BIOINFORMATICS

Vol. 21 Suppl. 1 2005, pages i57–i65
doi:10.1093/bioinformatics/bti1040

ExonHunter: a comprehensive approach to gene finding

Broňa Brejová, Daniel G. Brown, Ming Li and Tomáš Vinař*

School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, ON, Canada N2L 3G1

Received on January 15, 2005; accepted on March 27, 2005

ABSTRACT

Motivation: We present ExonHunter, a new and comprehensive gene finding system that outperforms existing systems and features several new ideas and approaches. Our system combines numerous sources of information (genomic sequences, expressed sequence tags and protein databases of related species) into a gene finder based on a 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 via a quadratic programming method, which we show to be an extension of 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. In addition, we give a new method for modeling the length distribution of intergenic regions in hidden Markov models.

Results: On a commonly used test set, ExonHunter performs significantly better than the existing gene finders ROSETTA, SLAM and TWINSCAN, with more than two-thirds of genes predicted completely correctly.

Availability: Supplementary material available at http://www.bioinformatics.uwaterloo.ca/supplements/05eh/

Contact: bbrejova@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 by Burge and Karlin, 1997] have given way to comparative approaches, where DNA sequence is supplemented by alignments of expressed sequence tags (ESTs) or cDNAs [e.g. GRAIL by Xu and Uberbacher (1997)], of known proteins [GenomeScan by Yeh *et al*. (2001)] or of a related genome [e.g. Twinscan by Korf *et al*. (2001); SGP2 by Parra *et al*. (2003); Projector by Meyer and Durbin (2004)] or by simultaneous analysis of syntenic regions of two organisms [e.g. ROSETTA by Batzoglou *et al*. (2000); SLAM by Alexanderson *et al*. (2003)]. Comparative approaches have demonstrated a 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 full transcript level (with the start site, all splice sites and the stop codon correct) is far from perfect (Brent and Guigo, 2004).

We propose a new, flexible framework to incorporate supplementary information sources 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 no one information source alone is sufficient to achieve the same performance as their combination.

We model individual sources of information as probabilistic statements with different levels of granularity. For example, a region covered by an EST match is probably a coding or untranslated region. On the other hand, for a protein–DNA alignment, we can make a stronger statement: it is likely that the region is coding. Such statements, with different granularity, cannot be combined by traditional methods such as linear combination (Tax *et al*., 2000). We have developed a new combination method based on quadratic programming, generalizing the linear combination method to such statements. Interestingly, the method used in TWINSCAN is a special case of our framework. When we have no comparative evidence at all, the system performs in the same way as the HMM alone.

The performance of comparative approaches to gene finding depends on the choice of informant genome (Brent and Guigo, 2004). 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 by Siepel and Haussler (2004)]. Such systems combine probabilistic models of the intron–exon structure of genes with phylogenetic models allowing for sequence

*To whom correspondence should be addressed.

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

divergence. They often require multiple alignments of syntenic DNA sequences from multiple species for both training and annotation of novel sequences. However, such datasets have only recently become available with the completion of the rat genome (Rat Genome Sequencing Project Consortium, 2004) and through targeted sequencing of sequences orthologous to a short section of the human genome (Thomas *et al.*, 2003). 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 (Krogh, 2000) and GENIE (Kulp *et al.*, 1997) are based on HMMs. Both differ from our framework significantly: instead of combining the available information, only one source is chosen to influence the score at each particular sequence position. EuGène (Schiex *et al.*, 2000) is based on probabilistically motivated directed acyclic graphs. The information from ESTs and protein matches is incorporated by direct modification of the edge weights in the graph. Despite their attempts, neither approach reports successful inclusion of ESTs in the predictions.

Combiner (Allen *et al.*, 2004), a new gene finder for *Arabidopsis thaliana*, combines the predictions of several gene finding programs with sequence alignments, using decision trees and dynamic programming. However, our approach allows the exploration of more possible gene structures, and it may ultimately choose one that does not appear optimal before incorporating additional evidence.

2 GENE PREDICTION WITH NUMEROUS INFORMATION SOURCES

Gene finders label each position in a given DNA sequence as intergenic, or from an intron, exon (in all 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 $\Pr(A^*|\text{sequence})$.

Combining an HMM with other supplementary sources of evidence (genome–genome sequence comparison, EST or protein alignments, etc.) is challenging. 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 $\Pr(A|\text{evidence})$ over all annotations A. The two probability distributions $\Pr(A|\text{evidence})$ and $\Pr(A|\text{sequence})$ are then combined using Bayesian principles, and we find the annotation A^* maximizing $\Pr(A^*|\text{sequence, evidence})$.

We use the two-step combination to avoid making independence assumptions about the advisors. Often, a linear combination approach is used to combine multiple predictions (Tax *et al.*, 2000). 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 gives only the level of detail appropriate for its 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 the 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 (Shafer, 1976), 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 a position being an 'intron or intergenic position' (i.e. a non-coding position), instead of specifying each label's probability separately. The partition π_a may be different at different positions in the sequence. For example, an advisor based on homology information may issue *vacuous* predictions ($\pi_a = \{\Sigma\}, p_a(\Sigma) = 1$) at unaligned positions.

2.2 Combination of advisors

Next, we integrate all advice into a single superadvisor prediction. 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 a probability distribution over all labels $x^* = (x_1, \ldots, x_n)$, where x_i is the probability of the ith label from Σ, given all advice.

We may view each advisor's advice as a restriction on the distribution x^*. If these restrictions were compatible, then for

each advisor a, and for each set S in partition π_a, the sum of the probabilities in x^* for all labels in S should equal $p_a(S)$.

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 the weighted L_2 distances between the advice and x^*. For one advisor a, this distance is defined as follows:

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

where $\mathrm{prior}(j)$ is the prior probability of label j, and $\mathrm{prior}(S) = \sum_{j \in S} \mathrm{prior}(j)$. We estimate the prior probability as the proportion of the genome annotated with a given label. Using $1/\mathrm{prior}(S)$ as the weight on labels in S gives greater impact to the same absolute change in the probability of labels with small prior. Now advisor combination can be formulated as a convex quadratic program (see Fletcher, 1987 for a reference):

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

Although 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 the advisors to represent their reliability.

LEMMA 1. *Consider two labels j and k, and suppose that all advisors assign them 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 prior(j)/prior(k).*

PROOF. Consider how the objective function of the quadratic program (2) changes as we vary the ratio of x_j and x_k while keeping their sum $x_j + x_k$ equal to a fixed s. The only terms in the objective function that change are the ones corresponding to the added advisor, predicting the prior with weight w:

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

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

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

Thus, when there is no information about the distribution of probability among a set of 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 in a reasonable way those cases where no information exists.

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

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

This expression can be rearranged as follows:

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

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 the values of $p_c(S)$ 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 (Tax *et al.*, 2000). Thus, our framework is a generalization of this linear opinion pool framework to predictions with incomplete distribution characterization.

In our implementation, we add more boundary conditions to the quadratic program to avoid extreme probability values for the x_j, 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 can tune the influence of the advisors on the overall prediction.

2.3 Combining the superadvisor and the HMM

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

$$\Pr(A|\text{seq},\text{ev}) = \frac{\Pr(\text{seq},\text{ev}|A)\cdot\Pr(A)}{\Pr(\text{seq},\text{ev})}. \tag{6}$$

We assume that the supplementary evidence and the information contained in the sequence alone are independent given annotation A. This assumption is not true in practice, but we try to limit dependencies by avoiding using the same features of the sequence in both the HMM and the advisors. Note that we do not make any independence assumptions between individual advisor predictions. In the HMM, we focus on short windows of the sequence (signals, local coding potential, etc.), whereas the advisors represent repeats and database searches.

Under this conditional independence assumption between HMM prediction and superadvisor prediction, $\Pr(\text{seq},\text{ev}|A) = \Pr(\text{seq}|A)\cdot\Pr(\text{ev}|A)$, and we may simplify Equation (6) to

$$\Pr(A|\text{seq},\text{ev}) \propto \Pr(A|\text{seq})\cdot\frac{\Pr(A|\text{ev})}{\Pr(A)}. \tag{7}$$

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, $\Pr(A|\text{ev})$ 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 $\text{prior}(j)$. In such a case, $\Pr(A|\text{ev})/\Pr(A) = 1$, so the prediction according Equation (7) will be the same as the prediction obtained using the HMM alone. This reasonable behaviour also extends 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 Equation (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 $x^{(i)}_{\ell(j)}/\text{prior}(\ell(j))$, 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.

Relationship to TWINSCAN. TWINSCAN (Korf *et al.*, 2001) 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 our architecture with a single advisor, making advice based only on 6mers in the conservation sequence. If the underlying HMM is the same, the predictions of both systems are also the same.*

PROOF. To incorporate the conservation sequence $c_1\cdots c_n$ into the predictions, TWINSCAN adds a separate emission probability in each state of the HMM, emitting the symbols of the conservation sequence independently of the DNA sequence, depending only on the five previous positions in the conservation sequence. The probability of annotation $A = \ell_1\cdots\ell_n$ given the DNA sequence and the conservation sequence defined by TWINSCAN is

$$\Pr_T(A|\text{seq},c) \propto \Pr(A|\text{seq})\cdot\prod_i \Pr(c_i|\ell_i, c_{i-5},\ldots,c_{i-1}). \tag{8}$$

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

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

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

$$\begin{aligned}\Pr_A(A|\text{seq},c) &\propto \Pr(A|\text{seq})\cdot\prod_i \frac{p^{(i)}(\ell_i)}{\text{prior}(\ell_i)}\\ &\propto \frac{\Pr_T(A|\text{seq},c)}{\prod_i Z(c_{i-5}\cdots c_i)} \propto \Pr_T(A|\text{seq},c).\end{aligned} \tag{10}$$

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

2.3.1 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. Somewhat analogously, GenomeScan uses only a few sites from each

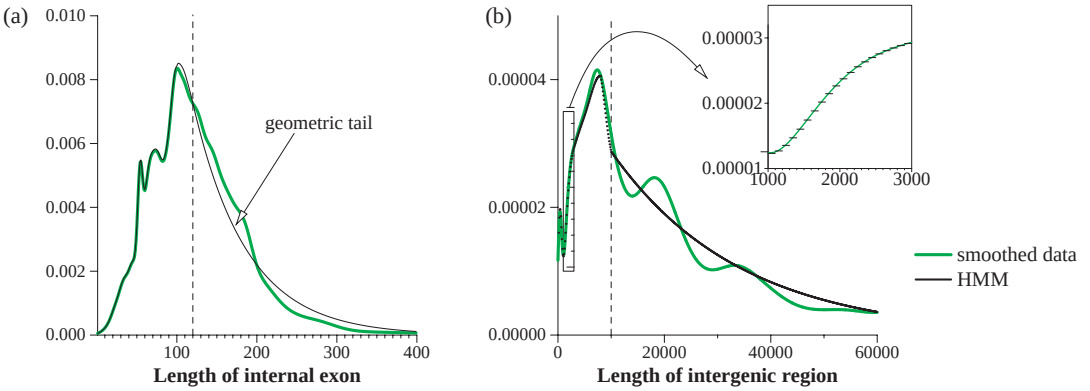

Fig. 1. *Length distributions.* (**a**) Length distribution of low GC content internal exons from the AUGUSTUS training set. The approximation used by the HMM is a combination of an arbitrary head distribution and a geometric tail. (**b**) Intergenic regions from chromosome 22 with low GC content. The inset shows part of the distribution in greater detail, highlighting the step-function nature of the approximation we use.

alignment. 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 $\max_j |\log(x_j/\text{prior}(j))|$. That is, we choose positions for the superadvisor that give a large change compared with the prior.

2.4 Extended HMM for gene structures

We have used a generalized HMM similar to that of GENSCAN (Burge and Karlin, 1997) or AUGUSTUS (Stanke and Waack, 2003) to model the 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.

2.4.1 GC content Model transition and emission probabilities depend on GC content level, estimated from a 1000 bp window around the 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.

2.4.2 Signal models We use higher order trees (HOT) models (Brejova *et al.*, 2003) of order 2 to model donor and acceptor site signals. HOT models capture significant non-adjacent intrasignal dependencies. Compared with 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.

2.4.3 Length distributions We model length distributions of exons and introns with a technique developed by Brejova and Vinar (2002). The length distribution is decomposed into two parts: a head with arbitrary distribution and a geometrically decaying tail (Fig.1). This decomposition allows use of a modified Viterbi algorithm running 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 intron length distributions; 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 produce only lengths that are multiples of k, in $O(nd/k)$ time with $O(n)$ preprocessing time. The submodel for an 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 a submodel requires $O(n(d/k+k))$ inference running time. Setting $d \approx 10\,000$, $k \approx 100$, we can now model non-geometrically distributed intergenic lengths in a practical running time (Fig.1). This was not possible in GENSCAN or AUGUSTUS.

3 CONSTRUCTION OF ADVISORS

The previous section describes a general framework for including 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 the future, we will include 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 the 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 vary with the source of the information and are described below.

The advisor will, for each position, give either vacuous advice, if it is not covered by any interval, or otherwise use a fixed binary partition $\pi = (X, \Sigma - X)$ for some subset X of all labels. For positions within an interval, we estimate the probability p_X that the true label is in set X, depending 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 because, 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 dataset. 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 prespecified number of buckets (in most cases, we have chosen five) by a simple dynamic programming algorithm to minimize the entropy in the bucket partitions similar to Fulton *et al*. (1995).

3.1.1 Proteins We used 11 072 human, 7778 mouse, 1085 chicken and 2047 fruit fly proteins from SwissProt Release 44 (July 2004). Each species yields a separate set of advisors.

We use BLASTX (Gish and States, 1993) 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 two codons from each side to further avoid non-coding regions. The first advisor uses these alignments to predict coding regions and the reading frame. Each alignment is turned into an interval with score corresponding to the BLOSUM62 score per position. The second advisor uses alignments from adjacent protein regions to non-adjacent genome regions to predict introns in the gaps between the alignments in the genomic sequence.

The third advisor predicts the start codon label if the alignment includes a protein's start codon. It also predicts the intergenic label at positions $-100 \ldots -1$. The fourth advisor predicts stop codons analogously.

3.1.2 Expressed sequence tags We used the TIGR human gene index (release 13, October 2003, consisting of 843 769 ESTs) and TIGR mouse gene index (release 12, October 2003, with 669 402 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 (Ma *et al*., 2002), with a seed enhancing homologous coding region sensitivity (Brejova *et al*., 2004). We realign ESTs with significant alignment using SIM4 (Florea *et al*., 1998), producing intervals of presumed exons and introns (gaps between neighbouring alignments from the same EST). The score is the percentage identity of SIM4 alignments.

ESTs often include untranslated regions, which are hard to separate from the coding parts of the ESTs. We experimented with ESTScan (Iseli *et al*., 1999) 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 from human to 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 the prior probability should be made for some position, that prediction is removed.

3.1.3 Genome to genome comparison We included two advisors based on genome–genome comparison: one for *Drosophila* (release 3 from fruitfly.org) and one for 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 six frames with the BLOSUM62 matrix, chose the best frame and located the highest-scoring segment after removing frameshifts (gaps that are not multiples of three). To avoid alignments in non-coding regions, we removed seven 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 the mouse genome, about one-third of the frame predictions by BLOSUM62 scoring were wrong on the training set; therefore, the advisor predicts 'exons' without specifying the 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.

3.2 Repeats

Gene finding programs usually either mask the original sequence for repeats or ignore repeats altogether. Instead,

Table 1. Comparison on the ROSETTA set (%)

	GENSCAN	ROSETTA	SLAM	TWINSCAN	TWINSCAN.p	SGP-1	EH	EH-nh
Gene Sn	44	—	—	—	—	—	74	68
Gene Sp	41	—	—	—	—	—	66	62
Exon Sn	82	83	78	84	86	70	91	89
Exon Sp	73	83	76	77	82	76	83	81
Nucleotide Sn	98	94	95	98	96	94	99	99
Nuclrotide Sp	88	98	98	89	94	96	93	93

The results for ROSETTA, SLAM, TWINSCAN, TWINSCAN.p (alignments from known orthologs only) and SGP-1 are from Alexanderson *et al*. (2003) (the authors did not report gene statistics). The EH column gives the results achieved by ExonHunter with all advisors. The EH-nh column corresponds to ExonHunter results without advisors originating in human datasets. We used standard definitions of sensitivity (Sn) and specificity (Sp) [e.g. Alexanderson *et al*. (2003)]. We evaluated the data with the eval program by Keibler and Brent (2003).

we base an advisor on a list of likely repeats produced by RepeatMasker (Smit *et al*., 2002, http://www.repeatmasker.org). 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. Satellites 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*. (2000). This data set was recently reused by Alexanderson *et al*. (2003) 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). We used half of the chromosome as a supplementary training set and the other half as a testing set.

We trained the HMM on a training set of 1284 human single-gene sequences created by Stanke and Waack (2003). We removed 81 sequences from this set owing to significant similarities 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 with other programs. Table 1 shows the comparison of ExonHunter with other gene finding programs evaluated by Alexanderson *et al*. (2003) on the ROSETTA dataset. ExonHunter used advisors based on human and mouse ESTs, human, mouse and chicken protein alignments, and mouse and *Drosophila* genome–genome comparison. On this dataset, we have outperformed all other tested programs at both exon and nucleotide levels, except for nucleotide specificity. At the gene level, our program identifies more than two-thirds of genes in the dataset completely correctly.

One could object to this test because 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. 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 with those of GENSCAN. Here, the general trend is similar to the observations for the other similarity-based gene finders by Parra *et al*. (2003) and Chatterji and Pachter (2004): the sensitivity stays roughly the same, but the number of exons and coding nucleotides predicted decreases significantly. The numbers from our experiments (data not shown) do not directly compare with those of Parra *et al*. (2003) or Chatterji and Pachter (2004) since we used different subsets of chromosome 22 and different version of the annotation.

Comparison with GenomeScan. To compare ExonHunter with GenomeScan (Yeh *et al*., 2001), we submitted the ROSETTA set to the GenomeScan web server, together with the protein sequences used by ExonHunter. In general, GenomeScan offered higher specificity at both exon level (sensitivity 91%, specificity 86%) and gene level (sensitivity 76%, specificity 74%). However, we note that the training set for GENSCAN (the HMM underlying GenomeScan) contains sequences with high similarity to 56 sequences in the ROSETTA testing set (about 48%). Our experience suggests that such a large overlap may artificially increase prediction accuracy as a result of overfitting. It is not feasible to exclude these sequences from the ROSETTA set, since predictions of other gene finders are not available for the smaller set, and such a small sample size would make statistical comparison of results much less possible.

Contribution of individual advisors. Table 2 shows that the most influence on the final result comes from the protein-based advisors, followed closely by the combination of EST advisors. Mouse ESTs work significantly better than human ESTs, most probably because of low conservation in untranslated regions between human and

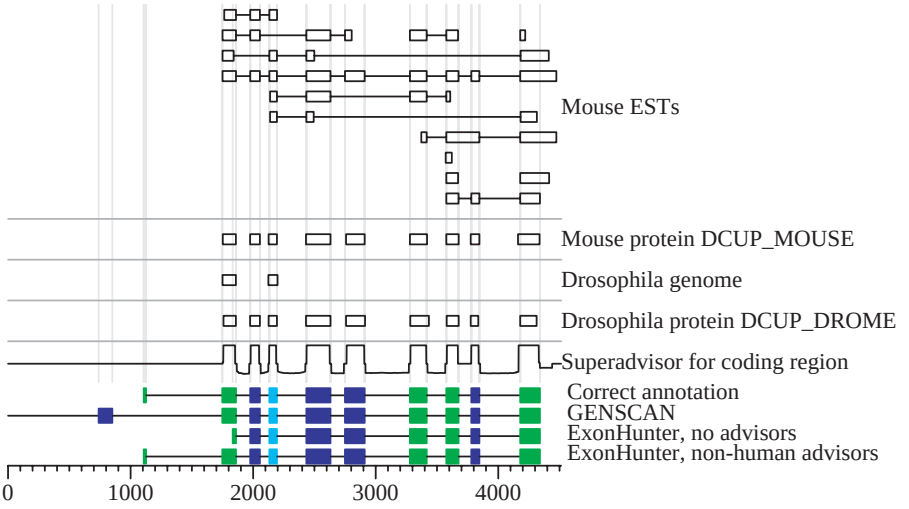

Fig. 2. *Predictions on human gene HSU30787.* Similarity matches processed by *Drosophila* and mouse advisors help to find the correct start site, even though it is not directly covered by any of the matches.

Table 2. Contribution of various combinations of advisors on the ROSETTA dataset (%)

Advisors used	Exon Sn	Exon Sp
Repeats only	77	74
Genomes: D	77	74
Genomes: M	80	74
ESTs: H	79	77
ESTs: M	85	79
ESTs: HM	85	80
Proteins: HMDC	88	82
Genomes: D; proteins: DC	80	76
ESTs: H; proteins: H	90	84
ESTs: M; proteins: M; genomes: M	89	81
All advisors	91	83

H—human; M—mouse; C—chicken; D—*Drosophila*. Proteins and ESTs alone contribute comparable amounts of information. The combination of all advisors performs better than advisors individually.

mouse. The contributions of *Drosophila* and chicken together appear comparable to the contribution of human ESTs. Finally, the combination of all advisors performs better than each of the advisors alone.

Cooperation of advisors with the HMM. Figure 2 shows the ExonHunter annotation on human gene HSU30787. Both GENSCAN and ExonHunter without advisors predict most splice sites correctly, but both annotations miss the first exon completely: GENSCAN extends the gene into the intergenic region, and ExonHunter starts 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 not covered by any alignment. 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 an 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, 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 a traditional linear combination approach, and adapted the Viterbi algorithm to our domain. TWINSCAN's approach to incorporating human–mouse comparison can 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 and ROSETTA, even if all supplementary information originating from human-based advisors is withdrawn. We also evaluated the contribution of individual advisors, finding that protein and EST databases are the two largest contributors toward the final result. However, no one source performs better alone than all in combination.

Although our method allows the incorporation of a wide range of information sources, we do not require all sources to be available. When no additional information is available, the system performs as a typical *ab initio* gene finder, and

adding more information helps to improve the prediction accuracy. This makes our system applicable to a wide range of datasets.

Our approach is becoming ever more relevant as more EST sequence collections for organisms related to humans are built (TIGR currently maintains libraries for eight such organisms). We implicitly allow for variability in handling informant species with varying evolutionary distance from 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 the 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 toward fully *in silico* analysis of newly sequenced genomes and their proteins.

ACKNOWLEDGEMENTS

This work was supported in part by the Natural Science and Engineering Research Council of Canada (NSERC), Canada Council Research Chair Program, the Killam Fellowship and the Human Frontier Science Program.

REFERENCES

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

Allen,J.E., Pertea,M. and Salzberg,S.L. (2004) Computational gene prediction using multiple sources of evidence. *Genome Res.*, **14**, 142–148.

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

Brejova,B., Brown,D.G. and Vinar,T. (2003) Optimal DNA signal recognition models with a fixed amount of intrasignal dependency. In *Algorithms and Bioinformatics: 3rd International Workshop (WABI)*, vol. 2812 of *LNBI*, pp. 78–94.

Brejova,B., Brown,D.G. and Vinar,T. (2004) Optimal spaced seeds for homologous coding regions. *J. Bioinform. Comput. Biol.*, **1**, 595–610.

Brejova,B. and Vinar,T. (2002) A better method for length distribution modeling in HMMs and its application to gene finding. In *Combinatorial Pattern Matching (CPM)*, vol. 2373 of *LNCS*, pp. 190–202.

Brent,M.R. and Guigo,R. (2004) Recent advances in gene structure prediction. *Curr. Opin. Struct. Biol.*, **14**, 264–272.

Burge,C. and Karlin,S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, **268**, 78–94.

Chatterji,S. and Pachter,L. (2004) Multiple organism gene finding by collapsed Gibbs sampling. In *International Conference on Computational Molecular Biology (RECOMB)*, pp. 187–193.

Fletcher,R. (1987) *Practical Methods of Optimization*. Wiley, Chichester, UK.

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

Fulton,T., Kasif,S. and Salzberg,S. (1995) Efficient algorithms for finding multi-way splits for decision trees. In *International Conference on Machine Learning (ICML)*, pp. 244–251.

Gish,W. and States,D.J. (1993) Identification of protein coding regions by database similarity search. *Nat. Genet.*, **3**, 266–272.

Iseli,C., Jongeneel,C.V. and Bucher,P. (1999) ESTScan: a program for detecting, evaluating, and reconstructing potential coding regions in EST sequences. In *International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pp. 138–148.

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

Korf,I., Flicek,P., Duan,D. and Brent,M.R. (2001) Integrating genomic homology into gene structure prediction. *Bioinformatics*, **17** (Suppl. 1), S140–S148.

Krogh,A. (2000) Using database matches with for HMMGene for automated gene detection in *Drosophila*. *Genome Res.*, **10**, 523–528.

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

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

Meyer,I.M. and Durbin,R. (2004) Gene structure conservation aids similarity based gene prediction. *Nucleic Acids Res.*, **32**, 776–783.

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

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

Schiex,T., Moisan,A. and Rouzé,P. (2000) EUGÈNE: an eukaryotic gene finder that combines several sources of evidence. In *Computational Biology. Selected papers from First International Conference on Biology, Informatics, and Mathematics*, vol. 2066 of *LNCS*, pp. 111–125.

Shafer,G.A. (1976) *Mathematical theory of evidence*. Princeton University Press, Princeton, NJ.

Siepel,A. and Haussler,D. (2004) Computational identification of evolutionarily conserved exons. In *International Conference on Computational Molecular Biology (RECOMB)*, pp. 177–186.

Smit,A.F.A., Hubley,R. and Green,P. (2002). RepeatMasker, www.repeatmasker.org.

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

Tax,D.M.J., van Breukelen,M., Duin,R.P.W. and Kittler,J. (2000) Combining multiple classifiers by averaging or multiplying? *Pattern Recogn.*, **33**, 1475–1485.

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

Xu,Y. and Uberbacher,E.C. (1997) Automated gene identification in large-scale genomic sequences. *J. Comput. Biol.*, **4**, 325–328.

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