GRAIL: A Multi-Agent Neural Network System for Gene Identification

YING XU, RICHARD J. MURAL, J. RALPH EINSTEIN, MEMBER, IEEE, MANESH B. SHAH, AND EDWARD C. UBERBACHER

Identifying genes within large regions of uncharacterized DNA is a difficult undertaking and is currently the focus of many research efforts. We describe a gene localization and modeling system, called GRAIL. GRAIL is a multiple sensor-neural network-based system. It localizes genes in anonymous DNA sequence by recognizing features related to protein-coding regions and the boundaries of coding regions, and then combines the recognized features using a neural network system. Localized coding regions are then "optimally" parsed into a gene model. Through years of extensive testing, GRAIL consistently localizes about 90% of coding portions of test genes with a false positive rate of about 10%. A number of genes for major genetic diseases have been located through the use of GRAIL, and over 1000 research laboratories worldwide use GRAIL on regular bases for localization of genes on their newly sequenced DNA.

I. INTRODUCTION

One of the most fundamental questions that can be asked about a deoxyribonucleic acid (DNA) sequence is whether or not it encodes protein. Localization of protein-coding regions in anonymous DNA sequence by pure biological means is both time-consuming and costly. A number of computational methods have been proposed and used to predict protein-coding regions and gene structures in the past few years [1]–[8]. Though the performance of these computational methods is currently imperfect, the computer-based approach may soon be the only one capable of providing analysis and annotation at a rate compatible with worldwide DNA sequencing throughput.

Computer-based gene prediction methods range from database searches for homology with known proteins to the more general and fundamental pattern recognition approaches. Though as more and more proteins are known and put into the database homology-based approaches will become increasingly useful, approximately 50% of the newly discovered genes have no detectable homologs in the protein databases. This means that pattern recognition methods will still play a crucial role in elucidating the locations and significance of genes throughout the genome.

The basis for most coding-region recognition methods is the positional and compositional biases imposed on the DNA sequence in coding regions by the genetic code and by the distribution of amino acids in proteins. Though recognition of each of these biases provides a useful indication of coding regions it is unrealistic to expect a single "perfect" indicator, given the incomplete state of our understanding of the underlying biological processes around genes. We previously proposed an approach to combine information from several coding-prediction algorithms, each designed to recognize a particular sequence property, using a neural network to provide more powerful coding recognition capabilities, and have implemented the algorithm as an e-mail server system, called the Gene Recognition and Analysis Internet Link (GRAIL) [2], [9]. While GRAIL has evolved considerably since its inception in 1991, the basic design principles are retained [10]–[12].

A D2NA can be considered as a sequence of four letters—A, C, G, T—representing four types of nucleotides. A typical DNA sequence could range from a few dozen bases to millions of bases. Portions of a DNA sequence may contain protein-coding regions, and consecutive coding regions may form a gene. A *gene* can be considered as an intermixed sequence of exons and introns, where *exons* represent coding regions and *introns* represent non-coding regions which separate exons; each intron starts with a *donor* splice junction and ends with an *acceptor* splice junction. During the process of mRNA maturation, introns are spliced out and exons are retained in the final message, which can then be translated into protein. Mathematically we can define that two exons are spliceable if the positions of their boundaries and the frames in which they are translated into protein satisfy an equation to be given later (each DNA segment has three possible

Manuscript received October 1, 1995; revised February 1, 1996. This work was supported by the Office of Health and Environmental Research, United States Department of Energy, Contract DE-AC05-84OR21400 (with Lockheed Martin Energy Systems).

Y. Xu, J. R. Einstein, M. B. Shah, and E. C. Uberbacher are with the Informatics Group, Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6364 USA (e-mail: xyn@ornl.gov).

R. J. Mural is with the Informatics Group, Biology Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6364 USA.

Publisher Item Identifier S 0018-9219(96)07178-2.

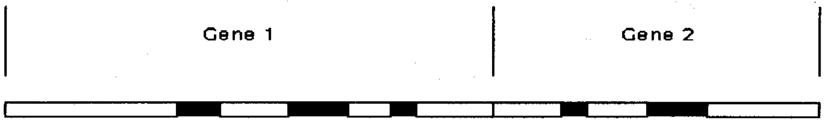


Fig. 1. A schematic of a DNA with two genes. Each solid rectangle represents an exon and a hollow rectangle represents an intron or intergenic region. The boundaries of an intron are donor and acceptor junctions.

translation frames).[1] The goal of gene recognition is to recognize exons and to group recognized exons, which are spliceable, to form a gene. Fig. 1 is a schematic of a DNA sequence showing genes, exons, introns and splice junctions.

To determine the likelihood that a DNA segment is an exon involves measuring coding potentials of the region and evaluation of the strength of boundary signals of the region, e.g., the strength of potential splice junctions bounding the region. A number of coding measures have been proposed based on the frequency of nucleotide "words" of a fixed length. Different types of DNA sequence (exons, introns, etc.) have different distributions of word occurrence [13]. In GRAIL, we have used a frame-dependent six-tuple preference model [2] and a fifth-order nonhomogeneous Markov chain model [14] to measure coding potentials. A number of measures including a five-tuple preference model, long-distance correlations between single bases, etc. have been used to measure the strength of a potential splice junction. These measures along with a number of correction factors are fed into a neural network for the final exon candidate evaluation. This neural network is trained, based on empirical data, to effectively weigh the various features in scoring the possibility of each sequence segment (candidate) being an actual exon. The use of empirical data for training allows the system to optimally utilize each feature in the presence of the others, without *a priori* assumptions about the independence of the features or their relative strengths.

Gene modeling involves selecting a set of most probable exon candidates that are spliceable to each other. While the neural network scores an exon candidate based on local information the gene modeling procedure makes the final exon prediction based on more global information, i.e., whether exon candidates are spliceable or not in addition to the neural network scores.

The GRAIL gene recognition algorithm can be outlined as the following four steps.

1) *Candidate generation.* The algorithm first generates a large candidate pool consisting of all possible exon candidates.
2) *Improbable candidate elimination.* A series of heuristic rules, each of which defines some necessary conditions a *probable* exon candidate should satisfy, are used to eliminate majority of the improbable candidates.
3) *Candidate evaluation.* The candidates which have passed the rules are then evaluated by a pretrained neural network.
4) *Gene modeling.* The algorithm selects, from the pool of scored exon candidates, a set of highest scoring candidates such that the adjacent candidates are spliceable to form a gene model.

Four types of exons are recognized based on their different boundary signals. We use the internal exons as examples to explain the basic ideas of exon recognition. Other types of exons, initial, terminal and single-exon, can be recognized similarly. An internal exon is bounded from left by an acceptor splice junction and from right by a donor splice junction.

II. Splice Junction Recognition

Evaluation of the donor and acceptor splice junctions is used in each of the first three steps of the GRAIL gene recognition algorithm. GRAIL recognizes acceptor junctions having the usual YAG (i.e., CAG or TAG) consensus, as well as the nonstandard AAG consensus, and also recognizes donor junctions containing the GT consensus.

Recognition of donor and acceptor splice junctions remains an imprecise art, due to a very significant background of nonfunctional sequences containing a splice consensus. Our recognition method is based on a number of relative frequency measures of nucleotide "words" appearing in the neighborhood of true splice sites versus false splice sites (containing minimal splice consensus) as each of those measures exhibits some discriminative power among true and false splice junctions. A large set of true and false splice sites are used to calculate these frequencies. As a result, a profile of frequencies is obtained for true and false splice sites, respectively. Then a vector of scores can be obtained for each true or false splice site based on the calculated profiles. For each of the three types of splice junctions mentioned above, a feedforward neural network is trained using the standard backpropagation learning algorithm, based on these vectors and their corresponding true or false labelings, to score a splice junction as being a true or false site. The neural network consists of seven inputs, one hidden layer of three nodes and one one output.

The seven frequency measures used in the YAG acceptor neural network recognition system are given as follows. Let $a_{-60} \cdots a_{35}$ represent the DNA segment containing a YAG consensus with $a_0 a_1 a_2 = \mathrm{YAG}$.

[1] A DNA sequence has two strands, forward and reverse complement. For each strand there are three possible translation frames. We only consider the forward strand in this discussion.

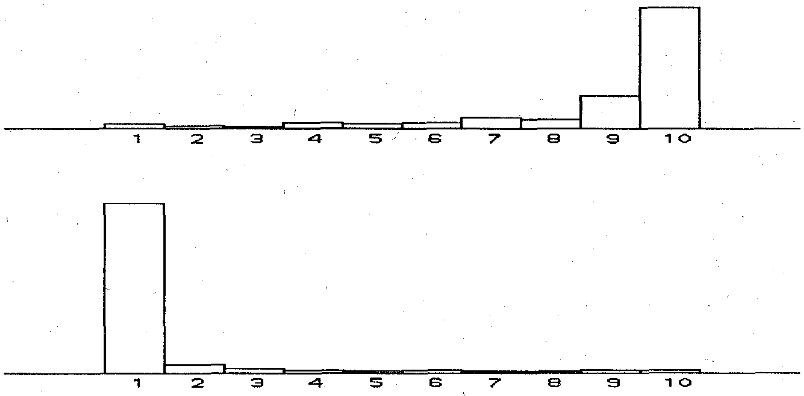


Fig. 2. YAG acceptor prediction. A total of 227 true (top) and 5127 false YAG acceptors (bottom) were tested. The height represents the percentage of acceptor candidates which were scored in the interval.

1)

$$\sum_{i=-23}^{-4} \log\left(F_t^i(a_i \cdots a_{i+4})/F_f^i(a_i \cdots a_{i+4})\right)$$

where $F_t^i()$ and $F_f^i()$ represent the positionally dependent (position i) five-tuple frequencies in true and false splice junction regions, respectively.

2)

$$\sum_{i=-27}^{0} \log(F_t(a_i)/F_f(a_i)) + \sum_{i=3}^{4} \log(F_t(a_i)/F_f(a_i))$$

where $F_t()$ and $F_f()$ are defined similarly to (1) except that they are not positionally dependent.

3)

$$\sum_{i=-27}^{0} PY(a_i)\sqrt{i+28}$$

where $PY(a_i)$ is 1 if a_i is a pyrimidine (C or T) otherwise 0.

4) The (normalized) distance between a_0 and the nearest upstream YAG.

5)

$$\sum_{i=-27}^{4} \sum_{j \geq i}^{4} \log\left(F_t^i(a_i a_j)/F_f^i(a_i a_j)\right)$$

where $F_t^i()$ and $F_f^i()$ are defined similarly to (1).

6), 7) Coding potentials in regions of $a_{-60} \cdots a_{-1}$ and $a_3 \cdots a_{35}$ measured using a frame-dependent six-tuple preference model (see Section III). This is to give an indication of a transition between noncoding and coding sequences.

Fig. 2 shows the performance statistics on an independent test set of YAG acceptor prediction system.

Acceptors with nonstandard AAG consensus are recognized using basically the same measures but with different frequency profiles. A separate neural network was trained for this type of acceptor. Similarly, donor splice junctions are recognized.

After evaluating all potential splice junctions GRAIL generates an exon candidate pool. Each exon candidate is a DNA segment with an open translation frame bounded by a pair of potential acceptor and donor junctions with scores larger than defined thresholds. Typically a few thousand of candidates are generated on a DNA sequence of 10 000 bases long. In the second step of the GRAIL gene recognition algorithm, the splice junction scores combined with several coding potential scores are used to design a number of heuristic rules. Each of these rules defines some necessary conditions that a probable exon candidate should satisfy. On average application of these rules eliminates over 90% of the original candidates with less than 3% of true exons being removed; Hence it greatly simplifies the learning process in the neural network evaluation step, and allows the neural network learning to focus on situations where the decision is more difficult.

III. Gene Recognition

A coding DNA encodes protein by encoding each amino acid of the protein into a triplet of nucleotides, also called a *codon*. Recognition of a coding region essentially involves a determination of whether the DNA sequence can be partitioned into segments of three and this sequence of nucleotide triplets may possibly correspond a "valid" protein, a sequence of amino acids. A number of models have been proposed to measure the coding potential of a DNA sequence, based on the distribution of consecutive amino acids in a protein. GRAIL uses two of those models, a frame dependent six-tuple preference model [2] and a fifth-order nonhomogeneous Markov chain model [14], as basic coding measures. The coding of amino acids in nucleotide triplets means that there are three possible ways to translate a DNA to protein, i.e., the three possible translation frames (two of which are incorrect).

A. Feature Extraction

The frame dependent six-tuple preference model consists of three preference values, $pf_0(X)$, $pf_1(X)$, $pf_2(X)$, for

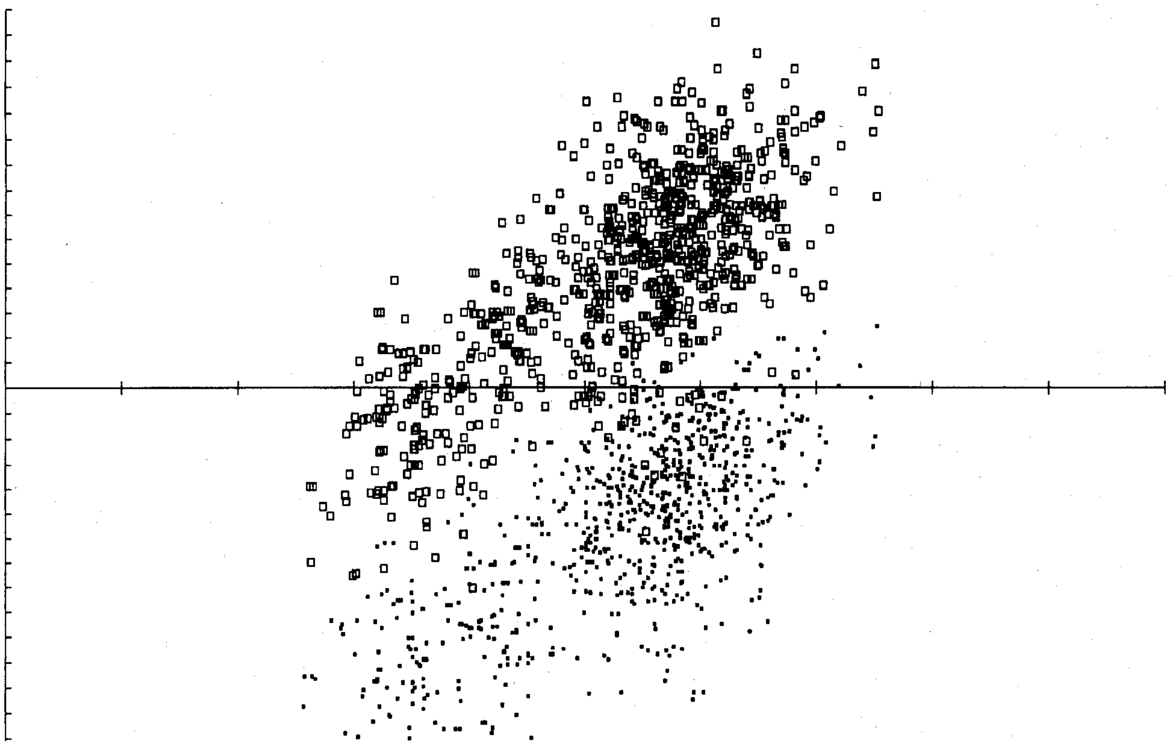


Fig. 3. The X-axis represents the G + C composition of an exon candidate and Y-axis represents the six-tuple scores measured by the frame-dependent preference model. Each tick mark on the horizontal axis represents 10% in G + C composition with 0% on the left and 100% on the right. The large squares represent the coding regions and the small dots represent the regions flanking coding regions.

each of the 4096 possible six-tuples X, which are defined as follows:

$$pf_r(X) = \log \frac{f_r(X)}{f_n(X)}, \quad \text{for } r = 0, 1, 2 \qquad (1)$$

where $f_r(X)$ is the frequency of six-tuple X appearing in a coding region and in the actual translation frame $+r$, for $r = 0, 1, 2$, and $f_n(X)$ is the frequency of X appearing in a noncoding region. In GRAIL, all the six-tuple frequencies were calculated from a large set of DNA sequences.[2]

Let $a_1 \cdots a_n$ be a DNA sequence of n bases long. The preference model calculates the coding potential of a segment $a_k \cdots a_m$ in each of the three possible translation frames, $r = 0, 1, 2$, as

$$\begin{aligned} &pf_r(a_k \cdots a_m) \\ &\quad = (pf_{(k+5-r)\mathrm{mod}3}(a_k \cdots a_{k+5}) \\ &\qquad + pf_{(k+6-r)\mathrm{mod}3}(a_{k+1} \cdots a_{k+6}) \\ &\qquad + pf_{(k+7-r)\mathrm{mod}3}(a_{k+2} \cdots a_{k+7}) + \cdots \\ &\qquad + pf_{(m-r)\mathrm{mod}3}(a_{m-5} \cdots a_m))/(m-k+1) \qquad (2) \end{aligned}$$

where mod is the modulo function.

Under the assumption that a DNA forms a fifth order nonhomogeneous Markov chain, GRAIL uses the Bayes formula to measure the coding potential of a DNA segment $a_k \cdots a_m$ in each of the three possible translation frames, $r = 0, 1, 2$, as follows (see (3) at the bottom of the page). where by the Markov chain assumption

$$\begin{aligned} &P_r(a_k \cdots a_m \mid \text{coding}) \\ &\quad = P_{(k+5-r)\mathrm{mod}3}(a_k \cdots a_{k+4} \mid \text{coding}) \\ &\qquad \times P_{(k+5-r)\mathrm{mod}3}(a_{k+5} \mid a_k \cdots a_{k+4}, \text{coding}) \\ &\qquad \times P_{(k+6-r)\mathrm{mod}3}(a_{k+6} \mid a_{k+1} \cdots a_{k+5}, \text{coding}) \\ &\qquad \cdots P_{(m-r)\mathrm{mod}3}(a_m \mid a_{m-5} \cdots a_{m-1}, \text{coding}). \qquad (4) \end{aligned}$$

and

$$\begin{aligned} &P_n(a_k \cdots a_m \mid \text{noncoding}) \\ &\quad = P_n(a_k \cdots a_{k+4} \mid \text{noncoding}) \\ &\qquad \times P_n(a_{k+5} \mid a_k \cdots a_{k+4}, \text{noncoding}) \end{aligned}$$

[2] The set contains 450 DNA sequences with 462 608 coding bases and 2 003 642 noncoding bases.

$$P_r(\text{coding} \mid a_k \cdots a_m) = \frac{P_r(a_k \cdots a_m \mid \text{coding})}{\sum_{f=0}^{2} P_f(a_k \cdots a_m \mid \text{coding}) + CP(a_k \cdots a_m \mid \text{noncoding})} \qquad (3)$$

$$\times P_n(a_{k+6} \mid a_{k+1} \cdots a_{k+5}, \text{noncoding}) \\ \cdots P_n(a_m \mid a_{m-5} \cdots a_{m-1}, \text{noncoding}). \quad (5)$$

and C is the estimate of the ratio of coding versus noncoding bases in DNA, $P_r(X \mid Y)$ and $P_n(X \mid Y)$ are the conditional probabilities of X in coding regions (in translation frame $+r$) in the presence of Y and in noncoding regions, respectively. These conditional probabilities can be estimated using the above pf_r and pf_n values.

Though not being totally independent measures, each of these two models has its own coding recognition strengths and weakness according to our test results. GRAIL uses both models as the basic coding feature extraction methods, and combines them along with other measures in the neural network coding recognition system.

Coding measures by the six-tuple preference model and the Markov chain model are also used to device heuristic rules for improbable exon candidate elimination in the second step of GRAIL gene recognition algorithm.

B. Information Fusion

In this subsection, coding measures refer to measures of coding potential using the six-tuple preference model and the Markov chain model. The goal of the exon recognition process is not just to discriminate exons from nonexonic regions but also to score the degree of correctness of an exon candidate that overlaps actual exons. For example, we consider a candidate which extends past one boundary of an exon, but otherwise overlaps it, to be partially correct. To achieve this scoring, we use coding measures in the flanking areas in addition to the coding measures of a candidate region. The rationale is that strong coding indication from the neighboring areas indicates that the candidate may be just a portion of an exon. As the candidate more closely approximates an actual exon, more noncoding elements will be included in its surrounding areas and hence the surroundings will exhibit a weaker coding score. GRAIL uses 60 bases on each side of an exon candidate as the flanking regions.

Splice junction scores are another set of measures used to help to determine the correct exon boundaries. Though false splice junction prediction may occur, in general true splice junctions have higher scores than nearby false splice junctions. By providing to the exon recognition neural network information from coding measures of an exon candidate, scores from flanking regions and the scores of its bounding splice junctions, GRAIL can fairly accurately score the degree of overlap (or correctness) of the candidate with the actual underlying exon.

One interesting observation we made indicates that shorter exons tend to have stronger splice junction sites and hence higher splice scores. Also short false exon candidates may by accident have high coding measures (because of statistical limitations). Based on these considerations, we have included the (normalized) exon candidate length as one of the inputs to the neural network recognizer. We have observed that the neural network can learn these relationships based on the training data.

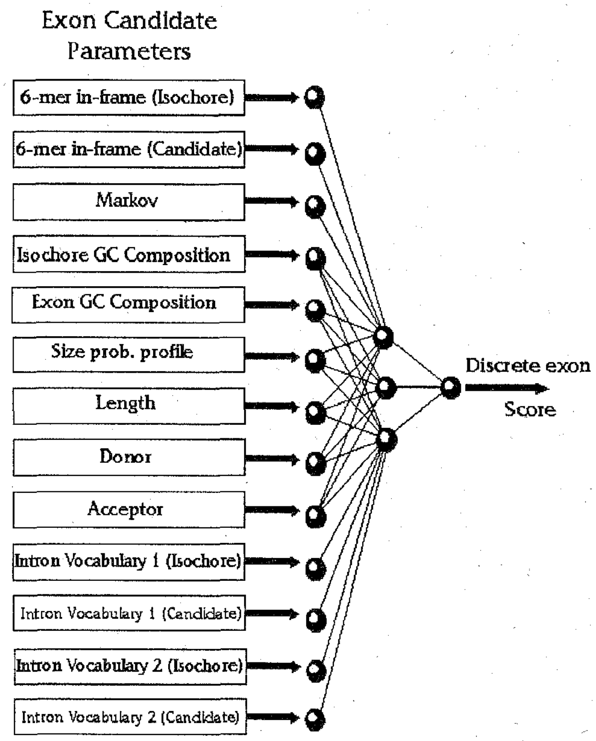

Exon Candidate Parameters
6-mer in-frame (Isochore)
6-mer in-frame (Candidate)
Markov
Isochore GC Composition
Exon GC Composition
Size prob. profile
Length
Donor
Acceptor
Intron Vocabulary 1 (Isochore)
Intron Vocabulary 1 (Candidate)
Intron Vocabulary 2 (Isochore)
Intron Vocabulary 2 (Candidate)
Discrete exon
Score

Fig. 4. A schematic of the neural network for evaluating internal protein coding exons in GRAIL.

The recognition of coding regions using the six-tuple (or in general k-tuple, for any fixed k) method is known to have strong dependence on the $\mathrm{G+C}$ (bases G and C) composition, and is more difficult in $\mathrm{G+C}$ poor domains. Our recent observation on the relationship of six-tuple coding measures and $\mathrm{G+C}$ composition supports this belief. If we estimate the frequencies of frame-dependent coding six-tuples and noncoding six-tuples in the high $\mathrm{G+C}$ domain, and use these frequencies to calculate coding measures for a set of coding regions and their 60-base flanking regions in all ranges of $\mathrm{G+C}$ composition, an unexpected pattern result is shown in Fig. 3. The coding measures for both the coding regions and their flanks are much lower in the $\mathrm{G+C}$ poor domain compared to the $\mathrm{G+C}$ rich domain. A very similar behavior is observed if the six-tuple frequencies are collected from low $\mathrm{G+C}$ DNA sequences. Interestingly, though the relative separation between coding regions and their flanking regions is similar at both ends of the $\mathrm{G+C}$ composition range, many nonexonic regions in high $\mathrm{G+C}$ isochore have higher coding measure than many coding regions in $\mathrm{G+C}$ poor regions. This certainly highlights the necessity to include the $\mathrm{G+C}$ composition as one piece of information in the neural network information fusion process. GRAIL uses the $\mathrm{G+C}$ compositions of both an exon candidate region and a 2000-base region centered around the candidate as two inputs to the neural network coding recognizer.

A schematic of the neural network used in GRAIL is shown in Fig. 4. This feedforward neural network has 13

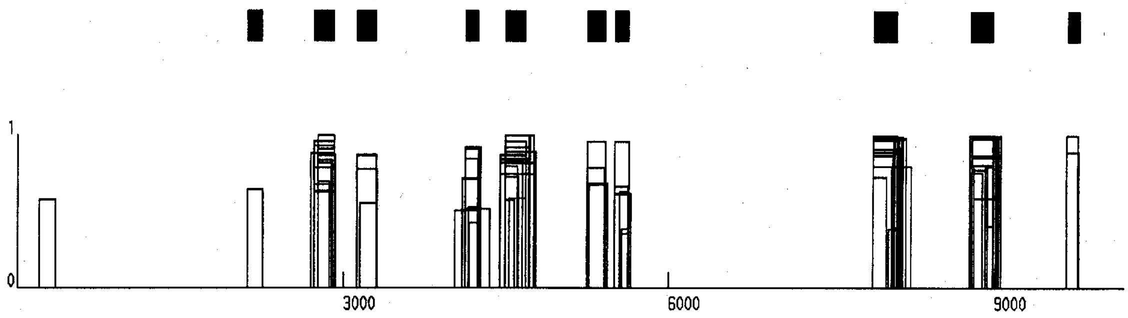


Fig. 5. Exon clusters. The X-axis is the sequence axis and Y-axis is the neural net score axis. The solid bars on the top represent the positions of the actual exons, and the hollow rectangles represent the predicted exon candidates with different boundary assumptions.

inputs, two hidden layers with seven and three nodes, respectively, and one output.

In training the neural network, our goal is to develop a network that can score the "partial correctness" of a potential exon candidate. A simple matching function $M()$ is used to represent the correspondence of a given candidate with the actual exon(s) during training.

$$M(\text{candidate}) = \frac{\sum_i m_i}{\text{length(candidate)}} \frac{\sum_i m_i}{\sum_j \text{length(exon)}_j} \quad (6)$$

where $\sum_i m_i$ is the total number of bases of the candidate that overlap some actual exons (in the same translation frame), and $\sum_j$ length (exon$_j$) is the total length of all the exons that overlap the candidate. Using such a function helps "teach" the neural network to discriminate candidates with different degrees of overlap with actual exons. The network was trained using the standard backpropagation algorithm on a training set containing about 2000 true, partially true and false exon candidates (a vector of features along with its corresponding $M()$ value for each candidate). All sequences used for training were from the Genome Sequence Database (GSDB) [15].

Fig. 5 shows a typical example of GRAIL neural network exon predictions. There could be more than one prediction for each actual exons. As can be seen, predictions for the same exon form a natural cluster, and in general the candidate that matches the actual exon exactly has the highest neural network score in the cluster.

C. Gene Modeling

The GRAIL gene modeling step takes as input the scored exon candidates generated by the coding recognition neural network and builds a single gene model in a specified region by appending a series of nonoverlapping exon candidates under the constraints that 1) the first candidate should start with a translation start codon ATG and the last candidate should end with an in-frame stop codon, TAA, TAG, or TGA, 2) adjacent candidates are spliceable (see below), 3) no in-frame stop codons can be formed when appending two adjacent exon candidates, and 4) the distance between two adjacent candidates has to be larger than the minimum intron size (60 bases are used in GRAIL).

Two candidates $a_j \cdots a_k$ and $a_m \cdots a_n$, $k < m$, with the preferred translation frames r_1 and r_2, respectively, are said to be *spliceable* if

$$r_2 = (m - k - 1 + r1) \text{mod} 3 \quad (7)$$

where the *preferred* translation frame refers to the frame exhibiting the highest coding potential.

GRAIL builds a gene model with the highest total neural network scores using a fast dynamic programming algorithm [3]. The basic idea of this algorithm is that it scans exon candidates in the increasing order of the indices of their boundaries, and builds an optimal (highest scored) partial gene model that ends with each exon candidate by extending the previous optimal partial gene models to include the current candidate. When expending an optimal partial gene model, the algorithm checks if the constraints (1)–(4) are satisfied. A globally optimal solution can be obtained when the algorithm finishes scanning all the candidates.

In addition to finding a set of highest scored candidates that forms a gene model, the algorithm also helps to eliminate false exon candidates as a result of enforcing the spliceability. Fig. 6 shows two examples of GRAIL gene prediction results.

D. Sequencing Error Handling

Performance of the GRAIL gene recognition algorithm depends on the correctness of the input DNA sequence. Insertion or deletion of DNA bases (or simply *indels*) may change the performance significantly as indels may disrupt the translation frames,[3] and GRAIL basic coding recognition methods are highly translation frame-dependent. We have developed an algorithm to detect and "correct" indels appearing in DNA coding regions [16]. While indel detection can be achieved using the information present in the DNA sequence, indel correction can be expected to be only imperfectly achieved since information may have been lost (for example, bases that have been deleted) when an indel error occurred. Hence the basic goal in indel correction is to recover a consistent translation frame within

[3] Here the translation frame of a coding region refers to the preferred translation frame of the region.

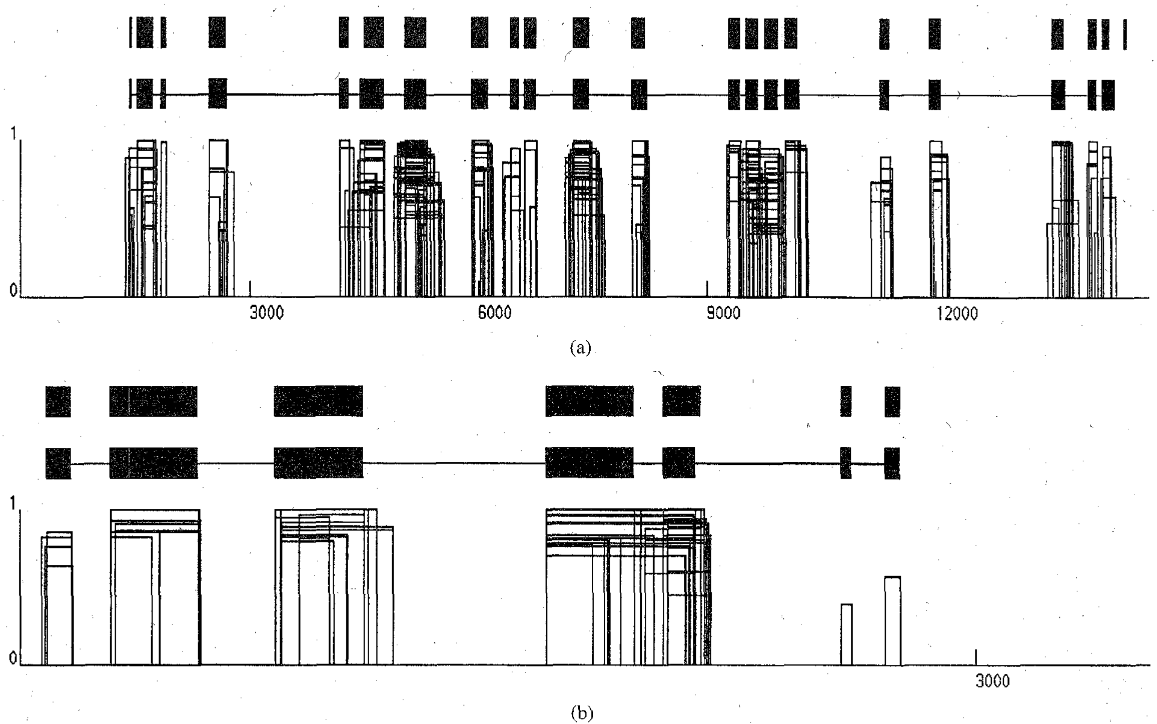


Fig. 6. GRAIL gene predictions. The X-axis is the sequence axis and the Y-axis is the neural net score axis. The solid bars on the top represent the actual exons. The solid bars on the second row represent predicted exons and gene model. Each hollow rectangle denotes an exon candidate: (a) sequence HUMATPGG and (b) sequence HUMMHB27D.

a (presumed) coding region. The key to the indel detection and correction algorithm is to localize the transition points of translation frames within each (presumed) coding region.

In GRAIL, a coding region is recognized along with its translation frame. The indel detection algorithm first finds all the transition points of the (preferred) translation frames along a given DNA sequence by discovering changes in the (preferred) translation frames, and then evaluates the coding potential on both sides of each transition point. If a transition point occurs between regions of high coding potential, the point is determined to be an indel. These indels are then corrected by adding a base "C" or deleting a base at each transition point to make a consistent translation frame (within each coding region). "C" is used to avoid the potential for creating stop codons.

To find the transition points, the algorithm divides a DNA sequence into segments in such a way that two adjacent segments have different translation frames, each segment has at least a minimum length (to prevent short range fluctuations), and the total coding potential along the translation frames are maximized. In this algorithm, coding potential is measured using the six-tuple frame-dependent preference model.

More specifically, we want to partition a given DNA D into segments $D_1, D_2, \cdots, D_m$ such that the following objective function is maximized

$$\sum_{i=1}^{m} pf_{r(i)}(D_i) \tag{8}$$

under the constraints that each D_i has as least K bases and there is no i, $1 \leq i < m$, with $r(i) = r(i+1)$, where $r(i)$ denotes the translation frame of segment D_i. Recall from Section III the definition of $pf_r(D_i)$.

This optimization problem is solved by a dynamic programming algorithm with $K = 30$. The potential for each transition point to be within a coding region is evaluated using the fifth order nonhomogeneous Markov chain model (see Section III) on 30 base regions before and after the transition point.

The indel detection and correction algorithm has greatly improved the prediction results of the GRAIL gene recognition system in the presence of sequencing errors. On a test set containing 202 DNA sequences with 1% randomly implanted indels, this algorithm has helped improve the GRAIL coding recognition from a true positive rate of 60%–81% with only 1% increase in false positive rate.

IV. Summary

The performance of multi-agent systems such as GRAIL depends critically on how the information from different agents is combined. Over a dozen of exon indicators and correction factors are used in the GRAIL gene recognition process. The relationship between these quantities and the presence of exons is complicated, incomplete and clearly nonlinear. To develop an effective mechanism to map these quantities, some of which may not be independent, to exon

Table 1 GRAIL Gene Recognition Performance

	DNA	Predictions				Gene Modeling			
	# Exons	TP	%	FP	%	TP	%	FP	%
Short	229	171	74.7	39	18.6	167	73.0	16	8.7
Long	600	575	95.8	30	4.9	564	94.0	13	2.3
Total	829	746	90.0	69	8.5	731	88.0	29	3.8
	# Bases								
Total	134814	122885	91.2	13048	9.6	122404	90.8	5972	4.7

TP and FT are the true and false positives, respectively. Short: 100 bases or less; long: otherwise. In the category of Prediction, the highest scoring candidate is selected from each cluster (See Section III-B) as the respresentative of the cluster.

and nonexonic regions is the main goal of our research. By training neural networks with hidden layers on empirical data, GRAIL seems to have captured some of the most essential part of this relationship based on its successful applications to gene recognition by molecular biologists worldwide over the past four years.

By using a neural network as the basic means to combine information from different sources, we have also obtained a flexible framework to include new information in our gene recognition system as deeper understanding and hence more information about genes are gained. Some recent work [17] has applied neural networks to combine information from recognized gene features and data base search information in a gene recognition algorithm.

Since its service being made available to public through an e-mail server in 1991 and also through a GUI-based client/server system in 1993, GRAIL has become one of the major tools used by molecular biologists. Over 1000 research labs worldwide use this system to intelligently select and design biological experiments where they are most needed and most useful. Among these applications, GRAIL has helped to locate a number of genes for major genetic diseases [18], [19].

Through the years, GRAIL has been extensively tested on its performance of gene recognition and modeling. On a recent test on 110 Human and Mouse DNA sequences consisting of 829 exons, 134 814 coding bases and 1 257 631 noncoding bases, GRAIL recognizes over 90% coding bases with about a 5% false positive rate as summarized in Table 1.

The high sensitivity and specificity of the GRAIL gene recognition and modeling system and its availability through the e-mail server and client/server system greatly increases the viability of the gene hunting strategies based on genomic sequencing and informatics analysis. We have shown that the detailed structure of genes can be characterized with considerable fidelity, and expect that, in terms of providing relatively complete information about uncharacterized regions of the genome, this overall technology will fair well when compared to experimental alternatives such as exon trapping and DNA-based methods. Computational characterization of genes in their genomic sequence context will increasingly provide an important framework for understanding aspects of gene regulation and larger questions related to the functional organization of the genome.

References

[1] J. W. Fickett, "Recognition of protein coding regions in DNA sequences," *Nucleic Acids Res.*, vol. 10, pp. 5303–5318, 1982.

[2] E. C. Uberbacher and R. J. Mural, "Locating protein-coding regions in human DNA sequences by a multiple sensors-neural network approach," in *Proc. Natl. Acad. Sci.*, 1991, vol. 88, pp. 11261–11265.

[3] Y. Xu, R. Mural, and E. C. Uberbacher, "Constructing gene models from a set of accurately-predicted exons: An application of dynamic programming," *Computer Applicat. in Biosci.*, vol. 10, pp. 613–623, 1994.

[4] M. S. Gelfand, "Computer prediction of exon-intron structure of mammalian pre-mRNA's," *Nucleic Acids Res.*, vol. 18, pp. 5865–5869, 1990.

[5] R. Guigo, S. Knudsen, N. Drake, and T. Smith, "Prediction of gene structure," *J. Molec. Biol.*, vol. 226, pp. 141–157, 1992.

[6] G. B. Hutchinson and M. R. Hayden, "The prediction of exons through an analysis of spliceable open reading frames," *Nucleic Acids Res.*, vol. 20, pp. 3453–3462, 1992.

[7] E. E. Snyder and G. D. Stormo, "Identification of coding regions in genomic DNA sequences: An application of dynamic programming and neural networks," *Nucleic Acids Res.*, Vol. 21, pp. 607–613, 1993.

[8] S. Dong and D. B. Searls, "Gene structure prediction by linguistic methods," *Genomics*, vol. 23, pp. 540–551, 1994.

[9] R. J. Mural *et al.*, "An artificial intelligence approach to DNA sequence feature recognition," *Trends in Biotechnol.*, vol. 10, pp. 66–69, 1992.

[10] E. C. Uberbacher, J. R. Einstein, X. Guan, and R. J. Mural, "Gene recognition and assembly in the GRAIL system: Progress and challenges," in *Proc. 2nd Int. Conf. on Bioinformatics, Supercomputing and Complex Genome Anal.*, H. A. Lim, J. W. Fickett, C. R. Cantor, and R. J. Robbins, Eds. New York: World Scientific, 1993, pp. 465–476.

[11] Y. Xu *et al.*, "An improved system for exon recognition and gene modeling in human DNA sequences," in *Proc. 2nd Int. Conf. on Intell. Syst. for Molecular Biol.*, R. Altman *et al.*, Eds. New York: AAAI, 1994, pp. 376–384.

[12] E. C. Uberbacher, Y. Xu, and R. J. Mural, "Discovering and understanding genes in human DNA sequence using GRAIL," *Methods in Enzymology*, vol. 266, pp. 259–281, 1996.

[13] J. M. Claverie, I. Sauvaget, and L. Bougueleret, "k-tuple frequency analysis: From intron/exon discrimination to T-cell epitope mapping," *Methods in Enzymology*, vol. 183, pp. 237–252, 1990.

[14] M. Borodovsky, Y. Sprizhitskii, E. Golovanov, and A. Aleksandov, "Statistical patterns in the primary structures of functional regions in E. Coli.," *Molekulyainaya Biologiya*, vol. 20, pp. 1390–1398, 1986.

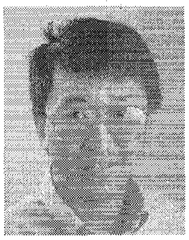
[15] H. S. Bilofsky and C. Burks, "The GenBank genetic sequence data bank," *Nucleic Acids Res.*, vol. 16, pp. 1861–1864, 1988.

[16] Y. Xu, R. Mural, and E. C. Uberbacher, "Correcting sequencing errors in DNA coding regions using dynamic programming," *Computer Applicat. in Biosci.*, vol. 11, pp. 117–124, 1995.

[17] E. E. Snyder and G. D. Stormo, "Identification of protein coding regions in genomic DNA," *J. Molec. Biol.*, vol. 248, pp. 1–18, 1995.

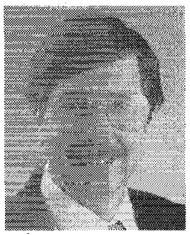
[18] R. Legouis *et al.*, "The candidate gene for the X-linked Kallmann syndrome encodes a protein related to adhesion molecules," *Cell*, vol. 67, no. 10, pp. 423–435, 1991.

[19] J. Mosser *et al.*, "Putative X-linked adrenoleukodystrophy gene shares unexpected homoloy with ABC transporters," *Nature*, vol. 361, pp. 726–730.



Ying Xu recieved the Ph.D. degree in computer science from the University of Colorado at Boulder in 1991.

He joined the Informatics Group of the Mathematics and Computer Science Division, Oak Ridge National Laboratory, Oak Ridge, TN, in 1993. From 1991 to 1993 he was a Visiting Assistant Professor at the Colorado School of Mines, Golden. His research interests include computational biology, image processing, and pattern recognition, applied combinatorial optimization and algorithms, and artificial intelligence.



Richard J. Mural received the B.S. degree in zoology from the University of Michigan, Ann Arbor, in 1969, and the Ph.D. degree in zoology from the University of Georgia, Atlanta, in 1978. He was a postdoctoral fellow at the Frederick Cancer Research Facility from 1978 to 1980.

He was a Staff Scientist at Frederick Cancer Research Facility from 1980 to 1985. In 1985 he joined the Protein Engineering Program in the Biology Division of Oak Ridge National Laboratory, where he has recently been involved in two major genome analysis efforts at ORNL. One such program develops computer methods (especially artificial intelligence) to analyze DNA sequence data.

Dr. Mural was a co-recipient of a 1992 R&D 100 Award from *R&D Magazine* for his work on GRAIL.

J. Ralph Einstein (Member, IEEE) received the B.S. degree in physics from Yale University, New Haven, CT, in 1944, and the Ph.D. degree in biophysical chemistry from Harvard University, Cambridge, MA, in 1959.

He is a Research Staff Member at Oak Ridge National Laboratory. His research interests include the development of computational methods for the detection of genes in DNA sequences, and the prediction of 3-D protein structures from their amino acid sequences.

Dr. Einstein is a member of the IEEE Computer Society and Sigma Xi.

Manesh B. Shah received the B.Tech. degree in electrical engineering from the Institute of Technology, Banaras Hindu University, India, in 1980 and the M.S. degree in computer science from the University of Tennessee, Knoxville, in 1988.

From 1988 to 1992 he was a Senior Computer Programmer in the Biology Division at Oak Ridge National Laboratory, where he worked on 3-D graphics, especially tomographic image reconstruction and analysis and modeling of chromosome structures from these images. in 1992 he joined the Informatics Group of the Mathematics and Computer Science Division, Oak Ridge National Laboratory, as a Research Associate. He has since been researching relational database design and access tools for biological databases; AI-based systems for recognizing biologically significant features in DNA sequences, and design and implementation of XGRAIL. He is also interested in distributed computing, GUI's, and application of pattern recognition to computational biology.

Dr. Shah is a member of the Association for Computing Machinery.

Edward C. Uberbacher received the B.A. degree from Johns Hopkins University, Baltimore, MD, in 1974, and the Ph.D. degree in chemistry from the University of Pennsylvania, Philadelphia, in 1979. In 1980 he began postdoctoral work in biophysics at the University of Pennsylvania.

He then joined the Biology Division of Oak Ridge National Laboratory—University of Tennessee Graduate School of Biomedical Sciences, where he became Research Assistant Professor in 1987. He later became an Investigator in the Oak Ridge National Laboratory, Biology Division, where his work focused on X-ray and neutron crystallography and scattering, and other biophysical methods. In 1991 he became the Informatics Group Leader at the ORNL Engineering Physics and Mathematics Division to develop AI and high-performance computing methods for genomic DNA sequence analysis.

Dr. Uberbacher received a R&D 100 Award from *R&D Magazine* for his work on the development of the GRAIL sequence analysis system.