Mining Genes in DNA Using GeneScout

Michael M. Yin
Department of Computer Science
New Jersey Institute of Technology
University Heights, Newark, NJ 07102, USA
mxy3100@njit.edu

Jason T. L. Wang
Department of Computer Science
New Jersey Institute of Technology
University Heights, Newark, NJ 07102, USA
wangj@oak.njit.edu

Abstract

In this paper, we present a new system, called GeneScout, for predicting gene structures in vertebrate genomic DNA. The system contains specially designed hidden Markov models (HMMs) for detecting functional sites including protein-translation start sites, mRNA splicing junction donor and acceptor sites, etc. Our main hypothesis is that, given a vertebrate genomic DNA sequence S, it is always possible to construct a directed acyclic graph G such that the path for the actual coding region of S is in the set of all paths on G. Thus, the gene detection problem is reduced to that of analyzing the paths in the graph G. A dynamic programming algorithm is used to find the optimal path in G. The proposed system is trained using an expectation-maximization (EM) algorithm and its performance on vertebrate gene prediction is evaluated using the 10-way cross-validation method. Experimental results show the good performance of the proposed system and its complementarity to a widely used gene detection system.

Keywords: Bioinformatics, Gene finding, Hidden Markov models, Knowledge discovery, Data mining

1 Introduction

Data mining, or knowledge discovery from data, refers to the process of extracting interesting, non-trivial, implicit, previously unknown and potentially useful information or patterns from data. In life sciences, this process could refer to finding clustering rules for gene expression, discovering classifications rules for proteins, detecting associations between metabolic pathways, predicting genes in genomic DNA sequences, etc. [6, 7].

Our research is targeted toward developing effective and accurate methods for automatically detecting gene structures in the genomes of high eukaryotic organisms. This paper presents a data mining system for automated gene discovery.

2 Our Approach

2.1 HMM Models for Predicting Functional Sites

Our proposed GeneScout system contains several specially designed HMM models for predicting functional sites as well as an HMM model for calculating coding potentials [8]. Often, the functional sites include (almost) invariant (consensus) nucleotides and other degenerate features. Thus, the invariant nucleotides themselves do not completely characterize a functional site. For example, a start codon is always a sequence of ATG and it is the start position in mRNA for protein translation, so the start codon is the first 3 bases of the coding region of a gene. ATG is also the codon for Methionine, a regular amino acid occurring at many positions in all of the known proteins. This means one is unable to detect the start codon by simply searching for ATG in a genomic DNA sequence.

It is reported that there are some statistic relations between a start codon ATG and the 13 nucleotides immediately preceding it and the 3 bases immediately following it [5]. We call these 19 bases containing a start codon a *start site*. We build an HMM model, called the Start Site Model, to model the start site. Figure 1 illustrates the model. As shown in Figure 1, there are 19 states for the Start Site Model. Except for states 14, 15 and 16, 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, 15 and 16 are constant states (representing a start codon), and the transitions from state 14 to 15 and from state 15 to 16 are also constant with a probability of 1. With the Start Site Model, we can use the HMM algorithms described in our previously published paper [9] to detect a start site.[1]

[1] In the previously published paper [9], we presented HMM models and algorithms for detecting splicing junction donor and acceptor sites. The HMM model for a donor site contains 9 states whereas the HMM model for an acceptor site contains 16 states. The algorithms used for training the HMM models for start sites, donor sites and acceptor sites and for detecting these functional sites are similar. Please see related publications [8, 9] for details.

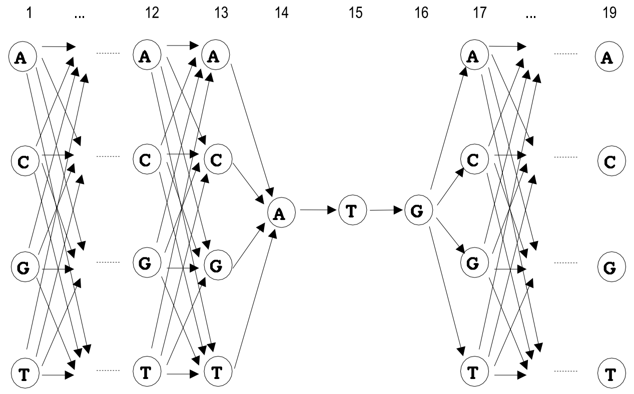

Figure 1. The Start Site Model.

2.2 Graph Representation of the Gene Detection Problem

The goal of GeneScout is to find coding regions. The main hypothesis used in our work is that, given a vertebrate genomic DNA sequence S, it is always possible to construct a directed acyclic graph G such that the path for the actual coding region of S is in the set of all paths on G. Thus, the gene detection problem is reduced to the analysis of paths in the graph G. We use dynamic programming algorithms to find the optimal path in G.

Consider a directed acyclic graph G where vertices are functional sites, and edges are exons and introns (Figure 2). All the edges from the top vertices to the bottom vertices in the graph G are candidate exons, and the edges from the bottom vertices to the top vertices are candidate introns. There must be a path on G representing real exons-introns as shown by the boldface edges in Figure 2. So, given a vertebrate genomic DNA sequence with detected sites, it is always possible to construct a directed acyclic graph G such that the path for real exons-introns is in the set of all paths on G. Thus the gene detection problem is reduced to the analysis of paths in the graph G [4].

2.3 A Dynamic Programming Algorithm

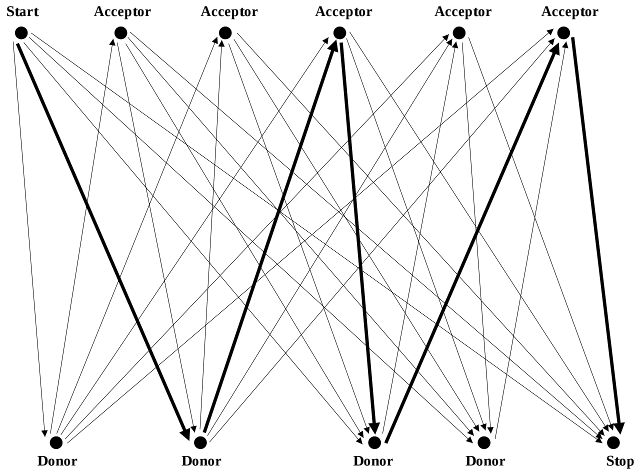

Figure 2. A site graph for gene detection with the boldface edges representing real exons-introns.

Consider again the graph G in Figure 2. A candidate gene is represented by a path in G. Let S_G denote the set of all paths in G. We assign a score to each functional site based on the HMM models and algorithms described in Section 2.1 [8, 9]. The score is used as the weight of the corresponding vertex v in S_G, and we denote that weight as $W(v)$. We associate each edge (v_1, v_2) in S_G with a weight $W(v_1, v_2)$. The weight $W(v_1, v_2)$ equals the coding potential of the candidate exon or intron corresponding to the edge (v_1, v_2) (the modeling and calculations of the coding potential are described in [8]).

Now, let v be a vertex in S_G and let $(v_1, v), \ldots, (v_k, v)$ be all edges entering v. Let $S(v)$ be the set of all paths entering the vertex v. We can calculate the weight of the optimal path in $S(v)$, denoted $\Theta(S(v))$, as follows:

$$\Theta(S(v)) = \max_{i=1}^{k}(\Theta(S(v_i)) + W(v_i) + W(v_i, v)) \qquad (1)$$

This recurrence formula can be used for computing $\theta(S(v))$ given the set of weights $\theta(S(v_i))$, $i = 1, \ldots, k$. Thus a dynamic programming algorithm can be used to find the weight of the optimal path and locate the path itself in the graph G. This path indicates the real exons (coding region) in the given genomic DNA sequence.

In the testing (prediction, respectively) phase where an unlabeled test (new, respectively) sequence S is given, GeneScout first detects the functional sites on S and then builds a directed acyclic graph G using the detected functional sites as vertices. Next, GeneScout finds the optimal path on G and outputs the vertices (functional sites) and edges on the optimal path, which displays the coding region on S.

3 Experimental Results and Discussion

In evaluating the accuracy of the proposed GeneScout system for detecting vertebrate genes, we adopted the database of human DNA sequences (570 sequences in total) originally collected by Burset and Guigo [2].

We applied the 10-way cross-validation method [9] to evaluating how well GeneScout performs when tested on sequences that are not in the training data set. The GeneScout 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) and then is tested on the sequences in the test data set. For each run, the training data set contains 90% of the total exons and the test data set contains 10% of the exons. Notice that each of the 570 sequences is used exactly once in the test data set.

Table 1 shows the results obtained in each run of cross-validation, and the average over all the ten runs. We estimate the prediction accuracy at both the nucleotide level and the exon level. At the nucleotide level, let TP_c be the number of true positives, FP_c be the number of false positives, TN_c be the number of true negatives, and FN_c be the number of false negatives. A *true positive* is a coding nucleotide that is correctly predicted as a coding nucleotide. A *false positive* is a non-coding nucleotide that is incorrectly predicted as a coding nucleotide. A *true negative* is a non-coding nucleotide that is correctly predicted as a non-coding nucleotide. A *false negative* is a coding nucleotide that is incorrectly predicted as a non-coding nucleotide. The sensitivity (S_c^n) and specificity (S_c^p) at the nucleotide level described in Table 1 are defined as follows:

$$S_c^n = \frac{TP_c}{TP_c + FN_c} \tag{2}$$

$$S_c^p = \frac{TP_c}{TP_c + FP_c} \tag{3}$$

The approximation correlation (AC) [2] is the measure that summarizes the prediction accuracy at the nucleotide level. AC ranges from -1 to 1. A value of 1 corresponds to a perfect prediction, while -1 corresponds to a prediction in which each coding nucleotide is predicted as a non-coding nucleotide, and vice versa. At the exon level, let TP_e be the number of true positives, FP_e be the number of false positives, TN_e be the number of true negatives, and FN_e be the number of false negatives. A *true positive* is an exon that is correctly predicted as an exon. A *false positive* is a non-exon that is incorrectly predicted as an exon. A *true negative* is a non-exon that is correctly predicted as a non-exon. A *false negative* is an exon that is incorrectly predicted as a non-exon. The sensitivity (S_e^n) and specificity (S_e^p) at the exon level described in Table 1 are defined as follows:

$$S_e^n = \frac{TP_e}{TP_e + FN_e} \tag{4}$$

$$S_e^p = \frac{TP_e}{TP_e + FP_e} \tag{5}$$

The result in Table 1 shows that, on average, GeneScout can correctly detect 86 percent of the coding nucleotides in the test data set. Among the predicted coding nucleotides, 79 percent are real coding nucleotides. At the exon level, GeneScout achieved a sensitivity of 51 percent and a specificity of 49 percent. This means GeneScout can detect 51 percent of exons in the test data set with both of their 5' and 3' ends being exactly correct.

	Nucleotide			*Exon*	
Run	S_c^n	S_c^p	AC	S_e^n	S_e^p
1	0.86	0.78	0.77	0.51	0.49
2	0.85	0.79	0.77	0.50	0.48
3	0.86	0.80	0.78	0.52	0.50
4	0.85	0.78	0.75	0.49	0.51
5	0.87	0.78	0.78	0.53	0.48
6	0.85	0.79	0.77	0.53	0.49
7	0.84	0.80	0.77	0.52	0.49
8	0.87	0.77	0.76	0.49	0.47
9	0.86	0.78	0.77	0.51	0.48
10	0.86	0.80	0.77	0.52	0.50
Average	0.86	0.79	0.77	0.51	0.49

Table 1. Performance evaluation of the proposed GeneScout system for gene detection.

	Nucleotide			*Exon*	
System	S_c^n	S_c^p	AC	S_e^n	S_e^p
GeneScout	0.86	0.79	0.77	0.51	0.49
VEIL	0.83	0.72	0.73	0.53	0.49
FGENEH	0.77	0.88	0.78	0.61	0.64
GeneID	0.63	0.81	0.67	0.44	0.46
GeneParser 2	0.66	0.79	0.67	0.35	0.40
GenLang	0.72	0.79	0.69	0.51	0.52
GRAIL 2	0.72	0.87	0.75	0.36	0.43
SORFIND	0.71	0.85	0.73	0.42	0.47
Xpound	0.61	0.87	0.68	0.15	0.18
GenScan	0.93	0.90	0.91	0.78	0.81

Table 2. Performance comparison between GeneScout and other systems for gene detection.

Table 2 compares GeneScout with other gene finding tools on the same 570 vertebrate genomic DNA sequences. The performance data for the other tools shown in the table are taken from the paper authored by Burset and Guigo [2] except for the VEIL system, whose data is taken from the paper authored by Henderson *et al.* [3] and GenScan, whose data is published by Burge and Karlin [1]. It can be seen from Table 2 that GeneScout beats or is comparable to these other programs except GenScan.

It can be seen from the table that GenScan is more accurate than GeneScout at both the nucleotide level and the

Prediction Results	# of coding nucleotides	% of coding nucleotides
GenScan predicted correctly	414,049	93.2%
GeneScout predicted correctly	382,712	86.1%
GeneScout and GenScan both predicted correctly	361,821	81.4%
GeneScout predicted correctly and GenScan missed	20,891	4.7%
GenScan predicted correctly and GeneScout missed	52,228	11.7%
GenScan and GeneScout both missed	9,558	2.2%

Table 3. Complementarity between GenScan and GeneScout.

exon level. However, as indicated by GenScan's inventors Burge and Karlin [1], many of the 570 sequences collected by Burset and Guigo [2] were used to train the GenScan system. This means that a portion of the test sequences were used in GenScan's training process. In contrast, GeneScout is tested on the sequences that are completely unseen in the training phase. We ran GenScan on the 570 sequences and got the same performance data as shown in Table 2. Table 3 shows the complementarity between GenScan and GeneScout. For the 570 sequences that contained 444,498 coding nucleotides totally, GenScan correctly predicted 93.2 percent of the coding nucleotides, while GeneScout correctly predicted 86.1 percent of the coding nucleotides. If both systems are used together, one can correctly predict (81.4% + 4.7% + 11.7%) = 97.8% of the total coding nucleotides. This is higher than the sensitivity of either individual system. We also found out that GenScan did not correctly predict any coding nucleotides in eight of the 570 sequences. In contrast, GeneScout did not miss any of the test sequences and , for the eight sequences GenScan missed, GeneScout correctly detected about 85% of the coding nucleotides. GeneScout runs much faster than GenScan. For example, it takes GeneScout 0.8 seconds to predict the gene structure on a 5 kb sequence. For the same sequence, GenScan needs several seconds to finish. Run time for the GeneScout program is $O(NV^2)$ where N is the length of the input sequence and V is the number of vertices on the site graph constructed during the gene predicting process. In practice, the run time grows approximately linearly with the sequence length for sequences of several kb or more. Typical run time for a X kb sequence on a Sun Sparc10 workstation is about $0.1 \times (X + 3)$ seconds.

Future work includes the incorporation of more parameters or criteria into GeneScout. One source of possible new parameters could be obtained from the analysis of potential coding regions, such as preferred exon and intron lengths, and positions of exon-intron junctions relative to the reading frame. We may also model more functional sites such as those in the upstream or downstream of a coding region. These efforts will further improve GeneScout's performance to make it more accurate for vertebrate gene detection.

References

[1] C. Burge and S. Karlin, "Prediction of complete gene structures in human genomic DNA," *J. Mol. Biol.*, 268:78–94, 1997.

[2] M. Burset and R. Guigo, "Evaluation of gene structure prediction programs," *Genomics*, 34(3):353–367, 1996.

[3] J. Henderson, S. Salzberg, and K. H. Fasman, "Finding genes in DNA with a hidden Markov model," *Journal of Computational Biology*, 4(2):127–141, 1997.

[4] M. A. Roytberg, T. V. Astakhova, and M. S. Gelfand, "Combinatorial approaches to gene recognition," *Computers Chem.*, 21(4):229–235, 1997.

[5] S. L. Salzberg, "A method for identifying splice sites and translational start sites in eukaryotic mRNA," *Computer Applications in the Biosciences*, 13(4):365–376, 1997.

[6] J. T. L. Wang, S. Rozen, B. A. Shapiro, D. Shasha, Z. Wang, and M. Yin, "New techniques for DNA sequence classification," *Journal of Computational Biology*, 6(2):209–218, 1999.

[7] J. T. L. Wang, B. A. Shapiro, and D. Shasha, editors, *Pattern Discovery in Biomolecular Data: Tools, Techniques and Applications*. Oxford University Press, New York, New York, 1999.

[8] M. M. Yin, *Knowledge Discovery and Modeling in Genomic Databases*. Ph.D. Dissertation, Department of Computer Science, New Jersey Institute of Technology, 2002.

[9] M. M. Yin and J. T. L. Wang, "Effective hidden Markov models for detecting splicing junction sites in DNA sequences," *Information Sciences*, 139(1-2):139–163, 2001.