The Conserved Exon Method for Gene Finding

Vineet Bafna **Daniel H. Huson**
Informatics Research
Celera Genomics Corp.
45 W. Gude Drive, Rockville, MD 20850
Tel.: 240-453-3350, Fax: 240-453-3324
e-mail: {Vineet.Bafna,Daniel.Huson}@celera.com

Abstract

A new approach to gene finding is introduced called the "Conserved Exon Method" (CEM). It is based on the idea of looking for conserved protein sequences by comparing pairs of DNA sequences, identifying putative exon pairs based on conserved regions and splice junction signals then chaining pairs of putative exons together. It simultaneously predicts gene structures in both human and mouse genomic sequences (or in other pairs of sequences at the appropriate evolutionary distance). Experimental results indicate the potential usefulness of this approach.

Keywords Gene finding, comparative genomics, dynamic programming

Introduction

The public and private efforts (Marshall 1999b; 1999a) to sequence the human genome promise to deliver at least a "rough draft" of the human genome later this year. Of all its potential benefits in health and medicine, arguably the most important one will be the ability to rapidly identify all human genes. Because of the size and complexity of the gene-finding problem, initial steps will undoubtedly be computational. However, sequencing the human genome is only the beginning. As an increasing number of eukaryotic genomes are sequenced, the accuracy of computational gene prediction becomes increasingly important.

Following convention, we define gene finding as identification of protein coding regions in genomic DNA. A number of computer programs have been designed for computational gene finding with varying levels of success. Broadly, they can be categorized into two classes. The *ab initio* methods include, but are not limited to GenScan (Burge & Karlin 1997), testcode (Fickett 1982), GeneID (Guigó *et al.* 1992), GENIE (Kulp *et al.* 1996; 1997) and FGENEH (Solovyev, Salamov, & Lawrence 1994). For a review of different methods, see e.g. (Burge 1997; Haussler 1998). Also see (Burset & Guigó 1996) for extensive tests of the accuracy of some of the earlier programs. Essentially, all of these programs look for translational and transcriptional features on the genome, such as coding regions, splice junctions, translation initiation or termination signals. These signals are then 'combined' to get predictions of gene structures. Much of the success of the newer methods, exemplified by GenScan, comes from the description of a gene structure through a probabilistic model (such as an "HMM"), and careful training of the model parameters.

The second class of gene finding programs, notably Procrustes (Gelfand, Mironov, & Pevzner 1996), and GeneWise (Birney 1999; Birney & Durbin 1997) rely on sequence similarity with known proteins to predict gene models explicitly. These homology based programs promise great accuracy albeit at significant expense of computational time. However, recent studies (Guigó *et al.* 1999) show that that even with the newer *ab initio* and homology based methods, the problem is far from being solved. The performance of GenScan drops when considering genomic regions with multiple genes, as compared to regions with single genes. Homology based programs work well when comparing against close homologs but their performance deteriorates with decreasing similarity.

The success of homology based methods may be understood by referring to studies in comparative genomics that compare humans against other species, e.g. (Hardison, Oeltjen, & Miller 1997). Coding regions are generally well conserved in species as far back as 450 Myrs. At evolutionary distances around 50 – 100 Myrs (human and mouse), the conservation also extends to other functional regions important for gene expression, maintaining genome structure, and so on. Two recent studies of human and mouse orthologous transcripts (Makalowski, Zhang, & Boguski 1996; Makalowski & Boguski 1999) show that the average level of identity at the amino acid level is greater than 85%. Given these facts, ideally one would like to be able to predict human gene structures using a homol-

ogy based algorithm comparing against mammalian orthologous protein sequences. Indeed, scientists often pull out the human and rodent transcripts of a gene at the same time. This makes it hard to find *novel* human genes using known rodent orthologs.

In this study, we describe a new gene-finding approach that simultaneously predicts complete gene structures in both human and mouse genomic sequences, which we call *Conserved Exon Method (*CEM*)*. It is based on the idea of looking for conserved protein sequences by comparing pairs of DNA sequences, identifying putative exon pairs based on sequence conservation, and splice site signals and then chaining pairs of putative exons together.

The first part of this approach is not new. For example, the TBLASTX program from the BLAST suite (Altschul *et al.* 1990; Gish & States 1993) performs precisely this task. A customized set of tools is also available for comparing two genomic sequences, and has been used to find coding exons and regulatory regions in human and mouse genomic sequences (Ansari-lari *et al.* 1998; Jang *et al.* 1999; Oeltjen *et al.* 1997). However, while these methods are useful in finding exons, they do not attempt to predict complete gene structures. Independent approaches to gene prediction in orthologous genomic sequences are considered in (Guigó 1999), and (Blayo, Rouzé, & Sagot 1999).

We describe the Conserved Exon Method in the following section. The section on Test Data and Methodology contains a brief description of a set of orthologous gene pairs of human and mouse (Ansari-lari *et al.* 1998; Jang *et al.* 1999; Oeltjen *et al.* 1997; Jareborg, Birney, & Durbin 1999) that serve as a useful test data set for this method. The statistics for testing are also described in that section. Finally, we describe the test results, and discuss future research in the Results section.

The Conserved Exon Method

The problem we solve is described as follows: given two genomic sequences with orthologous genes, predict the gene structures in both sequences. It is possible to model this purely as a sequence alignment problem. However, our implementation of the dynamic programming approach was resource intensive, and not suitable for a high-throughput gene annotation pipeline. For completeness, and because we use the full dynamic program in a limited context in our final algorithm, we describe some details of that approach in the subsection titled The Full Dynamic Program. In the following, we describe the Conserved Exon Method.

Figure 1 shows a comparison of two genomic regions

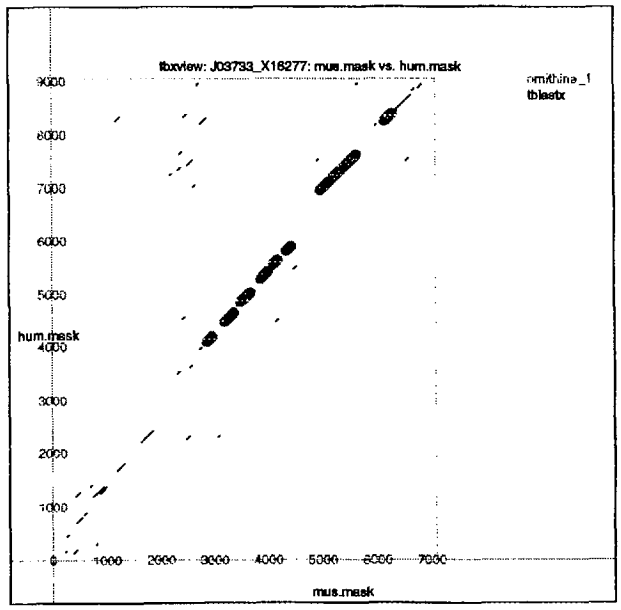

Figure 1: The horizontal axis represents 7100 bps of mouse DNA and the vertical axis represents 9043 bps of human DNA (Genbank accession numbers: J03733 and X16277). Thin black diagonal lines represent HSPs computed by TBLASTX (version 1.4.6) on the +/+ strands. Thick gray diagonal lies represent the exons of ornithine-1. RepeatMasker (Smit & Green 1995) was used to mask repeats in both sequences and only HSPs with score 17 or more are shown. This plot was produced by the author's own software.

(with Genbank accessions J03733 and X16277) using TBLASTX. The heavy grey diagonal lines describe matching exons of the orthologous gene ornithine-1. It is possible that the additional conserved regions could be just random matches, repeats, or other conserved functional sequence. This example reinforces the idea that one could look for coding and regulatory regions by comparing orthologous mammalian genomes for conserved regions. At the same time, it illustrates the difficulty of selecting true exons from among the conserved HSPs.

TBLASTX (or any similar local alignment algorithm) efficiently predicts putative conserved coding regions, but not actual splice boundaries. An obvious approach might be to restrict dynamic programming to a band around the local alignment hits. However, spurious hits can make the band quite large. Also, the dynamic programming calculation in each cell is much more expensive if we have to consider the possibility of introns. We work around the problem by extending (or shrinking) the local alignments to match putative splice junctions. Then, we construct gene models by

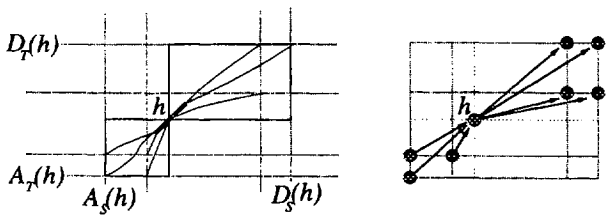

Figure 2: (a) From the mid-point $m(h)$ of each well-supported HSP h we run two dynamic programs: one starting at $m(h)$ and ending at $(D_S(h), D_T(s))$ and the other (in reverse direction) starting at $m(h)$ and ending at $(A_S(h), A_T(h))$. We maintain the best alignments from $m(h)$ to each of the splice-site pairs within the box defined by $(A_S(h), A_T(h))$ and $(D_s(h), D_T(s))$. (b) The splice sites of a pair of putative exons that intersect $m(h)$ give rise to nodes in the CEP-graph associated with h, if the corresponding alignment scores sufficiently well.

chaining together the extended local alignments in a consistent fashion. Before we describe the details of our method, we introduce some definitions:

1. An *HSP* h is a high-scoring local alignment (using BLAST terminology).
2. A *putative-exon* of a DNA sequence S is a sub-string of S flanked by a translation initiation or acceptor site at the 5' end, and a donor site or STOP codon at the 3' end.
3. An HSP h *intersects* an exon E in sequence S, or T, if the projection $m_S(h)$, or $m_T(h)$, respectively, of its mid-point $m(h)$ lies within the exon boundaries.
4. The *score* of a putative exon E in frame f, $\text{Score}_f(E)$ is the sum of scores of all HSPs with frame f that intersect with the putative exon.
5. $\text{Score}(E)$ is the maximum score of E in any frame. $\text{Ratio}(E) = \text{Score}(E)/\text{length}(E)$.
6. A *candidate-exon* is a putative-exon with length and Ratio within certain bounds.
7. Let G_1 be the transcript (spliced-message) of the gene containing exon E_1, and G_2 be the transcript of the gene containing exon E_2. (E_1, E_2) is a *conserved-exon-pair* if in the global alignment of G_1 with G_2, a portion of E_1 is aligned to a portion of E_2.
8. A conserved-exon-pair has *conserved splice junctions* if the splice boundaries of E_1 matches the boundaries of E_2 in the global alignment. For example, in Figure 3, (E_1, E_2), (E_3, E_4), (E_4, E_5), and (E_5, E_6) are orthologous-exon-pairs, and (E_1, E_2) have conserved splice junctions.

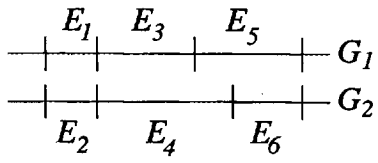

Figure 3: Conserved Exon pairs

9. For every HSP h, let $C_S(h)$ and $C_T(h)$ denote the list of candidate exons in sequences S and T, respectively, that intersect with h.
10. For every HSP h, let $\text{Range}_S(h) = (A_S(h), D_S(h))$, where $A_S(h)$ is the coordinate of 5'-most point of a candidate exon in $C_S(h)$, and $D_S(h)$ is the coordinate of the 3'-most point of a candidate-exon in $C_S(h)$.
11. The score of the alignment between G_1, and G_2 is the sum of the alignment scores of orthologous-exon-pairs minus the penalties for introns in S, and in T.

Putative Conserved-Exon-Pairs (CEPs)

To begin with, assume that true conserved-exon-pairs also have conserved splice junctions. Our algorithm proceeds roughly by identifying putative conserved-exon-pairs (CEPs), and pre-computing their alignments. By definition, each CEP should have a strong locally aligned region. It follows that each conserved exon-pair should be in the two candidate lists for some HSP h. Consider an arbitrary HSP h. See for example, Figure (a). Any pair of exons (E_1, E_2), where $E_1 \in C_S(h)$, and $E_2 \in C_T(h)$ is putatively a conserved-exon-pair. Pre-computing an alignment of each exon pair in $C_S(h) \times C_T(h)$ could be quite expensive. Also, we need to do the alignment at the protein level in all frames.

A key assumption that significantly simplifies our task is this: If an HSP h is the evidence for E_1 and E_2 being possible conserved-exon-pairs, then h should be a *portion* of the correct local alignment of E_1 and E_2. This allows us to get all the desired alignments in *two* local alignment computations per HSP. See Figure (a) and Procedure 1. By definition (see items 3 and 10 above), all donor sites of exons in $C_S(h)$ should be between $m_S(h)$ and $D_S(h)$, and all donor sites of exons in $C_T(h)$ are between $m_T(h)$ and $D_T(h)$. Correspondingly, all acceptor sites of $C_S(h)$ (respectively, $C_T(h)$) should be in the region $[A_S(h), m_S(h) - 1]$ (respectively, $[A_T(h), m_T(h) - 1]$).

The two alignment problems that we solve are the following: In the first, we do a computation over the

region defined by $[m_S(h), D_S(h)] \times [m_T(h), D_T(h)]$. In this single computation, we maintain the scores of the best alignments from the mid-point to all the putative donor site pairs. The second computation is over $[A_S(h), m_S(h) - 1] \times [A_T(h), m_T(h) - 1]$, and we maintain the scores of the best alignments from $(m_S(h) - 1, m_T(h) - 1)$ to all the putative acceptor-site pairs.

Now consider an arbitrary exon-pair $(E_1, E_2) \in C_S(h) \times C_T(h)$. The alignment of E_1 against E_2 has the following property: It must start at a point in the bounding box defined by $[A_S(h), m_S(h) - 1] \times [A_T(h), m_T(h) - 1]$, pass through the mid-point $(m_S(h), m_T(h))$ of h in the same frame as h, and end at a point in $[m_S(h), D_S(h)] \times [m_T(h), D_T(h)]$. The score of the alignment of (E_1, E_2) is simply the sum of the scores of the alignments from the appropriate acceptor-site pair to the mid-point of h, and from the mid-point of h to the donor-site pair. Thus, the two computations are sufficient.

The actual algorithms we use for local alignment computations are variants of standard algorithms and omitted. We do mention some subtleties. First, the alignments are forced to start in the frame defined by the HSP. Frameshifts are allowed subsequently (with an appropriate indel penalty). Each splice-junction pair is a cell in the dynamic programming matrix, and its score is maintained in a separate list. Let (i, j) be the coordinates of a cell corresponding to a splice-junction-pair $(a_S(h), a_T(h))$. The score assigned to $(a_S(h), a_T(h))$ is not Score$[i, j]$, but

$$\text{Score}(a_S(h), a_T(h)) = \max_{0 \le k_S(h), k_T(h) \le 2} \{\text{Score}[i - k_S(h)][j - k_T(h)]\}$$

This is to allow for the possibility of an intron splitting a codon. In this way, the alignment (which only scores codons) allows terminal nucleotide gaps without incurring a frameshift penalty. The amount of overhang

$$(o_S(h), o_T(h)) = \arg \max_{0 \le k_S(h), k_T(h) \le 2} \{\text{Score}[i - k_S(h)][j - k_T(h)]\}$$

is also stored along with the score. Finally, as the alignment is done at the protein level, there is a direction associated with it. The dynamic programming computation from the mid-point to the acceptor splice junctions is done by reversing each codon before scoring.

The results of these two dynamic programming computations are lists giving scores of the alignments from the mid-point to the appropriate splice-junctions. We use these two lists to build an associated *CEP-graph*. This procedure is summarized below:

Procedure 1 ComputeCEP (HSP h)

begin
  Let $(m_S(h), m_T(h))$ = *mid-point*(h)
  Let $(A_S(h), D_S(h))$ = *Range*$_S(h)$
  Let $(A_T(h), D_T(h))$ = *Range*$_T(h)$
  DonorScoreList
    =*ComputeDP*$(m_S(h), D_S(h), m_T(h), D_T(h))$
  AcceptorScoreList
    =*ComputeReverseDP*$(A_S(h), m_S(h) - 1, A_T(h), m_T(h) - 1)$
  buildCEPGraph(DonorScoreList,AcceptorScoreList)
end

Each node u in the CEP-graph corresponds to a coordinate pair (i, j), which is the start point, mid-point or terminating point of a candidate exon pair, see Figure (b). More precisely, u is one of the following:

- a *center* node, if (i, j) is the mid-point of h,
- a *donor* node if i and j are sites of donor splice signals in S, and T,
- an *acceptor* node if i and j are sites of acceptor signals,
- a *start* node if i and j are sites of translation initiation signals, or
- a *terminal* node if i and j are sites for a STOP codon.

Each node u has some additional information associated with it. The coordinates of the cell are maintained as (u_S, u_T). For each acceptor or donor node u, we maintain information on the nucleotide overhang at the boundary as overhang$(u) = (o_S(u), o_T(u))$. The edges in a CEP-graph come in two flavors: there is a directed edge from each acceptor or start node to the center, and from the center to each donor or terminal node. The weight of the edge is the score of the corresponding local alignment.

Constructing the CEM Graph

Each CEP-graph is a concise representation of alignments of pairs of exons. At most one pair can actually be a conserved-exon-pair in the true gene structures. The Conserved-Exon-Method takes the CEP-graphs of HSPs and chains them together, thus obtaining the full "CEM-graph". It builds gene models from this graph based on the assumption that the transcripts derived from correct orthologous gene structures will have the highest alignment score. Note that our treatment of the CEP-graph construction assumed that the conserved-exon-pairs also had conserved splice boundaries. We will maintain that assumption here, deferring the general case to the following section.

Let S and T be the two genomic sequences. First, we get a list of candidate exons in each sequence based on the scores of intersecting HSPs. In order to select candidate exons, we need signal sensors for splice-junctions, and translation initiation sites. Much research has gone into improving the signal sensors by modeling position specific nucleotide distributions (Staden 1984), correlations between positions (Agarwal & Bafna 1998; Burge & Karlin 1997; Salzberg 1997; Zhang & Marr 1993), and higher order effects using machine learning techniques (Pedersen & Nielsen 1997). In this paper, we make minimum use of signals. This is done first to study the power of homology in detecting true exons, and secondly to avoid over-training on the limited test data. We use WMM models (Salzberg 1997) for translation initiation, start and stop signals. However, the scores of these sensors are not explicitly used to build gene models. Instead all splice sites and initiation signals that score over a certain threshold are chosen to select candidate exons. Once the candidate exons are chosen, all splice junctions are considered equally strong. It is likely that a careful incorporation of splice-site scores will improve the power of predictions.

For each HSP h, compute the CEP-graph as described in the previous section. Next, we build a candidate exon graph $G = (V, E)$ (which we call the *CEM-graph*), as follows: V is the union of all the nodes in the CEP-graphs, and E contains all the edges in each CEP-graph. Further, add an edge from donor or start node u to an acceptor or terminal node v if both:

- $v_S >= u_S + M$, and $v_T >= u_T + M$, where M is a suitably chosen *minimum intron length*, and:
- Let $(o_S(u), o_T(u))$ = overhang(u), and $(o_S(v), o_T(v))$ = overhang(v). Then, $(o_S(u)+o_S(v)) = 0(\text{mod } 3)$, and $(o_T(u) + o_T(v)) = 0(\text{mod } 3)$,

The weight of the edge (u, v) is the score of aligning the amino-acids obtained by concatenating the overhangs on either side added to the penalty for an intron gap.

We summarize the method as follows:

Algorithm 1 (Conserved Exon Method)

1. *Determine a list of candidate exons for S, and T.*
2. *For every HSP h, determine the sets of intersecting candidate exons, $C_S(h)$, $C_T(h)$, and the ranges $Range_S(h)$, and $Range_T(h)$ of the candidate exons.*
3. *Sort HSPs lexicographically according to their ranges.*
4. *For each HSP h, call* BuildCEPGraph(h).
5. G = BuildCEMGraph().
6. getGeneModelScores(G).
7. BuildGeneModel(G).

We construct a gene model by traversing a high-scoring path through the candidate-exon graph, as described in the next section.

Obtaining a Gene Prediction from the CEM-Graph

By construction, a path in the CEM-graph corresponds to a prediction of orthologous gene structures in the two genomes. Based on the assumption that the correct gene models will have the highest alignment score, we can extract the correct gene structures simply by choosing the highest scoring path. As this is a directed acyclic graph, the highest scoring path can be computed via a topological sort. Procedure 2 describes the computation of score on each node. For an arbitrary node u, score(u) is the best score of an alignment of two genome prefixes $S[1..u_i]$, and $T[1..u_j]$, allowing for frame-shifts, amino-acid indels and intron penalties.

Procedure 2
getGeneModelScores(ConservedExonGraph $G(V, E)$)

```
begin
    OrderedNodeListL = TopologicalSort(G)
    for each v in L
        Initialize(Score(v))
        for each incoming edge e = (x,v)
            if (Score(v) < Score(x) + w(e))
                Score(v) = Score(x) + w(e)
                predecessor(v) = x
end
```

Once the scores on the nodes are computed the gene models are built by starting at the node with the highest score, and following the predecessors. The coordinates of start, terminal, donor and acceptor nodes reveal the gene structure in the two genomic sequences. As the boundaries of the path are not limited to start and terminal nodes, partial gene structures can be predicted.

Implementation Details

We have implemented the Conserved Exon Method in C++ using LEDA (Näher & Mehlhorn 1995). Our implementation takes as input (masked) DNA sequences of two species, HSPs computed by a program such as TBLASTX, and WMM matrices describing splice-site signals. For each HSP h in the input, using the given splice-site matrices, it computes a CEP-graph for h.

It then computes edges between different CEP-graphs as described above, and provides a number of different algorithms for computing a high scoring path through the constructed graph. The results can be interactively viewed and queried. Additionally, the program provides an interactive interface to external alignment programs and allows the exploration of the given data in any local region.

The run time of the algorithm is currently in the order of minutes for sequences of length 10 000 on a Pentium II laptop running Linux. This is slow compared to GenScan. The performance of our method is handicapped by the fact that the computation of the initial HSPs (using e.g. TBLASTX) and then the determination of possible splice-sites and the CEP-graphs occur separately. Doing both computations simultaneously should greatly improve the performance.

Extensions

It is reasonable to assume that in a large number of cases, orthologous mammalian genes also have conserved splice boundaries. The situation is slightly complicated if that is not the case. See Figure 3. Exons E_3, and E_4 overlap, but do not have conserved splice sites. We have two cases:

In the first case, the splice boundaries of E_3 and E_4 are close enough that the overlap between E_4 and E_5 is not significant. Then, in the CEP-graph aligning E_3, and E_4, the correct alignment should have a terminal gap (Note that amino-acid gaps are considered evolutionary changes and not penalized heavily). Let u be the donor node that represents the alignment of the donor sites of E_3 and E_4. Likewise, let v be the acceptor-node that represents the acceptor sites of E_5 and E_6. In computing the weight of the edge (u, v), we check to see if there are terminal gaps. If so, we call an alignment procedure to 'merge' the two gaps.

In the second case, the splice boundaries of E_3, and E_4 are far apart, and the overlap between E_4, and E_5 is significant. We resolve this through *binning*. Bin together HSPs h_1 and h_2 if they have the same frame in both sequences, and they share a candidate-exon in one sequence. Instead of considering HSPs in step 4 of Algorithm 1 we consider each bin, which has one or more HSPs. Order the HSPs in a bin according to their mid-point. Next, compute a modified CEP-graph for each HSP. A donor site u is connected to HSP h_1 only if there is no subsequent HSP h_2 to which it can be connected.

In addition to the alignment computation for each CEP-graph, we also do an alignment computation connecting the mid-points of adjacent ordered HSPs in the bin. These alignments are qualitatively different in that they allow intron gap penalties. The rest of the graph structure, and path construction can be easily extended to accommodate this structure.

The gene prediction can be extended to multiple genes as well. In the simplest level, one can add edges from a terminal node u to a start node v, as long as $v_j >= u_j + Q$, and $v_i >= u_i + Q$, where Q is an assumed minimum intergenetic length.

Alternative: The Full Dynamic Program

As mentioned in the beginning of the Methods section, we could model this purely as a sequence alignment problem. The objective would be to align the two sequences together so that the conserved exons are aligned. Specifically:

1. Only triplets of nucleotides, corresponding to codons, are aligned. To score matches of conceptual translations of codons, a standard score matrix, such as BLOSUM62 is used.

2. Three kind of gaps are allowed, with differing penalties. Nucleotide indels (due to sequencing errors), in-frame codon indels (evolutionary changes), and introns. Both indels and in-frame gaps are charged affine gap penalties, whereas introns either have a constant, or an appropriate concave gap penalty.

3. Intron gaps are allowed to split codons. The inserted intron must be flanked by appropriate donor and acceptor signals.

Recurrence equations for this problem have been worked out (Blayo, Rouzé, & Sagot 1999). These can be considered as a natural, but non-trivial extension of the algorithms used by homology based gene-finders that produce an alignment of genomic sequences with a homologous protein sequence (GeneWise), or genomic sequence against a transcript sequence (EST_GENOME (Mott 1997), sim4 (Florea *et al.* 1998)).

We derived our own recurrences for the problem, similar to (Blayo, Rouzé, & Sagot 1999), but did not implement a linear-space version. Our implementation of these recurrences are resource intensive in both time and space. Part of the reason is the size of the dynamic programming matrix. Further, the computation in each cell of the matrix is significantly more expensive as one has to consider the possibility of intron gaps. Improving the constants of this recurrence remains an interesting theoretical problem. We omit the details of the recurrences in this version, mentioning only that we use this procedure in a limited context in the main algorithm.

Test Data and Methodology

Until recently, there was no large-scale compilation of genomic data with orthologous gene pairs from different mammalian species. At the cDNA level, there have been two large scale studies comparing human and mouse genes (Makalowski, Zhang, & Boguski 1996; Makalowski & Boguski 1999). There have also been studies with extensive comparison of a few human and mouse genomic regions (Ansari-lari *et al.* 1998; Jang *et al.* 1999; Oeltjen *et al.* 1997; Jareborg, Birney, & Durbin 1999) have compiled a database of human and mouse genomic sequence with 77 orthologous gene pairs. They also map the gene features, including coding, untranslated, and upstream regions on the sequence, and performed a comparative analysis.

No explicit training was required for our approach. However, we selected 27 gene pairs to test the program for errors and to choose gap penalties, and a few other parameters. These were all genomic sequence pairs with single genes on them. For the final test, all the genomic sequence in the data-set were chosen, and split into data-sets, so that each pair contained a single orthologous gene pair. 300 nucleotides were chosen around the coding exon boundaries for the split. Pairs were excluded if the prediction was hypothetical, if the exons had repeat sequences, or if there was an error in annotation. The final test data-set had 60 gene pairs. We call the initial set of 27 sequence pairs, the *training* set, and the final set of 60 sequence pairs, the *test* set.

As CEM simultaneously predicts structure in both genomic sequences, the accuracy statistics are usually the same on the two genes. However, the results may be different if one of the genes has extra exons, or different splice boundaries. Therefore, in the test, we measure accuracy separately on each prediction.

Following standard practice (Burset & Guigó 1996), we measure the accuracy of predictions at both the nucleotide and exon level. As our algorithm is qualitatively different in that it requires two genomes with orthologous gene structures, it is hard to compare the results with other gene finding programs. We decided to compare performance against GenScan, a widely used *ab initio* program.

Exon Structure

At this level, we measure the accuracy of predictions by comparing predicted and true exons along the sequence. The set of correct and predicted exons can be classified into the following four types: Exons predicted correctly (*true positives (TP)*), Exons not predicted (*Missing Exons(ME)*), predicted exons with overlapping true exons (OV), and incorrectly predicted exons (*false positives* (FP)). Following (Burset & Guigó 1996), we consider an overlapping prediction to be incorrectly predicted. The two measures are

$$\text{Specificity(Sp)} = \frac{\text{TP}}{(\text{TP} + \text{FP} + \text{OV})}$$

and

$$\text{Sensitivity(Sn)} = \frac{\text{TP}}{(\text{TP} + \text{OV} + \text{ME})}$$

Nucleotide Level Accuracy

In this measure, we test the predicted coding value (coding or non-coding) against the true coding value for each nucleotide along the test sequence. We have four cases: coding nucleotides predicted correctly (the true positives,TP), non-coding nucleotides predicted correctly (true negatives, TN), coding nucleotides predicted incorrectly (false negatives, FN), and non-coding nucleotides predicted incorrectly (the false positives, FP). Sensitivity, and specificity are defined as follows:

$$\text{Specificity(Sp)} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$

$$\text{Sensitivity(Sn)} = \frac{\text{TP}}{(\text{TP} + \text{FN})}$$

Measuring accuracy at the exon level is complementary to evaluation at the nucleotide level. At the nucleotide level, we are measuring how well the sequence coding regions are located, while at the exon level, we measure how well the splice sites are identified. Following (Burset & Guigó 1996), we compute the specificity and sensitivity separately for each gene, and take the average.

Results and Discussion

Results on the 54 sequences (27 pairs) of the training set, and 120 sequences of the test set are shown in Table 1. CEM improves upon GenScan in all measures. As there was no explicit training of parameters, the results do not change much between training and test data. The results at the nucleotide level are better than the results at the exon level. This can be expected, as there was no explicit incorporation of splice-site scores in predicting the gene model. In many of the wrong predictions, the true and the wrong exon overlap often with one matching splice junction. Of a total of 934 exons, 734 were predicted correctly, 130 overlapped with true exons, and 70 were missing. An additional 48 false exons were predicted. In our strict measure of exon level accuracy, the 130 exons are considered to be both missing and incorrectly predicted. Therefore, the prediction should improve with incorporation of splice site scores. A large number of missing exons are due

	NumS	ExSn	ExSp	NuSn	NuSp
CEM	54	0.78	0.78	0.96	0.92
GenScan	54	0.72	0.70	0.96	0.89
CEM	120	0.76	0.8	0.94	0.95
GenScan	120	0.74	0.78	0.92	0.94

Table 1: Comparison of performance of CEM and GenScan on test data set of 60 pairs of genes. The following abbreviations are used. NumS: number of sequences; ExSn: Exon Sensitivity; ExSp: Exon Specificity; NuSn: Nucleotide Sensitivity; NuSp: Nucleotide Specificity. The specificity and sensitivity are calculated separately for each gene, and averaged.

to missing HSPs. Some of these are low scoring HSPs that could not be detected above noise, and others are missed by TBLASTX. We plan to implement a fast local alignment computation program to obtain HSPs and candidate exons in future versions of CEM.

As with other gene-prediction programs, initial and terminal exons are usually harder to predict accurately, because those exons are usually quite small, and because there is a fair amount of conservation in the 5' and 3' untranslated regions. It is interesting to note that both CEM and GenScan *improve* in exon level accuracy in going from training to test. This is partly explained by the following fact. In the test set, only 300 bp upstream of the first coding exon were chosen to do the analysis. The corresponding reduction in search space led to improved prediction for the first exon. It remains a topic of research to distinguish conserved UTR and upstream regions from the coding exons. This will be key to making accurate predictions in multi-gene sequences.

With *ab initio* programs, one usually has only a score or a P-value to decide if the gene model is correct or not, and subsequent verification steps often involve looking for similar sequences in a protein database. Like other homology based programs, a CEM prediction comes with an alignment of conserved sequences, and it is often possible to eye-ball the alignment to make an educated guess about the accuracy of the prediction. We illustrate this with a few examples: in gene TPI-7 (human genomic sequence described by Genbank accession U47924, and mouse sequence by Genbank accession AC002397); gene Ahsg-1 (D67013,AF007900); and gene H2-IAbeta2_4

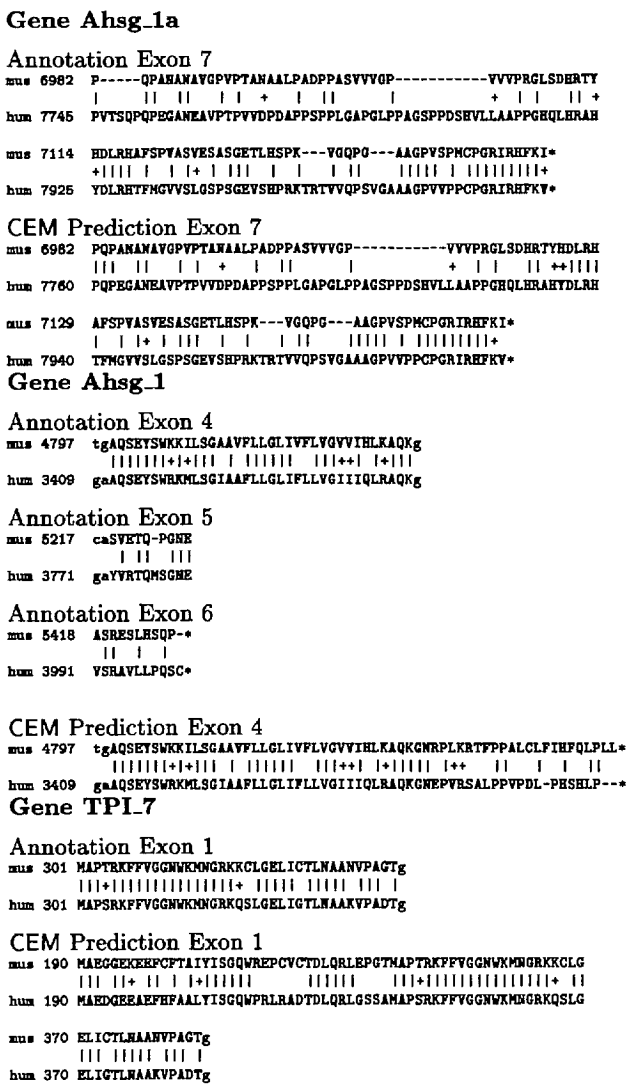

Figure 4: Comparison of annotation and CEM prediction.

(X87344,AF027865). See Figure 4. In each case, the prediction is different from the annotation, but offers a plausible, and possibly a better gene model, suggesting either erroneous annotation or alternative splicing. For the test, CEM predicted the model using the highest scoring path. However, it is possible to extract alternative gene models.

The trend in high-throughput sequencing now is to produce an abundance of draft sequence, and to assemble and finish it at a later stage. As the finishing is expensive, it is quite possible that the sequence for many model organisms (mouse, rat) will not be finished for many years. Gene-finders are thus faced with the double jeopardy of abundance of sequence, and sequence that is only available in small unordered pieces. We conclude by noting that CEM can be easily adapted

to deal with the case when one sequence is contiguous, and the other in unordered pieces, and could well be an important tool for whole genome comparisons in the early stages of the genome projects. We intend to further refine this method and to use it for high throughput annotation as the human and mouse genomic sequences become available.

Acknowledgements

Vineet Bafna would like to thank Roderic Guigó and Pankaj Agarwal for useful discussions at the beginning of this research.

References

Agarwal, P., and Bafna, V. 1998. Detecting non-adjoining correlations within signals in DNA. In *International Conference on Computational Molecular Biology (RECOMB)*, 1–7. ACM press.

Altschul, S.; Gish, W.; Miller, W.; Myers, E.; and Lipman, D. 1990. Basic local alignment search tool. *Journal of Molecular Biology* 215:403–410.

Ansari-lari, M. A.; Oeltjen, J.; Zhang, Z.; Muzny, D.; Liu, J.; Gorrell, J.; Chinault, A.; Belmont, J.; Miller, W.; and Gibbs, R. 1998. Comparative sequence analysis of a gene-rich cluster at human chromosme 12p13 and its syntenic region in mouse chromosome 6. *Genome Research* 8:29–40.

Birney, E., and Durbin, R. 1997. Dynamite: a flexible code generating language for dynamic programming methods used in sequence comparison. *Intelligent Systems in Molecular Biology (ISMB)* 5:56–64.

Birney, E. 1999. *Wise2.* http://www.sanger.ac.uk /Software/Wise2/wisedocs/wise2/wise2.html.

Blayo, P.; Rouzé, P.; and Sagot, M. 1999. Orphan gene finding - an exon assembly approach. Unpublished manuscript.

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

Burge, C. 1997. *Identification of genes in human genomic DNA.* Ph.D. Dissertation, Stanford University, Stanford, CA.

Burset, M., and Guigó, R. 1996. Evaluation of gene structure prediction programs. *Genomics* 34:353–367.

Fickett, J. W. 1982. Recognition of protein coding regions in DNA sequences. *Nucleic Acids Research.*

Florea, L.; Hartell, G.; Zhang, Z.; Rubin, G.; and Miller, W. 1998. A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Research* 8:967–974.

Gelfand, M. S.; Mironov, A.; and Pevzner, P. A. 1996. Gene recognition via spliced alignment. *Proceedings of the National Academy of Sciences* 93:9061–9066.

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

Guigó, R.; Knudsen, S.; Drake, N.; and Smith, T. F. 1992. Prediction of gene structure. *Journal of Molecular Biology* 226:141–157.

Guigó, R.; Agarwal, P.; Abril, J. F.; Burset, M.; and Fickett, J. W. 1999. Gene prediction accuracy in large DNA sequences. Unpublished manuscript.

Guigó, R. 1999. personal communication.

Hardison, R.; Oeltjen, J.; and Miller, W. 1997. Long human-mouse sequence alignments reveal novel regulatory elements: A reason to sequence the mouse genome. *Genome Research* 7:959–966.

Haussler, D. 1998. Computational genefinding. *Trends Guide to Bioinformatics* 12–15.

Jang, W.; Hua, A.; Spilon, S. V.; Miller, W.; and Roe, B. A. 1999. Comparative sequence of human and mouse BAC clones from the mnd2 region of chromosome 2p13. *Genome Research* 9:53–61.

Jareborg, N.; Birney, E.; and Durbin, R. 1999. Comparative analysis of noncoding regions of 77 orthologous mouse and human gene pairs. *Genome Research* 9:815–824.

Kulp, D.; Haussler, D.; Reese, M.; and Eeckman, F. 1996. A generalized hidden markov model for the recognition of human genes in DNA. In States, D.; Agarwal, P.; Gaasterland, T.; Hunter, L.; and Smith, R., eds., *Intelligent Systems in Molecular Biology (ISMB)*, 134–142. AAAI Press.

Kulp, D.; Haussler, D.; Reese, M.; and Eeckman, F. 1997. Integrating database homology in a probabilsitic gene structure mode. In *Proceedings of the Pacific Symposium on Biocomputing.*

Makalowski, W., and Boguski, M. 1999. Evolutionary parameters of the transcribed mammalian genome: An analysis of 2820 orthologous rodent and human sequences. *Proceedings of the National Academy of Sciences* 95:9407–9412.

Makalowski, W.; Zhang, J.; and Boguski, M. 1996. Comparative analysis of 1196 orthologous mouse and human full-length mRNA and protein sequences. *Genome Research* 6:846–857.

Marshall, E. 1999a. A high-stakes gamble on genome sequencing. *Science* 284(5422):1906–1909.

Marshall, E. 1999b. Sequencers endorse plan for draft in 1 year. *Science* 284(5419):1439–1441.

Mott, R. 1997. EST_GENOME: A program to align spliced DNA sequences to unspliced genomic DNA. *Comput. Appl. Biosci.* 13:477–478.

Näher, S., and Mehlhorn, K. 1995. LEDA, a platform for combinatorial and geometric computing. *Communications of the ACM* 38(1):96–102.

Oeltjen, J.; Malley, T.; Muzny, D.; Miller, W.; Gibbs, R.; and Belmont, J. 1997. Large-scale comparative sequence analysis of the human and murine bruton's tyrosine kinase loci reveals conserved regulatory domains. *Genome Research* 7:315–329.

Pedersen, A., and Nielsen, J. 1997. Neural network prediction of translation initiation sites in eukaryotes: perspectives for EST and genome analysis. In *Intelligent Systems in Molecular Biology (ISMB)*, 226–233. AAAI press.

Salzberg, S. 1997. A method for identifying splice sites and translational start sites in eukaryotic mRNA. *Comput. Appl. Biosci.* 13(4):365–376.

Smit, A., and Green, P. 1995. *Repeat Masker.* http://ftp.genome.washington.edu/cgbin/RepeatMasker.

Solovyev, V.; Salamov, A. A.; and Lawrence, C. B. 1994. Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. *Nucleic Acids Research* 22:5156–5163.

Staden, R. 1984. Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Research* 10:141–186.

Zhang, M., and Marr, T. 1993. A weight array method for splicing signal analysis. *Comput. Appl. Biosci.* 9(5):499–509.