ELSEVIER

Information Sciences 139 (2001) 139–163

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

Effective hidden Markov models for detecting splicing junction sites in DNA sequences

Michael M. Yin *, Jason T.L. Wang

Department of Computer and Information Science, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA

Received 10 September 1999; received in revised form 24 February 2001; accepted 1 April 2001

Abstract

Identification or prediction of coding sequences from within genomic DNA has been a major rate-limiting step in the pursuit of genes. Programs currently available are far from being powerful enough to elucidate the gene structure completely. In this paper, we develop effective hidden Markov models (HMMs) to represent the consensus and degeneracy features of splicing junction sites in eukaryotic genes. Our HMM system based on the developed HMMs is fully trained using an expectation maximization (EM) algorithm and the system performance is evaluated using a 10-way cross-validation method. Experimental results show that the proposed HMM system can correctly detect 92% of the true donor sites and 91.5% of the true acceptor sites in the test data set containing real vertebrate gene sequences. These results suggest that our approach provide a useful tool in discovering the splicing junction sites in eukaryotic genes.

Keywords: Hidden Markov models; Bioinformatics; Computational biology; Splicing junction; Gene finding

* Corresponding author.
E-mail addresses: yin@homer.njit.edu (M.M. Yin), jason@cis.njit.edu (J.T.L. Wang).

0020-0255/01/$ - see front matter
PII: S0020-0255(01)00160-8

1. Introduction

1.1. Background

A deoxyribonucleic acid (DNA) chain is a long, unbranched polymer composed of four types of nucleotides or bases: adenine (A), cytosine (C), guanine (G) and thymine (T). Genes, made of DNA, are the invisible information-containing elements that are distributed to each daughter cell when a cell divides. In general, genes are divided into two categories: *eukaryotic* and *prokaryotic*. Eukaryotic genes are from *eukaryotic cells* and prokaryotic genes are from *prokaryotic cells*. "Eu" means "good, well or true". "Karyote" (or "caryote") means "nucleus" ("caryon" in Greek). A eukaryotic cell, by definition, has a nucleus that contains the cell's DNA for all of its genes, enclosed by a double layer of membrane [1]. So, the eukaryotic gene category includes all kinds of genes from cells with a nucleus, such as those from any kind of animals, even yeast.

In the bioinformatics field, eukaryotic DNA (or gene) means the kind of genomic DNA with *introns*, such as the DNA from high level animals and human. Prokaryotic cells, in contrast to the eukaryotic cells, have relatively simple internal structures, specifically, without membrane enclosed nuclei [1]. Prokaryotic cells include those from the various types of bacteria such as *E. coli*. *E. coli* has simple genomic DNA and its cells are very easy to culture. So, *E. coli* is often used for research on prokaryotic DNA.

The basic gene structure for higher eukaryotes includes promoter, start codon, introns, exons, and stop codon, etc. (see Fig. 1). The exon sequences of a gene are also called the *coding sequences* of this gene, and the whole exon sequences of a gene are called the *coding region* of the gene (which is the region for making protein). In contrast, prokaryotic genes have no introns, and the gene structure includes only promoter, start codon, coding region and stop codon. Normally, if one can detect the promoter in a prokaryotic DNA sequence, one is able to find its gene coding region. Intron sequences range in size from about 80 nucleotides to 10 000 nucleotides or more. Introns in genes are of no function at all and are actually the genetic "junk" [1]. They differ dramatically from exons in that their exact nucleotide sequences seem to be unimportant. The only highly conserved sequences in introns are those required for intron removal.

The genetic information present in genes is expressed in organisms (*gene expression*) through the processes of *transcription* and *translation* (see Fig. 1). Transcription is the process for the production of a specific molecule, namely messenger RNA (mRNA), from a given sequence of DNA in a gene [3,4]. In this process, the genetic information (message) carried in the DNA is transcribed to (or written into) the mRNA. As its name implies, messenger RNA carries a message. The process by which mRNA directs the synthesis of a

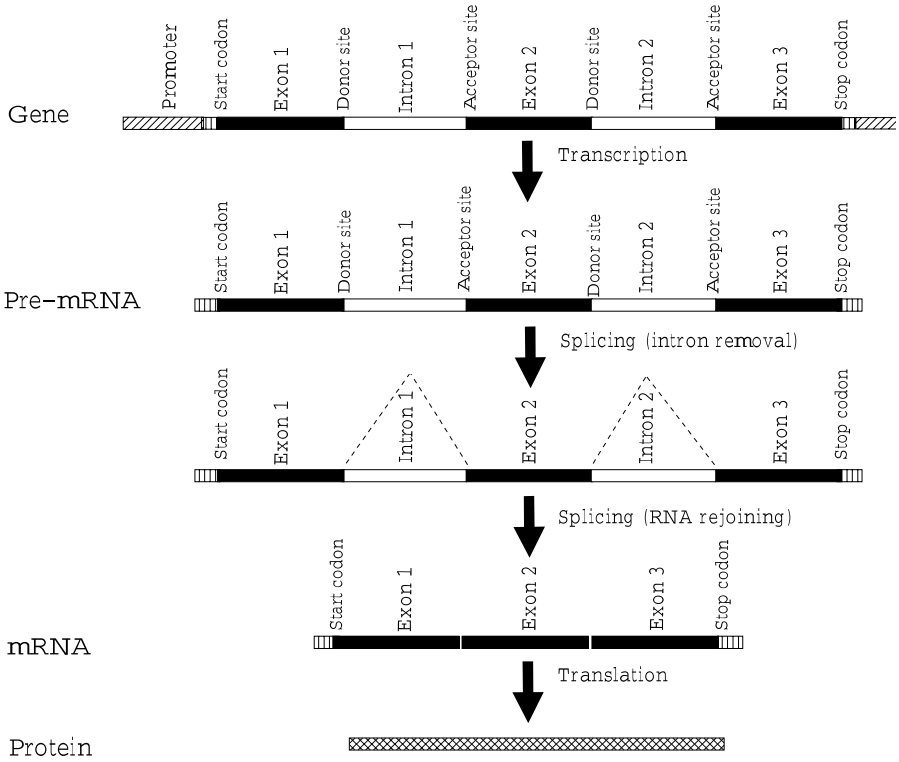

Fig. 1. Gene structure and gene expression processes. The basic gene structure for higher eukaryotes includes promoter, start codon, exons, introns and stop codon, etc. The boundaries between the exons and the introns are called 5′ donor sites, and the boundaries between the introns and the exons are called 3′ acceptor sites. During the DNA transcription process, the gene sequences (excluding the promoter region) are first transcribed into pre-mRNA.Then, the intron sequences in the pre-mRNA are removed and the RNA fragments are rejoined together by the RNA splicing process to get mRNA. The mRAN directs protein synthesis through the gene translation process.

specific protein is called translation. In this process, the information (message) carried in the mRNA sequence is translated into the amino acid sequence of the protein.

In the eukaryotic gene transcription process, the intermediate product is called pre-mRNA. Pre-mRNA is a direct copy of the DNA sequence in the eukaryotic gene and it contains the exon and intron sequences from the gene. The intron sequences will be removed from pre-mRNA, so a mature mRNA only consists of exon sequences, which will be translated into protein. The process for intron removal is called RNA splicing, and the positions for intron removal and RNA rejoining are called splicing junction sites (see Fig. 1). The consensus sequences at each end of an intron are nearly the same in all known

intron sequences. The conserved boundary sequence at the 5′ splice site is called a *donor site*, and the one at the 3′ splice site is called an *acceptor site*. The donor site and acceptor site are collectively referred to as splicing junction sites. The RNA breaking and rejoining (splicing) must be carried out precisely because an error of even one nucleotide would shift the reading frame in the resulting mRNA molecule and make nonsense of its message [1].

1.2. A bioinformatics approach

Identification or prediction of coding sequences from within a genomic DNA sequence has been a major rate-limiting step in the pursuit of genes. For eukaryotic gene detection, we have to detect the start codon, exons, introns and stop codon. How to find out exons/introns? The most important step is to detect splicing junction sites including donor and acceptor sites, because once the splicing junction sites are detected, the exon/intron boundaries are found. Then we can remove the introns from the DNA sequence to get coding regions. Biologists study gene structures based on lab experiments such as PCR on cDNA libraries, Northern blot, sequencing, etc. However, characterizing the 60 000–100 000 genes thought to be hidden in the human genome by means of merely lab experiments is not feasible. A current trend is to complement the lab study with a bioinformatics approach.

The bioinformatics approach for gene detection means using computer programs to elucidate a gene structure from DNA sequence signals, including start codon, splicing junction donor sites and acceptor sites, stop codon, etc. Since 1990s, a number of programs have been developed for locating gene coding regions (exons). However, the vertebrate DNA sequence signals involved in gene determination are usually ill defined, degenerate and highly unspecific. Given the current detection methods it is usually impossible to distinguish the signals truly processed by the cellular machinery from those that are apparently nonfunctional [8]. Furthermore, the inherent conservatism of the currently popular methods such as similarity search, GRAIL, etc., greatly limits our ability for making unexpected biological discoveries from increasingly abundant genomic data. Except for a very limited subset of trivial cases, the automated interpretation without experimental validation of genomic data is still a myth [7]. Unlike the situation in bacteria and yeast organisms, in which computer systems have substantially contributed to the automatic analysis of genomes, automatic sequence analysis and structure elucidation for the genomes of high eukaryotic organisms are far from being a reality [8].

Our research is targeted toward developing effective and accurate methods for identifying gene structures in the genomes of high eukaryotic organisms. The first phase of our research, introduced in this paper, is for gene structure signal detection. Then, during the second phase, we will combine the gene

structure signal information with global gene structure information to develop a full gene structure detection system. Splicing junction donor and acceptor sites are the most important functional gene structure signals. Earlier we developed a Motif model and used pattern matching techniques for donor prediction [12,13]. We also reported our case studies and preliminary results for predicting splicing junction acceptors [14–16]. In this paper, we systematically introduce our approach that uses hidden Markov models (HMMs) to represent the degeneracy features of the splicing junction sites. We developed **TEM**, an EM-like algorithm, to train our HMM system. Then we use the 10-way cross-validation method to evaluate our system for detecting splicing junction sites in unlabeled DNA sequences.

HMMs have been used extensively to describe sequential data or processes such as speech recognition. Researchers in computational biology have recently started to use HMMs for biological sequence analysis. Lukashin, Borodovsky [10] and their colleagues [5] successfully applied HMMs to the detection of protein coding regions in prokaryote. Audic and Claverie [2] reported their use of Markov transition matrices to detect eukaryotic promoters. Salzberg [11] used HMMs to identify splice junction sites and translational start sites in eukaryotic genes; his group also developed an HMM system, called Viterbi exon–intron locator (VEIL), for finding eukaryotic genes [9]. Our approach differs from Salzberg's by using a different topology of HMMs and by employing two modules for implementing the HMMs: one for true sites, and the other for false sites. Even though the current systems are far from being powerful enough for gene structure elucidation, the information these researchers provide is valuable, and the research on automated gene detection using HMMs is of great potentiality.

2. Using HMMs to model splicing junction sites

2.1. The Donor Model

Splicing junction sites in vertebrate DNA include donor and acceptor sites. Donor sites are conserved boundary sequences at the 5′ splicing sites in DNA. The conserved sequences include 9 nucleotide bases with GT (GU in mRNA) almost invariable to all donor sites [1]. An example of a donor site is shown below:

$$\overbrace{\cdots CAG}^{\text{exon}}\overbrace{\text{GT}GAGA\cdots}^{\text{intron}}$$

The nucleotide G occurs at position 4 and the nucleotide T occurs at position 5 in a donor site. We refer to a 9-base sequence that exists as a donor in a real gene sequence as a *true donor site*. Note that in all true donor sites, G and T

occur at position 4 and position 5, respectively. We refer to a 9-base nondonor sequence in which G and T also occur at position 4 and position 5, respectively, as a *false donor site*. Notice that we do not consider those sequences without G, T being at position 4 and position 5, respectively, because they are deemed to be nondonor sequences. Given an unlabeled 9-base sequence with G, T being at position 4 and position 5, respectively, referred to as a *candidate donor site*, our algorithm tries to determine whether the candidate sequence is a true donor site or a false donor site. We design a Donor Model, defined below, based on HMMs to describe the consensus and degenerate properties occurring in true donor sites.

An HMM with 9 states and a set of transitions is used for modeling a true donor site, which is represented as a digraph where states correspond to vertices and transitions to edges. At each state, the HMM generates a base b in {A, G, C, T} according to the state and transition probabilities, with the exception of state 4 and state 5. At state 4, the HMM constantly generates base $b = \mathrm{G}$, and at state 5, the HMM constantly generates base $b = \mathrm{T}$. Each state s is associated with a discrete probability distribution, $P(s)$. For state 4 and state 5, $P(s) = 1$. Except at state 3 and state 4, each base b at a state has four possible transitions to the next state. Each transition has a probability, $P(t)$, which represents the probability that the HMM makes that transition. Each base at state 3 has a fixed transition, namely $P(t) = 1$, to the base G at state 4. Similarly, at state 4, the base G has a fixed transition, namely $P(t) = 1$, to the base T at state 5. Fig. 2 illustrates the Donor Model.

2.2. The Acceptor Model

Acceptor sites are conserved boundary sequences at the 3′ splicing sites in DNA. The conserved sequences include 16 nucleotide bases with AG almost invariable to all acceptor sites [1]. An example of an acceptor site is shown below:

$$\overbrace{\cdots CTATCCTTCTCACAG}^{\text{intron}}\overbrace{G\cdots}^{\text{exon}}$$

The nucleotide A occurs at position 14 and the nucleotide G occurs at position 15 in an acceptor site. We refer to a 16-base sequence that exists as an acceptor in a real gene sequence as a *true acceptor site*. Note that in all true acceptor sites, A and G occur at position 14 and position 15, respectively. We refer to a 16-base nonacceptor sequence in which A and G also occur at position 14 and position 15, respectively, as a *false acceptor site*. Given an unlabeled 16-base sequence with A, G being at position 14 and position 15, respectively, referred to as a *candidate acceptor site*, our algorithm tries to determine whether the candidate sequence is a true acceptor site or a false acceptor site. We use an

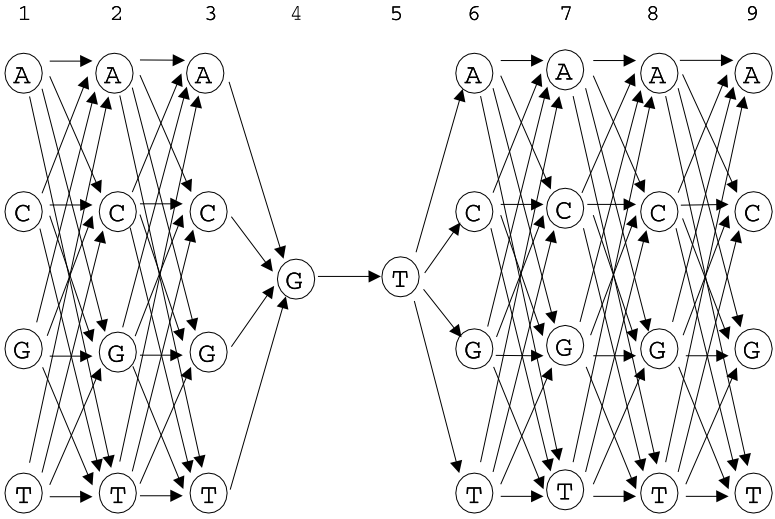

Fig. 2. The Donor Model. There are 9 states in this model. Except for state 4 and state 5, there are four possible bases at each state, and a base at one state may have four possible ways to transit to the next state. States 4 and 5 are a constant, and the transition from state 4 to state 5 is also a constant with the probability of 1. In a gene sequence, states 1–3 belong to an exon and states 4–9 are part of an intron.

Acceptor Model, defined below, to describe the consensus and degenerate properties occurring in true acceptor sites.

An HMM with 16 states and a set of transitions is developed for modeling a true acceptor site, which is represented as a digraph where states correspond to vertices and transitions to edges. At each state, the HMM generates a base b in {A, G, C, T} according to the state and transition probabilities, with the exception of state 14 and state 15. At state 14, the HMM constantly generates base $b = \mathrm{A}$, and at state 15, the HMM constantly generates base $b = \mathrm{G}$. Each state s is associated with a discrete probability distribution, $P(s)$. For state 14 and state 15, $P(s) = 1$. Except at state 13 and state 14, each base b at a state has four possible transitions to the next state. Each transition has a probability, $P(t)$, which represents the probability that the HMM makes that transition. Each base at state 13 has a fixed transition, namely $P(t) = 1$, to the base A at state 14. Similarly, at state 14, the base A has a fixed transition, namely $P(t) = 1$, to the base G at state 15. Fig. 3 illustrates the Acceptor Model. There are 16 states in this model. Except state 14 and state 15, there are four possible bases at each state, and a base at one state may have four possible ways to transit to the next state. States 14 and 15 are a constant, and the transition from state 14 to state 15 is also a constant with a probability of 1. In a gene sequence, states 1–15 belong to an intron and state 16 is the first base of an exon.

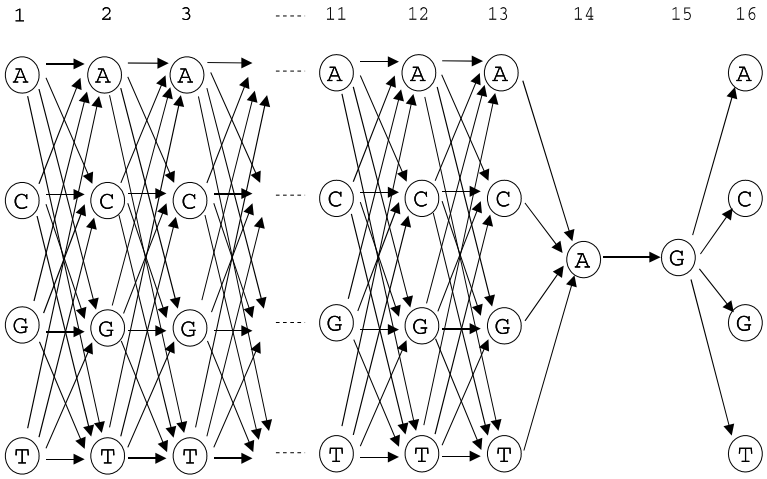

Fig. 3. The Acceptor Model. There are 16 states in this model. Except for state 14 and state 15, there are four possible bases at each state. Except for state 13 and state 14, a base at one state may have four possible ways to transit to the next state. States 14 and 15 are a constant, and the transition from state 14 to state 15 is also a constant with the probability of 1. In a gene sequence, states 1–15 belong to an intron and state 16 is part of an exon.

2.3. Two modules for each model

In vertebrate DNA sequences, there are much more false splicing junction sites than true sites. The ratio between the number of false sites and the number of true sites is about 100–1. In order to mine out the differences between the true sites and false sites, we implement two programs, True Donor Module and False Donor Module, based on the Donor Model and another two programs, True Acceptor Module and False Acceptor Module, based on the Acceptor Model. The True Donor Module and True Acceptor Module are collectively referred to as *true site modules*. The False Donor Module and False Acceptor Module are collectively referred to as *false site modules*. We train the true site modules using the true sites in the training data set, and train the false site modules using the false sites in the training data set. A given candidate site is tested by these modules. Let S_{cand} be a candidate site. Let $P(Y = 1 \mid S_{\text{cand}}, M^{(\text{t})})$ be the probability of S_{cand} being a donor (acceptor) sequence given that it is processed by a true site module. Let $P(Y = 0 \mid S_{\text{cand}}, M^{(\text{f})})$ be the probability of S_{cand} being a nondonor (nonacceptor) sequence given that it is processed by a false site module. In the above specification, $M^{(\text{t})}$ is for the true site modules and $M^{(\text{f})}$ is for the false site modules. In the splicing junction detection phase, these true site modules and false site modules are used to classify candidate sequences into right categories. For example, for a candidate donor site, we

first pass it through True Donor Module to get $P(Y = 1 | S_{\text{cand}}, M^{(\text{t})})$, the probability of this candidate site being a donor sequence. Then we pass it through False Donor Module to get $P(Y = 0 | S_{\text{cand}}, M^{(\text{f})})$, the probability of this candidate site being a nondonor sequence. Comparing these two values, we assign a score to the candidate sequence. This candidate sequence is assigned to the true donor category or false donor category depending on its score obtained.

3. Algorithms

The algorithms described in this section can be used for both the Donor Model and the Acceptor Model. For illustration purposes, we focus on the Donor Model and its corresponding modules, True Donor Module and False Donor Module. The algorithms for the Acceptor Model are essentially the same.

3.1. Training algorithm

We develop a modified expectation maximization (EM) algorithm, called **TEM**, for training the modules. The original EM method takes, as the input, a set of unaligned sequences and a motif length, and returns a probabilistic model for the motif [3]. Because our data set contains splicing junction sites with the same length, and all these sites can be aligned to each other, we design **TEM** specifically for training a HMM with fixed topology.

Let M represent the set of sequences that are randomly picked from our positive training data set and negative training data set. (In the study presented here, M contains about 200 true donor sites and 14 000 false donor sites.) Each sequence in M is labeled as *positive* or *negative* depending on whether it is from the positive training data set or the negative training data set. Let E^{t} be the set containing the remaining sequences in the positive training data set, and let E^{f} represent the set containing the remaining sequences in the negative training data set. There are much more true (false, respectively) donor sites in E^{t} (E^{f}, respectively) than those in M. (In our study presented here, the total number of the sequences in E^{t} and E^{f} is about nine times of the number of sequences in M.) Let P be a subset of M.

In the training phase, the **TEM** algorithm proceeds iteratively to converge. At each iteration, the algorithm removes some sequences from E^{t} and E^{f} and inputs those sequences into True Donor Module and False Donor Module. The algorithm then uses these modules to determine which sequences are placed in P as we will explain later. We use S_n^{em} to represent the *sensitivity* and use S_p^{em} to represent the *specificity* during the **TEM** training. S_n^{em} is the ratio between the number of true donor sites in P and the total number of true donor sites in M; note that $P \subseteq M$. S_p^{em} is the ratio between the number of true donor sites in P and the total number of sequences in P. The goal of our **TEM** training is, given

a fixed value of S_n^{em}, we train the modules iteratively to get a maximal value of S_p^{em}, or until E^{t} and E^{f} become empty. In this research, we use $S_n^{\mathrm{em}} = 0.90$ for training the modules.

Specifically, let T_{states} represent the total number of states in the Donor Model. Let b_i ($b_i \in \{\mathrm{A}, \mathrm{G}, \mathrm{C}, \mathrm{T}\}$) be the base at state i, $1 \leqslant i \leqslant T_{\mathrm{states}}$. Let $tr_i(b_i, b_{i+1}), 1 \leqslant i \leqslant T_{\mathrm{states}} - 1$, be the transition from state i to state $i+1$. The topology for the Donor Model is fixed, and all of the transition probabilities and state probabilities are initialized to random values. Then we pick one-tenth of the sequences from E^{t} and input them into True Donor Module. At the same time, we pick one tenth of the sequences from E^{f} and input them into False Donor Module. We record the number of the individual bases, b_i, at each state and the number of individual transitions, $tr_i(b_i, b_{i+1})$, from one state to the next state. We then compute the post probabilities for all the states and transitions in True Donor Module and False Donor Module. Let $T^{(\mathrm{t})}tr_i(b_i, b_{i+1})$ be the total number of transitions from a base b_i at state i to a base b_{i+1} at state $i+1$ in True Donor Module. Let $T_{\mathrm{in}}^{(\mathrm{t})}$ be the total number of true donor sites that have been input into True Donor Module. The state transition probabilities, $f\,tr_i^{(\mathrm{t})}(b_i, b_{i+1})$, in True Donor Module can be calculated as follows:

$$f\,tr_i^{(\mathrm{t})}(b_i, b_{i+1}) = \frac{T^{(\mathrm{t})}tr_i(b_i, b_{i+1})}{T_{\mathrm{in}}^{(\mathrm{t})}}. \tag{1}$$

Similarly, let $T^{(\mathrm{f})}tr_i(b_i, b_{i+1})$ be the total number of transitions from a base b_i at state i to a base b_{i+1} at state $i+1$ in False Donor Module. Let $T_{\mathrm{in}}^{(\mathrm{f})}$ be the total number of false donor sites that have been input into False Donor Module. The state transition probabilities, $f\,tr_i^{(\mathrm{f})}(b_i, b_{i+1})$, in False Donor Module can be calculated as follows:

$$f\,tr_i^{(\mathrm{f})}(b_i, b_{i+1}) = \frac{T^{(\mathrm{f})}tr_i(b_i, b_{i+1})}{T_{\mathrm{in}}^{(\mathrm{f})}}. \tag{2}$$

Next, we treat all the sequences in M as unlabeled sequences and input them into True Donor Module and False Donor Module. Let $P(True\,|\,S,\ M^{(\mathrm{t})})$ denote the probability of a sequence S in set M being a donor sequence, and let $P(False\,|\,S,\ M^{(\mathrm{f})})$ denote the probability of S being a nondonor sequence. In order to calculate $P(True\,|\,S,\ M^{(\mathrm{t})})$, we must calculate the probability of the sequence S given the sequence is a true donor site using True Donor Module. This can be written as

$$P(S\,|\,True,\ M^{(\mathrm{t})}) = \prod_{i=1}^{T_{\mathrm{states}}-1} f\,tr_i^{(\mathrm{t})}(b_i, b_{i+1}), \quad b_i \in \{\mathrm{A}, \mathrm{G}, \mathrm{C}, \mathrm{T}\}. \tag{3}$$

Our **TEM** algorithm uses Bayes' rule to estimate $P(True\,|\,S,\ M^{(\mathrm{t})})$ from $P(S\,|\,True,\ M^{(\mathrm{t})})$. Bayes' rule states that

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{4}$$

so,

$$P(True\,|\,S,\ M^{(\mathrm{t})}) = \frac{P(S\,|\,True,\ M^{(\mathrm{t})})P(True)}{P(S)}. \tag{5}$$

$P(True)$ is the prior probability that is assumed to be a constant, and $P(S)$ is the product of the individual base probabilities in the sequences. $P(S)$ can be written as

$$P(S) = \prod_{i=1}^{T_{\mathrm{states}}} P(b_i\,|\,True,\ M^{(\mathrm{t})}). \tag{6}$$

In the same way, we can write equations for calculating $P(False\,|\,S,\ M^{(\mathrm{f})})$ as follows:

$$P(S\,|\,False,\ M^{(\mathrm{f})}) = \prod_{i=1}^{T_{\mathrm{states}}-1} f\,tr_i^{(\mathrm{f})}(b_i, b_{i+1}), \quad b_i \in \{\mathrm{A}, \mathrm{G}, \mathrm{C}, \mathrm{T}\}, \tag{7}$$

$$P(False\,|\,S,\ M^{(\mathrm{f})}) = \frac{P(S\,|\,False,\ M^{(\mathrm{f})})P(False)}{P(S)}, \tag{8}$$

$$P(S) = \prod_{i=1}^{T_{\mathrm{states}}} P(b_i\,|\,False,\ M^{(\mathrm{f})}). \tag{9}$$

Let *pratio* be the probability ratio of sequence S in set M, and

$$pratio = \frac{P(True\,|\,S,\ M^{(\mathrm{t})})}{P(False\,|\,S,\ M^{(\mathrm{f})})}. \tag{10}$$

The *pratio* is calculated for each sequence in set M. We then sort the sequences in set M, in the descending order, according to their *pratio* values. Suppose the total number of positive sequences in set M is N. Then we select the *pratio* value for the $N \times S_n^{\mathrm{em}}$th positive sequence and use that *pratio* value as the positive lower bound, denoted L_p. (In the study presented here, there are 200 positive sequences in set M and the sensitivity S_n^{em} is 0.9. Therefore, L_p is the *pratio* value of the 180th positive sequence in set M.) The **TEM** algorithm assigns a sequence $S \in M$ into set P if the *pratio* value for S is greater than or equal to L_p. Let $T_{(TP)}$ be the number of positive sequences in set M that are assigned into set P. Let $T_{(P+N)}$ be the total number of positive sequences in M. Then, by definition,

$$S_n^{\mathrm{em}} = \frac{T_{(TP)}}{T_{(P+N)}}. \tag{11}$$

Let $T_{(PP)}$ be the total number of sequences in M that are assigned into P. Then, by definition,

$$S_p^{\mathrm{em}} = \frac{T_{(TP)}}{T_{(PP)}}. \tag{12}$$

The re-estimation procedure then adjusts all of the probabilities hidden in the Donor Model in order to increase S_p^{em}. New sequences in E^{t} and E^{f} are picked and removed from E^{t} and E^{f}. These sequences are then run through True Donor Module and False Donor Module again and the probabilities are further refined. This process is iterated until the S_p^{em} is maximized or until E^{t} and E^{f} become empty. This algorithm is guaranteed to converge to a locally optimal estimate of all the probabilities in the Donor Model. The positive lower bound L_p that maximizes S_p^{em} will be an output and used in the splicing junction sites detection phase. Fig. 4 summarizes the **TEM** algorithm used in the training phase.

3.2. Algorithm for detecting splicing junction sites

As described in Section 2, a candidate donor site refers to a 9-base sequence fragment with bases G, T being at position 4 and position 5, respectively. The input of the site detection algorithm is a fragment, denoted S_{cand}, of 9 bases extracted from a genomic DNA sequence S with a minimum length of 9 bases. In this research, the longest DNA sequence used is about 50 000 bases long. The S_{cand} has G, T at position 4 and 5, respectively, and is considered as a candidate donor site. We refer to the 9 bases in S_{cand} as $b_1, b_2, \ldots, b_9$, respectively. The output of the site detection algorithm is a flag, $KIND_i$, indicating whether the S_{cand} starting at positing i of the genomic DNA sequence S is a true donor site or not.

Let $f\,tr_j^{(\mathrm{t})}(b_j, b_{j+1})$ be the probability of a transition from base b_j to base b_{j+1}, $1 \leqslant j \leqslant 8$, of S_{cand} using True Donor Module. We define a flag variable Y to be 1 if S_{cand} belongs to a true site category, and 0 otherwise. Let n be the length of the candidate site S_{cand} (n is 9 for donor sites and 16 for acceptor sites). Let $P(S_{\mathrm{cand}}|Y = 1, M^{(\mathrm{t})})$ be the probability of the candidate site S_{cand} given that it is a donor site processed by True Donor Module. Then

$$P(S_{\mathrm{cand}} \,|\, Y = 1, M^{(\mathrm{t})}) = \prod_{j=1}^{n-1} f\,tr_j^{(\mathrm{t})}(b_j, b_{j+1}), \quad b_j \in \{\mathrm{A}, \mathrm{G}, \mathrm{C}, \mathrm{T}\}. \tag{13}$$

As defined before, $P(Y = 1 \,|\, S_{\mathrm{cand}}, M^{(\mathrm{t})})$ is the probability of S_{cand} being a donor site given that it is processed by True Donor Module. According to Bayes' rule [cf. Eq. (4)]:

INPUT:

Untrained HMM site modules (including a true site module and a false site module);

Positive training data set, E^t;

Negative training data set, E^f;

TEM testing data set, M;

OUTPUT:

Fully trained HMM site modules and L_p;

ALGORITHM:

unmaximized := true;

while unmaximized **do begin**

unmaximized := false;

if E^t is not empty **then begin**

remove one tenth of the sequences from E^t and input them into the true site module;

for $i = 1$ **to** $T_{states} - 1$

calculate $ftr_i^{(t)}(b_i, b_{i+1})$ as in Equation (1);

end;

if E^f is not empty **then begin**

remove one tenth of the sequences from E^f and input them into the false site module;

for $i = 1$ **to** $T_{states} - 1$

calculate $ftr_i^{(f)}(b_i, b_{i+1})$ as in Equation (2);

end;

for each sequence $S \in M$ **do begin**

calculate $P(True \mid S,\ M^{(t)})$ as in Equation (5);

calculate $P(False \mid S,\ M^{(f)})$ as in Equation (8);

calculate *pratio* as in Equation (10);

end;

select L_p;

calculate S_p^{em} according to L_p;

if (S_p^{em} is not maximized) and (either E^t or E^f is non-empty) **then**

unmaximized := true;

end;

Fig. 4. The **TEM** algorithm ued in the training phase.

$$P(Y=1\,|\,S_{\text{cand}},M^{(\text{t})})=\frac{P(S_{\text{cand}}|Y=1,M^{(\text{t})})P(Y=1)}{P(S_{\text{cand}})}. \tag{14}$$

When examining a set of sequences to detect true donor sites, we can treat the underlying prior $P(Y=1)$ as a constant [11]. $P(S_{\text{cand}})$ is the product of the individual base probabilities for $b_1,b_2,\ldots,b_n$ in S_{cand}:

$$P(S_{\text{cand}})=\prod_{j=1}^{n}P(b_j\,|\,Y=1,M^{(\text{t})}),\quad b_j\in\{\text{A},\text{G},\text{C},\text{T}\}. \tag{15}$$

Similarly, we use False Donor Module to compute $P(Y=0\,|\,S_{\text{cand}},M^{(\text{f})})$, the probability of S_{cand} being a false donor site given that it is processed by False Donor Module. So, we can write the false donor site counterparts of the above equations:

$$P(S_{\text{cand}}\,|\,Y=0,M^{(\text{f})})=\prod_{j=1}^{n-1}f\,tr_j^{(\text{f})}(b_j,b_{j+1}),\quad b_j\in\{\text{A},\text{G},\text{C},\text{T}\}, \tag{16}$$

$$P(Y=0\,|\,S_{\text{cand}},M^{(\text{f})})=\frac{P(S_{\text{cand}}\,|\,Y=0,M^{(\text{f})})P(Y=0)}{P(S_{\text{cand}})}, \tag{17}$$

$$P(S_{\text{cand}})=\prod_{j=1}^{n}P(b_j\,|\,Y=0,M^{(\text{f})}),\quad b_j\in\{\text{A},\text{G},\text{C},\text{T}\}. \tag{18}$$

Given the candidate donor site S_{cand} starting at position i in the genomic DNA sequence S, our algorithm will find the two most likely sets of states through the two HMM modules for S_{cand}. Then the algorithm calculates $P(Y=1\,|\,S_{\text{cand}},M^{(\text{t})})$ and $P(Y=0\,|\,S_{\text{cand}},M^{(\text{f})})$. A score, *sratio*, is assigned to the candidate site based on the scoring function below:

$$sratio=\frac{P(Y=1\,|\,S_{\text{cand}},M^{(\text{t})})}{P(Y=0\,|\,S_{\text{cand}},M^{(\text{f})})}. \tag{19}$$

Comparing *sratio* with the L_p obtained from the training phase, we assign a flag, $KIND_i$, to the candidate site S_{cand} based on the following formula:

$$KIND_i=\begin{cases}1 & \text{if } sratio\geqslant L_p,\\ 0 & \text{otherwise.}\end{cases} \tag{20}$$

The candidate site S_{cand} is classified as a true donor site if $KIND_i$ has a value of 1. S_{cand} is classified as a false donor site if $KIND_i$ has a value of 0. Fig. 5 illustrates the site detection algorithm.

INPUT:

A candidate donor site S_{cand} starting at position i of an unlabeled genomic DNA sequence

OUTPUT:

/* $KIND_i$ is a flag indicating whether S_{cand} is a true donor site or not. */

$KIND_i$;

ALGORITHM:

present S_{cand} to True Donor Module and calculate

$P(Y = 1 \mid S_{cand}, M^{(t)})$ as in Equation (14);

present S_{cand} to False Donor Module and calculate

$P(Y = 0 \mid S_{cand}, M^{(f)})$ as in Equation (17);

calculate $sratio$ as in Equation (19);

calculate $KIND_i$ as in Formula (20);

Fig. 5. Algorithm for classifying splicing junction donor sequences.

4. Experiments and results

4.1. Sequence data and evaluation method

In order to evaluate the accuracy of our HMM system for splicing junction site detection, we used the database of DNA sequences originally collected by Burset and Guigo [6], who used this database to compare a number of major gene-finding programs. The sequences in this database were obtained from the vertebrate divisions of GenBank release 85.0 (October, 1994). There are 570 vertebrate sequences in the database and they all have simple and standard gene structures. Each entry contains a complete protein coding sequence with no in-frame stop codons. There are at least one exon and one intron in all entries in the database. There are 2079 true donor sites and 2079 true acceptor sites, all of which are standard splicing junction sites. This means that all the donor sites have 'GT' and all the acceptor sites have 'AG' at the right positions. This database now becomes the standard data set for evaluating gene-finding programs.

We applied the 10-way cross-validation method [14] to evaluate how well our HMM system performs when tested on data that are not in the training dataset. Cross-validation is a standard experimental technique for determining how well a classifier performs on unseen data [9]. Specifically, we randomly partition the 570 sequences at hand into 10 sets. These sets have roughly the same number of true donor sites; the sets also have roughly the same number of true acceptor sites. For each iteration in the 10-way cross-validation experiment, we use nine out of the ten sets as the training data set, and use the

remaining one set as the test data set. The HMM system is trained using the training data set (i.e., all sequences excluding those in the test data set are used as the training data). The system is then tested on the sequences in the test data set. Thus, the training data set consists of 90% and the test data set consists of 10% of the sequences. Each time in the 10-way cross-validation experiment, the HMM system is trained with sequences containing about 1871 true sites and 135 000 false sites. The HMM system is tested on the sequences containing about 208 true sites and 14 000 false sites.

4.2. Experimental results

The state transition probabilities for the Donor Model and the Acceptor Model are shown in Tables 1–4. Comparing the state transition probabilities of the true site modules with those of the false site modules, we see that the proposed HMM system maximizes the differences between the true sites and false sites. The results for detecting splicing junction sites are summarized in Tables 5 and 6. The results for each of the 10 test sets of the cross-validation

Table 1
State transition probabilities for True Donor Module

	$f\,tr_i^{(t)}(b_i, b_{i+1})$							
i	1	2	3	4	5	6	7	8
A → A	0.21	0.04	Null	Null	Null	0.32	0.04	0.01
A → G	0.05	0.51	0.08	Null	Null	0.06	0.63	0.03
A → C	0.02	0.01	Null	Null	Null	0.06	0.02	0.01
A → T	0.04	0.04	Null	Null	Null	0.06	0.02	0.02
G → A	0.13	0.02	Null	Null	Null	0.37	0.01	0.12
G → G	0.02	0.11	0.81	Null	Null	0.04	0.10	0.13
G → C	0.03	0.00	Null	Null	Null	0.02	0.01	0.12
G → T	0.02	0.01	Null	1.00	Null	0.01	0.00	0.46
C → A	0.23	0.02	Null	Null	Null	0.02	0.02	0.01
C → G	0.02	0.07	0.02	Null	Null	0.00	0.03	0.00
C → C	0.04	0.01	Null	Null	Null	0.00	0.02	0.02
C → T	0.05	0.02	Null	Null	Null	0.01	0.02	0.02
T → A	0.02	0.00	Null	Null	0.50	0.01	0.00	0.00
T → G	0.04	0.12	0.08	Null	0.44	0.02	0.07	0.02
T → C	0.03	0.01	Null	Null	0.03	0.00	0.01	0.01
T → T	0.03	0.01	Null	Null	0.03	0.00	0.00	0.01

The state transition probability values are of 'double' type in our computer programs. In order to save space, the values are rounded to the second position following the decimal point to fit into this table. For example, a probability value of 0.13293 is shown in this table as 0.13, but 0.13593 is shown here as 0.14. Theoretically, the sum of the transition probabilities from one state to the next state should equal to 1.00. Because of the rounding, the sum of the values in each column in this table may be slightly smaller or greater than 1.00. This holds in Tables 2–4 as well.

Table 2
State transition probabilities for False Donor Module

	$f\,tr_i^{(f)}(b_i, b_{i+1})$							
i	1	2	3	4	5	6	7	8
A → A	0.08	0.08	Null	Null	Null	0.05	0.06	0.06
A → G	0.07	0.08	0.28	Null	Null	0.05	0.07	0.07
A → C	0.05	0.02	Null	Null	Null	0.04	0.05	0.04
A → T	0.05	0.08	Null	Null	Null	0.05	0.05	0.06
G → A	0.07	0.07	Null	Null	Null	0.09	0.05	0.06
G → G	0.07	0.08	0.27	Null	Null	0.10	0.07	0.07
G → C	0.06	0.02	Null	Null	Null	0.07	0.05	0.05
G → T	0.05	0.09	Null	1.00	Null	0.09	0.05	0.07
C → A	0.08	0.08	Null	Null	Null	0.06	0.07	0.07
C → G	0.02	0.02	0.08	Null	Null	0.01	0.02	0.02
C → C	0.07	0.02	Null	Null	Null	0.07	0.07	0.07
C → T	0.06	0.12	Null	Null	Null	0.08	0.08	0.09
T → A	0.05	0.04	Null	Null	0.18	0.04	0.05	0.05
T → G	0.09	0.09	0.37	Null	0.35	0.06	0.09	0.08
T → C	0.07	0.02	Null	Null	0.22	0.06	0.07	0.06
T → T	0.07	0.09	Null	Null	0.25	0.09	0.08	0.08

are shown, so are the average results for all the 10 test sets. In Tables 5 and 6, TP is the number of true positives. FP is the number of false positives. TN is the number of true negatives. FN is the number of false negatives. A true positive is a true donor (true acceptor, respectively) site that is also classified as a true donor (true acceptor, respectively) site. A false positive is a false donor (false acceptor, respectively) site that is mis-classified as a true donor (true acceptor, respectively) site. A true negative is a false donor (false acceptor, respectively) site that is also classified as a false donor (false acceptor, respectively) site. A false negative is a true donor (true acceptor, respectively) site that is mis-classified as a false donor (false acceptor, respectively) site. S_n^{true} is the ratio between the number of correctly classified true donor (true acceptor, respectively) sites and the total number of true donor (true acceptor, respectively) sites in the test data set, i.e.,

$$S_n^{\text{true}} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{21}$$

We also did similar calculations to evaluate the performance of the proposed HMM system in predicting the false splicing junction sites. S_n^{false} is the ratio between the number of correctly classified false donor (false acceptor, respectively) sites and the total number of false donor (false acceptor, respectively) sites in the test data set, i.e.,

Table 3
State transition probabilities for True Acceptor Module

	$f\,tr_i^{(t)}(b_i, b_{i+1})$														
i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A → A	0.01	0.01	0.00	0.01	0.02	0.01	0.00	0.01	0.01	0.00	0.02	0.01	0.02	Null	Null
A → G	0.01	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	Null	1.00	Null
A → C	0.04	0.04	0.03	0.02	0.03	0.03	0.03	0.03	0.04	0.03	0.02	0.17	Null	Null	Null
A → T	0.05	0.03	0.03	0.03	0.03	0.03	0.05	0.01	0.03	0.01	0.00	0.04	Null	Null	Null
G → A	0.02	0.01	0.02	0.00	0.01	0.01	0.00	0.01	0.00	0.01	0.01	0.00	0.01	Null	0.30
G → G	0.03	0.03	0.04	0.03	0.03	0.02	0.03	0.03	0.01	0.02	0.03	0.00	Null	Null	0.48
G → C	0.04	0.04	0.05	0.03	0.03	0.03	0.03	0.05	0.03	0.01	0.02	0.23	Null	Null	0.14
G → T	0.05	0.05	0.04	0.05	0.05	0.04	0.06	0.04	0.04	0.01	0.01	0.03	Null	Null	0.09
C → A	0.04	0.03	0.03	0.04	0.03	0.05	0.02	0.04	0.02	0.03	0.13	0.01	0.79	Null	Null
C → G	0.03	0.03	0.01	0.01	0.02	0.01	0.03	0.01	0.01	0.01	0.06	0.00	Null	Null	Null
C → C	0.14	0.14	0.14	0.14	0.14	0.16	0.17	0.18	0.24	0.25	0.16	0.24	Null	Null	Null
C → T	0.18	0.15	0.17	0.21	0.18	0.16	0.15	0.19	0.17	0.21	0.08	0.07	Null	Null	Null
T → A	0.02	0.02	0.03	0.02	0.02	0.02	0.02	0.03	0.01	0.02	0.06	0.00	0.18	Null	Null
T → G	0.07	0.07	0.05	0.06	0.05	0.09	0.06	0.04	0.03	0.04	0.15	0.00	Null	Null	Null
T → C	0.14	0.15	0.18	0.17	0.18	0.16	0.19	0.19	0.18	0.14	0.12	0.14	Null	Null	Null
T → T	0.15	0.19	0.17	0.16	0.20	0.17	0.14	0.14	0.16	0.21	0.11	0.06	Null	Null	Null

Table 4
State transition probabilities for False Acceptor Module

	$f\,tr_i^{(f)}(b_i, b_{i+1})$														
i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A → A	0.05	0.05	0.05	0.06	0.06	0.06	0.06	0.08	0.05	0.06	0.05	0.04	0.28	Null	Null
A → G	0.08	0.10	0.10	0.08	0.11	0.08	0.08	0.09	0.13	0.11	0.09	0.06	Null	1.00	Null
A → C	0.05	0.05	0.04	0.05	0.07	0.03	0.06	0.06	0.04	0.04	0.05	0.04	Null	Null	Null
A → T	0.05	0.04	0.04	0.03	0.05	0.04	0.04	0.03	0.03	0.03	0.03	0.01	Null	Null	Null
G → A	0.08	0.07	0.08	0.11	0.07	0.08	0.08	0.07	0.09	0.07	0.04	0.07	0.36	Null	0.19
G → G	0.10	0.10	0.13	0.11	0.11	0.13	0.12	0.12	0.12	0.19	0.12	0.12	Null	Null	0.43
G → C	0.08	0.07	0.07	0.07	0.07	0.08	0.07	0.05	0.06	0.07	0.05	0.08	Null	Null	0.18
G → T	0.06	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.04	0.19	0.03	Null	Null	0.20
C → A	0.08	0.07	0.07	0.07	0.06	0.07	0.07	0.07	0.07	0.05	0.04	0.07	0.28	Null	Null
C → G	0.03	0.02	0.04	0.03	0.03	0.03	0.02	0.03	0.04	0.03	0.03	0.03	Null	Null	Null
C → C	0.06	0.07	0.06	0.06	0.06	0.08	0.07	0.08	0.07	0.06	0.08	0.10	Null	Null	Null
C → T	0.08	0.07	0.08	0.06	0.07	0.06	0.07	0.06	0.06	0.06	0.06	0.02	Null	Null	Null
T → A	0.04	0.03	0.02	0.04	0.03	0.03	0.04	0.03	0.03	0.03	0.03	0.09	0.08	Null	Null
T → G	0.07	0.10	0.08	0.09	0.09	0.09	0.07	0.08	0.09	0.07	0.07	0.15	Null	Null	Null
T → C	0.04	0.06	0.05	0.04	0.04	0.05	0.05	0.06	0.03	0.04	0.04	0.05	Null	Null	Null
T → T	0.04	0.04	0.04	0.04	0.04	0.04	0.03	0.04	0.04	0.03	0.03	0.01	Null	Null	Null

Table 5
Performance evaluation of the proposed HMM system used in detecting donor sites

Set	# of true donors	# of false donors	TP	FP	TN	FN	S_n^{true}	S_n^{false}	S_n
1	209	16,259	191	634	15,625	18	0.914	0.961	0.960
2	210	13,411	191	643	12,768	19	0.910	0.952	0.951
3	203	12,942	185	677	12,265	18	0.911	0.948	0.947
4	200	15,473	183	654	14,819	17	0.915	0.958	0.957
5	208	17,245	192	815	16,430	16	0.923	0.952	0.952
6	213	15,817	205	809	15,008	8	0.962	0.948	0.949
7	206	15,895	191	762	15,133	15	0.927	0.951	0.952
8	212	13,206	194	748	12,458	18	0.915	0.942	0.953
9	209	14,334	192	702	13,632	17	0.919	0.950	0.951
10	209	14,651	190	702	13,949	19	0.909	0.951	0.952
Average							0.921	0.951	0.952

Table 6
Performance evaluation of the proposed HMM system used in detecting acceptor sites

Set	# of true acceptors	# of false acceptors	TP	FP	TN	FN	S_n^{true}	S_n^{false}	S_n
1	209	21,553	198	1428	20,125	11	0.947	0.933	0.934
2	210	19,169	197	1371	17,798	13	0.938	0.928	0.929
3	203	19,995	183	1404	18,591	20	0.901	0.929	0.929
4	200	22,683	181	1364	21,319	19	0.905	0.939	0.940
5	208	24,721	194	1416	23,305	14	0.933	0.942	0.943
6	213	23,871	194	1392	22,479	19	0.911	0.941	0.941
7	206	22,877	186	1388	21,489	20	0.903	0.938	0.939
8	212	19,012	192	1400	17,612	20	0.906	0.925	0.926
9	209	20,798	189	1398	19,400	20	0.904	0.932	0.932
10	209	18,221	189	1377	16,844	20	0.904	0.924	0.923
Average							0.915	0.934	0.934

$$S_n^{\text{false}} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \tag{22}$$

where S_n is the proportion of the candidate sites in the test data set that are classified correctly. S_n tells how well the proposed HMM system can assign true sites and false sites into the right categories; it is calculated by the following formula:

$$S_n = \frac{N_c}{N_t} \tag{23}$$

where N_c is the number of the candidate sites in the test data set that are classified correctly and N_t is the total number of the candidate sites in the test data set.

The results in Table 5 show that, on average, our system can correctly detect 92% of the true donor sites in the test data set, and 95% of the false donor sites in the test data set are predicted as false sites. Overall, 95% of the candidate donor sites are classified into the right categories. The results for acceptor classification are shown in Table 6. The proposed HMM system can correctly predict 91.5% of the true acceptor sites in the test data set and 93% of the false acceptor sites in the test data set. In general, the system can assign 93% of the candidate acceptor sites into the right categories.

To investigate how well the proposed HMM system can discriminate true splicing junction sites from false splicing junction sites when a group of

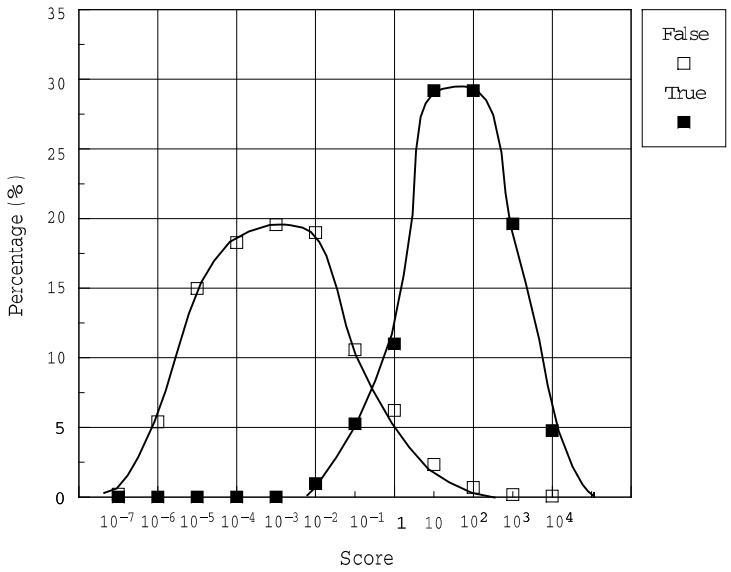

Fig. 6. Score distributions for true donor sites and false donor sites.

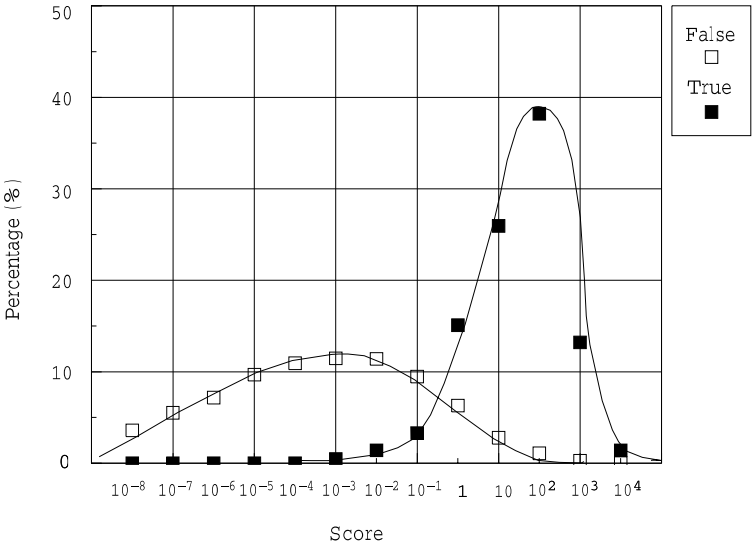

Fig. 7. Score distributions for true acceptor sites and false acceptor sites.

candidate sequences are presented to the system, we performed some statistic analysis on the scores the HMM system assigned to each candidate site in the 10-way cross-validation experiment. Fig. 6 shows the score distribution of true donor sites and false donor sites in one test data set. Fig. 7 shows the score distribution of true acceptor sites and false acceptor sites in the same test data set. Striking differences can be observed by comparing the curves in these figures. The scores for the true donor sites can be higher than 10 000, with about 85% of the true donor sites scoring more than 10. For the false donor sites, only about 5% of the sequences score more than 1, with the majority of the false donor sites scoring between 0.1 and 0.00001. More than 10% of the false donor sites score less than 0.00001. The score distribution for the true acceptor scores in Fig. 7 shares a similar pattern as the one for the true donor sites shown in Fig. 6. The scores for the false acceptor sites are more scattered, but again, there are only 5–6% of the sequences scoring more than 1. The results suggest that the proposed HMM system can be used to discover the degenerate features of the splicing junction sites to a great degree.

5. Conclusions

In this paper we have developed HMMs to represent the consensus and degenerate features of splicing junction sites in eukaryotic genes. The proposed Donor Model and Acceptor Model have a different topology from those

previously reported for splicing junction site detection. To capture the consensus and degenerate features of the splicing junction sites, we introduced constant states and constant state transitions into the HMMs. This innovative approach conceptually simplifies the splicing junction site models and the computation process of using the models. The results from the 10-way cross-validation experiment show that the proposed HMM system can correctly detect 92% of the true donor sites and 91.5% of the true acceptor sites in the standard sequence data set composed by Burset and Guigo.

It is worth to point out that we only use the local information in the proposed Donor Model and Acceptor Model. When combining our HMM system with the global gene structure information, it is likely that one can achieve even better results for site recognition. This is the research underway in our group. Currently, we are trying to integrate our HMM system with other gene structure information to develop an effective and accurate system for full gene structure detection.

References

[1] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, J.D. Watson, Molecular Biology of the Cell, third ed., Garland Publishing, New York and London, 1989.
[2] S. Audic, J. Claverie, Detection of eukaryotic promoters using Markov transition matrices, Comput. Chem. 21 (1997) 223–227.
[3] T.L. Bailey, M.E. Baker, C. Elkan, An artificial intelligence approach to motif discovery in protein sequences: application to steroid dehydrogenases, J. Steroid Biochem. 62 (1) (1997) 29–44.
[4] J.J.W. Baker, G.E. Allen, The Study of Biology, fourth ed., Addison-Wesley, Reading, MA, 1982.
[5] M. Borodovsky, J. McIninch, GENMARK: parallel gene recognition for both DNA strands, Comput. Chem. 17 (1993) 123–133.
[6] M. Burset, R. Guigo, Evaluation of gene structure prediction programs, Genomics 34 (3) (1996) 353–367.
[7] J. Claverie, O. Poirot, F. Lopez, The difficulty of identifying genes in anonymous vertebrate sequences, Comput. Chem. 21 (4) (1997) 203–214.
[8] R. Guigo, Computational gene identification: an open problem, Comput. Chem. 21 (4) (1997) 215–222.
[9] J. Henderson, S. Salzberg, K.H. Fasman, Finding genes in DNA with a hidden Markov model, J. Comput. Biol. 4 (2) (1997) 127–141.
[10] A.V. Lukashin, M. Borodovsky, GeneMark.hmm: new solutions for gene finding, Nucl. Acids Res. 26 (4) (1998) 1107–1115.
[11] S.L. Salzberg, A method for identifying splice sites and translational start sites in eukaryotic mRNA, Comput. Appl. Biosci. 13 (4) (1997) 365–376.
[12] J.T.L. Wang, S. Rozen Shapiro, B.A. Shasha, Z. Wang, M. Yin, New techniques for DNA sequence classification, J. Comput. Biol. 6 (2) (1999) 209–218.
[13] M.M. Yin, J.T.L. Wang, Algorithms for splicing junction donor recognition in genomic DNA sequences, in: Proceedings of the IEEE International Joint Symposiam on Intelligence and Systems, Rockville, Maryland, May 1998, pp. 169–176.

[14] M.M. Yin, J.T.L. Wang, Application of hidden Markov models to gene prediction in DNA, in: Proceedings of the IEEE International Conference on Information, Intelligence and Systems, Bethesda, Maryland, November 1999, pp. 40–47.
[15] M.M. Yin, J.T.L. Wang, Recognizing splicing junction acceptors in eukaryotic genes using hidden Markov models and machine learning methods, in: Proceedings of the Fifth Joint Conference on Information Sciences, Atlantic City, New Jersey, February 2000, pp. 786–789.
[16] M.M. Yin, J.T.L. Wang, Application of hidden Markov models to biological data mining: a case study, in: B.V. Dasarathy (Ed.), Data Mining and Knowledge Discovery: Theory, Tools, and Technology II, Proceedings of SPIE, vol. 4057, The International Society for Optical Engineering, USA, 2000, pp. 352–358.