ESTScan: a program for detecting, evaluating, and reconstructing potential coding regions in EST sequences

Christian Iseli*†, C. Victor Jongeneel*†, Philipp Bucher*‡

* Swiss Institute of Bioinformatics, † Office of Information Technology, Ludwig Institute for Cancer Research, and ‡ Swiss Institute for Experimental Cancer Research
Chemin des Boveresses 155, 1066 Epalinges, Switzerland
Tel: +41-21-692-5991; FAX +41-21-692-5945
Christian.Iseli, Victor.Jongeneel, Philipp.Bucher@isb-sib.ch

Abstract

One of the problems associated with the large-scale analysis of unannotated, low quality EST sequences is the detection of coding regions and the correction of frameshift errors that they often contain. We introduce a new type of hidden Markov model that explicitly deals with the possibility of errors in the sequence to analyze, and incorporates a method for correcting these errors. This model was implemented in an efficient and robust program, ESTScan. We show that ESTScan can detect and extract coding regions from low-quality sequences with high selectivity and sensitivity, and is able to accurately correct frameshift errors. In the framework of genome sequencing projects, ESTScan could become a very useful tool for gene discovery, for quality control, and for the assembly of contigs representing the coding regions of genes.

Introduction

Background

Many complementary approaches are being used to characterize the genes encoded in the genome of any individual species. While the ultimate goal of most genome sequencing projects is to produce a complete sequence with as low an error rate as possible, it has proven enormously useful to also generate large numbers of single-pass, low fidelity sequences from the expressed portion of the genome. This approach, dubbed Expressed Sequence Tag (EST) sequencing (Adams, Kelley et al. 1991), is of great value in characterizing the transcriptome, in the discovery and assembly of the coding regions of new genes, and in providing unique markers for physical mapping. The widely recognized contribution of ESTs to gene discovery has spurred the production of very large numbers of these sequences, both from the academic and the private sector. There are currently over 1.5×10^6 human EST sequences in the public databases, and ESTs make up more than 60% of all of the database entries. An even larger number are thought to be available from private sources.

Exploitation of the EST data still lags far behind their production. Many biologists use the *blastn* or *tblastn* programs (Altschul, Madden et al. 1997) to find EST sequences that may belong to the same family as a query sequence, usually in the hope of finding new genes that may play a role in their field of experimental interest. Motif-based searches based on hidden Markov models or profiles are performed only in a handful of well-equipped biocomputing groups, and large-scale analyses of the EST databases are still very rare. It is also noteworthy that there are still no publicly available, comprehensive assembled EST contig databases, which would be of great help in gene discovery programs. The Unigene databases produced by the NCBI (http://www.ncbi.nlm.nih.gov/UniGene), which group human, mouse and rat ESTs in clusters likely to be derived from the same gene, are currently the most useful collections in this regard.

For several reasons, the detection and assembly of new coding sequences from ESTs is not a trivial task. First, because most libraries used as a source of ESTs were constructed by oligo(dT)-primed cDNA synthesis, regions derived from the poly(A)-proximal regions of mRNAs, which are overwhelmingly non-coding 3'UTRs, are over-represented. Second, the inherently low quality of EST sequences very often results in errors that impede the proper recognition of coding regions: shifts in the reading frame caused by missing or erroneously inserted bases, stop codons introduced by sequencing errors, and ambiguous bases precluding accurate translation (Ouellette and Boguski 1997). Therefore, standard database comparison programs that rely on an automated translation of target sequences in the six possible frames often miss significant similarities. It thus seemed desirable to develop a tool that could recognize potential coding regions in poor quality sequences, reconstruct these coding regions in their proper reading frame, and discriminate between ESTs with coding potential and those derived from non-coding regions.

Potential applications

Applications of a program capable of reliably detecting coding potential in EST-quality data are many. To cite a few:

1. **Quality assessment of cDNA libraries**. Methods based on approaches other than oligo(dT) priming, e.g. random priming, low-stringency priming followed by PCR, and 3' or 5' RACE, are subject to multiple artifacts, the most common of which is contamination with genomic sequences. The program should be able to discriminate between bona fide ESTs (mostly coding) and genomic contaminants (non-coding).
2. **Gene discovery**. The direct screening of EST databases with sequence or motif-based queries is computationally expensive and prone to artifacts. The program should ideally allow the creation of coding sequence-only nucleotide databases and/or predicted protein databases (with embedded corrections for frameshift errors), making the searches much more efficient and sensitive. A *blastp* search of an EST-derived, frameshift-corrected predicted protein database would be both more sensitive and faster than a *tblastn* search of the full EST database with the same query. The predicted protein databases could themselves provide a source of new sequences without similarity to any known gene families, and thus spur the discovery of new families.
3. **Exon detection in genome survey sequences**. Many current genome projects are generating low-quality genomic "shotgun" sequences similar to ESTs (Venter, Adams et al. 1998). It would be useful to flag potential coding exons in these data, both as an aid to gene mapping and as an alternative gene discovery tool. Current gene prediction programs, which expect high-quality data as input and are thus not error-tolerant, are not well suited to this task.

Algorithms for predicting coding regions

The detection of coding regions is by no means a novel problem (reviewed by Fickett, 1996). As it is a central issue in gene prediction, many techniques have been devised to address it, and more specifically to detect coding exons in genomic sequences. Approaches relying on the detection of similarity to existing database entries (see e.g. Brown, Sander et al. 1998), or on lexical analysis involving splice donor and acceptor signals, are outside the scope of our project. The most powerful independent predictor of coding potential is probably the known associated bias in hexanucleotide composition, imposed by species-dependent codon usage biases and amino acid composition inhomogeneities. This hexanucleotide bias, which is used as a component in many gene prediction algorithms, was formalized as an inhomogeneous 3-periodic fifth-order Markov model in the GENMARK program (Borodovsky and McIninch 1993), and incorporated into the exon model used by the GENSCAN program (Burge and Karlin 1997). Hidden Markov models have become the most widely used descriptors for the diagnostic features of genes, including not just coding regions but also introns, exons, splice junctions, etc. (Henderson, Salzberg et al. 1997, Fickett and Hatzigeorgiou 1997, Burge and Karlin 1998)

In order to take into account errors often found in EST sequences, the coding sequence model should accommodate three additional possibilities normally ignored in gene prediction algorithms: (1) frameshift errors that would destroy the periodicity of the Markov chain; (2) sequencing errors that would introduce erroneous stop codons; (3) the presence of a sizeable number of ambiguous nucleotides.

These possibilities can be taken into account by embedding the exon model for the coding regions into a hidden Markov model, a principle introduced by Krogh, Mian et al. (1994) and which is used as the basis of many modern gene prediction programs including GENSCAN (Burge and Karlin 1997) and newer versions of GeneMark (Lukashin and Borodovsky 1998). In this application, the additional states of the Markov chain model frame-shift errors rather than RNA processing and translational control signals. Recent work by Audic and Claverie (1998) introduced an error-tolerant HMM for the identification of genes in bacterial genomes, but their method did not include a specific method for correcting sequencing errors.

The ESTScan program is thus an implementation of a coding region detection method based on an inhomogeneous 3-periodic fifth-order hidden Markov model, extended to allow for various types of sequencing errors, and normalized to correct for biases introduced by the length of the sequence and its G+C isochore group. We show here that this produces an efficient and robust method for the detection, evaluation and reconstruction of coding regions in poor quality sequence data.

Datasets and Methods

Sequence sets

For length- and isochore-dependent score normalization, we extracted from the human EST databases four isochore-separated sets (<43% G+C, 43-51%, 51-57%, and > 57%), each containing about 10^7 nucleotides. These were concatenated, cut into segments of 200 nt, internally shuffled in windows of 10 nt, reconcatenated, and finally split into pieces of variable length. This produced a series of normalization databases that preserved most of the compositional biases of the original ESTs while being devoid of coding potential.

Using the SRS system (Etzold, Ulyanov et al. 1996) to index and parse the EMBL database, we created a database of 1549 human 3' UTR regions to serve as negative controls. We also prepared a set of 6342 EST-derived 3' UTRs representing the five best matches in the human EST database for each of the members of the 3' UTR database, and truncated the search results to the non-coding matching

regions using the *xblast* program (Claverie and States 1993). As positive controls, we used the isochore-separated coding sequence collections prepared by the GENIO group at the University of Stuttgart (http://ipvr2.informatik.uni-stuttgart.de/GENIO), and a set of ESTs matching each of the isochore groups, using the same method as for the 3' UTRs (*blastn* of each sequence against the human EST database, extract 5 best hits, *blastn* against the original sequence set followed by *xblast* to extract the coding region).

Hidden Markov model

ESTScan uses a novel type of hidden Markov model (HMM) for prediction and reconstruction of coding regions. The description presented here assumes that the reader is familiar with the basic concepts of HMMs (tutorial introductions can be found in Rabiner (1989), and Durbin, Eddy et al. (1998)). The ESTScan HMM, whose basic architecture is shown in Fig. 1, includes as a submodel for the coding region the *n*th order 3 periodic Markov process previously used for exon recognition in gene prediction algorithms. The additional states model frame-shift errors and flanking non-coding sequences. In its present version, ESTScan does not use a particular sensor module for translation start sites nor a specific model for non-coding sequences.

The model we designed is different from previously described hidden Markov models combining higher order states with deletion and insertion states, in that the emission probabilities of the higher order states do not depend on the preceding symbols of the emitted sequences but on the preceding symbols of a frame-shift corrected hypothetical true coding sequence. The stochastic process by which this model generates EST-like sequences is best described by pretending that the model generates two sequences in parallel: a hidden sequence $x_1 \ldots x_n$ having the properties of a true coding region, and a visible sequence $e_1 \ldots e_m$ being an error-containing copy of the true coding region possibly flanked by non-coding sequences. The model contains four different state types producing a visible symbol, a hidden symbol, both types of symbols, or none, at any one time. If both types of symbols are produced, they are always identical meaning that our model does not explicitly describe a substitution error-generating process. The exact properties of each state are listed in the right-hand part of Fig. 1.

A visible EST-like sequence is generated by a random walk through the model formally described by a series of states $q_1, \ldots, q_k$. Knowing the parameters of the model one can compute the probability by which a given pair of visible and hidden sequences is generated by a particular path. The problem solved by the ESTScan algorithm can thus be described as finding the most probable path $q_1 \ldots q_k$,

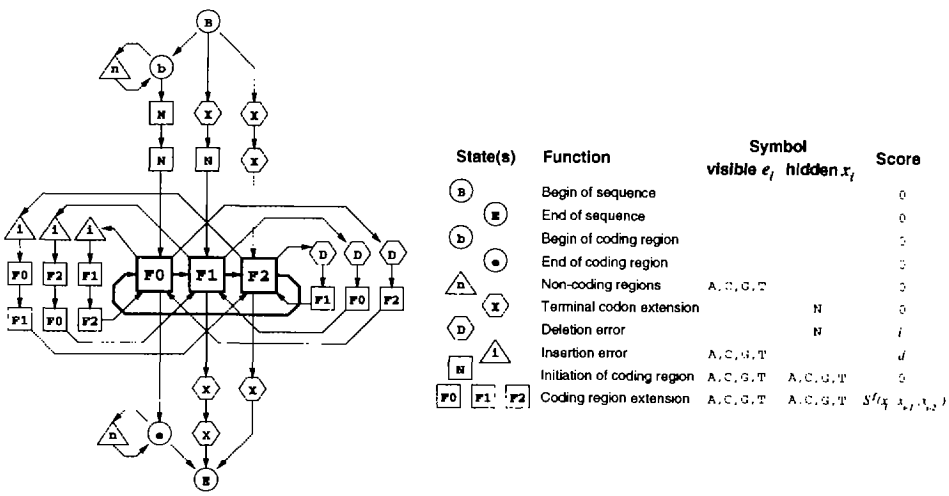

Figure 1. HMM underlying the ESTScan algorithm

The model shown represents a novel type of HMM generating two sequences at once, a hidden sequence corresponding exactly to the true coding region, and a visible copy of the same sequence, which may contain a few insertion and deletion errors, as well as non-coding sequences at the beginning and at the end. There are four types of states represented by different shapes: silent states (circles), states emitting only a visible symbol (triangles), states emitting only a hidden symbol (hexagons), and states adding the same type of symbol to the visible as well as to the hidden sequence (squares). The central part of the graph (emphasized by thick lines) represents the coding model, which in this example, is a 2nd order 3 periodic inhomogeneous Markov process. The two wings on the left and right side of the coding region model produce insertion and deletion errors, respectively, in the visible sequence. Special F states following frame-shift error states are needed because the scores assigned to visible bases by these states depend in different manners on the preceding visible symbols. For instance, if a visible symbol e_i is matched with an F state of the deletion wing, its score will be conditioned on the preceding symbols e_{i-2}, e_{i-1}, rather than e_{i-1}, e_{i-2} as for the main coding states. The loops around states b and e produce leading and trailing non-coding sequences. The additional states above and below the central coding region belong to entry and exit modules which make sure that the coding region starts and ends in the correct codon position. A visible sequence is generated by a random walk through the model. The states do not only produce visible or hidden symbols, they also define scores for such events related to their occurrence probabilities. The exact properties of each state are defined in the table on the right-hand side.

generating a given visible sequence $e_1 \ldots e_m$. Due to the fact that the HMM never emits two different symbols at a time, the hidden sequence is uniquely defined by the emitted sequence and the corresponding path. This property also makes it possible to find the most likely path which determines the most likely coding region, with the aid of a standard Viterbi-type dynamic programming algorithm (see Durbin et al. 1998, page 58).

Since ESTScan uses a Viterbi approach (computation of a score corresponding to the most likely path) rather than a full probabilistic approach (summing over the probabilities of all possible paths), the parameters of the model states can be defined as additive scores interpretable as the logarithm of the ratio of the actual probability over the corresponding null model probability. Using such additive scores, a number of sticky issues can be avoided, such as for instance the definition of specific probabilities for wildcard symbols, or the choice of a particular geometric length distribution for the flanking regions. For practical purposes, it is also not necessary that the implied emission probabilities of a given state sum to one. In the case of the ESTScan model, all scores for the transitions can be normalized to zero by transferring the corresponding costs to subsequent destination states. The transition terms can thus be eliminated from the expression for the total score of a path generating a particular sequence, which then has the following form:

$$S(e_1 \ldots e_m \mid q_1 \ldots q_k) = \sum_i S(e_{j(i)} \mid q_i)$$

Here $e_{j(i)}$ is the visible symbol emitted by the ith element of the path. Note that this term is only relevant for states that emit a visible symbol. For the states implementing the nth order 3 periodic inhomogeneous Markov process modeling the coding region, the score depends on the emitted symbol, the current reading frame, and the n preceding symbols of the hidden sequence (see Fig. 1). All other scores are independent of the symbol produced.

Parameters used by the HMM

ESTScan uses exactly the same type of coding region model as GENSCAN, and the same parameter file format. A particularly useful feature of this model is that it offers the possibility to use different Markov processes for different isochores (G+C content ranges). The order of the inhomogeneous 3-periodic Markov chain is a variable. The human exon model included in the GENSCAN distribution and used in this work consists of two 5th-order Markov chains for sequences of high and low G+C content (in Fig. 1, an HMM for a 2nd order Markov chain is shown for simplicity). In the GENSCAN parameter tables, the Markov transition probabilities are expressed as log-odds ratios of the higher-order Markov probabilities of the coding regions and the null model probabilities of the corresponding bases. The absolute values of the probabilities are thus unknown. The results shown in Table 1 clearly indicate that the GENSCAN exon scoring tables favor G+C-rich sequences, suggesting that the corresponding null model was derived from total genomic sequences rather than exons only, the latter tending to have a higher G+C-content than the surrounding non-coding sequences.

The order of the Markov process also determines the number of the parameters to follow. The GENSCAN parameter table defines scores for all $n+1$-tuples of the alphabet {A,C,G,T} for the three alternative codon positions. (The $n+1$th symbol of the tuple is the one which is actually produced by the state, the preceding n symbols are those upon which its score depends.) For most gene prediction programs, it is not known how ambiguous base symbols (rarely occurring in genomic sequences) are handled. For ESTScan, an appropriate treatment of ambiguous symbols is essential, not only because such symbols frequently occur in EST sequences, but also because the underlying HMM produces such symbols to accommodate hypothetical bases presumably missing in the input sequence. A precise description of how such symbols are treated is therefore desirable.

ESTScan uses a five-letter alphabet for internal sequence representation, including the wildcard character 'N' in addition to the four standard base symbols. During sequence input, any ambiguous IUPAC base symbol is automatically converted into 'N'. At the same time, the GENScan score tables for the coding region model are expanded to a five-letter alphabet. The scores of all $n+1$-tuples containing an N at the last position are set to zero. The scores for $n+1$-tuples containing exactly one N between position 1 and n are assigned the average of the four corresponding $n+1$-tuples in which 'N' is replaced by A, C, G or T. The current version of ESTScan offers in addition six alternative ways to compute these values, consisting of averaging over only the 1,2,3 highest, or the 1,2,3 lowest values of the four corresponding $n+1$-tuples. The scores for $n+1$-tuples containing more than one 'N' between positions 1 and n, are recursively computed using the same averaging procedure.

So far, we have not yet described a mechanism accounting for substitution errors occurring in the input sequence. As mentioned before, these errors which in most cases have far less drastic consequences on the deduced amino acid sequence, are not dealt with by the HMM architecture. There is however one particular case, the accidental generation of a stop codon, which makes a mechanism to deal with such errors mandatory. Because the GENSCAN exon models assigns prohibitively low scores to all $n+1$-tuples containing a stop codon in the critical frame, such an error would cause immediate termination of a coding region if no modification to the scoring system were applied. ESTScan deals with this problem by raising all prohibitively low scores to a threshold value defined by a command line parameter (the min matrix value parameter, see next Section). This threshold value must however remain substantially below the scores of the most unfavorable stop codon-free hexanucleotides in order to preserve the capability to detect intervening non-coding regions in coding exons. The only two additional non-zero scores of the HMM shown in Fig. 1, which do not belong

to the coding region model itself, are the insertion and deletion penalties, which are also specified on the command line.

Finding multiple matches

ESTScan also has the capability of finding multiple coding sequences separated by intervening non-coding regions in the input sequence. This is useful because ESTs may indeed contain such intervening sequences as a result of cloning artifacts or genomic contaminations in the cDNA libraries. Alternatively, true coding regions could look like non-coding sequences because of a high local concentration of sequencing errors or atypical sequence properties. In either case, it would probably be impossible to reconstruct the correct coding region from a negatively scoring sequence segment, which is sufficient reason for excluding it from the coding region prediction.

In order to accommodate the possibility of multiple coding regions, the Viterbi algorithm recursively computes the optimal coding region for subsequences starting at position 1 of the input sequence and ending at current position k. During this scanning process, it keeps track of the score of the best coding region ending at the current position, as well as the globally maximal score obtained at a previous position. Each time the current best score becomes negative or null, the procedure is re-initialized. A coding region is defined as the portion of sequence comprised between the procedure starting point and the point where the score was maximum; i.e., the portion of sequence that contributed mostly positive scores (provided the maximum score was above a specified cutoff value). The scanning procedure restarts at the nucleotide following the one where the score was maximum. To take into account the fact that there can be multiple fluctuations in the score between the start and maximum points, and to provide optimal tuneability, ESTScan introduces a configurable maximum drop parameter (-D). After a maximum score (greater than the cutoff) has been seen, the current best score will not be allowed to drop more than this specified value below the last maximum (i.e., we do not allow a valley deeper than the maximum drop, after having seen a proper maximum score). When the score goes below this maximum drop, the same thing happens as when the score reaches zero: production of a coding region and restart of the scanning procedure. Note that the absolute drop-off value is typically lower than the cut-off value for an individual coding region. We have not found a way to formulate an algorithm implementing this behavior in an HMM framework.

Score normalization

To classify short EST sequences into likely coding and non-coding sequences, one has to make adjustments to the scores to account for the effect of the sequence length. Assuming that the ESTScan scores are distributed according to the Karlin-Altschul statistics (Karlin and Altschul 1990) - and we have no reasons to believe that this is not the case - a simple length normalization procedure is suggested by the well-known formula used to compute the expectation values for BLAST matches: $E = K N^{\lambda S}$, where E is the expected number of matches with scores above S for sequence length N. K and λ are two parameters that could be determined by a single simulation experiment. However, this approximate formula applies only to situations where the length of the match is much smaller than the length of the sequence. Because we knew from preliminary tests with negative control databases that this condition is not satisfied for short EST sequences, we expected a more complex relationship between the expected false positive rate and the sequence length.

Therefore, rather than relying on a theoretical model, we decided to compensate for the effects of length and isochore class by generating a table of empirical data documenting the score boundaries (*cutoffs*) obtained at various false positive values f when using our algorithm on the normalization databases described above. The program with default parameters was run against each database, and the *cutoff* scores reached by a proportion of $(1-f)$ of all sequences were recorded. The data obtained are shown in Table I; they indicate that scores obtained from non-coding EST-type sequences strongly depend on the length and the isochore class of the sequence. Interestingly, there was a much stronger length dependence for the scores obtained from sequences with high G+C content.

To take these effects into account, the score normalization procedure reads a set of matrices of the type shown in Table I, selects the matrix corresponding to the sequence's GC content, and then uses linear interpolation on the sequence length, and logarithmic interpolation on the accepted false positive rate, to determine the proper cutoff value. The normalized score is then calculated as $100 \times \log$ (*score / cutoff*). Thus, a negative score is obtained if the probability that a query sequence is coding is less than the accepted false positive rate (0.01 by default).

Implementation and availability

The program was written as a Perl script, which calls the main algorithm (the alignment of the Markov model to the query sequence) as a compiled C module. In order to maximize flexibility, many parameters can be passed on the command line, using the syntax:

```
ESTScan [options] <file>
  where options are:
-m <int>    min value in matrix [-50]
-d <int>    deletion penalty [-50]
-i <int>    insertion penalty [-50]
-D <int>    maximum drop value [200]
-M <file>   score matrices file
-p <float>  GC select correction for score
            matrices [4]
-N <int>    how to compute the score of N [0]
-w <int>    width of the FASTA sequence output
-a          all in one sequence output [0]
-O          report header information for best
            match only [0]
-b          show results for both strands [0]
```

```
-t <file>  Translate to protein.
-o <file>  send output to file.  - means stdout.
-f <float> expected false positive rate [0.01]
-F <file>  false positive rate matrices file
-s <int>   Skip sequences shorter than length [1]
-c <int>   absolute cutoff value [undefined]
```

The input is a FASTA-formatted file containing an arbitrary number of sequences to be analyzed. The output is normally a FASTA-formatted file containing the predicted coding region, padded with X characters at the 5' end (if necessary) to force frame 1 to be the correct reading frame; insertions are indicated by X and deletions by showing the nucleotide to be removed in lower case. Alternatively, coding regions can be output in uppercase against a background of lowercase non-coding sequence (-a option), or the output can be restricted to statistics only (-O option). In addition to the nucleotide sequence(s) of the predicted coding region(s), ESTScan can produce predicted protein sequences (-t option). The sequence's ID and description, the normalized score, absolute score, computed cutoff value, and the begin and end position of the predicted coding region are indicated on the FASTA header line.

The format and values of the score matrices for the Markov model are those used by Burge and Karlin's GENSCAN program, and ESTScan can read GENSCAN's matrices without modifications. For score normalization, we use a table similar to Table I, distributed with the program. For the time being, this table is available only for human sequences.

We also provide utilities for generating the normalization databases and tables: sorting of ESTs according to isochore class, shuffling of the sequences, output of uniformly-sized pseudo-sequences after shuffling, and generation of a table suitable for input into ESTScan. These tools can be used to generate databases and tables for various species, or scoring systems of varying stringency.

The program, including source code, is freely available from the authors upon request. Individuals or institutions interested in redistributing the program or in incorporating its code into a commercial product should contact the authors.

Results

Initial optimization

There is no easy method to perform an optimization of all parameters for all classes of sequences (high-quality sequences and ESTs derived from different isochore classes) in an objective way. Therefore, we only investigated the effects of a few parameters on the ability of the program to discriminate between coding and non-coding sequences, and we have not yet attempted to optimize the matrix cutoff, ambiguous nucleotide scoring, and drop values.

First, we found that score normalization is absolutely necessary to produce an objective criterion by which to evaluate the probability of a sequence to be coding. As the raw scores were highly dependent on both the length of the sequence and its isochore class (Table I), no single cutoff value could be assigned that would distinguish coding from non-coding sequences. On the other hand, a normalization table that considered only two isochore classes (with a boundary at 47% G+C) performed almost as well as the four-class table presented here.

Second, and surprisingly, we found that the indel penalties did not have a very significant effect on the effectiveness of the discrimination over a wide range of negative values. We have not yet investigated in detail how much they would affect the accuracy of the coding region detection, but preliminary results indicate that high penalties result in the elimination (as opposed to frame correction) of out-of frame regions. It should be noted that our scoring system awards an average score of about 1.5 to each nucleotide found to be within a predicted coding region. Unsurprisingly, on the "high-quality" dataset extracted from non-EST entries of the GenBank/EMBL databases, the discrimination power was essentially independent of the

		False positive rate (*f*)					
Class	**len**	**0.2**	**0.1**	**0.05**	**0.02**	**0.005**	**0.002**
I	100	47	59	70	84	105	118
I	250	63	77	92	111	143	164
I	400	71	88	104	128	166	192
I	625	81	98	117	142	184	211
I	1000	91	111	131	160	206	244
I	5000	134	160	185	221	283	323
II	100	57	73	87	105	130	146
II	250	82	105	129	159	203	227
II	400	98	128	158	198	253	290
II	625	115	151	188	236	303	349
II	1000	138	181	226	280	371	414
II	5000	243	307	372	453	563	693
III	100	65	83	100	121	148	165
III	250	99	131	162	199	248	279
III	400	124	169	212	263	333	376
III	625	155	211	267	334	424	470
III	1000	199	273	345	434	543	614
III	5000	447	610	761	932	1254	1494
IV	100	76	98	118	140	172	190
IV	250	129	172	209	252	309	344
IV	400	173	237	290	350	432	481
IV	625	235	319	391	470	574	639
IV	1000	331	444	542	647	784	859
IV	5000	1166	1491	1750	2034	2334	2507

Table I. Cutoff values used for score normalization. Shuffled EST sequence databases with entries of uniform length were produced as described in Methods. The isochore classes were: **I:** <43% G+C; **II:** 43-51%, **III:** 51-57%, and **IV:** > 57%. The scores for all entries in each database were calculated, and the cutoff score for false positive rates of *f* calculated. The actual table used by ESTScan contains a larger number of entries, based on more values for *f* and *length*.

penalties used. On the EST-derived datasets, optimal discrimination between coding and non-coding sequences was achieved using indel penalties of around -50, which is the default value for the program. However, penalties as high as -15 and as low as -100 still produced acceptable discrimination levels (data not shown).

Sensitivity and selectivity of coding region detection

In order to estimate the sensitivity and selectivity of our method, we ran the program on 3' UTR sequences as well as isochore-separated collections of coding regions, derived either from high-quality database entries or from ESTs. Results were collected with the –O option, which reports for each sequence the highest score obtained on either strand. In normal mode, only those sequences producing positive scores would be reported.

The results are shown in Figure 2, either as relative frequencies of normalized score intervals (upper panels), or as the cumulative fraction of sequences scoring above (or below, for the negative control sets) a specific score (lower panels). Clearly, the score distributions of the 3' UTRs are well separated from those of the coding regions. As expected, this separation is better for the high-quality sequences than for the EST sequences. The scores for the high-quality sequences were obtained using an expected false positive rate of 0.01; at that rate, between 3.5 and 4.5% of the coding sequences scored below zero (false negatives), while less than 3% of the 3' UTR sequences scored above zero (false positives). In other words, about 94% of all good-quality sequences given to the program were flagged accurately to be coding or non-coding. For the EST sequences, we chose an expected false positive rate of 0.1; at that level, 5.74% of isochore class I, 4.72% of class II, 4.37% of class III, and 5.92% of class IV were scored as false negatives, while 18.17% of 3' UTRs were scored as false positives. In other words, it is possible to flag and extract about 95% of true coding sequences from EST databases while accepting a tolerable rate of false positives. In contrast, GENSCAN found only about 35% of the coding sequences in the positive EST set. It was interesting to note that the false positive rates measured for 3'UTRs were higher than those calculated from the normalization databases. We suspect that this is due to the lower G+C content of 3'UTRs, which causes them to undergo a less stringent normalization than high isochore sequences (see Table I).

We did not perform systematic tests on the accuracy of the assignment of boundaries between coding and non-coding regions in EST sequences. However, we did ascertain that the presence of non-coding regions did not affect either the sensitivity or the selectivity of coding region detection (data not shown).

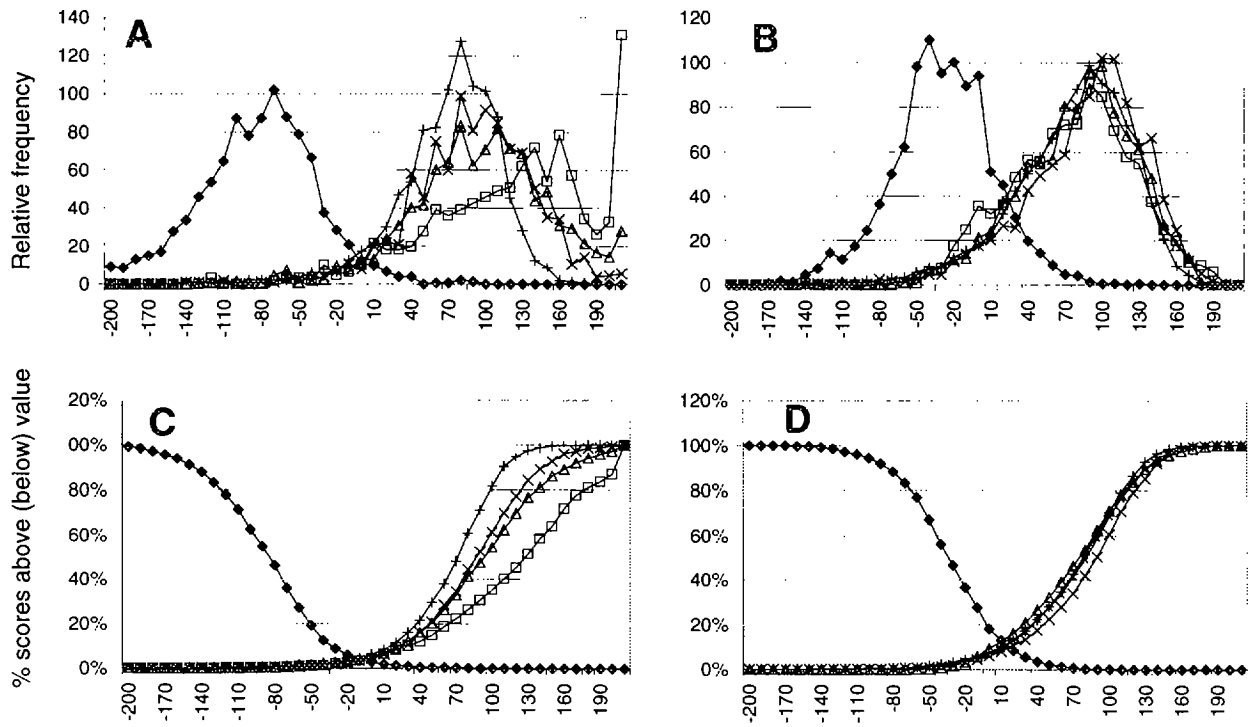

Figure 2. Discrimination between coding and non-coding sequences

Normalized scores were computed for collections of 3' UTR sequences (diamonds), and of coding sequences from isochores I (squares), II (triangles), III (crosses) and IV (plus signs). Panels A and C: results from high-quality sequences; panels B and D: results from EST sequences. Panels A and B: histograms of the distribution of the scores, in 10-unit intervals. Panels C and D: cumulative percentages of the scores above (or below, for the 3' UTRs) the interval indicated on the abscissa.

Figure 3. Example of a reading frame reconstruction by ESTScan

a) Reverse-complement of an EST sequence (Ac-nr: AA857478) containing most of the human ferritin light subunit mRNA sequence. The part corresponding to the coding region is shown in upper case. Conflicts with the published mRNA sequence (Ac-nr :M10119) are indicated above the sequence.

b) Frame-shift corrected coding region predicted by ESTScan. Note that bases missing in the input sequence are represented by 'X'.

c) Alignment of the human ferritin light chain protein sequence with the translated coding region predicted by ESTScan. The alignment was generated with the program align0 from the fasta2 package (W.R. Pearson, ftp://ftp.virginia.edu/pub/fasta/).

d) EST coding region prediction with GENSCAN. Shown is an alignment between the human ferritin and the translated amino acid sequence from two consecutive exons predicted by GENSCAN for the same EST sequence. Note that GENSCAN accommodates the two frameshift errors in the central part of the coding region by predicting a short intron.

Accuracy of reading frame corrections

One of the design goals of the program was to detect and correct frameshift errors present in EST sequences. We tested this functionality on two specific examples: (1) an EST containing most of the mRNA sequence of the light chain of human ferritin, with two sequencing errors in the coding region that each cause the deletion of one nucleotide; (2) GenBank/EMBL entry AB000215, the sequence of the mRNA for the rat CCA1 protein, which was found to contain a sequencing error (deletion of 1 nt) near its 3' end that affects the sequence of the last 40 amino acids of the protein (K. Hofmann, personal communication). This second case is made more difficult by the fact that the artefactual reading frame is not interrupted by a stop codon before the position of the true stop codon.

ESTScan performed extremely well in both cases. The first one is illustrated in Figure 3: the program, run with default parameters, detected both frameshifts and corrected them by inserting an "X" within less than 10 nt of the actual sequencing error. The GENSCAN program, which was not designed to handle insertions and deletions leading to frameshifts but otherwise uses an algorithm very similar to ours, accommodated the double frameshift by converting the intervening region to an intron. This is still a remarkably good result considering the very different design goals of GENSCAN. In the second example, ESTScan again detected the frameshift and corrected it within 10 nt, despite the fact that it was preceded by over 1800 nt of correct coding sequence and that the correction did not affect the total length of the ORF. It should be noted that the GENIO program also detected this frameshift, while GENSCAN did not.

To get a more quantitative evaluation, we tried to assess ESTScan's accuracy in coding region reconstruction on a larger scale. The design of an adequate test for this purpose is not trivial as coding regions and sequencing errors are of course not annotated in the EST sequences. We therefore chose an indirect approach, comparing the sensitivity of a *tblastn* search of an EST DNA database with that of a *blastp* search of the corresponding protein database extracted by ESTScan. The rationale of this experiment is as follows: If ESTScan would miss many true coding regions, select the wrong reading frame, or be unable to localize frame-shift errors with sufficient precision, then the *blastp* search would miss a substantial fraction of the sequences found by the *tblastn* search. Conversely, if ESTScan would precisely restore all open reading frames, one would expect some increase in sensitivity due to the error correction and reduction of the search space. The database chosen for this experiment consisted of all human EST sequences not assigned to a cluster in Unigene version 74. ESTScan extracted from these 299,191 sequences totaling 97.6 million residues, 78,387 ORFs totaling 6.0 million amino acids. Taking into account that a *tblastn* search considers both strands of a DNA sequence, this amounts to a 32-fold reduction of the

search space, resulting in a proportional decrease in search time. As protein queries we used to sequences expected to hit many targets in the EST database: a yeast protein kinase (YPK1, SWISS-PROT entry O12688) and human epidermal growth factor (EGF, SWISS-PROT entry P01133). With the protein kinase query, the *tblastn* search of the DNA database produced 126 hits with E-values lower than 10^{-4}; the *blastp* search of the protein database detected 123 sequence at the same significance threshold. With the EGF query, the *blastp* search was even slightly more effective than the *tblastn* search (48 versus 44 hits). A more detailed analysis of the hit lists revealed that about 10% of the sequences found by *tblastn* were missed by *blastp* because they were not detected by ESTScan, and that a similar fraction of sequences found by *blastp* reached the significance threshold thanks to frame-shift corrections made by ESTScan. In summary, these results indicate that most of the reading frames detected by ESTScan are also reconstructed with high accuracy, and that applying this procedure can reduce the execution time of an EST database search with a protein query by a factor of 30 without net loss of sensitivity.

Discussion

While coding exon recognition is an important component of any gene prediction program, the recognition and correction of sequencing errors leading to frameshifts and erroneous stop codons has only been addressed in the context of homology-based approaches (Brown, Sander et al. 1998; Birney, Thompson and Gibson, 1996)) in the bioinformatics literature. The only program we know of that appears to be based on principles similar to ESTScan is GENIO/frame (N. Mache, unpublished), which generates a graphical representation of the probability that each of the three forward frames in a coding sequence is the coding frame. This allows a quick visual detection of frameshifts, but does not at present propose a model for their automated correction. The GENIO suite also includes an explicit model for non-coding regions, which may improve its ability to discriminate between coding and non-coding regions. To our knowledge, the frameshift detection and coding region recognition algorithms of GENIO have not yet been integrated into a single program.

We approached the problem of reconstructing correct coding regions from error-containing EST sequences by a new type of HMM combining higher order Markov states with insertion and deletion states. Although simpler HMMs with these properties and corresponding algorithms have been described before (Durbin, Eddy et al. 1998) we felt that a new model type was required to solve this problem for two reasons: First, in the previously described model type, the higher order Markov probabilities were dependent on the emitted (visible) sequence context, which appears to be inconsistent with the goal of this algorithm to correct for errors. Obviously, probabilities that are supposed to describe the statistical properties of true coding regions should not be made dependent on potentially erroneous preceding base sequences. Secondly, there is an ambiguity with the previously described models regarding the exact placement of insertions and deletions during path reconstruction. Since the higher order Markov probabilities apply to overlapping $n+1$-mers (in case of the GENSCAN exon model to hexamers overlapping by 5 bases), it is by no means clear where to delete or where to insert a single base within the overlap region. Most disturbingly, by blindly applying some arbitrary convention, a stop codon could be generated by accident. Therefore, we introduced a new model type where the higher order Markov probabilities depend not on the visible sequence context, but on preceding symbols of a supposedly correct hypothetical coding region. We are aware of the fact that the error-generating process implemented in this model (see Fig. 1) may not be entirely realistic as it does not allow for double insertions or deletions at nearby locations (within the dependence range of the Markov process). However, this aspect is irrelevant from a practical viewpoint because with reasonable gap penalty values such unlikely events could never be correctly reconstructed anyway, as a single base insertion would always be a cheaper way to restore the coding frame than a double deletion and vice-versa. We thus believe that the model proposed in this work could serve as paradigm for other sequence analysis applications where appropriate modeling of sequencing errors is crucial for good performance.

The experiments with shuffled databases revealed the expected complex relationship between sequence length and false positive rate. We were more surprised to find a strong isochore dependence of the ESTScan scores, apparently reflecting an inherent property of the exon model copied from GENSCAN, and were wondering why this property seems not to negatively affect the performance of the model in gene prediction. There may be a simple explanation for this paradox. The GENSCAN algorithm assigns each part of the sequence to one of three sequence classes, exon, intron, or intergenic regions, each of which is modeled by a specific Markov process. The exon model is thus in competition with other models, and if these other models exhibit the same type of isochore dependence, the net effect would be zero. In contrast, ESTScan does not use a particular model for non-coding regions, which is tantamount to comparing the exon model to a simple null model consisting of a zero-order Markov process with equal base frequencies. These considerations suggest that the isochore normalization could be tackled by introducing a more appropriate model for non-coding regions into the ESTScan HMM. While such an approach seems more elegant and intellectually more satisfying, we cannot be sure at the moment whether it would work equally well or better than the simple and robust procedure we currently use.

The practical implementation of ESTScan should make it a widely useful tool. We have designed a simple Web interface to the program, for the use of biologists who are interested in checking individual sequences for coding potential and sequencing errors (at

http://www.ch.embnet.org). On the other hand, it can be used in batch mode to scan arbitrarily large numbers of sequences and produce databases of error-corrected coding regions or of predicted protein sequences. These databases could then be used as a much-improved source of raw data for efforts aimed at producing contigs from existing EST collections. Currently, a major obstacle to the automated production of such contigs is the presence in the EST databases of large amounts of non-coding sequence, including repetitive elements, introns, and 3' UTRs, all of which should be removed by ESTScan.

In conclusion, we feel that the ESTScan program fills a new niche among the panoply of methods available to the biocomputing community, and that it introduces in the process potentially interesting variations to the Markov models used in biological sequence analysis.

References

Adams, M. D., J. M. Kelley, et al. (1991). "Complementary DNA sequencing: expressed sequence tags and human genome project." Science **252**(5013): 1651-6.

Altschul, S. F., T. L. Madden, et al. (1997). "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." Nucleic Acids Research **25**(17): 3389-402.

Audic, S. and J. M. Claverie (1998). "Self-identification of protein-coding regions in microbial genomes." Proceedings of the National Academy of Sciences of the United States of America **95**(17): 10026-31.

Birney, E., Thompson, J. D., and T. J. Gibson (1996). "PairWise and SearchWise: finding the optimal alignment in a simultaneous comparison of a protein profile against all DNA translation frames." Nucleic Acids Research **15**: 2730-2739.

Borodovsky, M. Y. and J. D. McIninch (1993). "GENMARK: parallel gene recognition for both DNA strands." Comput. Chem. **17**: 123-133.

Brown, N. P., C. Sander, et al. (1998). "Frame: detection of genomic sequencing errors." Bioinformatics **14**(4): 367-71.

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

Burge, C. B. and S. Karlin (1998). "Finding the genes in genomic DNA." Current Opinion in Structural Biology **8**(3): 346-54.

Claverie, J.-M. and D. J. States (1993). "Information enhancement methods for large scale sequence analysis." Comput. Chem. **17**: 191-201.

Durbin, R., S. Eddy, et al. (1998). Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge, UK, Cambridge University Press.

Etzold, T., A. Ulyanov, et al. (1996). "SRS: information retrieval system for molecular biology data banks." Methods in Enzymology **266**: 114-28.

Fickett, J. W. (1996). "Finding genes by computer: the state of the art." Trends in Genetics **12**(8): 316-20.

Fickett, J. W. and A. G. Hatzigeorgiou (1997). "Eukaryotic promoter recognition." Genome Research **7**(9): 861-78.

Henderson, J., S. Salzberg, et al. (1997). "Finding genes in DNA with a Hidden Markov Model." Journal of Computational Biology **4**(2): 127-41.

Karlin, S. and S. F. Altschul (1990). "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes." Proc. Natl. Acad. Sci. U.S.A. **87**: 2264-2268.

Krogh, A., I. S. Mian, et al. (1994). "A hidden Markov model that finds genes in E.coli DNA." Nucleic Acids Res. **22**: 4768-4778.

Lukashin, A. V. and M. Borodovsky (1998). "GeneMark.hmm: new solutions for gene finding." Nucleic Acids Research **26**(4): 1107-15.

Ouellette, B. F. and M. S. Boguski (1997). "Database divisions and homology search files: a guide for the perplexed." PCR Methods & Applications **7**(10): 952-5.

Rabiner, L. R. (1989). "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE **77**: 257-286.

Venter, J. C., M. D. Adams, et al. (1998). "Shotgun sequencing of the human genome." Science **280**(5369): 1540-2.