Gene Identification and Analysis: An Application of Neural Network-based Information Fusion

Sherri Matis, Ying Xu, Manesh B. Shah, Richard J. Mural, and Edward C. Uberbacher

Informatics Group
Computer Science and Mathematics Division and Biology Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6364

"The submitted manuscript has been authorized by a contractor of the U.S. Government under contract No. DE-AC05-96OR22464. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes."


*Research was supported by the Office of Health and Environmental Research, U.S. Department of Energy under contract No. DE-AC05-96OR22464 with Lockheed Martin Energy Research Corporation.
DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
Gene Identification and Analysis: An Application of Neural Network-based Information Fusion

Sherri Matis, Ying Xu, Manesh B. Shah, Richard J. Mural, J. Ralph Einstein, and Edward C. Uberbacher
Informatics Group
Computer Science and Mathematics Division and Biology Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6364
matisgrail.epm.ornl.gov

Abstract

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 gene features related to protein-coding regions and the boundaries of coding regions, e.g. splice sites, and then combines the recognized features using a neural network system. Localized coding regions are then “optimally” parsed into a gene model. RNA polymerase II promoters can also be predicted. 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.

1 Introduction

One of the most fundamental questions that can be asked about a 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 [7, 21, 24, 8, 9, 10, 19, 6]. 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 the world-wide DNA sequencing effort.

Although a number of strategies for computer gene prediction exist, the most fundamental and general methods employ pattern recognition. These methods exploit the positional and composition bias in coding vs non coding DNA, and 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 GRAIL (Gene Recognition and Analysis Internet Link) [21, 16]. While GRAIL has evolved considerably since its inception in 1991 the basic design principles are retained [22, 25, 27].

DNA is composed of 2 strands of 4 nucleotide bases and it is represented as a sequence of four letters, A, C, G, T. The chromosomes consist of long strands of DNA which are organized in discreet transcription units or genes. The entire human genome is composed of 23 chromosomes pairs composed of approximately 3 billion basepairs. The purpose of a gene is to code for a specific protein, and at the DNA level, the gene is made up of protein coding regions or exons interspersed with non-coding DNA regions or introns, and is preceded by the promoter which contains the sequence signals to initiate transcription. When the gene is expressed, the semi-conserved signal sequences in the promoter bind proteins which transcribe the
Figure 1: A schematic of a gene. The promoter region precedes the gene and is represented by an open rectangle. The sequence elements are within the promoter region. Each solid rectangle represents an exon and the diagonally striped rectangles represent introns or inter-genic regions. The boundaries of an intron are donor and acceptor junctions.

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 pre-trained 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 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.

Promoters are recognized by combining statistical information for the sequence signals, the TATA box, GC signal, CAAT box, the cap and translation initiation sites, found within or near to promoters, with distance information for signal pairs. All of these scores are input to a back propagation neural network. The output of the neural network is then evaluated using a set of biologically relevant rules which incorporate the gene model, predicted by GRAIL, with known promoter-exon constraints.

2 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 non-standard 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 non-functional 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 feed-forward neural network is trained using the standard back-propagation 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 3 nodes and one one output.

The seven frequency measures used in the YAG acceptor neural network recognition system are given as follows. Let \( a_{-90} \ldots a_{35} \) represent the DNA segment containing a YAG consensus with \( a_{0}a_{1}a_{2} \equiv \text{YAG} \).

\[
(1) \sum_{i=-23}^{-1} \log(F_{i}^{t}(a_{i} \ldots a_{i+1})/F_{i}^{f}(a_{i} \ldots a_{i+1})),
\]

where \( F_{i}^{t}() \) and \( F_{f}^{t}() \) represent the positionally dependent (position \( i \)) 5-tuple frequencies in true and false splice junction regions, respectively.

\[
(2) \sum_{i=-27}^{-1} \log(F_{i}^{t}(a_{i})/F_{f}(a_{i})) + \sum_{i=3}^{4} \log(F_{i}^{t}(a_{i})/F_{f}(a_{i})),
\]

where \( F_{i}^{t}() \) and \( F_{f}^{t}() \) 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.

\[
(5) \sum_{i=-27}^{4} \sum_{j=2}^{1} \log(F_{i}^{t}(a_{i}a_{j})/F_{f}^{t}(a_{i}a_{j})),
\]

where \( F_{i}^{t}() \) and \( F_{f}^{t}() \) are defined similarly to (1).

(6, 7) Coding potentials in regions of \( a_{-90} \ldots a_{-1} \) and \( a_{3} \ldots a_{35} \) measured using a frame-dependent 6-tuple preference model (see next section). This is to give an indication of a transition between noncoding and coding sequences.

Acceptors with non-standard AAG consensus are recognized using basically the same measures but with different frequency profiles. A separate neural network was trained based 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 10000 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.

3 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 6-tuple preference model [21] and a 5th order non-homogeneous Markov chain model [2], 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).

3.1 Feature extraction

The frame dependent 6-tuple preference model consists of three preference values, \( pf_0(X) \), \( pf_1(X) \), \( pf_2(X) \), for each of the 4096 possible 6-tuples \( X \), which are defined as follows:

\[
pf_r(X) = \frac{f_r(X)}{f_0(X)}, \quad r = 0, 1, 2, \]

where \( f_r(X) \) is the frequency of 6-tuple \( X \) appearing in a coding region and in the actual translation frame +\( r \). for \( r = 0, 1, 2 \), and \( f_0(X) \) is the frequency of \( X \) appearing in a noncoding region. In GRAIL, all the 6-tuple frequencies were calculated from a large set of DNA sequences\(^1\).

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

\[
pf_r(a_k \ldots a_m) = pf_{(k+5-r)\text{mod}3}(a_k \ldots a_{k+5}) + pf_{(k+6-r)\text{mod}3}(a_{k+1} \ldots a_{k+6}) + pf_{(k+7-r)\text{mod}3}(a_{k+2} \ldots a_{k+7}) + \cdots + pf_{(m-r)\text{mod}3}(a_{m-5} \ldots a_m)
\]

where \( \text{mod} \) is the modulo function.

Under the assumption that a DNA forms a 5\( ^{th} \) order non-homogeneous Markov chain, GRAIL uses the Bayes formula to measure the coding potential of a DNA segment \( a_k \ldots a_m \) in each of the three possible translation frames, \( r = 0, 1, 2 \), as follows.

\[
P_r(a_k \ldots a_m \mid \text{coding}) = \frac{P_r(a_k \ldots a_m \mid \text{coding})}{\sum_{j=0}^{2} P_j(a_k \ldots a_m \mid \text{coding}) + C P(a_k \ldots a_m \mid \text{noncoding})},
\]

where by the Markov chain assumption,

\[
P_r(a_k \ldots a_m \mid \text{coding}) = pf_{(k+5-r)\text{mod}3}(a_k \ldots a_{k+4}) pf_{(k+6-r)\text{mod}3}(a_{k+5} \ldots a_{k+1}, \text{coding}) \times P_{(k+6-r)\text{mod}3}(a_{k+1} \ldots a_{k+5}, \text{coding}) \ldots
\]

\[
P_{(m-r)\text{mod}3}(a_{m-5} \ldots a_m, \text{coding})
\]

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 weaknesses 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 6-tuple preference model and the Markov chain model are also used to devise heuristic rules for improbable exon candidate elimination in the second step of GRAIL gene recognition algorithm.

3.2 Information fusion

In this subsection, coding measures refer to measures of coding potential using the 6-tuple preference model and the Markov chain model. The goal of the exon recognition process is not just to discriminate exons from non-exonic 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 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.

The recognition of coding regions using the 6-tuple (or in general k-tuple, for any fixed k) method is known to have strong dependence on the G+C (bases G and C) composition, and is more difficult in G+C poor domains. Our recent observation on the relationship of 6-tuple coding measures and G+C composition supports this belief. If we estimate the frequencies of frame-dependent coding 6-tuples and noncoding 6-tuples in the high 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 G+C composition, an unexpected pattern result is shown in Figure 2. The coding measures for both the coding regions and their flanks are much lower in the G+C poor domain compared to the G+C rich domain. A very similar behavior is observed if the 6-tuple frequencies are collected from low G+C DNA sequences. Interestingly, though the relative separation between coding regions and their flanking regions is similar at both ends of the G+C composition range, many non-exonic regions in high G+C isochores have higher coding measure than many coding regions in G+C poor regions. This certainly highlights the necessity to include the G+C composition as one piece of information in the neural network information fussion process. GRAIL uses the 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 Figure 3. This feed-forward neural network has 13 inputs, two hidden layers with 7 and 3 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.

$M(\text{candidate}) = \frac{\sum_i m_i}{\text{length(candidate)}} \frac{\sum_i m_i}{\sum_i \text{length(exon}_i)},$

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 $\sum_i \text{length(exon}_i)$ 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 back-propagation 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) [1].

3.3 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 non-overlapping 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 $a_j...a_k$ and $a_m...a_n$, $k < m$, with the preferred translation frames $r_1$ and $r_2$, respectively, are said to be spliceable if

$$r_2 = (m - k - 1 + r_1) \mod 3,$$

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 [24]. 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.

Figure 4 shows two examples of GRAIL gene prediction results.

**4 Promoter Recognition**

Prediction of RNA polymerase II promoters has been attempted with only marginal success. The lack of a good method for promoter recognition reflects the variable and complex nature of the promoter region. One can simplistically define a promoter as a collection of protein binding sites, each for a specific protein, found at a constrained distance with respect to one
another and to the initial or starting exon. Although there are hundreds of documented protein binding sites found within different promoters, many of these sites bind protein only under certain conditions, and are found in only a small minority of promoters. The situation is further complicated by the fact that only a few binding sites are well characterized experimentally and even these can vary in sequence. Finally, because experimental analysis of promoter regions is laborious and time consuming, only a modest number of promoters have been studied.

In order to efficiently predict promoters, a method was developed which uses a neural network to fuse weak information with some strong constraints. Although this work is described in greater detail in [13, 14], we offer a brief description.

Promoters are detected on the basis of the presence of 5 of the most studied binding sites: the TATA box, TATAA: the GC signal, GGGCGG; the CAAT box, CAAT; the cap or transcription initiation site: the translation initiation site, ATG; and their distances with respect to one another[3, 5]. A candidate promoter region is defined as a region of 600 bases with an ATAT position 465-475 and is based on the format of experimentally determined promoters present in EPD [3, 13, 14].

A vector was generated for a 13 input standard back propagation neural network. The vector was composed of scores for each of the five elements, calculated using a statistical frequency matrix constructed from positional nucleotide frequencies known for functional elements and flanking regions of 10 bases. In order to use the strong distance constraint for the TATA-cap pair, the cap site was positioned so the maximum cap frequency matrix score, within 25-35 bases to the right of the TATA-box, was used for the input vector. Distance scores for the relative positions of the TATA-GC, TATA-CAAT, TATA-cap, cap-ATG, and GC-CAAT pairs were calculated using distance histograms. In some cases, multiple CAAT and or GGGCGG elements were present so the number of each of these elements was included in the input vector as well as the GC content for the 600 base region.

The output from the neural network was further refined and validated using a set of rules. These rules utilize gene model information from GRAIL and known biological constraints about promoter position relative to the gene. The rules help to eliminate many false positive predictions made by the neural network. These rules are described in detail in [13, 14].

Figure 5: A schematic of the neural network used for promoter prediction. The neural network consisted of 13 input nodes 2 hidden layers of 3 and 2 hidden nodes and a single output. The network was trained and tested using a true set composed of functional promoters from EPD, the Eucaryotic Promoter Database [3] and a false set composed of 600 base regions, containing a minimal TATAA and found within genes.

5 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 non-linear. To develop an effective mechanism to map these quantities, some of which may not be independent, to exon and non-exonic 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 parts 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 [20] has applied neural networks to combine information from recognized gene features and database search information in a gene recognition algorithm.

Since its service being made available to public through an email 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 [11, 15].

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, 134814 coding bases and 1257631 non-coding bases, GRAIL recognizes over 90% coding bases with about a 5% false positive rate as summarized in the following table.

### Table I: GRAIL gene recognition performance.

<table>
<thead>
<tr>
<th>DNA</th>
<th>Predictions</th>
<th>Gene Modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Exons</td>
<td>TP</td>
</tr>
<tr>
<td>Short</td>
<td>229</td>
<td>171</td>
</tr>
<tr>
<td>Long</td>
<td>600</td>
<td>375</td>
</tr>
<tr>
<td>Total</td>
<td>829</td>
<td>746</td>
</tr>
<tr>
<td>Total</td>
<td>134814</td>
<td>122885</td>
</tr>
</tbody>
</table>

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 3.2) as the representative of the cluster.

For the largest group of RNA polymerase II promoters, 66% were detected with 1 false positive for every 23,407 bases of DNA. Although these results are not as impressive as for gene modeling, the false positive rate is much lower than other systems, which find from 1 in 1500 [12] to 1 in 6000 [17] false positives per base while finding a comparable number of correct promoters [18] (58-70% for the other systems).

The high sensitivity and specificity of the GRAIL gene recognition and modeling system and its availability through 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 cDNA 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.

### Acknowledgment

This research was supported by the Office of Health and Environmental Research, United States Department of Energy, under contract DE-AC05-84OR21400 with Lockheed Martin Energy Systems, Inc.
References


DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.