

Integrating Database Homology in a Probabilistic Gene Structure Model

David Kulp, David Haussler

*Baskin Center for Computer Engineering and Computer Science
University of California, Santa Cruz CA, 95064, USA
dkulp@cse.ucsc.edu, haussler@cse.ucsc.edu*

Martin G. Reese, Frank H. Eeckman

*Genome Informatics Group
Lawrence Berkeley National Laboratory
1 Cyclotron Road, Berkeley, CA 94720
mgreese@lbl.gov, fheeckman@lbl.gov*

We present an improved stochastic model of genes in DNA, and describe a method for integrating database homology into the probabilistic framework. A generalized hidden Markov model (GHMM) describes the grammar of a legal parse of a DNA sequence. Probabilities are estimated for gene features by using dynamic programming to combine information from multiple sensors. We show how matches to homologous sequences from a database can be integrated into the probability estimation by interpreting the likelihood of a sequence in terms of the bit-cost to encode a sequence given a homology match. We also demonstrate how homology matches in protein databases can be exploited to help identify splice sites. Our experiments show significant improvements in the sensitivity and specificity of gene structure identification when these new features are added to our gene-finding system, Genie. Experimental results in tests using a standard set of annotated genes showed that Genie identified 95% of coding nucleotides correctly with a specificity of 91%, and 77% of exons were identified exactly.

1 Introduction

Our research is born out of the need for complete, automated, genefinding systems for long unannotated sequences, now being produced at a very high volume. Such genefinders attempt to synthesize statistical and database information. Our genefinding system, called Genie, is a generalized hidden Markov model (GHMM) – a hidden Markov model¹ whose states are arbitrary sub-models emitting variable length sequences. The basic methodology used in the Genie system is described more completely in previous work by Kulp, *et al.*² This paper expands on this work and introduces three new additions: an improved probability estimation scheme that combines evidence from multiple sources, a general sensor class to interpret database matches probabilistically, and a simple method to aid in splice site identification by using protein database homology. Our system is similar in design to GeneParser,³ but is based on a rigorous probabilistic framework.

We first present a brief overview of the general methodology, including the use

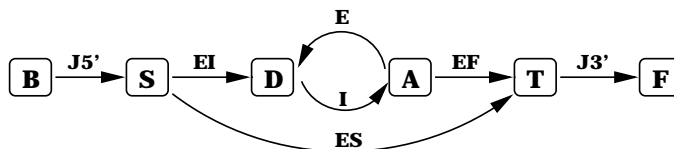


Figure 1: A simple GHMM for a sequence containing a multiple exon gene. The arcs represent multi-symbol states and nodes represent transitions between states. The state labels are J5' : 5' UTR, EI : Initial Exon, E : Exon, I : Intron, E : Internal Exon, EF: Final Exon, ES : Single Exon, and J3' : 3' UTR. The node labels are B : Begin, S : Start Translation, D : Donor, A : Acceptor, T : Stop Translation, F : Final. The arrows imply a generation of bases from 5' to 3'.

of a GHMM to describe the gene structure syntax, signal sensors to identify transitions, and content sensors to “score” candidate regions. A detailed discussion follows describing a simplified method for combining evidence from multiple sources. One such source of evidence is a sequence database. We explain how a database match is interpreted, according to information theory, in terms of encoding cost. Last, we describe an enhancement to the parsing that aids in splice site identification by limiting splice site candidates to those sites that correspond to adjacent hits in a protein database.

2 Methodology

2.1 Basic System Framework

A generalized hidden Markov model is an enhancement to the standard hidden Markov model popularly used in time-sequence pattern recognition such as speech and computation biology. (See, among others, the tutorial from Rabiner and Juang⁴ and the introduction to HMMs in biosequence analysis by Krogh, *et al.*¹) In a standard hidden Markov model, viewed as a generator, each state emits a single symbol. A GHMM describes a more general model in which each state can emit one or more symbols according to an arbitrary distribution. Each state represents an independent sub-model which may, itself, be a hidden Markov model or any statistical model. Figure 1^a shows a simple model of the eukaryotic gene structure. The states in the model are shown on the arcs of the graph. Nodes in the graph represent transitions between states. Each state corresponds to a sub-model of an abstract gene feature such as an “Internal Exon” (E) or an “Intron” (I). For any candidate sequence, x , and state, q , we can estimate a likelihood of the sequence given the sub-model corresponding to that state’s feature, which we denote $P(x|q)$. The method for likelihood estimation is described later in section 2.4.

For any state q , the node that the arc for state q leads to is denoted $node(q)$. Once in this node, a next state is chosen from among the outgoing arcs from this

^aFigures 1 and 2 used by permission. Copyright 1996, AAAI Press.

node. The probability of choosing the next state r is denoted $P(r|node(q))$. For example, in figure 1, the state I (Intron) leads to the node A (Acceptor). After the acceptor can come either the internal exon state (E) or the final exon (EF). The former is chosen with probability $P(E|A)$ and the latter with probability $P(EF|A)$ where $P(E|A)+P(EF|A) = 1$. These parameters are determined from training data.

Given a complete DNA sequence, we assert that the sequence was derived according to some combination of states, following the syntactic framework of our graph, but the actual ordered set of states is hidden. The predicted gene structure is simply the ordered set of states – the path through the graph – determined to be most likely. We formalize this concept by defining a parse and sequence likelihood. A parse is composed of an ordered set of states, $\{q_1, q_2, \dots, q_k\}$, and a corresponding ordered set of subsequences, $\{x_1, x_2, \dots, x_k\}$, where q_1 is a “state arc” coming out of the unique begin node (B), the source of the graph, and q_k is a “state arc” leading to the unique final node (F), the sink for the graph. The likelihood of the sequence $X = x_1, \dots, x_k$ and the parse $\phi = (q_1, \dots, q_k; x_1, \dots, x_k)$, according to a hidden Markov model, is the joint independent probability of the subsequences given the corresponding states and the probability of transitioning between states. That is,

$$P(X, \phi) = P(q_1|B) \left(\prod_{i=1}^k P(x_i|q_i) \right) \left(\prod_{i=1}^{k-1} P(q_{i+1}|node(q_i)) \right). \quad (1)$$

Using a variant of the Viterbi algorithm⁴, we can deduce the parse that maximizes equation 1.

While figure 1 represents the basic ordering of gene features, it fails to fully capture the syntactic restrictions of a “legal parse”. In an ideal DNA sequence, the parse is “frame consistent”, i.e., the total number of coding nucleotides is a multiple of three and the reading frame is consistent from exon to exon. We can add additional states to the model graph such that only frame consistent parses are allowed. Figure 2 shows the model graph representing a frame consistent GHMM. The three levels represent the three frames. Exon lengths are restricted in the various exon transitions in such a way to enforce frame consistency (see Kulp, *et al*²).

2.2 Sensors

A sensor is an mechanism for recognizing or scoring a subsequence according to a model of an abstract gene feature. There are two types of sensors used in the Genie system: signal sensors and content sensors.

2.3 Signal Sensor Models

Signal sensors are used to recognize transitions between states in a GHMM. This type of sensor is used in a pre-processing step to identify candidate sites where state transitions can occur. In the GHMMs shown in figures 1 and 2, the nodes correspond to gene features such as acceptor sites, donor sites, and positions of start

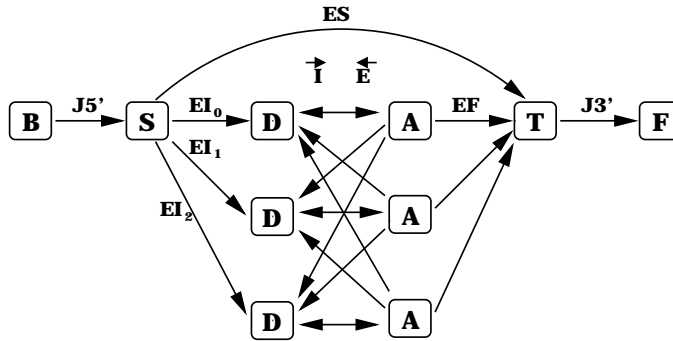


Figure 2: A GHMM including frame constraints. The additional acceptor and donor transition nodes ensure that only syntactically correct parses are considered.

and stop translation. A typical signal sensor might be a neural network to recognize an acceptor site.

2.4 Content Sensor Models

Content sensors are used to estimate the likelihood of a subsequence given a particular state in the GHMM. Typically, the model of a content sensor integrates evidence contributed from multiple sources and estimates a likelihood of a subsequence from the combined information. By applying a simple dynamic program we can estimate the likelihood of a region based on multiple contributions.

Each source of evidence is called a component; a component is trained to recognize a specific feature. For example, an intron content sensor may include a nucleotide component that estimates the likelihood of each nucleotide given an intron nucleotide model. Another component might estimate the likelihood of a subsequence as an intronic repeat given an intron repeat model. The content sensor outputs the likelihood estimation of a subsequence as the maximum likelihood obtained from combinations of components.

Figure 3 shows an example of a fictitious subsequence scored by an internal exon content sensor. It is composed of several components: a nucleotide component, a codon component, end-region components representing the regions adjacent to the acceptor and donor sites, and a database homology match component. A component returns a likelihood for each potential feature occurrence, called an “extent”. In the figure, the maximum likelihood is determined by the joint probability of the extents shown in the bottom of the figure, i.e. an acceptor extent, followed by two nucleotide extents, a database match extent, and three codon extents.

Again we use dynamic programming to calculate the joint probability of all extents. The dynamic program used here employs a simpler scheme than that used in the standard Viterbi algorithm for a hidden Markov model as used in our GHMM

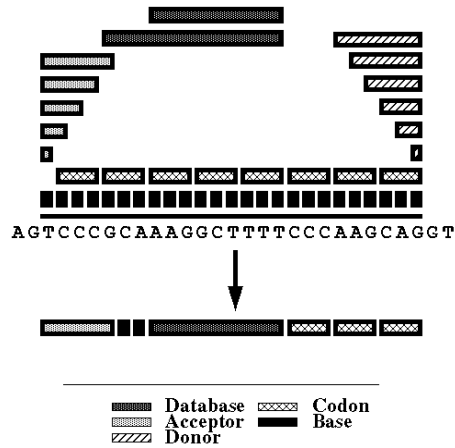


Figure 3: A sample content sensor combines evidence from multiple components to derive a maximum likelihood of the sequence. The arrow shows the combination of component features corresponding to the maximum likelihood.

gene model. Partial extents are prohibited and the transition probabilities between components are assumed to be uniform. The dynamic program is straightforward: the maximum likelihood is memoized at each nucleotide position in the sequence. Let i be the current position, from $1 \dots m$, in the subsequence of length m . For each extent ending at i , the likelihood of the sequence from $1 \dots i$ is calculated as the product of the likelihood of the extent times the maximum likelihood at the starting position of the extent. The extent which produced the maximum likelihood at position i is memoized, and the current position advances to $i + 1$. The result, is an $O(m)$ algorithm for estimating the maximum likelihood of the subsequence given the set of components in the content sensor.

This simple, efficient method encourages a modular approach to developing an effective gene-finding system because components can be easily added to or subtracted from a content sensor.

2.5 Integrating Homology Matches

Each component of a content sensor represents a model for a specific gene feature. The component estimates a likelihood for each possible extent. One such component is a database homology match component. Database homology raises the problem of assigning a “fair score” for a match relative to other scoring components such as the codon component. The typical solution is to experimentally weight database scores in an ad-hoc fashion. In the Genie system, the likelihood of a subsequence that is

found to be homologous to a subsequence in a database is theoretically interpreted in terms of the bit-cost to encode the subsequence given the database match. Let the “subject” be a database homolog and the “target” be the matching subsequence in the DNA sequence that is being analyzed. We estimate the encoding cost, C , of the target as the sum of the following costs:

- The offset into the database
- Translation Cost – the encoding cost of the target given the subject

The likelihood of the target is calculated as a probability of 2^{-C} .

The offset is described as the encoding cost for uniquely specifying where the match in the database is located. If the number of starting positions for matches in the database is S and assuming that all positions are uniformly likely, then the cost of encoding the offset of the subject in the database is $-\log(1/S) = \log(S)$. The translation cost of the target is database specific, but typically involves a substitution matrix to translate from the subject to the target. We show in section 3.3 how this value is derived for two different types of databases.

2.6 Identifying Splice Sites Using Homology

If two database homology matches are found in a single protein sequence and the two subsequences are adjacent, then we can conclude that the “match pair” implies an insertion in the target sequence of protein coding or non-coding nucleotides. If the insertion is of non-coding nucleotides, then a pair of splice sites can be inferred from a pair of homology matches. To reduce the chance that a match pair is the result of an insertion of coding nucleotides, we only consider those match pairs with suitable splice sites patterns at the match boundaries. To force the splice site to be used in a parse, the dynamic program (Viterbi) mentioned in section 2.1 is modified slightly. If one of the homology matches of a match pair is used to derive the likelihood in a candidate internal exon content sensor, then we infer that the candidate exon must splice to an exon that contains the partner in the match pair. During the dynamic program, if such a condition arises, only parses that conform to the match pair restriction are considered.

Figure 4 shows the situation in which a homology match pair, labeled M_1 and M_2 , are found near two candidate splice sites, D_2 and A_2 , respectively. The dynamic program disregards all other splice sites, i.e. D_1 , A_1 , D_3 , and A_3 , which ensures that *both* splice sites D_2 and A_2 are included or *neither* is included in the parse.

3 Experiments

The GHMM graph used in Genie is as shown in Figure 2. Here, we describe the specific implementation details of the sensors and components used in the Genie system.

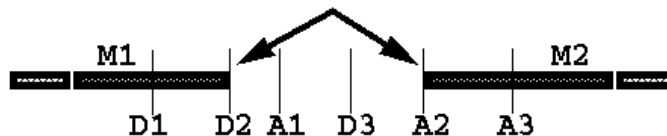


Figure 4: A match pair, M_1 and M_2 imply a pair of splice sites, D_1 and A_1 . The dynamic program is constrained such that D_1 is used in the final parse if and only if A_1 is used.

3.1 Signal Sensors

Three feedforward neural networks were trained to recognize acceptor sites, donor sites, and the beginning of translation. Each neural network uses a window of nucleotides – four binary inputs per nucleotide. See Kulp, *et al*⁸ for more details. The experimental thresholds for these neural networks were set low to ensure that very few false negatives were allowed.

The signal sensor for stop translation is trivial: all occurrences of the three possible stop codons, “TAA”, “TGA”, and “TAG” are considered as candidates.

Signal sensors serve as an initial pre-processing step to reduce the search space. As such, false negatives will add to the running time of the dynamic program, but are not a serious concern in terms of predictive accuracy because they will likely be discarded as improbable during the parsing step.

3.2 Content Sensors

There are five content sensors used in the Genie system: a non-coding sensor (for J5' and J3'), an intron model, and initial, internal, and final exon sensors. The components used are:

- Nucleotide component – a window of ± 150 bases estimates the local nucleotide frequency.
- Codon component – a 16-input neural network. The previous three nucleotides are encoded as 12 binary inputs and the local nucleotide frequency are the four remaining real valued inputs. The neural network has 61 outputs corresponding to the estimated likelihood of the 61 possible codons. This feedforward network has a single hidden layer of 17 units; softmax is used as the error function.
- BLASTX^{5,6} database component – a homology match component for a protein database described in section 3.3.
- BLOCKS database component – a homology match component for the protein motifs in the BLOCKS database⁷ described in section 3.3.
- Exon 5' end-region component – a simple profile trained on the 20 nucleotides downstream of the “AG” consensus.

- Exon 3' end-region component – a profile of the seven nucleotides upstream of the “GT” consensus.
- Intron 5' end-region component – a profile of the six nucleotides downstream of the “GT” consensus.
- Intron 3' end-region component – a profile of the 19 nucleotides upstream of the “AG” consensus.

All of the end-region components allow extents of lengths up to the size of the profile.

Some components are used in multiple sensors. The non-coding sensor contains only the nucleotide component. The intron sensor contains the nucleotide component, and the intron 5' and 3' end-region components. The internal exon sensor contains the codon component, the BLASTX and BLOCKS database components, and the exon 5' and 3' end-region components. The initial and final exon sensors contain the same components as the internal exon, except the 5' end-region and 3' end-region components are not included, respectively.

3.3 Translation Cost for Homology Match Components

The Genie system currently uses two different types of protein databases, the BLOCKS database and a non-redundant sequence database^b.

The BLOCKS database is a collection of over 2000 highly conserved protein motifs without insertions or deletions. We derived a profile for each motif using a nine-component Dirichlet mixture.⁸ The translation cost of a target nucleotide sequence given a BLOCKS homology is the combined cost of encoding the target protein product using the BLOCKS motif profile and the cost of translating the target protein product from an amino acid sequence into a nucleotide sequence. The profile cost is simply the product of the probability of each residue in each column. The protein-to-nucleotide cost is computed using the codon component described in section 3.2, but the probability of a codon is normalized to sum to one over all possible codons that translate to the same amino acid.

The other data we use is the non-redundant protein database, “nr”. To avoid trivial use of homology in our experiments we remove from nr any sequence with an estimated 50% or greater sequence identity with a protein coded by a sequence in our test set. We use BLASTX with $E = 1$ to identify potential homology matches in the non-redundant protein database. Given a potential match, the translation cost is similar to that used for the BLOCKS database, but the BLOSUM-62 substitution matrix is used instead of the motif profile. We set the BLASTX expectation parameter relatively high, resulting in a large number of false homology matches, but this does not mean that each potential match is actually used in the parse. Most are rejected when evidence from different components is combined during the dy-

^bThe “nr” database supplied by the NIH, available at <ftp://ncbi.nlm.nih.gov/blast/db/>, includes all protein sequences from GenBank, EMBL, DDBJ, and PDB, but excludes STSs and ESTs.

namic program step. On the other hand, an otherwise weak homology match may be sufficiently strong in a certain context to help identify a true exon.

3.4 Cross-validation Experiments

The data set used during the train/test experiments is a collection of 301 annotated, multiple-exon human DNA sequences from the GenBank sequence database.^c The data set was randomly partitioned into seven test sets of uniform size to be used in cross-validation experiments. For each test set, the content sensors were trained on the remaining training data and predictions were recorded for the sequences in the test set.

Additional tests were performed on a data set of 570 vertebrate genes. This data set was used by Bursett and Guigo as a benchmark for the comparison of many different gene-finders.⁹ The data set is similar to ours, but it includes all vertebrates and similar sequences are not removed.

4 Results

Tables 1, 3, and 2 show the results of running Genie using different configurations on all seven test sets, the average results over the entire data set, and the results using the Bursett/Guigo data set. In accordance with the testing scheme established by Bursett and Guigo, we report sensitivity and specificity with respect to per-base prediction of coding/non-coding and with respect to exact prediction of exons. The per-base sensitivity is the fraction of true coding bases predicted as coding, and the specificity is the fraction of all predicted coding bases that were correct. Similarly, the exon sensitivity is the fraction of true exons predicted exactly, and the specificity is the fraction of predicted exons that were correct. In these tests, correct exon prediction requires identification of the exact position of splice sites. Fully or partially overlapping predictions are not accepted. The approximate coefficient (AC) is described by Bursett and Guigo as a preferred alternative over the correlation coefficient and defined by

$$AC = \frac{1}{2} \left(\frac{TP}{TP+FN} + \frac{TP}{TP+FP} + \frac{TN}{TN+FP} + \frac{TN}{TN+FN} \right) - 1$$

where TP, FP, TN, and FN are true positives, false positives, true negatives, and false negatives.

In addition, we also report the fraction of true exons that were not identified either exactly or overlapping (Missing Exons) and the fraction of predicted exons that did not overlap any true exon (Wrong Exons).

^cThe data set is available at <ftp://genome.lbl.gov/pub/genesets/>.

Table 1: Prediction results on seven test sets and the Bursett/Guigo data set. “Per base” statistics refer to the ability to predict whether a nucleotide is coding or non-coding. “Per exon” statistics refer to the ability to predict a complete exon exactly.

Data Set	<i>Per Base</i>			<i>Exact Exon</i>				
	Sn	Sp	AC	Sn	Sp	Avg	ME	WE
Part 0	0.92	0.87	0.88	0.75	0.73	0.74	0.05	0.17
Part 1	0.65	0.73	0.63	0.46	0.47	0.46	0.26	0.28
Part 2	0.64	0.82	0.68	0.53	0.57	0.55	0.19	0.21
Part 3	0.73	0.81	0.72	0.56	0.59	0.57	0.19	0.22
Part 4	0.76	0.83	0.75	0.63	0.63	0.63	0.13	0.13
Part 5	0.70	0.83	0.73	0.56	0.55	0.55	0.19	0.20
Part 6	0.78	0.77	0.76	0.66	0.65	0.65	0.21	0.24
Average	0.74	0.81	0.74	0.59	0.59	0.59	0.17	0.21
B/G	0.87	0.88	0.85	0.69	0.70	0.69	0.10	0.15

Table 2: Prediction results on seven test sets and the Bursett/Guigo data set using homology matches and match pairs for splice site identification. In this table, we also include, for comparison, predictive results of GeneID+ and GeneParser3 as reported in Bursett and Guigo. Only sequences of length less than 8000 were tested in the latter data set to provide comparable results with the other genefinders.

Data Set	<i>Per Base</i>			<i>Exact Exon</i>				
	Sn	Sp	AC	Sn	Sp	Avg	ME	WE
Part 0	0.94	0.95	0.93	0.83	0.83	0.83	0.05	0.09
Part 1	0.73	0.79	0.71	0.55	0.55	0.55	0.23	0.29
Part 2	0.77	0.88	0.79	0.65	0.68	0.67	0.16	0.18
Part 3	0.79	0.82	0.77	0.60	0.61	0.61	0.19	0.28
Part 4	0.83	0.90	0.83	0.68	0.69	0.69	0.14	0.14
Part 5	0.76	0.83	0.76	0.64	0.64	0.64	0.14	0.20
Part 6	0.83	0.74	0.76	0.70	0.63	0.67	0.15	0.32
Average	0.80	0.84	0.79	0.66	0.66	0.66	0.15	0.21
B/G Data Set								
Genie	0.95	0.91	0.91	0.77	0.74	0.76	0.04	0.13
GeneID+	0.91	0.90	0.88	0.73	0.70	0.71	0.07	0.13
GeneParser3	0.86	0.91	0.86	0.56	0.58	0.57	0.14	0.09

Table 3: Prediction results on seven test sets and the Bursett/Guigo data set using homology matches, but no match pair splice site site identification.

Data Set	<i>Per Base</i>			<i>Exact Exon</i>				
	Sn	Sp	AC	Sn	Sp	Avg	ME	WE
Part 0	0.94	0.90	0.91	0.81	0.78	0.80	0.04	0.17
Part 1	0.72	0.77	0.70	0.50	0.50	0.50	0.21	0.30
Part 2	0.77	0.87	0.79	0.65	0.68	0.66	0.16	0.19
Part 3	0.75	0.80	0.74	0.55	0.58	0.56	0.22	0.28
Part 4	0.83	0.90	0.83	0.68	0.69	0.69	0.14	0.14
Part 5	0.76	0.83	0.76	0.63	0.63	0.63	0.14	0.20
Part 6	0.84	0.74	0.77	0.71	0.64	0.67	0.15	0.32
Average	0.80	0.83	0.78	0.65	0.64	0.64	0.15	0.22

5 Discussion

The work presented here extends the work and results reported in Kulp, *et al.* We have developed a simple probabilistic interpretation of database homology matches and described a means of combining database information and other sources of evidence into an integrated probability estimation. With the addition of information from database homology matches, overall accuracy in predicting exact exons increased by approximately 7%, and the number of missed and wrong exons dropped significantly. Per base sensitivity increased about 7% as well, and specificity rose 3%. The overall performance of Genie compares quite favorably with the other gene finders on the Bursett and Guigo dataset.

Genie appears to perform better on the Bursett and Guigo dataset than on our own dataset. This is especially interesting, since that dataset contains sequences from all vertebrates, whereas Genie was trained on human DNA sequences only, which are all that appear in our dataset. However, this may not be so surprising, given the variance in performance we observe even between different parts of our own dataset. We do not believe that this variance results from the difference in the training sets used for the seven different parts. Rather, it seems there is a wide range of difficulty in genefinding from one sequence to another, depending mostly on the number of exons in the gene and the length of the introns, so some test datasets just end up containing more “hard to parse” sequences than others. For this reason it is especially important to compare the performance of different gene finders on the same benchmark set of test sequences. It would be desirable to also use the same partition of the dataset for cross validation experiments, if possible. For this reason we have made our dataset available with a defined seven part partition.

It is worth noting that no correlation was detected between GC content and predictive accuracy, although additional work is needed in this regard. To illustrate the effectiveness of homology, Figure 5 and 6 show how Genie exploits homology to refine gene structure prediction. In figure 5, a protein fragment from a canine kidney

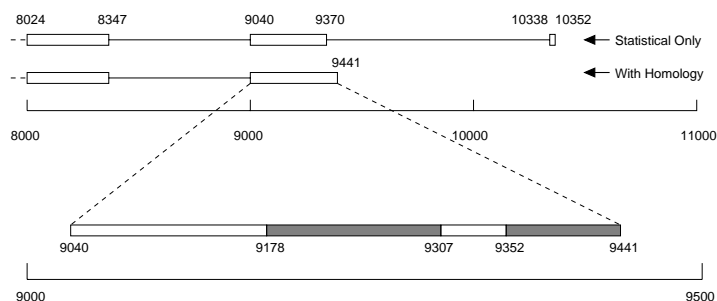


Figure 5: The diagram shows the gene prediction for the final 3000 bases of the 11kb DNA for the human osteopontin six exon gene (GenBank Accession D14813). The first prediction is the result of running Genie without homology information. Genie fails to identify the complete final exon and predicts an additional small final exon. The second prediction includes BLAST searches against the “nr” protein database. Here, two segments from a strong osteopontin-related homolog (PIR Accession A38646) are found, shown shaded, with a small insertion between them.

cell line is used to identify the final exon of an 11Kb DNA sequence. In figure 6, a sheep hypothalamic protein fragment was used by Genie to improve the prediction of a human transcript regulator gene. In the second example, a match pair constrained the parse and ensured correct prediction of the final exon. Homology information does not always improve predictions, however. In some cases, a false match may result in a worse genefinder prediction, but overall performance is improved for the entire test set using homology.

In future work we plan to extend Genie so that it can find multiple genes in a single DNA sequence. We also plan to improve the statistical model used in the intron state of Genie, as well as the model for intergenic DNA. This can be accomplished by incorporating sensors for promoters, the transcription start site, DNA repeat sequences, and the overall structure of 5' and 3' untranslated regions. Further work is also in progress to improve the splice site sensors. Finally, a WWW interface to Genie is also planned.

6 Acknowledgments

We would like to thank Kevin Karplus for his contributions to this work, especially the splice site profiles he built, and his valuable discussion. We also extend our gratitude to Gary Stormo, Nomi Harris, Greg Helt, and Richard Hughey for their assistance in the development of Genie. This work was supported in part by DOE grant no. DE-FG03-95ER62112 and DE-AC03-76SF00098. D. Haussler acknowledges support of the Aspen Center for Physics, Biosequence Analysis Workshop.

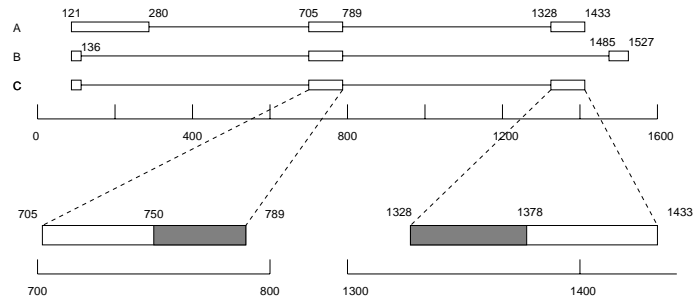


Figure 6: The diagram shows the gene prediction for a small transcription regulation gene (GenBank Accession U20325). (A) shows the correct annotated sequence, (B) is the prediction using statistical methods only, and (C) is the prediction using the “nr” protein database. A homology “match pair” (two adjacent matches in the protein fragment PIR B61322), shown shaded, ensured that the splice sites at positions 789 and 1328 were used in the prediction.

References

1. A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *JMB*, 235:1501–1531, February 1994.
2. D. Kulp, D. Haussler, M. Reese, and F. Eeckman. A generalized hidden Markov model for the recognition of human genes in DNA. In *ISMB-96*, St. Louis, June 1996. AAAI Press.
3. E. Snyder and G. Stormo. Identification of protein coding regions in genomic dna. *JMB*, 248, 1995.
4. L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16, January 1986.
5. S. F. Altschul, W. Gish, W. Miller, Myers E. W., and Lipman D. J. Basic local alignment search tool. *J. Mol. Biol.*, 215, 1990.
6. W. Gish and D. J. States. Identification of protein coding regions by database similarity search. *Nature Genetics*, 3, 1993.
7. Steven Henikoff and Jorja G. Henikoff. Automated assembly of protein blocks for database searching. *NAR*, 19(23):6565–6572, 1991.
8. K. Sjölander, K. Karplus, M. P. Brown, R. Hughey, A. Krogh, I. S. Mian, and D. Haussler. Dirichlet mixtures: A method for improving detection of weak but significant protein sequence homology. *CABIOS*, 12(4), 1996.
9. M. Burset and R. Guigo. Evaluation of gene structure prediction programs. *Genomics*, 34(3):353–367, 1996. Data set and evaluation results can be found at <http://www.imim.es/GenelDentification/Evaluation/Index.html>.
10. R. Guigo, S. Knudsen, N. Drake, and T. Smith. Prediction of gene structure. *J. Mol. Biol.*, 226:141–157, 1992.