ExonHunter: A Comprehensive Approach to Gene Finding

Broňa Brejová[*] Daniel G. Brown Ming Li Tomáš Vinař

School of Computer Science, University of Waterloo,
200 University Ave West, Waterloo, ON N2L 3G1, Canada
{bbrejova,browndg,mli,tvinar}@cs.uwaterloo.ca

Technical Report CS–2004–57

Abstract

We present ExonHunter, a new and comprehensive gene finder system that outperforms existing systems, featuring several new ideas and approaches. Our system combines numerous sources of information (genomic sequences, ESTs, and protein databases of related species) with a gene finder based on hidden Markov model in a novel and systematic way. In our framework, various sources of information are expressed as partial probabilistic statements about positions in the sequence and their annotation. We then combine these into the final prediction with a quadratic programming method extending existing methods. Allowing only partial statements is key to our transparent handling of missing information and coping with the heterogeneous character of individual sources of information. As well, we give a new method for modeling length distribution of intergenic regions in hidden Markov models. On a commonly used test set, ExonHunter performs significantly better than ROSETTA, SLAM, or TWINSCAN, and more than two thirds of genes were predicted completely correctly.

Keywords: gene prediction, comparative genomics, hidden Markov models

[*] Corresponding author. E-mail: bbrejova@cs.uwaterloo.ca

1 Introduction

Gene finding, predicting the exon-intron structure of genes, is a basic task in DNA sequence analysis. With more than 400 eukaryotic sequencing projects under way[1], gene finding tools easily adapted to new organisms are greatly needed. Early successes based on hidden Markov models (HMMs) (*e.g.,* GENSCAN [7]), have given way to comparative approaches, where DNA sequence is supplemented by alignments of expressed sequence tags or cDNAs (*e.g.,* GRAIL [26]), of known proteins (GenomeScan [27]), of a related genome (*e.g.,* Twinscan [13], SGP2 [17]), or by simultaneous analysis of syntenic regions of two organisms (*e.g.,* ROSETTA [2], SLAM [1]). Comparative approaches have demonstrated significant increase in performance at the level of exons (*i.e.,* correctly identifying matching pairs of splice sites on exon boundaries), but their accuracy at the transcript level (starting with the correct start site, through all splice sites, and ending at the correct stop codon) is far from perfect [6].

We propose a new flexible framework to incorporate supplementary sources of information into an HMM-based gene finder. Our new gene finder exploits information from proteins, ESTs, genome-genome comparisons, and sequence repeats to achieve better performance than several well-known programs (ROSETTA, SLAM, TWINSCAN). Our experiments also show that none of the information sources alone is sufficient to achieve the same performance as their combination.

We model individual sources of information as probabilistic statements with different level of granularity. For example, a region covered by an EST match is with high probability coding or untranslated region. On the other hand, for a protein alignment we can make a stronger statement: it is with high probability coding. Such statements with different granularity cannot be combined by traditional methods, such as linear combination [24]. We developed a new combination method based on quadratic programming, which generalizes the linear combination method to such statements. Interestingly, the method used in TWINSCAN is also a special case of our framework. When we have no comparative evidence at all, the system performs the same as the HMM alone.

Performance of comparative approaches to gene finding depends on the choice of informant genome [6]. In our system we can handle sources at different evolutionary distance differently, assigning them different levels of granularity or probability values.

Related work. Recently, systems that exploit syntenic genomic sequences from multiple species have emerged (*e.g.,* ExoniPhy [21]). Such systems combine probabilistic models of intron-exon structure of genes with phylogenetic models allowing for sequence divergence. They often require multiple alignments of syntenic DNA sequences from multiple species for both training and annotation of novel sequences. However, such data sets have only recently become available with the completion of rat genome [18] and through targeted sequencing of sequences orthologous to a short section of human genome [25]. The scarce availability of such data is a major obstacle in practical gene finding using such methods, especially for species attracting less interest than human.

Some other gene finding programs use several sources of information. HMMGene [14] and GENIE [15] are based on HMMs. Both differ from our framework significantly: instead of combining available information, only one source is chosen to influence score at each particular sequence position. EuGène [19] is based on probabilistically motivated directed acyclic graphs. The information from ESTs and protein matches is incorporated by direct modification of edge weights in the graph. Despite their attempts, neither approach reports successful inclusion of ESTs in the predictions.

[1] Genomes Online Database `http://www.genomesonline.org`, October 2004

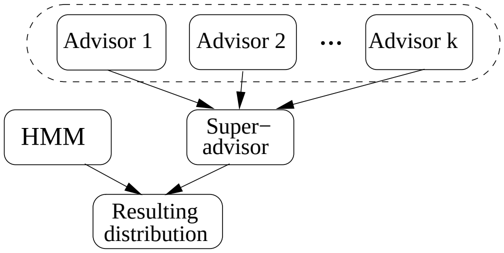

Figure 1: **Overview of model architecture.** Advisors, based on various sources of information, are combined into the superadvisor by a quadratic programming method. We modified Viterbi algorithm to find the most probable annotation in a combined distribution of superadvisor and HMM.

2 Gene Prediction with Numerous Information Sources

Gene finding seeks to label each position in a given DNA sequence as intergenic, intron, exon (in six different reading frames), donor site, acceptor site, start codon, or stop codon. An HMM for gene finding defines a conditional probability distribution over all possible *annotations* (sequences of labels) of a specific sequence. Biologically meaningless annotations receive zero probability. To predict genes, we find the annotation A^* that maximizes $\Pr(A^*|\mathit{sequence})$.

Combining an HMM with other supplementary sources of evidence (genome-genome sequence comparison, EST or protein alignments, etc.) is challanging. In our framework, each source of evidence yields one or several *advisors*. These are first combined into a *superadvisor*, which gives a probability distribution defining $\Pr(A|\mathit{evidence})$ over all annotations A. The two probability distributions, $\Pr(A|\mathit{evidence})$ and $\Pr(A|\mathit{sequence})$, are then combined with Bayesian principles, and we find the annotation A^* maximizing $\Pr(A^*|\mathit{sequence}, \mathit{evidence})$.

We use the two-step combination to avoid making independence assumptions about the advisors. Often, a linear combination is used to combine multiple predictions [24]. Here, we extend this principle to allow advisors to provide only partial information about the probability distribution they are supposed to predict. In this way, each advisor only gives the reliable information from its own source of evidence, and other advisors or the HMM complete the rest.

We demonstrate how to incorporate the two probability distributions by extending the well-known Viterbi algorithm. The resultant variation still takes time linear in the length of the sequence to find the most probable annotation of the sequence.

2.1 Advisors

In our model, supplementary information is represented in *advisors*. For each position in the sequence, an advisor specifies a probability distribution over annotation labels. For simplicity, we assume that the labels of different positions in the sequence are independent. (This assumption is, of course, false; we address this point in Section 2.3.) Then, the probability of a particular annotation is the product of the probabilities of labels at individual positions.

Some sources of information make it hard to estimate a complete probability distribution. For example, an advisor based on genome-genome comparison may reliably distinguish between coding and non-coding bases, but may not identify, for non-coding bases, whether they are intronic or intergenic. Therefore, loosely motivated by the Dempster–Shafer theory of evidence [20], we allow the advisor to provide only partial information as follows.

Definition 1 (Advice of an advisor). *Let Σ be the set of labels. The* advice *of advisor a at position i is a partition π_a of the set Σ and a probability distribution $p_a(S)$ over all partition elements $S \in \pi_a$. The value $p_a(S)$ is an estimate of the probability that the correct label at position i is in set S, given the information available to advisor a.*

In the genome comparison example above, the advisor can specify the probability of position being an "intron or intergenic position" (*i.e.,* non-coding position) instead of specifying the probability of each label separately. The partition π_a may be different for different positions in the sequence. For example, an advisor based on homology information may issue *vacuous* predictions ($\pi_a = \{\Sigma\}, p_a(\Sigma) = 1$) for unaligned positions.

2.2 Combination of Advisors

Next, we integrate all advice into a single *superadvisor* prediction. In this step, for each position in the sequence independently, we create a full probability distribution over all labels, resolving conflicts between the advice of different advisors. The superadvisor prediction at a particular position is probability distribution $x^* = (x_1, \dots, x_n)$ over all labels.

We may view each advisor's advice as a restriction on the distribution x^*. If all these restrictions were compatible, then for each advisor a, and for each set S in the partition π_a, the sum of the probabilities in x^* for all labels in S, should equal to $p_a(S)$.

In practice, however, the advice often conflicts, and no distribution x^* can satisfy all constraints. Therefore, we want to recover a distribution x^*, which is close to all advice. We have chosen to do this by minimizing the sum of weighted L_2 distances between the advice and x^*. For one advisor a, this distance is defined as follows:

$$\mathit{dist}_a(x^*) = \sum_{S \in \pi_a} \frac{1}{\mathit{prior}(S)} \cdot \left(p_a(S) - \sum_{j \in S} x_j \right)^2 , \tag{1}$$

where $\mathit{prior}(j)$ is the prior probability of label j, and $\mathit{prior}(S) = \sum_{j \in S} \mathit{prior}(j)$. We estimate the prior probability as the proportion of the genome annotated with a given label. Using $1/\mathit{prior}(S)$ as the weight on labels in S gives greater impact to the same absolute change in the probability for labels with small prior. Now advisor combination can be formulated as a convex quadratic program:

$$\begin{aligned} \text{minimize} \quad & \sum_a w_a \cdot \mathit{dist}_a(x^*) \\ \text{subject to} \quad & \sum_{j \in \Sigma} x_j = 1 \\ & x_j \geq 0 \text{ for all labels } j \in \Sigma \end{aligned} \tag{2}$$

While using quadratic programming to combine the advisors may not seem intuitive, we observe that in important special cases, our approach exhibits reasonable behaviour[2]. The quadratic program also allows for non-negative weights w_a to be assigned to advisors to represent their reliability.

Lemma 1. *Suppose that all advisors assign labels j and k to the same partition element. If we add an advisor whose advice is the prior probability for each label, then in the superadvisor prediction, x_j/x_k will be equal to $\mathit{prior}(j)/\mathit{prior}(k)$.*

Proof. Let us observe how the objective function of the quadratic program (2) changes when we vary ratio of x_j and x_k while keeping their sum $x_j + x_k$ fixed at some value s. The only terms

[2] This was not the case for other combination methods and distance measures we attempted.

in the objective function that change are the ones corresponding to the added advisor predicting prior with weight w:

$$\frac{w}{\mathit{prior}(j)}(\mathit{prior}(j) - x_j)^2 + \frac{w}{\mathit{prior}(k)}(\mathit{prior}(k) - x_k)^2. \tag{3}$$

By setting $x_k = s - x_j$ and taking derivative, this expression is minimized when $x_j = s \cdot \mathit{prior}(j)/(\mathit{prior}(j) + \mathit{prior}(k))$. Then the ratio of x_j and x_k is $\mathit{prior}(j)/\mathit{prior}(k)$. □

Thus, when there is no information about the distribution of probability between these labels, the probability in the superadvisor prediction is distributed according to the prior probabilities. We have added an advisor whose advice is this prior probability for each label, with small weight w_a. Because of its small weight, it does not influence the final prediction much, but it allows us to resolve these cases where no information exists reasonably.

The following lemma investigates the special case when several advisors predict probability distributions over the same partition of labels.

Lemma 2. *Assume that all members of a group of advisors issue advice over the same partition of labels. They can be replaced by a single advisor whose advice is a linear combination of the predictions of all the advisors in the group.*

Proof. We will prove the lemma for two advisors a and b with the same partition; the proof for more advisors follows directly. Let us assume that advisors a and b both have set S in their respective partitions π_a and π_b. The objective function thus contains terms

$$\frac{w_a}{\mathit{prior}(S)} \cdot \left(p_a(S) - \sum_{j \in S} x_j\right)^2 + \frac{w_b}{\mathit{prior}(S)} \cdot \left(p_b(S) - \sum_{j \in S} x_j\right)^2. \tag{4}$$

This expression can be rearranged as follows:

$$\frac{w_a + w_b}{\mathit{prior}(S)} \cdot \left(\frac{p_a(S)w_a + p_b(S)w_b}{w_a + w_b} - \sum_{j \in S} x_j\right)^2 + \frac{w_a w_b (p_a(S) - p_b(S))^2}{\mathit{prior}(S)(w_a + w_b)}. \tag{5}$$

The constant term not depending on x can be dropped without changing the solution of the quadratic program. Thus, advisors a and b can be replaced by a single advisor c with weight $w_c = w_a + w_b$, partition $\pi_c = \pi_a = \pi_b$ and predictions $p_c(S) = (p_a(S)w_a + p_b(S)w_b)/(w_a + w_b)$. Note that this defines a proper advisor since values $p_c(S)$ are non-negative and sum to one. □

For example, suppose all advisors use a complete partition of labels. Then the superadvisor is a linear combination of individual advisors. Often, linear combination is used to combine distributions when independence assumptions cannot be made [24]. Thus, our framework is a generalization of the linear opinion pool to predictions with incomplete distribution characterization.

In our implementation, we add more boundary conditions to the quadratic program to avoid extreme probability values for x_j's, so none is set to zero or one. In particular, we enforce the rule that changes of label probabilities with respect to the prior values may be at most 100-fold. Changing this limit is one way to tune the influence of the advisors on the overall prediction.

2.3 Combining the superadvisor and the HMM

For a given *sequence*, the hidden Markov model defines $\Pr(A|\mathit{sequence})$ for annotations A. Similarly, for given supplementary *evidence*, the superadvisor defines $\Pr(A|\mathit{evidence})$. In sequence-based prediction using HMMs, we seek the annotation A^* maximizing $\Pr(A^*|\mathit{sequence})$. In our case, we seek the most probable annotation given the sequence and the supplementary evidence. Such probability can be computed using Bayes' rule:

$$\Pr(A|\mathit{sequence},\mathit{evidence}) = \frac{\Pr(\mathit{sequence},\mathit{evidence}|A)\cdot\Pr(A)}{\Pr(\mathit{sequence},\mathit{evidence})}. \tag{6}$$

We assume that the supplementary evidence and the information contained in the sequence alone are independent. This assumption is not entirely true in practice, but we try to limit the dependencies by avoiding using the same features of the sequence in both the HMM and advisors. Thus, in the HMM, we focus on short windows in the sequence (signals, local coding potential, *etc.*), while advisors represent information from database search. Under this assumption, $\Pr(\mathit{evidence},\mathit{sequence}|A) = \Pr(\mathit{evidence}|A)\cdot\Pr(\mathit{sequence}|A)$, and we may simplify formula (6) to:

$$\Pr(A|\mathit{sequence},\mathit{evidence}) \propto \Pr(A|\mathit{sequence})\cdot\frac{\Pr(A|\mathit{evidence})}{\Pr(A)} \tag{7}$$

Since we seek only the most probable annotation A^*, we need not compute the normalization factor. Also, since we have made our previous positional independence assumption in Section 2.1, $\Pr(A|\mathit{evidence})$ can be computed by multiplying superadvisor probabilities position by position. The prior probability of annotation A is computed similarly.

If there is no non-vacuous advice available for the sequence, according to Lemma 1, the prediction $x_j^{(i)}$ of the superadvisor for label j at position i is equal to $\mathit{prior}(j)$. In such case, $\Pr(A|\mathit{evidence})/\Pr(A) = 1$, so the prediction according to the formula (7) will be the same as the prediction obtained by the HMM alone. This reasonable behaviour extends also to more complicated cases, where absence of supplementary information does not allow reliable advisor predictions on some subsets of labels at some positions in the sequence.

The most probable annotation according to the formula (7) can be recovered using a simple modification of the well-known Viterbi algorithm, given the positional independence assumption in our superadvisor model. It is sufficient to multiply the emission probability at position i in the state j by the factor $x_{\ell(j)}^{(i)}/\mathit{prior}(\ell(j))$, where $\ell(j)$ is the label assigned to state j. The running time of the modified algorithm remains linear in the length of the sequence.

Relation to TWINSCAN. TWINSCAN [13] enhances the prediction of an HMM by addition of a separate *conservation sequence* composed of characters representing matched, mismatched, and unaligned bases in the alignments with the informant genome. This can be seen as a special case of the advisor framework.

Lemma 3. *TWINSCAN can be implemented in the advisor architecture with a single advisor, making advice based only on 6-mers in the conservation sequence. If the underlying HMM is the same, the predictions of both systems are identical.*

Proof. To incorporate the conservation sequence into the predictions, TWINSCAN adds a separate emission probability in each state of the hidden Markov model, emitting the symbols of conservation sequence independently of the DNA sequence, depending only on five previous positions in the conservation sequence. Denote the conservation sequence $c = c_1 \dots c_n$. The probability of annotation $A = \ell_1 \dots \ell_n$ given *sequence* and the conservation sequence c, defined by TWINSCAN is simply:

$$\Pr_T(A|\mathit{sequence}, c) \propto \Pr(A|\mathit{sequence}) \cdot \prod_i \Pr(c_i|\ell_i, c_{i-5}, \dots, c_{i-1}) \tag{8}$$

Let us create an advisor, which at position i uses the 6-mer $c_{i-5}, \dots, c_i$ from the conservation sequence. The advisor will predict complete partition of labels, where for each label ℓ, the probability is defined as follows:

$$p^{(i)}(\ell) = \frac{\Pr(c_i|\ell, c_{i-5}, \dots, c_{i-1}) \cdot \mathit{prior}(\ell)}{Z(c_{i-5} \dots c_i)}, \tag{9}$$

where $Z(c_{i-5} \dots c_i)$ is a normalization constant needed to achieve that $\sum_{\ell'} p^{(i)}(\ell') = 1$. Note that $Z(c_{i-5} \dots c_i)$ does not depend on ℓ. When we combine this advisor with an HMM by our combination rule, conditional probability of annotation A will be:

$$\Pr_A(A|\mathit{sequence}, c) \propto \Pr(A|\mathit{sequence}) \cdot \prod_i \frac{p^{(i)}(\ell_i)}{\mathit{prior}(\ell_i)} \propto \frac{\Pr_T(A|\mathit{sequence}, c)}{\prod_i Z(c_{i-5} \dots c_i)} \propto \Pr_T(A|\mathit{sequence}, c) \tag{10}$$

Notice that $\prod_i Z(c_{i-5} \dots c_i)$ is a constant for a fixed conservation sequence, *i.e.,* it does not depend on annotation A. Thus, the two conditional distributions defined by TWINSCAN and by the advisor model above are the same. □

Positional independence assumption. So far we have assumed positional independence in the advisor predictions. However, this assumption is obviously false. For example, homology information comes in intervals, with strong dependencies between nearby positions in the sequence.

To deal with this problem, we replace most predictions of the superadvisor with vacuous predictions, so that all non-vacuous predictions are at least 50 positions apart. We choose the set of non-vacuous predictions with dynamic programming, maximizing the sum of scores measuring the "informativeness" of each position, where each position's score is $\max_j |\log(x_j/\mathit{prior}(j))|$. That is, we choose positions for the superadvisor that give large change compared to the prior.

2.4 Extended Hidden Markov Model for Gene Structures

We have used a generalized Hidden Markov model similar to that of GENSCAN [7] or AUGUSTUS [23] to model basic gene structure and sequence composition properties of different sequence elements, length distributions, and signals. We limit our description of the model to a few notable differences between traditional models and ours.

GC content. Model transition and emission probabilities depend on GC content level, estimated from a 1000bp window around current position. Other gene finders, such as GENSCAN, vary parameters based on the GC content level of the whole input sequence. Our approach is appropriate because even within a single gene, GC content level can vary significantly between coding and non-coding parts. We use four GC content levels; each covers roughly 25% of the sequence.

Signal models. We use higher order trees (HOT) models [3] of order 2 to model donor and acceptor site signals. HOT models capture significant non-adjacent intrasignal dependencies. Compared to other models, HOT models offer only a small improvement in discrimination power. However, they provide more accurate probability estimates than other models, and thus are appropriate for use with generative probabilistic models such as HMMs.

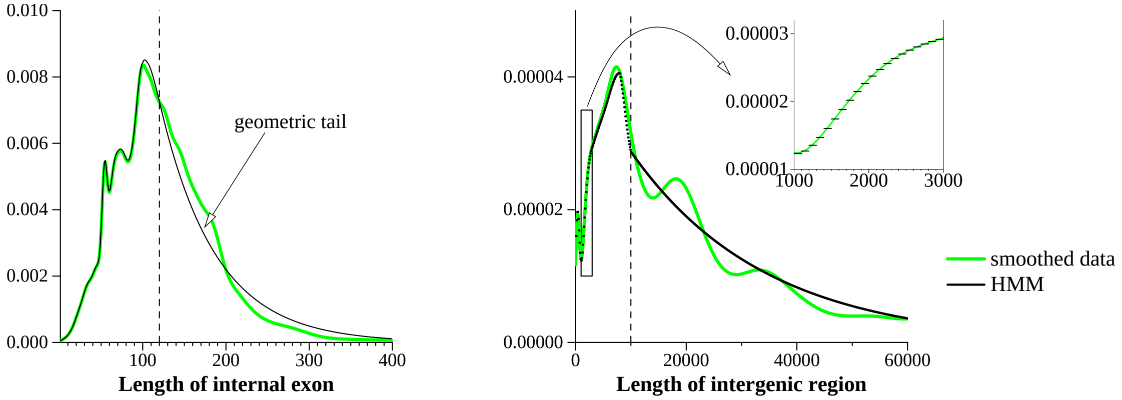

Figure 2: **Length distributions. Left:** Length distribution of internal exons from the AUGUSTUS training set with low GC content. The approximation used by the HMM is a combination of arbitrary head distribution and geometric tail. **Right:** Intergenic regions from chromosome 22 with low GC content. Inset shows part of the distribution in greater detail, highlighting step-function nature of the approximation used by the HMM.

Length distributions. We model length distributions of exons and introns with a technique developed in Brejova and Vinar [5]. The length distribution is decomposed into two parts: a head (with arbitrary distribution) and a geometrically decaying tail—see Fig.2. This decomposition allows use of a modified Viterbi algorithm that runs in $O(nd)$ time, where d is the length of the head region of the distribution. Both exon and intron lengths can be modeled accurately with small values of d, making the algorithm practical. GENSCAN cannot model non-geometric length distributions for introns; an intron model similar to ours was recently used by AUGUSTUS.

We further extended this approach to states that generate k characters at a time, and thus only produce lengths that are multiples of k, in $O(nd/k)$ time with $O(n)$ pre-processing. Now the state for intergenic region consists of two states: a generalized state generating a non-geometrically distributed number of k-mers, followed by a state with emission length distributed uniformly over the range from 1 to k. Such an intergenic region requires $O(n(d/k+k))$ inference running time. Setting $d \approx 10000$, $k \approx 100$, we can now model non-geometrically distributed intergenic lengths in practical running time—see Fig.2. This was not possible in GENSCAN or AUGUSTUS.

3 Construction of Advisors

In the previous section we described a general framework for including information from various supplementary sources of evidence into an HMM-based gene finder as advisors. Here, we present the specific advisors used in our human gene finder, ExonHunter.

ExonHunter currently incorporates information from protein databases (human, mouse, chicken, and Drosophila), ESTs (human and mouse), genome-genome comparison (mouse and Drosophila), and repeats found by RepeatMasker. In future work, we will incorporate information from more sources, both traditional (more ESTs and proteins from various organisms related to humans) and less traditional (protein families, transcription factor binding sites, CpG islands, *etc.*).

3.1 Homology Search Results as Intervals

For each homology search (*e.g.,* genome-genome comparisons, EST search, protein similarity), we represent its results as a set of intervals with their associated scores, with each interval typically corresponding to the region covered by a local alignment. The parameters of the homology search program and the definition of the score varies with the source of the information.

The advisor will, for each position, give either a vacuous advice, if it is not covered by any interval, or use a fixed binary partition $\pi = (X, \Sigma - X)$ for some subset X of all labels, otherwise. For positions in an interval, we estimate the probability p_X that the true label is in set X. This estimate depends on the score of the interval, and the distance to the nearest interval boundary. The score represents the overall quality of an alignment. The distance from the boundary is used since, for example, alignments covering exons may extend to neighbouring non-coding regions. If the site is covered by multiple intervals, we choose the interval maximizing p_X.

The values of p_X are estimated from the training data set. To limit the number of parameters, the score range and the distance range are each partitioned into several buckets, creating a two-dimensional bucket grid. For each two-dimensional bucket (i, j), the probability p_X is estimated as the true positive rate, or the fraction of sites in the bucket labeled by a label from X. To bucket the distances, we first pick a threshold T so that 40% of intervals achieve length at least $2T$. Then we make one bucket for each value from 1 to T, and one for all distances greater than T. The scores are then divided into a pre-specified number of buckets (in most cases, we have chosen 5) by a simple dynamic programming algorithm to minimize the entropy in the bucket partitions.

Proteins. We used 11072 human, 7778 mouse, 1085 chicken, and 2047 Drosophila proteins from SwissProt Release 44 (July 2004). Each species yields separate set of advisors.

We use BLASTX [10] with increased gap penalties to find regions of the input DNA sequence homologous to the proteins. We discard alignments containing long gaps potentially spanning introns, and remove 2 codons from each side to further avoid avoid non-coding regions. We represent resulting alignments as intervals with scores corresponding to BLOSUM62 score per position. The corresponding advisors include the frame implied by the alignment in the advice.

If the beginning of a protein is in the alignment, the start codon label at the corresponding position, and the intergenic label at positions $-100 \ldots -1$ are advised, using the interval framework and the same score. The third advisor predicts stop codons if the alignment has the end of a protein.

Expressed Sequence Tags. We used the TIGR human gene index (release 13, October 2003, consisting of 843769 ESTs) and TIGR mouse gene index (release 12, October 2003, with 669402 ESTs), with each creating two EST-based advisors: one for exons, one for introns.

To improve speed, we filter ESTs against the input sequence using PatternHunter [16], with a seed enhancing homologous coding region sensitivity [4]. ESTs with a significant alignment are realigned using SIM4 [9], producing intervals of presumed exons and introns (gaps between neighbouring alignments from the same EST). The score is the percent identity of SIM4 alignments.

ESTs often include untranslated regions, which are hard to separate from the coding parts of the ESTs. We have experimented with ESTScan [11] with unsatisfactory results. Therefore, in human ESTs, the exon intervals are used to predict "exon or intergenic", and intron intervals predict "intron or intergenic". Untranslated regions are not as well conserved between human and mouse: far fewer intron intervals from mouse ESTs occurred in intergenic regions. Therefore the mouse intron advisor predicts only the "intron" label.

Another problem arises from alternative splicing. The same position can be covered by both intron and exon intervals. Instead of attempting to isolate ESTs corresponding to a single splicing variant, we remove the EST predictions for sites covered by both intron and exon intervals. In this way, we leave the choice of the splicing variant to the HMM. If a prediction lower than prior probability should be made for some position, such prediction is removed.

Genome to Genome Comparison. We included two advisors based on genome to genome comparison: one based on the Drosophila genome sequence (release 3 from fruitfly.org), and one based on the mouse genome (from genome.ucsc.edu, October 2002).

We used PatternHunter with the coding region detection seed to locate significant alignments between the genome and the input sequence. We rescored the alignments in all 6 frames with the BLOSUM62 matrix, chose the best frame, and located the highest-scoring segment after removing frameshifts (gaps that are not multiple of three). To avoid alignments in non-coding regions, we removed 7 codons from each side. The score of each interval is the BLOSUM62 score per position.

Advice from the Drosophila genome includes the frame implied by the re-scoring. For mouse genome, about one third of frame predictions by BLOSUM62 scoring were wrong on the training set; therefore, the advisor predicts "exons" without specifying frame. Moreover, the intervals in the training data often included non-coding parts; therefore, we used only very strong matches, removing all advice that raised the exon probability less than 10 times above the prior.

As we demonstrate in Section 4, our approach to human-mouse genome comparison is contributing little to the final result (in fact, in many experiments, we abandon the mouse genome advisor altogether). Other authors [2, 13, 1] have obtained better results based on local patterns observed in a human-mouse alignment, rather than on global properties of the alignment. Such method can be easily incorporated into our framework, given for example, Lemma 3 that shows how to build an advisor equivalent to TWINSCAN.

3.2 Repeats

Compared to other gene finding programs, which either mask the original sequence for repeats, or ignore repeats altogether, we use a different method to deal with sequence repeats. We base an advisor on a list of likely repeats produced by RepeatMasker [22]. We have divided the repeats into four categories, each handled separately. Low complexity repeats and simple repeats whose periodicity is a multiple of three are ignored: significant portions of these occur in coding regions. The only class of repeats found to be a good predictor of intergenic regions are satellites, which form the second category. At positions annotated as satellites, the repeat advisor predicts the "intergenic" label. Simple repeats whose periodicity is not a multiple of three form the third category, boosting the probability of a position being "intron or intergenic". Finally, the fourth category consists of all other repeats. At such positions we again predict "intron or intergenic".

4 Experimental Results

Our primary testing set is the ROSETTA set of 117 human single-gene sequences developed by Batzoglou et al. [2]. This data set was recently reused by Alexandersson et al. [1] to compare SLAM against other gene finders, and we reuse the results of their experiments for comparison. We also experimented with human chromosome 22 (Sanger annotation, release 3.1b, 2002). Half of the chromosome was used as a supplementary training set, and the other half as a testing set.

We trained the hidden Markov model on a training set of 1284 human single-gene sequences created by Stanke et al. [23]. We removed 81 sequences from this set due to significant similarities

	GENSCAN	ROSETTA	SLAM	TWINSCAN	TWINSCAN.p	SGP-1	EH
Gene Sn	44%	—	—	—	—	—	68%
Gene Sp	41%	—	—	—	—	—	63%
Exon Sn	82%	83%	78%	84%	86%	70%	90%
Exon Sp	73%	83%	76%	77%	82%	76%	83%
Nucl. Sn	98%	94%	95%	98%	96%	94%	99%
Nucl. Sp	88%	98%	98%	89%	94%	96%	93%

Table 1: **Comparison on ROSETTA set.** Results for ROSETTA, SLAM, TWINSCAN, TWINSCAN.p (alignments from known orthologs only), and SGP-1 are from [1] (the authors did not report gene statistics). The EH column gives the best results achieved by ExonHunter with all advisors except mouse genome. We used standard definitions of sensitivity (Sn) and specificity (Sp), see, for example, [1]. We evaluated the data with program of Keibler and Brent [12].

	with mouse genome		w/o mouse genome		repeats
	all	non-human	all	non-human	only
Gene Sn	68%	62%	68%	63%	45%
Gene Sp	59%	54%	63%	59%	43%
Exon Sn	90%	88%	90%	88%	77%
Exon Sp	81%	79%	83%	81%	74%
Nucl. Sn	99%	99%	99%	98%	94%
Nucl. Sp	93%	91%	93%	93%	92%

Table 2: **Comparison ExonHunter variants.** Even without advisors from human information, ExonHunter outperforms established gene finders.

to the ROSETTA set. We trained intergenic region lengths and all parameters for advisors on the chromosome 22 training set. For these, we need a significant amount of intergenic sequence.

Comparison to other programs. Table 1 shows the comparison of ExonHunter with other gene finding programs evaluated by Alexandersson et al. [1] on the ROSETTA data set. ExonHunter used advisors based on human and mouse ESTs, human, mouse, and chicken protein alignments, and Drosophila genome-genome comparison. On this data set, we have outperformed all other tested programs at both exon and nucleotide levels, except for nucleotide specificity. We excluded the mouse genome advisor, as the results slightly worsened when it was added, mostly affecting gene specificity (see Table 2, discussion in Section 3.1). At the gene level, our program identifies more than two thirds of genes in the data set completely correctly.

One could object to this test since many of the genes in the ROSETTA set are also found in the database of human ESTs or proteins. Therefore, we also evaluated the program without advisors based on human information (Table 2). We still maintain the highest sensitivity on both exon and nucleotide levels, with only a 2% drop in exon specificity; the change mostly affects the gene statistics.

ExonHunter on chromosome 22. To test ExonHunter on longer genomic sequences, we ran the program on the testing set from human chromosome 22 and compared the results to GENSCAN (Table 3). Here, the general trend is similar to the observations for the other similarity based gene finders by Parra et al. [17] and Chatterji and Pachter [8]: the sensitivity stays roughly the same, while the number of exons and coding nucleotides predicted decreases significantly. The numbers from our experiments (data not shown) do not directly compare to [17, 8] since we used different subsets of chromosome 22, and different version of the annotation.

	GENSCAN	ExonHunter
Gene Sensitivity	11%	12%
Gene Specificity	4%	3%
Exon Sensitivity	72%	73%
Exon Specificity	30%	35%
Nucleotide Sensitivity	91%	91%
Nucleotide Specificity	36%	51%

Table 3: **GENSCAN and ExonHunter on chromosome 22 testing set.** Specificity increases, while sensitivity remains the same. The same trend is observed by other authors on similar datasets.

Advisors used	Exon Sn.	Exon Sp.
repeats only	77%	74%
genomes: D	77%	74%
genomes: M	78%	72%
ESTs: H	79%	77%
ESTs: M	85%	78%
ESTs: HM	86%	81%
Proteins: HMDC	86%	79%
genomes: D; proteins: DC	80%	75%
ESTs: H; proteins: H	87%	81%
ESTs: M; proteins: M	88%	81%
all advisors	90%	81%
all except M genome	90%	83%

Table 4: **Contribution of various combinations of advisors** on ROSETTA data set. H - human; M - mouse; C - chicken; D - Drosophila. Proteins and ESTs alone contribute comparable amounts of information towards the final results. The combination of all advisors performs better than advisors individually.

Contribution of individual advisors. Table 4 shows that the most influence on the final result comes from the EST-based advisors, followed closely by combination of protein advisors. Mouse ESTs work significantly better than human ESTs, most likely due to low conservation in untranslated regions between human and mouse. The contributions of Drosophila and chicken together appears comparable to the contribution of human ESTs. The most useful organism as a source of supplementary information is mouse, closely followed by the set of advisors originating in human. Finally, the combination of all advisors performs better than each of the advisors alone.

Cooperation of Advisors with HMM. Figure 3 shows example of ExonHunter annotation on human gene HSU30787. Both GENSCAN and ExonHunter without advisors predict most of the splice sites correctly but both annotations miss the first exon completely with GENSCAN extending the gene into intergenic region and ExonHunter starting inside the second exon. We added advisors based on mouse EST alignments, mouse and Drosophila protein alignments, and Drosophila genome-genome comparison, resulting in very clean superadvisor advice for all exons except the first, which was still not covered by any alignment. However, this helped the HMM to extend the second exon correctly and locate the alternative start site and the first exon, resulting in a completely correctly predicted gene.

5 Concluding Remarks

We have introduced a probabilistic framework for incorporating many sources of supplementary information into HMM-based gene finder, resulting in a practical gene finder with promising performance on human sequences.

Our framework is based on probabilistic statements made using various sources of information,

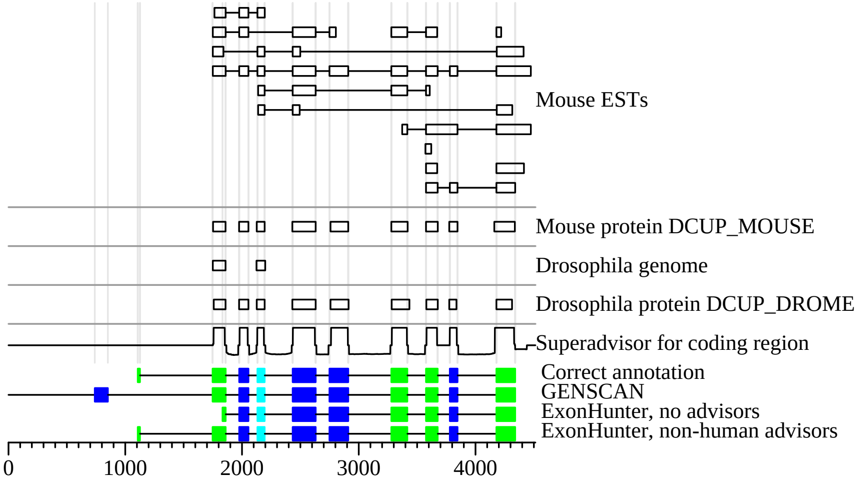

Figure 3: **Predictions on human gene HSU30787.** Similarity matches processed by Drosophila and mouse advisors help to find correct start site, even though it is not directly covered by any of the matches.

called advisors. Advisors create advice with varying granularity and forcefulness, to avoid making uninformed predictions. Thus, they cannot be combined by traditional expert combination methods. We developed a quadratic programming-based method, extending the traditional linear combination approach, and adapted the Viterbi algorithm to our domain. TWINSCAN's approach to incorporate human-mouse comparison can also be seen as a special case of our framework. We also developed a novel method for modeling intergenic length distributions in HMMs.

Our gene finder, ExonHunter, outperforms several other programs such as SLAM, TWINSCAN, or ROSETTA, even if all supplementary information originating from human-based advisors is withdrawn. We also evaluated contribution of individual advisors, finding that protein and EST databases are the two largest contributors towards the final result. However, neither of those sources performs better alone than in combination.

Our approach is becoming ever more relevant, as more EST sequence collections for organisms related to humans are built. For example, TIGR currently maintains EST libraries for 8 such organisms. We implicitly allow for variability in handling informant species with varying evolutionary distance with respect to the reference organism. The method easily transfers to other species, since it does not require special species-specific data sets.

Finally, in our experiments on ROSETTA set we observed that more than two thirds of genes were predicted exactly correctly. Improvement in this measure allows better analysis of structure and function of the encoded protein, for example using computational protein folding. As such, our results are a tangible step in moving towards fully *in silico* analysis of newly sequenced genomes and their proteins.

References

[1] M. Alexandersson, S. Cawley, and L. Pachter. SLAM: cross-species gene finding and alignment with a generalized pair hidden Markov model. *Genome Research*, 13(3):496–502, 2003.

[2] S. Batzoglou, L. Pachter, J. P. Mesirov, B. Berger, and E. S. Lander. Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Research*, 10(7):950–958, 2000.

[3] B. Brejova, D. G. Brown, and T. Vinar. Optimal DNA signal recognition models with a fixed amount of intrasignal dependency. In G. Benson and R. Page, editors, *Algorithms and Bioinformatics: 3rd International Workshop (WABI)*, volume 2812 of *Lecture Notes in Bioinformatics*, pages 78–94, Budapest, Hungary, September 2003. Springer.

[4] B. Brejova, D. G. Brown, and T. Vinar. Optimal spaced seeds for homologous coding regions. *Journal of Bioinformatics and Computational Biology*, 1(4):595–610, 2004.

[5] B. Brejova and T. Vinar. A better method for length distribution modeling in HMMs and its application to gene finding. In A. Apostolico and M. Takeda, editors, *Combinatorial Pattern Matching, 13th Annual Symposium (CPM)*, volume 2373 of *Lecture Notes in Computer Science*, pages 190–202, Fukuoka, Japan, July 3–5 2002. Springer.

[6] M. R. Brent and R. Guigo. Recent advances in gene structure prediction. *Current Opinion in Structural Biology*, 14(3):264–272, 2004.

[7] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268(1):78–94, 1997.

[8] S. Chatterji and L. Pachter. Multiple organism gene finding by collapsed Gibbs sampling. In *Proceedings of the 8th Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 187–193. ACM Press, 2004.

[9] L. Florea, G. Hartzell, Z. Zhang, G. M. Rubin, and W. Miller. A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Research*, 8(9):967–974, 1998.

[10] W. Gish and D. J. States. Identification of protein coding regions by database similarity search. *Nature Genetics*, 3(3):266–272, 1993.

[11] C. Iseli, C. V. Jongeneel, and P. Bucher. ESTScan: a program for detecting, evaluating, and reconstructing potential coding regions in EST sequences. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB'99)*, pages 138–148. AAAI press, 1999.

[12] E. Keibler and M. R. Brent. Eval: a software package for analysis of genome annotations. *BMC Bioinformatics*, 4(1):50, 2003.

[13] I. Korf, P. Flicek, D. Duan, and M. R. Brent. Integrating genomic homology into gene structure prediction. *Bioinformatics*, 17(Suppl.1):S140–8, 2001.

[14] A. Krogh. Using database matches with for HMMGene for automated gene detection in Drosophila. *Genome Research*, 10(4):523–528, 2000.

[15] D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman. Integrating database homology in a probabilistic gene structure model. *Pacific Symposium on Biocomputing (PSB)*, pages 232–234, 1997.

[16] B. Ma, J. Tromp, and M. Li. PatternHunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440–445, 2002.

[17] G. Parra, P. Agarwal, J. F. Abril, T. Wiehe, J. W. Fickett, and R. Guigo. Comparative gene prediction in human and mouse. *Genome Research*, 13(1):108–117, 2003.

[18] Rat Genome Sequencing Project Consortium. Genome sequence of the Brown Norway rat yields insights into mammalian evolution. *Nature*, 428(6982):493–521, 2004.

[19] T. Schiex, A. Moisan, and P. Rouzé. EUGÈNE: An eukaryotic gene finder that combines several sources of evidence. In O. Gascuel and M.-F. Sagot, editors, *Computational Biology. Selected papers from First International Conference on Biology, Informatics, and Mathematics*, volume 2066 of *LNCS*, pages 111–125, 2000.

[20] G. A. Shafer. *Mathematical theory of evidence*. Princeton University Press, 1976.

[21] A. Siepel and D. Haussler. Computational identification of evolutionarily conserved exons. In *Proceedings of the 8th Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 177–186. ACM Press, 2004.

[22] A. F. A. Smit, R. Hubley, and P. Green. RepeatMasker. `http://www.repeatmasker.org`, 2002.

[23] M. Stanke and S. Waack. Gene prediction with a hidden Markov model and a new intron submodel. *Bioinformatics*, 19(Suppl.2):II215–II225, 2003.

[24] D. M. J. Tax, M. van Breukelen, R. P. W. Duin, and J. Kittler. Combining multiple classifiers by averaging or multiplying? *Pattern Recognition*, 33:1475–1485, 2000.

[25] J. W. Thomas et al. Comparative analyses of multi-species sequences from targeted genomic regions. *Nature*, 424(6950):788–793, 2003.

[26] Y. Xu and E. C. Uberbacher. Automated gene identification in large-scale genomic sequences. *Journal of Computational Biology*, 4(3):325–328, 1997.

[27] R. F. Yeh, L. P. Lim, and C. B. Burge. Computational inference of homologous gene structures in the human genome. *Genome Research*, 11(5):803–806, 2001.