BIOINFORMATICS

Vol. 19 Suppl. 2 2003, pages ii103–ii112
DOI: 10.1093/bioinformatics/btg1067

Modeling sequencing errors by combining Hidden Markov models

C. Lottaz [1, 2,*], C. Iseli [1, 3], C. V. Jongeneel [1, 3] and P. Bucher [1, 2]

[1]Swiss Institute of Bioinformatics, Switzerland, [2]Swiss Institute for Experimental Cancer Research, Switzerland and [3]Office of Information Technology, Ludwig Institute for Cancer Research, chemin des Boveresses 155, CH-1066 Epalinges s/Lausanne, Switzerland

Received on March 17, 2003; accepted on June 9, 2003

ABSTRACT

Among the largest resources for biological sequence data is the large amount of expressed sequence tags (ESTs) available in public and proprietary databases. ESTs provide information on transcripts but for technical reasons they often contain sequencing errors. Therefore, when analyzing EST sequences computationally, such errors must be taken into account. Earlier attempts to model error prone coding regions have shown good performance in detecting and predicting these while correcting sequencing errors using codon usage frequencies. In the research presented here, we improve the detection of translation start and stop sites by integrating a more complex mRNA model with codon usage bias based error correction into one hidden Markov model (HMM), thus generalizing this error correction approach to more complex HMMs. We show that our method maintains the performance in detecting coding sequences.

Keywords: coding region prediction, sequencing errors, expressed sequence tags, hidden Markov models.

Contact: Claudio.Lottaz@molgen.mpg.de

INTRODUCTION

The millions of expressed sequence tags (ESTs) available in public and proprietary databases for various species sample their respective transcriptomes in different tissues, in various cell types and under varying conditions (Adams *et al.*, 1991; Camargo *et al.*, 2001; Strausberg *et al.*, 2000). Thereby ESTs are typically produced by single-pass sequencing in order to keep costs moderate. Many ESTs contain parts of coding sequences and may therefore allow to discover so far unknown proteins as well as to infer the locations where and the conditions under which certain proteins are expressed.

*To whom correspondence should be addressed. Current address: Max-Planck-Institute for Molecular Genetics, Ihnestr. 73, D-14195 Berlin (Germany)

Coding regions in ESTs

ESTs are nucleotide sequences of limited length (several hundreds of nucleotides) obtained from single-pass sequencing of cDNA clones. Usually these clones are sequenced from 5′ and 3′ ends. ESTs may span short coding sequences entirely, but frequently 5′ ESTs start in the middle of a coding region, while 3′ ESTs often only contain a piece of the 3′ untranslated region. However, one outstanding feature of ESTs is a considerable rate of sequencing errors, since they are not verified by repeated sequencing. Moreover, the reliability is high for the first nucleotides but drops after a few hundred nucleotides. In average, we expect about one insertion, deletion or substitution in 100 nucleotides.

In addition to the obvious application of error corrected coding region prediction, finding hypothetical coding sequences and the corresponding putative proteins, the following applications are worth mentioning:

- Discrimination between ESTs containing large coding regions and those containing none can be helpful to sort out genomic contaminations with little coding potential from raw experimental data.

- Searching entire databases of EST sequences for features expected in coding regions is slow due to the amount of data, and has weak sensitivity due to sequencing errors. Frameshift corrected databases of predicted coding regions and translations of these into protein sequences have the potential to improve search speed and accuracy substantially.

- Error-tolerance is also an issue when mapping ESTs on genomic data. The difference in quality between the two kinds of data makes mapping more difficult. Frameshift-corrected predictions are likely to avoid some of the ambiguities encountered.

Error-tolerant modeling of coding sequences

Various approaches have been applied in gene finding tools (Fickett, 1996), and modeling of coding regions has been an important element in gene prediction. According to the species-dependent bias in codon usage and amino acid frequencies, *n*-tuples of nucleotides have particular distributions in coding regions. Therefore coding potential can be measured as conformance of a region to such a distribution. One approach to compute such conformance is formalized as a 3-periodic inhomogeneous fifth order hidden Markov model and variants are used in programs such as GENMARK (Borodowsky and McIninch, 1993), GENSCAN (Burge and Karlin, 1997) and GLIMMER (Salzberg *et al.*, 1998). A tutorial on Markov models in general is given in (Rabiner, 1989).

Since sequencing errors are common in ESTs, error-tolerant and error correcting methods are needed to analyze such sequences. An error modeling HMM for genomic sequences of E-Coli has been proposed in (Krogh *et al.*, 1994). `ESTScan`, an earlier attempt to detect and predict coding regions in ESTs, has focused on error-tolerant modeling of coding regions (Iseli *et al.*, 1999). `ESTScan` uses a modified hidden Markov model which not only generates a sequence but also reads a sequence at the same time. The read sequence is supposed to be an EST while the predicted frameshift corrected coding sequence is generated. `ESTScan`'s model has states which only read nucleotides, states which only write nucleotides and states which do both. Using these particular states insertions and deletions are modeled explicitly and penalized with a user specified penalty such that errors are predicted with reasonable frequency. Thereby, neither start and stop translation sites nor untranslated regions are modeled explicitly.

This approach performs well for the detection of ESTs containing a coding region. Elaborate fine-tuning allows to keep the false positive acceptable while most of ESTs containing coding sequences are recognized. It has also been shown that reading frame is successfully recognized and frameshift errors are corrected. However, the detection of coding region boundaries is more delicate and `ESTScan` often inaccurately predicts the start and stop sites of a coding region.

Since it has been shown that translation initiation start sites show detectable sequence patterns in vertebrates (Kozak, 1987), various gene predictors model coding region start sites. Start and stop sites are explicitly modeled in GENSCAN by position specific scoring matrices. However, GENSCAN and other gene predictors cannot cope with sequencing errors. Other tools focus on detecting sequencing errors. Most of these are based on statistical models of codon frequency (Fichant and Quentin, 1995; Xu *et al.*, 1995) or sequence homology (Brown *et al.*, 1998; Guan and Überbacher, 1996; Sze and Pevzner, 1997). In contrast, DIANA-EST (Hatzigeorgiou *et al.*, 2001; Hatzigeorgiou, 2002) relies on artificial neural networks to predict coding regions. In this approach start and stop sites are modeled explicitly in order to improve the prediction quality.

Combining HMMs

Preliminary studies have shown that by adding profiles for translation start and stop sites to the original `ESTScan` approach significant improvement in the detection of coding region boundaries can be achieved. Therefore, in the new version of `ESTScan`, we combine a hidden Markov model for complete messenger RNA sequences including a model for untranslated regions as well as profiles for start and stop sites with our former HMM based approach to sequence error modeling. While this is not expected to improve neither specificity nor sensitivity, the intention is to improve the detection of coding regions boundaries.

In order to evaluate the performance of `ESTScan 2.0` and compare it to `ESTScan 1.3`'s results, we devised new evaluation methods also described here. Thereby, not only the classification of ESTs into coding region containing or not, but also the correct classification of single nucleotides as "part of a coding region" or "part of an untranslated region" is considered. Moreover, the precision of start site and stop site detection is evaluated and significant improvement is observed.

In this article we next describe the models used for modeling messenger RNA and ESTs. In Section implementation issues are discussed. We report some results in Section before we finally draw conclusions.

MODELING ERRORS IN ESTS

The new version of `ESTScan` is based on a classical Viterbi algorithm to determine the optimal path though a hidden Markov model. Each state of the model is thereby attributed either to the coding or to the untranslated region. The coding sequence can then be collected from the outputs of the coding region states. The HMM we use in `ESTScan 2.0` is described in this section.

Modeling mRNA

Our suggestion to model complete messenger RNAs using a high order hidden Markov model is illustrated in Figure 1. The classical 3-periodic inhomogeneous hidden Markov model for coding regions still represents the core of the coding sequence (states F0, F1 and F2 in Figure 1). In addition, profiles at both ends of this coding sequence core model the particular codons often observed at start and stop sites (ATG starts and TAA/TGA/TAG stops for instance). Finally, the model for untranslated regions may

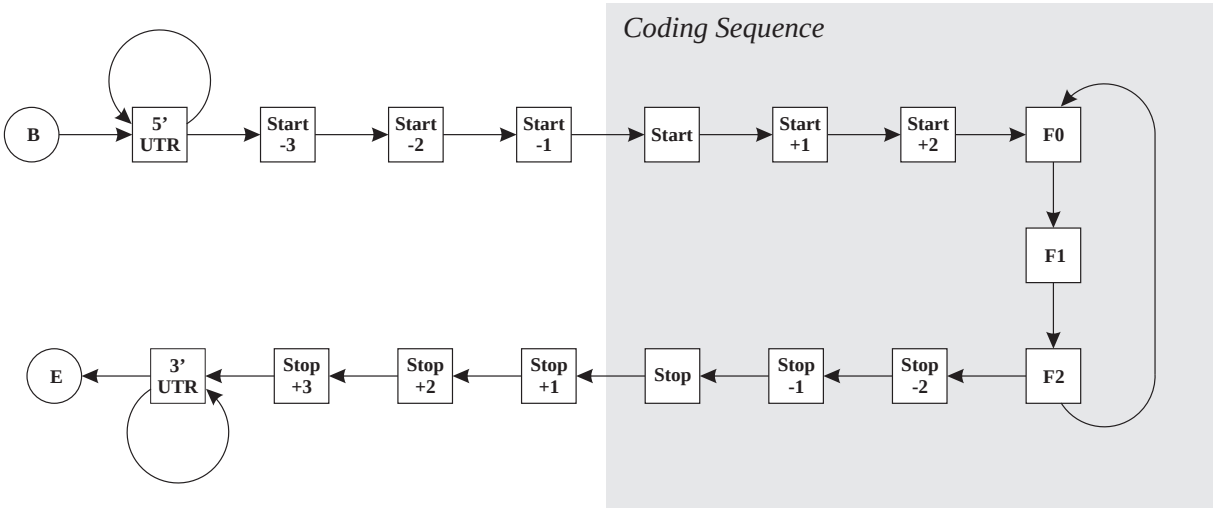

Fig. 1. Order 3 hidden Markov model for complete mRNA sequences. Only states in the gray area are attributed to the coding region.

also contribute to best position transitions from coding to non-coding and vice versa.

We suggest to use an order one HMM for transition probabilities while using higher orders for emission probabilities. The start profile contains at least n positions in the coding region where n is the order for emission probabilities. This ensures that when the model first enters the state F0, all nucleotides that determine its emission probability are coding. An analogous justification requires n untranslated states in the stop profile.

When using order n to determine emission probabilities, a problem arises when computing emission probabilities for the first n nucleotides $x_0 \dots x_{n-1}$ of the sequence to be analyzed. These emission probabilities depend on nucleotides $x_{-n} \dots x_{-1}$ which are not known and therefore assumed to be 'N'. That is, emission probabilities are averaged over all possibilities in nucleotides $x_{-n} \dots x_{-1}$.

A most probable path through this model attributes a state to each nucleotide of an analyzed sequence. The corresponding predicted coding sequence consists of all nucleotides attributed to states belonging to the second half of the start profile, to the coding sequence core states and to the first half of the stop profile. The gray region in Figure 1 indicates the states to which coding nucleotides are supposed to be attributed.

Modeling sequencing errors

Given that sequencing errors are frequent in EST sequences, we suggest to model these as shown in Figure 2. Similar to the approach developed for the former version of `ESTScan` this model simultaneously reads and writes a sequence. It contains different types of states represented in Figure 2 by the following symbols:

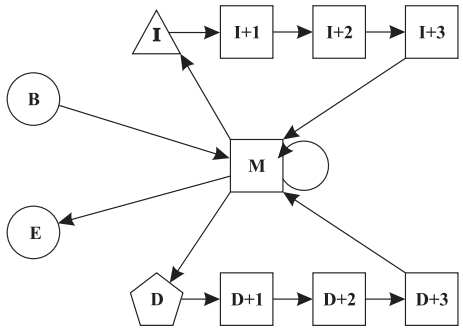

Fig. 2. Error model of order 3 for EST sequences. Circles represent silent states, triangles stand for read-only states, pentagons represent write-only states and squares are match states.

- Circles: silent states, used to represent begin and end states.
- Squares: match states, they read a symbol from the input and write a symbol onto the output.
- Triangles: read-only states, used to model insertion errors.
- Pentagons: write-only states, used to model deletion errors.

Transition probabilities are chosen such that in average once in 100 times a match state is followed by either the insert or delete state. Write-only states write an 'X' symbol on the output sequence as a place-holder for an unknown nucleotide.

States $I + 1$ to $I + n$ and $D + 1$ to $D + n$ where n is the order of the model are needed because their

emission probabilities do not depend on the last n input symbols due to the introduction of an insertion or deletion within these last n symbols. The particularity about our error model is the suggestion to attribute a context to each state. A context indicates which of the recent input symbols are used to compute the emission probabilities of the given state. In usual hidden Markov models all states have context $(-1, -2, \ldots, -n)$. It indicates that when reading nucleotide x_i, its emission probability is computed depending on $x_{i-1} \ldots x_{i-n}$.

Computing emission probabilities like this is not adequate in our error model for states following hypothetical insertions or deletions. For instance, emission probabilities of state $I+1$ in the order 3 model of Figure 2 depend on $x_{i-2}x_{i-3}x_{-4}$, since x_{i-1} is considered an insertion error. Thus state $I+1$ has context $(-2, -3, -4)$. Hence state $I+2$ has context $(-1, -3, -4)$ and $I+2$ has context $(-1, -2, -4)$. An additional extension is needed to model deletions. We allow the symbol X in contexts, which indicates that an unknown nucleotide should be used at the corresponding position. Thus state $D+1$ has context $(X, -1, -2)$ indicating the probabilities are computed based on the subsequence $Nx_{i-1}x_{i-2}$. Similarly, state $D+2$ has context $(-1, X, -2)$ and state $D+3$ has context $(-1, -2, X)$.

Combine RNA and error models

In order to convert the mRNA model illustrated in Figure 1 into an error tolerant and correcting version, we suggest to combine it with the error model of Figure 2 in the following manner:

1. To each state S of the mRNA model the context $(-1, \ldots, -n)$ is attributed.
2. To each state S a read-only state S^I is connected with transition probability according to how many insertions are expected.
3. To each state S a write-only state S^D is connected with transition probability according to the expected frequency of deletions. It always emits `X`.
4. For each S call $insertion\_add(S, S^I, 1)$
5. For each S call $deletion\_add(S, S^D, 1)$

The procedures `insertion_add` (Fig. 3) and `deletion_add` (Fig. 4) recursively copy the original structure of the hidden Markov model for the first n states following state S given as argument.

In the pseudo-code n is the model order, $e(S)$ denotes the emission probabilities of state S, $t(S \rightarrow T)$ is the probability of transition S to T and E stands for the end state.

This procedure generates at least $p(2n+3)$ new states where p is the number of states in the original model and

Proc$insertion_add(S, I, j)$ $T \leftarrow$ successors of X;
if $j > n$ **then**
  foreach T_i **do** $t(I \rightarrow T_i) \leftarrow t(S \rightarrow T_i)$
else
  $c \leftarrow (-1, \ldots, -j+1, -j-1, \ldots, -n-1)$;
  foreach T_i **do**
    if $T_i = E$ **then**
      $t(I \rightarrow E) \leftarrow t(S \rightarrow T_i)$
    else
      generate S_i^{I+j} with context c;
      $e(S_i^{I+j}) \leftarrow e(T_i)$;
      $t(I \rightarrow S_i^{I+j}) \leftarrow t(S \rightarrow T_i)$;
      $insertion\_add(T_i, S_i^{I+j}, j+1)$;

Fig. 3. Generates the extensions needed after hypothetical insertion errors

Proc$deletion_add(S, D, j)$ $T \leftarrow$ successors of S;
if $j > n$ **then**
  foreach T_i **do** $t(D \rightarrow T_i) \leftarrow t(S \rightarrow T_i)$
else
  $c \leftarrow (-1, \ldots, -j+1, 0, -j, \ldots, -n+1)$;
  foreach T_i **do**
    if $T_i = E$ **then**
      $t(I \rightarrow E) \leftarrow t(S \rightarrow T_i)$
    else
      generate S_i^{D+j} with context c;
      $e(S_i^{D+j}) \leftarrow e(T_i)$;
      $t(D \rightarrow S_i^{D+j}) \leftarrow t(S \rightarrow T_i)$;
      $deletion\_add(T_i, S_i^{D+j}, j+1)$;

Fig. 4. Generates the extensions needed after hypothetical deletion errors

n is the model's order. The lowest number of states is generated in the case where no state in the original model has multiple successors. For our model in Figure 1 p is $4n+5$, thus the number of states for a complete error-tolerant model according to our approach would contain more than $(4n+5)(2n+3)$. In the common case of $n=5$ we expect more than 325 states.

Modeling EST structure

So far we have built an error-tolerant mRNA model. In Section‘ we have mentioned that ESTs may contain the whole coding region, hold only parts of it or may even be entirely untranslated. This is the last piece of information we need to integrate in order to present a model for EST sequences. Figure 5 illustrates the structure to be considered. We have chosen transition probabilities empirically. Since they are highly dependent on the

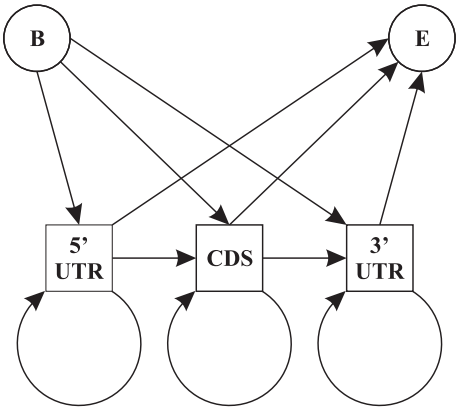

Fig. 5. Modeling coding region structure in ESTs.

technology which produced the EST to be analyzed, these parameters may be fine tuned from case to case.

Figure 5 implies for instance that all states shown in Figure 1 can be reached directly from the begin-state and that transitions from any state directly into the end-state are added.

IMPLEMENTATION

Similar to its predecessor we implemented this new version of `ESTScan` as a Perl-script calling compiled C-routines for time-critical computation. Input and output data formats are, also similar to `ESTScan 1.x`, FASTA files, possibly with several entries. The parameters of the mRNA model are read from a model file before starting the analysis. Tools to extract training data, remove redundancy and train parameters are provided with the package. Namely the script `build_tables` reads a training configuration from a configuration file and writes model parameters to be read by `ESTScan` in a model file. A training configuration includes information such as the name of the organism to be analyzed, the file where training sequences can be extracted, the order of the model and the number of states in start and stop profiles. `ESTScan 2.0` can be downloaded from directory `/sib-isrec/ESTScan/` on ftp://ftp.isrec.isb-sib.ch. Table 1 enumerates some of the files included in the distribution. Online queries can be submitted at http://www.ch.embnet.org following the link to `ESTScan2` on the entry page.

Training data and redundancy

For several species large amounts of sequence data are available. Although we concentrate on mRNA data for training, thousands of entries are found for the commonly analyzed species. We prefer to extract training data from curated databases like RefSeq (Pruitt and Maglott, 2001), but also use EMBL entries (Stoesser *et al.*, 2001) for species not found in RefSeq. However, a major issue with sequence data is redundancy.

Table 1. `ESTScan 2.0` distribution, a selection of files provided with short descriptions

Files	Description	I/O
`ESTScan`	main program	in: ESTs, model out: coding region
`maskred`	redundancy masker	in: unmasked, out: masked
`build_table`	extract data, split isochores adjust model	in: config-file(s) in: db-flatfiles out: model file
`*.conf` and `*.smat`	configuration/ model files	human, rat,mouse, fly, arabidopsis...

Most nucleotide sequence databases contain a considerable amount of redundancy for mainly two reasons:

- Evolutionary mechanisms: Duplication and mutation of bits of DNA to develop new functional elements is a very common mechanism in evolution. However, from an information theoretical point of view, this mechanism generates redundancy.
- Behavior of researchers: Researchers often work on similar, trendy topics and therefore analyze similar genes. Quite often, almost exactly the same sequence is submitted several times to nucleotide sequence databases.

When using redundant data for training hidden Markov models, a bias toward the overrepresented sequences is introduced. This bias is unfortunate, because we end up with a model, which is an expert for what we already know, but does not recognize the unknown sites or regions of interest. Therefore we suggest to remove chunks of redundant mRNA sequence from the training dataset.

Common techniques for removing redundancy from training sets are based on the removal of entire sequences which are highly similar to others in the set. A fast procedure to do this for protein sequences is presented in (Brendel, 1992). (Hobohm and Sander, 1994) contains a description of a redundancy avoiding selection method in the context of protein structures. We conjecture that in order to avoid biases in the training set also redundant pieces of sequences longer than a given threshold (our default is $m = 30$) should be masked. Since it is not possible to store all tuples longer than a threshold and thus detect reoccurrence exactly, we mask continuous runs of reoccurring tuples short enough to keep in memory (our default is $n = 12$). Our redundancy masking algorithm works as follows:

1. Initialize the current position to 0.
2. Find next reoccurring n-tuple, set a to its start and e to its end.
3. While the n-tuple ending at position $e + 1$ is reoccurring set e to $e + 1$.
4. If $e - a \geq m$ then mask nucleotides a to e.
5. Goto step 2 until the end of the training database is reached.

The size n of the reoccurring tuples detected is limited by the computer's memory. It is not possible to store occurrence of tuples substantially larger than 15. The minimum size m of runs to be masked should be chosen large enough to avoid masking biologically relevant signals and patterns.

It has been shown before, for instance in (Burge and Karlin, 1997), that for certain species codon usage depends on the overall GC-content of the sequence under investigation. Therefore `ESTScan` can split its training data into isochores, sets of sequences with similar GC-content. Redundancy reduction and training are then performed on these subsets independently, thus yielding a complete parameter set for each isochore. When `ESTScan` searches for coding regions in a new sequence, it first computes its GC-content and then sets parameters according to the appropriate isochore.

Simplified error tolerant EST-model

The model described in Section is complex and contains many states and transitions. We have chosen to implement a simplified version with the following features:

- Sequencing errors are neither modeled in untranslated regions nor in start and stop profiles.
- Start and stop profiles contain $2n$ states where n is the order of the model. These states have order 0.
- The coding region is modeled with insertion and deletion errors.
- Sequencing errors within the coding region are modeled without branches, thus no errors closer than n nucleotides to the stop profile are modeled.
- 5′ and 3′ untranslated regions are modeled with the same emission probabilities.

Let us give a few justifications for these simplifications. We have found that tuple distribution is not very characteristic for untranslated regions. Therefore in UTRs errors are unlikely to be distinguishable from noise. Similar to GENSCAN (Burge and Karlin, 1997), states in the start and stop codons have order 0. Errors less than n nucleotides before the first state of the stop profile are expected to be moved by a few nucleotides, causing only one or two wrong amino acids predicted. Finally, we have chosen to use the same emission probabilities for 5′ and 3′UTRs, since only little data on 5′UTRs is available.

Training and evaluation methods

We have developed a script which facilitates the training and evaluation of model parameters. The following steps are performed for the training task:

1. Extract mRNA entries from EMBL or RefSeq data files. Only entries with complete coding sequences annotated are accepted.
2. Split the data into user definable isochores.
3. Mask redundant pieces of sequence from training data.
4. Compute nucleotide usage tables for the mRNA model of Figure 1. Frequencies are used to estimate probabilities and converted to log-odds scores.

Unobserved tuples when training emission probabilities of high order models cause perfectly plausible pieces of coding sequence to be rejected. Small pseudocounts are an adequate means to cure this problem. Therefore we use 1 pseudocount by default in our training.

A second script has been devised to simplify the evaluation process. During evaluation we want to measure the following criteria to reflect the quality of our approach and the chosen parameters:

- False negative rate (sensitivity).
- False positive rate (specificity).
- Accuracy of start and stop site prediction.

These values are measured in test sets consisting of partially coding and entirely non-coding EST sequences. In our context, the false negative rate f_n is the ratio of partially coding ESTs predicted as non-coding among all partially coding ESTs in the test set. Likewise, the false positive rate f_p is the ratio of non-coding ESTs wrongly classified among all non-coding ESTs. Sensitivity is $1 - f_n$ and specificity is $1 - f_p$. In addition to the evaluation of `ESTScan` in (Iseli *et al.*, 1999), sensitivity and specificity are not only computed per sequence but also per nucleotide. For computing false positive and negative rates on nucleotide level, each nucleotide is classified as false or true negative or positive. Evaluation of start/stop-site detection is done by measuring the distances between annotated and predicted sites.

The following steps are performed by the above mentioned script for evaluation:

1. Find UniGene clusters corresponding to an mRNA set not used for training.

2. Match ESTs from these UniGene clusters to corresponding mRNAs to choose ESTs for evaluation and to annotate start and stop sites.
3. Predict coding regions.
4. Compute sensitivity and specificity on nucleotide and sequence level.
5. Compare predicted and annotated start and stop sites.

Choosing ESTs for evaluation is a delicate task because ESTs are often redundant, sometimes of poor quality and never annotated with features such as start and stop sites. In order to cover with our choice most of the mRNAs put aside for evaluation, we rely on UniGene (Schuler, 1997). In UniGene clusters for the given mRNAs, we find ESTs which can be aligned with these. By choosing only ESTs with very good alignments with their mRNA (`megablast` E-value lower than 1e-20, less than 5% mismatches), we avoid poor quality ESTs. By choosing per mRNA exactly one EST aligned to parts of the coding region and one aligned to the mRNA's UTR only, we avoid heavy redundancy. Finally, the alignment with the corresponding mRNA allows us to annotate the start and stop sites for the chosen ESTs.

After the prediction of coding regions through the hidden Markov model, specificity and sensitivity can be computed on the nucleotide as well as the sequence level. Finally also the distances between predicted and annotated start and stop sites are evaluated in histograms and pie charts.

EVALUATION

In this section we elaborate, if the new version of `ESTScan 2.0` substantially improves coding region prediction compared to its predecessor.

Training and evaluation data

In order to compare the two modeling approaches of `ESTScan` versions 1 and 2, we use the same data, redundancy reduction and isochore splitting. The performance gain has been measured using human data. mRNA sequences are needed in training to compute codon frequencies and start/stop profiles. We have extracted 17'037 human mRNA entries from the RefSeq database. These curated entries have little or no redundancy due to duplicate submission or the like and are expected to have careful annotations of coding regions.

Similar to GENSCAN, we have split the mRNA data into four isochores: GC-content below 43%, 43% to 47%, 47% to 51%, and above 51%. The high GC-content isochore turns out to be the largest containing 48% of all mRNA sequences. The low-GC-content isochore contains 24% and the middle Isochores 15% and 14% respectively. Redundancy reduction was performed with a minimum mask-length of 30 nucleotides. It masks a surprisingly high amount of data. In fact, the amount of redundancy reduction is strikingly high for high GC-content (24%) and still considerable for the other isochores (12% to 14%). The presence of many splice variants in RefSeq may explain this high degree of redundancy, since our redundancy masking approach masks repetitions of common exons.

In order to evaluate the performance of both versions of `ESTScan`, EST data with information about coding regions they contain have to be made available. For the 17 037 mRNAs 15 719 UniGene clusters have been retrieved. Together, these contain 940 793 ESTs each of which matches one mRNA. According to the pairwise alignment of each EST with its corresponding mRNA, 14 554 coding sequence containing ESTs with annotated start and/or stop sites and 12 284 ESTs containing only untranslated regions have been determined.

Evaluation setup

In order to separate evaluation and training in an unbiased manner, we evaluate in a tenfold cross validation like setup. The 17 037 mRNAs are split into ten buckets of equal size in a round robin process, that is, mRNA-bucket i contains mRNAs number $i + 10n, n < 1703$. For each of ten evaluation runs nine of these buckets are used as training set for the two `ESTScan` versions, while the ESTs linked to the tenth bucket are used for evaluation. Thus EST sequences used for evaluation have no direct link to mRNAs used in training.

The predictions have been computed with default parameters in both version of `ESTScan`. Table 2 shows all penalties which have been used. These values have been fine-tuned by hand using rat data. Moreover, predictions smaller than 50 nucleotides have been filtered in both programs, since we consider them as noise. In addition, `ESTScan 1.3` uses 0.1 as the expected false positive rate on random sequences (see (Iseli *et al.*, 1999) for explanations). This parameter has no more meaning in `ESTScan 2.0`.

In order to evaluate the accuracy of coding region predictions, we have computed sensitivity and specificity per nucleotide and per sequence as well as the precision in start and stop site detection. Specificity per sequence is computed using 12'284 entirely untranslated ESTs by counting the predictions generated on these. Sensitivity per sequence is computed likewise on the 14'554 ESTs containing coding parts. The measures on nucleotide levels are computed by counting misclassified nucleotides across both EST sets.

Although the sensitivity and specificity on a nucleotide level also partially reflect the accuracy of start and stop site prediction, we have computed distances between predicted

Table 2. ESTScan default penalties, also used in evaluation

Transition penalties				
	To:			
From:	5′UTR	CDS	3′UTR	END
START	−10	−10	−5	-
5′UTR	0	−80	-	−40
CDS	-	0	−80	−40
3′UTR	-	-	0	−20
Error penalties:				
Insertion:		50		
Deletion:		50		
Stop in coding region:		100		

Table 3. Compare prediction accuracy of ESTScan's previous and new version on EST data. Sensitivity and specificity are computed per nucleotide (nt.) and per sequence (sq.)

	ESTScan 1.3	ESTScan 2.0
Sensitivity (nt.)	78.1%	94.5%
Specificity (nt.)	53.7%	69.2%
Sensitivity (sq.)	79.5%	97.1%
Specificity (sq.)	63.3%	55.9%

and annotated start and stop sites in order to illustrate this aspect of prediction. Frequencies of given distances show in more detail, how the predictors behave close to the boundaries of coding regions.

Results

The results computed for sensitivity and specificity are shown in Table 3. When considering the measurements on nucleotide level ESTScan 2.0 outperforms its predecessor. However, specificity of both versions is rather low. The number of false positives has actually slightly worsened for the new ESTScan version when considering the per sequence level. The excellent performance in sensitivity suggests fine-tuning parameters in order to lower the rate of false positives at the cost to loose some sensitivity.

The weak point of ESTScan 1.x has always been the exact detection of start and stop sites. The improvement achieved through explicitly modeling start and stop sites as suggested in ESTScan 2.0 is illustrated in Figure 6.

The most striking improvement is the increase in exact matches, that is, cases where the prediction coincides with the annotation. While the first version of ESTScan is only able to find start sites exactly in 17.1% of the cases, ESTScan 2.0 does so in almost two thirds of the coding regions. For stop sites the new version brings the ratio of exact matches from 6.4% to 55.1%. ESTScan 2.0 provides predictions closer than 50 nucleotides to the annotation for start and stop sites in more then three quarters of the cases.

DISCUSSION

In this article we have shown a systematic approach to introduce error tolerance into a hidden Markov model. In order to improve such an error-tolerant model, it is enough to improve the underlying HMM. Thereby we concentrate on insertions and deletions since their correction is crucial in predicted coding region translation. Since our approach is particularly interesting for feature prediction in error prone EST sequences, we have used it in the newest version of ESTScan to detect coding regions in ESTs. Instead of focusing only on modeling the coding regions, the new version's model covers the whole messenger RNA. Compared to ESTScan's predecessor, normalization on window segment shuffled sequences is no longer needed. ESTScan now relies on a Viterbi optimal path algorithm to determine coding regions instead of using a cut-off for a coding potential score.

Our approach to error modeling is similar to the one used in earlier versions of ESTScan. It is limited to the modeling of sequencing errors with pairwise distances larger than the order of the HMM, although the probability for this to happen is quite high: Given an error rate of $p_e = 1\%$ the probability to encounter at most 1 error in an 6-tuple (for model order $n = 5$) is

$$p_t = (1 - p_e)^{n+1} + (n + 1)\, p_e (1 - p_e)^n = 99.85\%$$

When more than one error is observed in a 6-tuple these would be at least as close as $n = 5$ nucleotides and can thus not be modeled. The probability for this to occur is $(1 - p_t)$. In a typical EST of length 600 there are 100 tuples where this can happen independently, hence the probability for errors at distance closer or equal to 5 is higher than $1-(1-p_t)^100 = 13.6\%$. However, we do not expect much harm of the resulting prediction errors. The reading frame is lost for very few nucleotides only. Two close deletions are modeled by one insertion, two close insertions by one deletion and the close combination of deletion/insertion is just ignored. All of these situations lead to very few misinterpreted nucleotides and just two or three wrong amino acids. Moreover, even if we modeled two close indels, such a pair of errors would be penalized so heavily that the Viterbi algorithm would always predict one or no error instead. For similar reasons it is not possible to detect substitution errors. Similar to ESTScan 1.x we only treat the case when substitutions lead to spurious stop codons by allowing stops with small probabilities within coding regions.

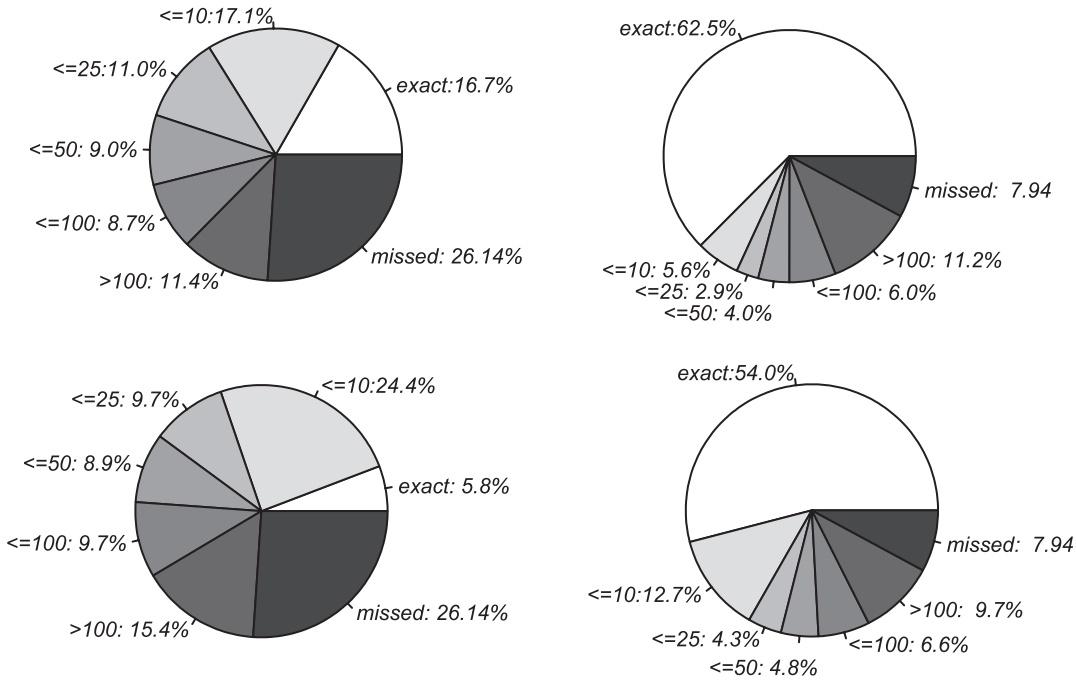

Fig. 6. Compare start (top) and stop site (bottom) prediction accuracy of ESTScan's previous (left) and new version (right) on EST data. The pie charts represent the fraction of predictions within the given distance range.

We have shown in our evaluation that the performance achieved by our more complete model is promising for start and stop site prediction as well as for detecting coding regions. We have shown in a cross validation experiment on human data that ESTScan 2.0 can exactly predict 64.2% of start and 55.1% of stop sites. Moreover, high sensitivity of 94.5% on nucleotide level is observed. In these disciplines the new version of ESTScan outperforms its predecessor. In contrast, the specificity of 69.2% needs improvement. Apparently spurious hits of the start or stop profile trick the new approach into misinterpreting short runs of untranslated regions as coding. However, some of the cases here counted as false positives may actually be not yet known coding regions.

Also a new attempt to cope with redundancy in training data has been proposed. While classical approaches only eliminate redundancy by removing entire sequences, we explore the usefulness of masking redundant pieces of sequence. This approach, however, raises the question, how and if elimination of conserved signals, domains, pieces of splice variants and other patterns should be performed. During our tests we have observed no dramatic changes in predictions due to redundancy reduction. Even when extracting training data from nucleotide databases with considerable amounts of redundancy and actually removing up to 24% of the training set, very small changes in discrimination potential are observed. However, evaluation of redundancy reduction is particularly difficult, since we can only evaluate our models on known data. The models which best fit this possibly highly redundant data, will get best marks, while those trained on less redundant data will tend to perform weaker on redundant evaluation data. Nevertheless, they may represent more accurately the biological issue to be modeled and therefore perform better on new data.

The work presented here is limited to a simplified version of the error-tolerant model proposed. Modeling of errors in the start and stop-profiles as well as in the *n* nucleotides before the stop profile should be attempted and evaluated. Furthermore, the use of order zero states in start and stop profiles causes the following weakness: about 10% of the predicted coding sequences end with TGG, since this combination of nucleotides scores high in our stop profile. Modeling start and stop sites using order 1 or 2 states may correct this weakness. Further suggestions for improvement include more dynamic choice of error penalties according to the current position in the sequence or quality factors from trace files.

ACKNOWLEDGMENTS

The authors would like to thank Anders Krogh and his colleagues from the Center for Biological Sequence Analysis of the Technical University in Denmark for helpful comments and discussions.

REFERENCES

Adams,M.D., Kelley,J.M., Gocayne,J.D., Dubnick,M., Polymeropoulos,M.H., Xiao,H., Merril,C.R., Wu,A., Olde,B., Moreno,R.F. *et al.* (1991) Complementary DNA sequencing: expressed sequence tags and human genome project. *Science*, **252**, 1651–1656.

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

Brendel,V. (1992) PROSET—a fast procedure to create nonredundant sets of protein sequences. *Math. Comp. Modell.*, **16(6/7)**, 37–43.

Brown,N.P., Sander,C. and Bork,P. (1998) Frame: detection of genomic sequencing errors. *Bioinformatics*, **14(4)**, 367–371.

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

Camargo,A.A., Samaia,H.P.B., Dias-Neto,E., Simo,D.F., Migotto,I.A., Briones,M.R.S., Costa,F.F., Aparecida Nagai,M., Verjovski-Almeida,S., Zago,M.A. *et al.* (2001) From the cover: the contribution of 700 000 ORF sequence tags to the definition of the human transcriptome. *Proc. Natl Acad. Sci. USA*, **98**, 12103–12108.

Fichant,G.A. and Quentin,Y. (1995) A frameshift error detection algorithm for DNA sequencing projects. *Nucleic Acid Res.*, **23(15)**, 2900–2908.

Fickett,J.W. (1996) Finding genes by computer: the state of the art. *Trends Genet.*, **12(8)**, 316–320.

Guan,X. and Überbacher,E.C. (1996) Alignments of DNA and protein sequences containing frameshift errors. *Comput. Appl. Biosci.*, **12(1)**, 31–40.

Hatzigeorgiou,A.G. (2002) Translation initiation start prediction in human cDNAs with high accuracy. *Bioinformatics*, **18(2)**, 343–350.

Hatzigeorgiou,A.G., Fiziev,P. and Reczko,M. (2001) DIANA-EST: a statistical analysis. *Bioinformatics*, **17(10)**, 913–919.

Hobohm,U. and Sander,C. (1994) Enlarged representative set of protein structures. *Protein Sci.*, **3(3)**, 522–524.

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

Kozak,M. (1987) An analysis of 5′-noncoding sequences from 699 vertebrate messenger RNAs. *Nucleic Acid Res.*, **15(20)**, 8125–8148.

Krogh,A., Mian,I.S. and Haussler,D. (1994) A hidden Markov model that finds genes in *E.coli* DNA. *Nucleic Acid Res.*, **22(22)**, 4768–4778.

Pruitt,K.D. and Maglott,D.R. (2001) RefSeq and LocusLink: NCBI gene-centered resources. *Nucleic Acid Res.*, **29(1)**, 137–140.

Rabiner,L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77(2)**, 257–285.

Salzberg,S., Delcher,A., Kasif,S. and White,O. (1998) Microbial gene identification using interpolated Markov models. *Nucleic Acid Res.*, **26(2)**, 544–548.

Schuler,G.D. (1997) Pieces of the puzzle: expressed sequence tags and the catalog of human genes. *Mol. Med.*, **75(10)**, 694–698.

Stoesser,G., Baker,W., van den Broek,A., Camon,E., Garcia-Pastor,M, Kanz,C., Kulikova,T., Lombard,V., Lopez,R., Parkinson,H., Redaschi,N., Sterk,P., Stoehr,P. and Tuli,M.A. (2001) The EMBL nucleotide sequence database. *Nucleic Acid Res.*, **29(1)**, 17–21.

Strausberg,R.L., Buetow,K.H., Emmert-Buck,M.R. and Klausner,R.D. (2000) The cancer genome anatomy project: building an annotated gene index. *Trends Genet.*, **16(3)**, 103–106.

Sze,S.U. and Pevzner,P.A. (1997) Las Vegas algorithms for gene recognition: suboptimal and error-tolerant spliced alignment. *Comput. Biol.*, **4(3)**, 297–309.

Xu,Y., Mural,R.J. and Überbacher,E.C. (1995) Correcting sequencing errors in DNA coding regions using a dynamic programming approach. *Comput. Appl. Biosci.*, **11**, 117–124.