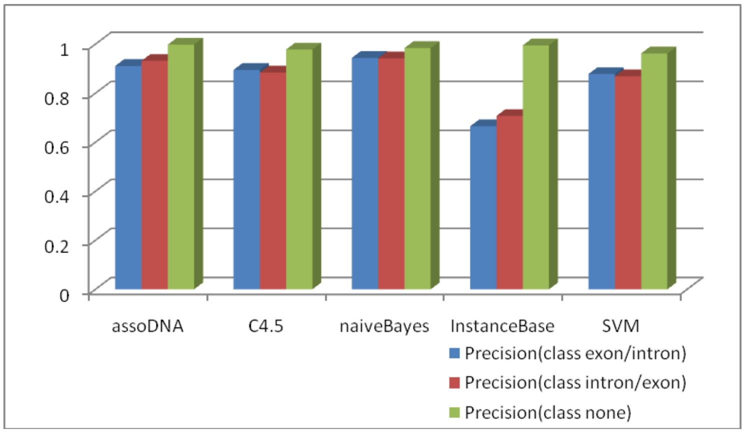


Fig. 6 Precision comparisons of different recognition methods

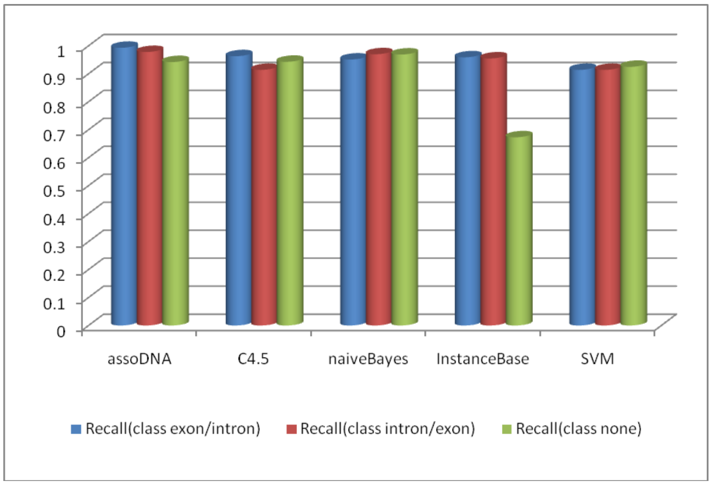


Fig. 7 Recall comparisons of different recognition methods

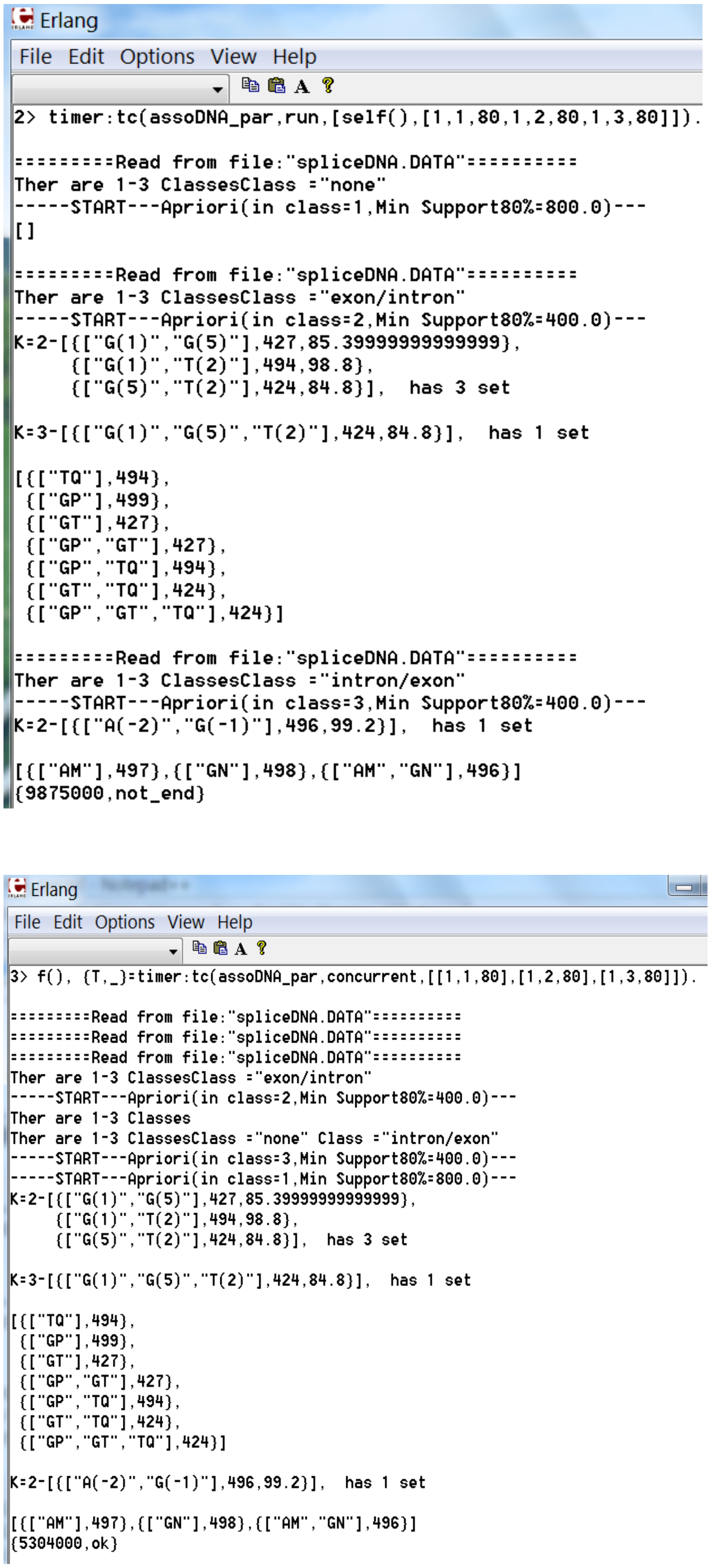
C. Performance Improvement via Concurrency

To improve the computational performance of the proposed assoDNA method, we employ the concept of concurrent programming. Erlang is a functional language that provides facilities for designing and implementing concurrent program. Those facilities include the spawn, send, and receive primitives to handle concurrent programming. Concurrent processes in Erlang communicate through asynchronous message passing with dedicated memory space for each process, rather than a thread concept and shared memory. The example of transforming sequential assDNA program to be a concurrent assoDNA is showed in Fig. 8. Reduction in running time can be compared through the screenshots in Fig. 9 in which the last line on a upper screen is sequential running time (in a unit of microseconds), whereas the last line on a lower screen is concurrent running time. Time reduction in this example is around 46.29%.

```
-module(assoDNA_par).
concurrent(P1, P2, P3) ->
        spawn(assoDNA_par, run, [self(),P1]),
        spawn(assoDNA_par, run, [self(),P2]),
        spawn(assoDNA_par,run,[self(),P3]),
        receive
              my_end -> ok
        end.

run(MasterID, InputL) ->
        R = main2(any, 3, InputL),
        file:delete("out.txt") ,
        AD = lists:last(R),
        [ADD|_] = AD,
        Rules = lists:sublist(R, length(R)-1),
        PrintRules = map(fun( {D, S, Per, Class} ) ->
                              {to_Col3(notLast(D)), S, Per,
                               transformBack(Class) }
                        end,
                        Rules),
        ADP = lists:map(fun(Data) ->
                              {Data, checkRules(Data, Rules) }
                              end,
                              AD),
        ADPprint = map(fun({Data, V}) ->
                              Predict = transformBack(V),
                              {Data, [last(Data), Predict,
                                mark(last(Data), Predict) ] }
                        end,
                        ADP),
        Predict = map(fun( {F, S} ) ->
                              {to_Col3(notLast(F)), S}
                     end,
                     ADPprint),
        writeToFile(Predict),
        [_,Stop|_] = InputL,
        if Stop ==2 -> MasterID ! my_end ;
               true -> MasterID ! not_end
        end.
```

Fig. 8 Program coding for running concurrent assoDNA



```
2> timer:tc(assoDNA_par,run,[self(),[1,1,80,1,2,80,1,3,80]]).

=========Read from file:"spliceDNA.DATA"==========
Ther are 1-3 ClassesClass ="none"
-----START---Apriori(in class=1,Min Support80%=800.0)---
[]

=========Read from file:"spliceDNA.DATA"==========
Ther are 1-3 ClassesClass ="exon/intron"
-----START---Apriori(in class=2,Min Support80%=400.0)---
K=2-[{["G(1)","G(5)"],427,85.39999999999999},
     {["G(1)","T(2)"],494,98.8},
     {["G(5)","T(2)"],424,84.8}],  has 3 set

K=3-[{["G(1)","G(5)","T(2)"],424,84.8}],  has 1 set

[{["TQ"],494},
 {["GP"],499},
 {["GT"],427},
 {["GP","GT"],427},
 {["GP","TQ"],494},
 {["GT","TQ"],424},
 {["GP","GT","TQ"],424}]

=========Read from file:"spliceDNA.DATA"==========
Ther are 1-3 ClassesClass ="intron/exon"
-----START---Apriori(in class=3,Min Support80%=400.0)---
K=2-[{["A(-2)","G(-1)"],496,99.2}],  has 1 set

[{["AM"],497},{["GN"],498},{["AM","GN"],496}]
{9875000,not_end}
```

```
3> f(), {T,_}=timer:tc(assoDNA_par,concurrent,[[1,1,80],[1,2,80],[1,3,80]]).

=========Read from file:"spliceDNA.DATA"==========
=========Read from file:"spliceDNA.DATA"==========
=========Read from file:"spliceDNA.DATA"==========
Ther are 1-3 ClassesClass ="exon/intron"
-----START---Apriori(in class=2,Min Support80%=400.0)---
Ther are 1-3 Classes
Ther are 1-3 ClassesClass ="none" Class ="intron/exon"
-----START---Apriori(in class=3,Min Support80%=400.0)---
-----START---Apriori(in class=1,Min Support80%=800.0)---
K=2-[{["G(1)","G(5)"],427,85.39999999999999},
     {["G(1)","T(2)"],494,98.8},
     {["G(5)","T(2)"],424,84.8}],  has 3 set

K=3-[{["G(1)","G(5)","T(2)"],424,84.8}],  has 1 set

[{["TQ"],494},
 {["GP"],499},
 {["GT"],427},
 {["GP","GT"],427},
 {["GP","TQ"],494},
 {["GT","TQ"],424},
 {["GP","GT","TQ"],424}]

K=2-[{["A(-2)","G(-1)"],496,99.2}],  has 1 set

[{["AM"],497},{["GN"],498},{["AM","GN"],496}]
{5304000,ok}
```

Fig. 9 Screenshots of sequential assoDNA (upper screen) versus concurrent assoDNA (lower screen)

V. Conclusion

Splice site prediction from the fragment of DNA sequence is an extensively studied problem in computational molecular biology and bioinformatics. Splicing is the modification process that occurs during the transcription of gene expression, which is the process of transcribing a DNA template into mRNA sequence and then translating the mRNA to protein structure.

In organisms with cell nucleus, called eukaryotes, the transcription is not straightforward from DNA template to mRNA. It instead involves the intermediate step from DNA to pre-mRNA, then to mRNA. Splicing occurs at the stage of changing from pre-mRNA to the mRNA transcript. This change involves removing introns and then joining the exon parts together to form a continuous genetic sequence ready for the translation to a functional polypeptide chain of protein structure. To accurately predict the splice site of the DNA sequence fragment is an important task of gene prediction.

In this paper, we present a new method to splice site prediction from genomic sequence by means of association analysis. Association mining is unsupervised learning task that has been successfully applied to the marketing and business applications. It is rarely used in the area of bioinformatics. We thus devise a method to apply association mining technique to induce prediction model from the DNA sequence dataset. From the experimental results comparative to other classification learning techniques, we found that the prediction model based on association analysis can produce a significantly high recall prediction, whereas the precision rate is as good as other learning techniques.

The proposed method is however unable to generate frequent patterns when the support metric is set lower than 10%. This is because high memory consumption during the candidate pattern generation steps. We are planning to solve this problem in our future work.

Acknowledgment

This research has been funded by grants from the National Research Council of Thailand (NRCT) and the Thailand Research Fund (TRF, grant number RMU5080026). The DEKD Research Unit has been supported by Suranaree University of Technology.

References

[1] R. Agrawal, T. Imielinski, and A. Swami, Mining association rules between set of items in large databases, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1993, pp. 207-216.

[2] S. Brunak, J. Engelbrecht, and S. Knudsen, Prediction of human mRNA donor and acceptor sites from the DNA sequence, *Journal of Molecular Biology*, Vol. 220, Issue 1, 1991, pp. 49-65.

[3] F. Crick, Central dogma of molecular biology, *Nature*, Vol. 227, 1970, pp. 561-563.

[4] P. Cristea, V. Mladenov, R. Tuduce, G. Tsenov, and S. Petrakieva, Neural networks for prediction of nucleotide

sequences by using genomic signals, *WSEAS Transactions on Systems*, Vol.7, Issue 7, 2008, pp. 637-644.

[5] R. Dama evicius, Splice site recognition in DNA sequences using k-mer frequency based mapping for support vector machine with power series kernel, *Proceedings of International Conference on Complex, Intelligent and Software Intensive Systems*, 2008, pp. 687-692.

[6] S. Degroeve, B. De Baets, Y. Van de Peer, and P. Rouze, Feature subset selection for splice site prediction, *Proceedings of the European Conference on Computational Biology (ECCB)*, 2002, pp. 75-83.

[7] R. Dogan, L. Getoor, and W. Wilbur, Characterizing RNA secondary-structure features and their effects on splice-site prediction, *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops*, 2007, pp. 89-94.

[8] S. Guo and Y. Zhu, Predicting splice site by improved Bayesian classifier, *Proceedings of the Fifth International Conference on Natural Computation*, 2009, pp. 82-85.

[9] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, The WEKA data mining software: An update, *SIGKDD Explorations*, Vol.11, Issue 1, 2009, pp.10-18.

[10] R. Islamaj, L. Getoor, and W. Wilbur, A feature generation algorithm for sequences with application to splice-site prediction, *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, 2006, pp. 553-560.

[11] D. Krane and M. Raymer, *Fundamental Concepts of Bioinformatics*, San Francisco: Pearson Education International, 2003.

[12] A. Lorena and A. de Carvalho, Human splice site identification with multiclass support vector machines and bagging, *Proceedings of the Joint International Conference on Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP)*, 2003, pp. 234-241.

[13] A. Lumini and L. Nanni, Identifying splice-junction sequences by hierarchical multiclassifier, *Artificial Intelligence and Applications*, 2005, pp. 416-420.

[14] The Machine Learning Database Repository. http://mlearn.ics.uci.edu/databases/molecula-biology/splice-junction-gene-sequences.

[15] R. Mubark, H. Keshk, and M. Eladawy, Different species classifier and hemoglobin structure predictor based on DNA sequences, *International Journal of Biology and Biomedical Engineering*, Vol. 2, Issue 2, 2008, pp. 49-58.

[16] C. Nantasenamat, T. Naenna, C. Isarankura-Na-Ayudhya, and V. Prachayasittikul, Recognition of DNA splice junction via machine learning approaches, *EXCLI Journal*, Vol.4, 2005, pp. 114-129.

[17] M. Noordewier, G. Towell, and J. Shavlik, Training knowledge-based neural networks to recognize genes in DNA sequences, *Proceedings of the Conference on Advances in Neural Information Processing Systems*, 1990, pp. 530-536.

[18] Y. Qi, F. Lin, and K. Wong, High performance computing in protein secondary structure prediction, *WSEAS Transactions on Circuits and Systems*, Vol. 2, Issue 3, 2003, pp. 619-624.

[19] S. Rampone, Recognition of splice junctions on DNA sequences by BRAIN learning algorithm, *Bioinformatics*, Vol.14, No.8, 1998, pp. 676-684.

[20] H. Sun, Q. Peng, Q. Zhang, and D. Mou, Splice site prediction based on characteristics of sequential motifs and C4.5 algorithms, *Proceedings of the Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, 2008, pp. 417-422.

[21] Y. Yamada and K. Satou, Prediction of genomic methyllation status on CpG islands using DNA sequence features, *WSEAS Transactions on Biology and Biomedicine*, Vol. 5, Issue 7, 2008, pp. 153-162.

[22] X. Zhang, K. Heller, I. Hefter, C. Leslie, and L. Chasin, Sequence information for the splicing of human pre-mRNA identified by support vector machine classification, *Genome Research*, Vol. 13, No. 12, 2003, pp. 2637-2650.

Nittaya Kerdprasop is an associate professor at the school of computer engineering, Suranaree University of Technology, Thailand. Her current address is School of Computer Engineering, Suranaree University of Technology, 111 University Avenue, Nakhon Ratchasima 30000, Thailand, e-mail: nittaya@sut.ac.th.

She received her B.S. from Mahidol University, Thailand, in 1985, M.S. in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, USA, in 1999. She is a member of ACM and IEEE Computer Society. Her research of interest includes Knowledge Discovery in Databases, AI, Logic Programming, Deductive and Active Databases.

Kittisak Kerdprasop is an associate professor at the school of computer engineering, Suranaree University of Technology, Thailand. He received his bachelor degree in Mathematics from Srinakarinwirot University, Thailand, in 1986, master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in computer science from Nova Southeastern University, USA., in 1999. His current research includes Data mining, Artificial Intelligence, Functional Programming, Computational Statistics.