Journal of Bioinformatics and Computational Biology

ANGLE: A sequencing errors resistant program for predicting protein coding regions in unfinished cDNA.

Kana Shimizu

Department of Computer Science, Graduate school of Waseda University
17 301 kikui-cho, Shinjuku-ku, Tokyo, 162-0044, Japan
kana@muraoka.info.waseda.ac.jp

Jun Adachi

The Institute of Statistical Mathematics
4-6-7 Minami-Azabu Minato-ku Tokyo, 106-8569, Japan
Department of Biosystems Science, The Graduate University for Advanced Studies
240-0193 Shonan Village Hayama Kanagawa, Japan
adachi@ism.ac.jp

Yoichi Muraoka

Department of Computer Science, Graduate school of Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan
muraoka@waseda.jp

Received (24 5 2005)
Revised (18 9 2005)
Accepted (20 11 2005)

In the process of making full-length cDNA, predicting protein coding regions helps both in the preliminary analysis of genes and in any succeeding process. However, unfinished cDNA contains artifacts including many sequencing errors, which hinder the correct evaluation of coding sequences. Especially, predictions of short sequences are difficult because they provide little information for evaluating coding potential. In this paper, we describe ANGLE, a new program for predicting coding sequences in low quality cDNA. To achieve error-tolerant prediction, ANGLE use a machine-learning approach, which makes better expression of coding sequence maximizing the use of limited information from input sequences. Our method utilizes not only codon usage, but also protein structure information that is difficult to use for stochastic model-based algorithms, and optimizes limited information from a short segment when deciding coding potential, with the result that predictive accuracy does not depend on the length of an input sequence. The performance of ANGLE is compared with ESTSCAN on four dataset each of them has a different error rate (one frame-shift error or one substitution error per 200-500 nucleotides) and on one dataset which has no error. ANGLE outperforms ESTSCAN by 9.26% in average Matthews's correlation coefficient on short sequence dataset (<1000 bases). On long sequence dataset, ANGLE achieves comparable performance.

Keywords: cDNA; sequencing errors; AdaBoost; coding sequence; EST

1. Introduction

The study of cDNAs remains an essential approach for structural and functional genome annotations. Especially, full-length cDNA-sequencing is a powerful tool for accurate annotations, and many large-scale projects have been successful[1,2,3].

Several time-consuming steps are required to determine full-length cDNAs[4]. Before the finishing step, predicting protein coding regions in uncorrected sequences helps in both the preliminary analysis of genes and any succeeding process. However, these sequences are such low quality that some sequences are truncated, and many sequences have considerable sequencing errors including deletion/insertion errors that lead to frame-shift errors, and substitution errors that can introduce unexpected stop codons in protein coding regions . We can expect about one frame-shift error or one substitution error in every 200-500 nucleotides. To make prediction accurate, such errors must be corrected; doing so also helps in designing primers. Moreover, to check whether a sequence is truncated is important.

Thus, an error-tolerant method is indispensable for predicting the coding sequence in unfinished cDNA.

1.1. *Related works*

Many studies related to the gene-finding problem have been conducted[5]. Here, we categorize those especially related to our study into three areas.

(1) Detecting frame-shift errors.
Pro-Frame[6] is a similarity-based program. FSED[7], Xu *et al.*'s work[8] and FrameD[9] are based on statistical models of codon frequency. These studies focus on detecting frame-shift errors rather than predicting protein-coding regions.

(2) Predicting coding regions from DNA.
There are mainly two approaches for gene finding. One is based on noncomparative methods such as Markov chain models which is powerful models for defining coding regions of a new DNA sequences. GeneMark[10] and Glimmer[11] are programs for predicting genes in prokaryotic DNA, while GENSCAN[12] and GlimmerM[13] are those for eukaryotic DNA.

The other approach is identifying coding frames by comparative analysis of homologous transcripts. CRITICA[14] is a hybrid algorithm of noncomparative methods and comparative methods. CSTminer [15] is based on cross-species genome comparisons, which can also identifying noncoding conderved sequences.

These programs do not detect sequencing errors, because their target is high-quality DNA.

(3) Predicting coding regions from cDNA, EST.
ESTSCAN[16,17] is a Hidden Markov Models (HMM) based program for Expressed Sequence Tags (ESTs)[18], or cDNAs, that can detect sequencing errors. DECODER[19] is a program based on a simple method of scoring codon usage

that can account for the quality of input sequences to detect sequencing errors.

Among related works, DECODER was mainly used to analyze mouse cDNA sequences in RIKEN Mouse Genome Encyclopedia Project[1], while GeneMark has been used for human cDNA sequences in the Kazusa cDNA sequencing project which gave rise to HUGE database[3].

1.2. *Modeling of coding sequences*

Our goal is to correct sequencing errors and to predict coding regions simultaneously. If additional information, such as sequencing quality of each base is available, positions where sequencing errors occurs may be predictable without using the general feature of protein coding sequence (CDS). DECODER used such additional information to help prediction. However, sequencing quality is not always available for every researcher. Therefore, prediction from only sequences data is more convenient to users.

The natural strategy to predict CDS using only sequences data is extending a modeling of CDS to treat sequencing errors. For example, ESTSCAN just adds insertion and deletion states to hidden Markov model of dicodon, while DECODER evaluates all Open Reading Frames via codon usage, after it artificially insert/delete bases into/from target sequences. Namely, prediction of sequencing errors positions depends on a basic modeling of CDS without sequencing errors. Thus better expression of CDS is essential for predicting both the coding regions and sequencing errors positions. Especially on short sequences, correct evaluation of coding potential is difficult because they provide little information. To overcome the shortage, the modeling method must maximize the use of limited information.

Various approaches have been taken to modeling coding sequences[5]. Particularly, Markov chain models have been used frequently[20]. In the ideal case of having a training sequence of unlimited length, a higher-order Markov model is a better predictor [20], because the model has more information with which to indicate biological knowledge. For example, an 8th-order model can express the correlation among three adjacent amino acids, while a 5th-order model can only express the correlation between two adjacent amino acids.

However, many tools cannot use higher-order models, even though they are powerful, because higher-order models require a much larger amount of training data to estimate reliable parameters in real cases[20]. If the order is m and the length of a training sequence is N, only $N-m$ strings of size $m+1$ are available to estimate the parameters related to no less than 4^m types of oligonucleotides. Therefore, an attempt to derive a model of too high an order will result in overfitting. For this reason, instead of a higher-order model, many gene-finding tools rely on a fixed 3-periodic-5th-order Markov model that forms dicodon compositions. Of the tools mentioned above in section 1.1, GENSCAN, GENEMARK, and ESTSCAN use this model.

To cope with sparse data, a variable-order Markov model is discussed in a report

about the interpolated Markov model (IMM)[11]. Glimmer is an implementation of IMM that combines several Markov models from order zero to given order k to decide transition probabilities according to the available training-sequence information. Also, an interpolated context model (ICM) is introduced in GlimmerM [13] that is a more sophisticated model of IMM. Still, IMM and ICM cannot model interrelated information simultaneously (e.g., motifs and codon usage are interrelated information), because they are based on a stochastic model.

Programs that are not based on a Markov model usually use statistical models of codon usage or dicodon usage, as do other programs based on a Markov model. However, many other kinds of biological knowledge are available, such as information about a composition of three or four adjacent amino acids that shows interaction between separate amino acids in a protein secondary structure. Such compositions occur because protein coding sequences will be translated into amino acid sequences that form protein secondary structures in cells. We think this kind of information can help making a more efficient model of coding sequences.

In this paper, we propose a hybrid approach of machine learning and Markov model which optimizes the information of input sequences utilizing not only codon usage but also protein structure information. We used statistical bias of diamino acid k-composition (composition of a pair amino acids A, B. A is located k residues away from B. If $k = 1$, that means the composition of adjacent amino acids) as a measure of protein structure. To calculate optimal coding potential, we used boosting algorithm [21] that can produce an accurate prediction rule by combining rough and moderately inaccurate rules.

In the following section, we describe the proposed method and draw conclusions after evaluating the performance of ANGLE when it was used to analyze four human mRNA dataset each of them has a different error rate (one frame-shift error or one substitution error against 200-500 nucleotides) and one human mRNA dataset which has no error. The performance of ANGLE is 9.26% better than that of ESTSCAN in average Matthews's correlation coefficient on short sequence dataset (<1000 nucleotides). On long sequence dataset, ANGLE achieves comparable performance.

2. Methods

2.1. *Overview of ANGLE*

ANGLE has three steps. In the first step, ANGLE calculates coding potential of all codons using the information of a short region around the target codon. (All codons means all codon of every frame of an input sequence.) Short regions are brought using sliding-window. All codons are labeled CDS or ELSE according to its coding potential. In the next step, the most probable path is traced using a Markov chain model with dynamic programming. And then, positions where a frame is changed are detected as rough positions of frame-shift errors in the path. In the final step, each rough position is modified by selecting the most probable position

Fig. 1. Overview of ANGLE. Step 1 : An input sequence is divided into short segments via sliding-window, and each window is labeled CDS or ELSE. Step 2 : The optimal path is calculated using the Markov Model. Step 3 : Find a frame-shift error from each candidate area.

from candidate positions located near the rough positions. Fig.1 shows an image of the process.

2.2. *Utilizing protein structure information*

Since a protein forms a structure, those effects can be observed in amino acid sequence as statistic bias. Especially, we paid attention to the interaction between one amino acid and other amino acids located several residues away, because such interactions occur on structures. For example, on an alpha helix structure, hydrogen bonds occur between positions i and i+4 that make a compositional bias of diamino acid 4-composition. The same things can be seen on loop structure, beta sheet structure, or coils. Each structure has its own frequent feature of amino-acid sequence, and some previous works have discussed predicting secondary structures[22,23]. However, our goal is not predicting structures but coding regions. Therefore, all diamino acid k-compositions are used for scores, so as to obtain comprehensive features of secondary structures.

Fig. 2 show the strong bias of diamino acid k-composition in CDS against that in random sequences. Following Score E_k is used in the graph for comparison. If there is no correlation between amino acid A and B in the pair, each amino acids appears independently and the joint probability $P(A, B)$ should be $P(A) \cdot P(B)$. If not, E_k should be a large value.

$$D_k(A, B) = \left| \frac{P(A, B)}{P(A) \cdot P(B)} - 1.0 \right|$$

$$E_k = \sum_i \sum_j (D_k((A_i, B_j))^2$$

(B is an amino acid k residues away from A.)

In the graph, k=1,. . . ,4 shows high scores comparing to the others. A pair acids with larger number of k is less informative because effect of k=1,. . . ,$n-1$ are mixed when k=n. Table 1 shows top 20 pairs of score D_k.

Detail use of diamino acid k-composition is shown in section 2.5.

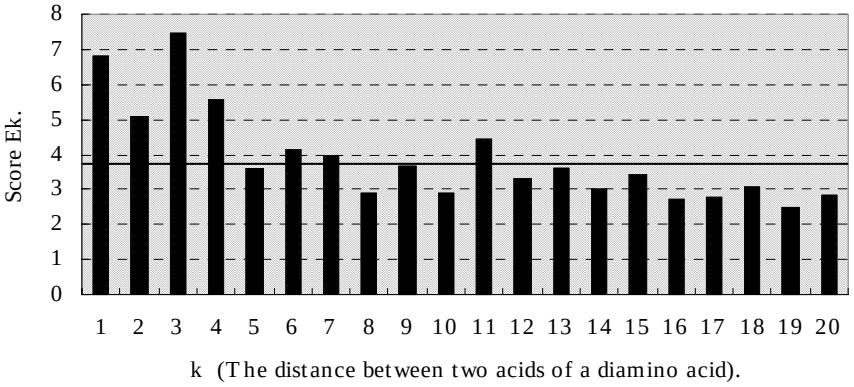

Fig. 2. Comparing diamino acid k-compositions of CDS with those of random sequences. The cross line on the graph shows an average value of all. Scores are calculated on human mRNA entries from RefSeq. Detail information is shown in Table 2.

Table 1. Top 20 pairs of score D_k.

top x	k	pairs	score	top x	k	pairs	score
1	3	Cys-Cys	1.69	11	3	Trp-Trp	0.47
2	4	His-His	0.85	12	2	Trp-Trp	0.45
3	4	Cys-Cys	0.71	13	2	Lys-Lys	0.45
4	2	Cys-Cys	0.71	14	4	Cys-Phe	0.44
5	3	Pro-Pro	0.69	15	1	Lys-Lys	0.43
6	2	Pro-Pro	0.68	16	2	Gln-Gln	0.43
7	4	Pro-Pro	0.62	17	1	Ala-Ala	0.42
8	3	Gly-Gly	0.6	18	1	Trp-Trp	0.42
9	1	Glu-Glu	0.58	19	3	Lys-Lys	0.42
10	1	Pro-Pro	0.52	20	2	Tyr-Cys	0.41

Scores are calculated on human mRNA entries from RefSeq. Detail information is shown in Table 2.

2.3. *AdaBoost*

We used AdaBoost[24] to classify each short segment of an input sequence as either CDS or ELSE. AdaBoost is a meta-learning algorithm that repeatedly runs a simple

learning algorithm called the weak learner on the same training data and then combines its hypotheses into one final hypothesis to achieve higher accuracy than a single weak learner hypothesis would have. The main idea of AdaBoost is to assign each example of the given training set a weight. At the beginning all weights are equal, but in each round the weak learner returns a hypothesis, and the weights of all examples classified wrongly by that hypothesis are increased. In that way, the weak learner is forced to focus on the difficult examples of the training set. The final hypothesis is a combination of the hypotheses of all rounds, namely a weighted majority vote in which hypotheses with lower classification errors have higher weight.

We define each example as a pair of x_i and y_i, while $X = \{x_i, \ldots, x_n\}$ is a set of short segments of an input sequence, and $Y = \{y_i, \ldots, y_n\}$ is a set of labels of X with $y \in \{1, -1\}$. (If x_i is CDS, $y_i = 1$; otherwise $y_i = -1$.).

2.4. *Weak learner*

The weak learner of AdaBoost is an algorithm that produces an appropriate classifier in every boosting iteration. We use a very simple algorithm WL that tries several prepared classifiers $C_i\{i = 0, \ldots, N\}$ on examples S and select the most accurate of all under the condition that each example (x_j, y_j) has a different weight $w_{t,j}$, as follows.

$$S = \{(x_1, y_1), \ldots, (x_n, y_n)\}.$$
$$\mathrm{WL}(S) = \mathrm{argmax}(C_i(S)), (i = 0, \ldots, N).$$

Prepared classifiers C_i have a score V_i based on biological knowledge and produce an optimal threshold $T_{i,t}$ in every boosting round t. Values of scores V_i are calculated from the sequence information of x_i, and then x_i is classified according to $T_{i,t}$.

In the next section, we explain how the score V_i is calculated.

2.5. *Scores*

We prepared three types of scores for weak learner, (1) a score based on amino acid composition, (2) a score based on codon composition, and (3) a score based on diamino acid k-composition. We obtained 21 scores from (1), including all amino acids plus stop codons, 64 scores from (2), and $n \times 21 \times 21$ from (3). (n is the number of k. If k=1 and k=2 are selected, n=2.) Comments on the scores are given below.

(1) Amino acid composition
Needless to say, the compositional bias of amino acids is a basic indicator for detecting coding sequence. The scores counted all amino acids that appear in the sample.

(2) Codon composition
Several codons were translated into the same amino-acid. For example, both

CAT and CAC were coded as histidine. Such codon redundancy results in compositional bias, because mutations that do not change the amino acid translation are accumulated more easily. Thus, this is helpful knowledge for distinguishing coding sequence from untranslated regions. Scores are calculated as follows.

$$P(C_i)/P(\alpha(C_i))$$

(i = 0,. . . ,63, $\alpha(C_i)$ is an amino acid translated by codon C_i).

(3) Diamino acid k-composition
To avoid the effect of amino acid composition, we normalize the joint probability by each amino acid composition as follows.

$$\frac{P(A_i, B_j)}{P(A_i) \cdot P(B_j)}$$

(i=0,. . . ,20, j=0,. . . ,20, A_i and B_j stand for one of twenty amino acids or a stop codon.)

2.6. *Markov model*

Now we have three frames that have all codons labeled CDS or ELSE. The next mission is to find the optimal path from frames. Fig. 3 shows the topology of a Markov model. We temporarily detect frame-shift errors at the positions where a frame is changed on the final path.

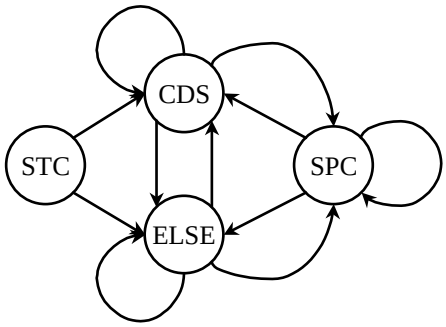

Fig. 3. The topology of the Markov model. Four states (CDS, ELSE, STC (start codon), SPC (stop codon)) are defined, since each codon is classified into CDS or ELSE if the codon is not a stop codon. 'ATG' is alternatively labeled STC, CDS or ELSE according to the current score.

2.7. *Modifying frame-shift error positions*

Since ANGLE uses sliding-windows, the classification task is vague on the border between CDS and ELSE. To overcome this problem, ANGLE corrects a temporal frame-shift position p in a way that compares all candidate positions near p. That

is, ANGLE deletes bases so as to correct detected frame-shifts at the positions from p-w to p+w and recalculates the boosting score of a window of p; and then it chooses the position that returns the best scores.

2.8. *Parameters*

For the classification process, we have to fix window size. The longer the size, the higher the accuracy of classification is because we can obtain more information from samples. However, the longer window is less sensitive to frame-shift positions. ANGLE uses a heuristic value of 32 codons for window size as a default parameter.

The next parameter is k for diamino acid k-composition. Considering E_k of Fig. 2, we set k=1,...,4.

On tracing the best path process, we have to fix some parameters for DP calculation of Markov chains. We set heuristic values of 10 for the start value, -20 for the frame changing penalty value. If the 'ATG' appears on the path and the current score is lower than the start value, the 'ATG' is set to be a start codon, and the current score is reset to 10, if not, the 'ATG' is counted as just a 'ATG'.

2.9. *Pseudocounts*

In ANGLE, the system has to classify short segments of the target sequence by using related information that is not always sufficient. For example, scores for codon composition require probabilities $P(\alpha(C_i))$ for a denominator. If a target segment has strong bias for $\alpha(C_i)$, the value is 0 and score will be infinite. Moreover, we have 441 pairs over 20 amino acids and a stop codon; this high number requires a longer window to obtain sufficient information. To make up for any shortage, we employ pseudocounts which modifies probabilities using background distribution. For example, when a normal probability of amino acid or a stop codon A_i (i=0,...,20) on a given window is calculated,

$$P(A_i) = \frac{\text{Counts A}_\text{i} \text{ on the window}}{\text{windowsize}}$$

is used. Instead of using it, a probability with pseudocounts $P'(A_i)$ is calculated as follows.

$$\frac{\text{counts A}_\text{i} \text{ on the window} + \alpha * \text{counts A}_\text{i} \text{ on learning data}}{\text{windowsize} + \alpha * \text{ learning data size}}$$

(α is an arbitrary parameter).

2.10. *Separate learning from GC pressure*

Some previous works[12,16,17] reported GC compositional bias. Thus, ANGLE divides data into four according to GC pressure: $p < 43\%$, $43\% \leq p < 51\%$, $51\% \leq p < 57\%$, $57\% \leq p$, then trained classifiers separately.

3. Materials

We extracted human mRNA entries from RefSeq database[25]. Table 2 presents details of the data. Since RefSeq data has high quality and no sequencing errors, we artificially insert/delete bases into/from original RefSeq data at random positions to evaluate the accuracy of frame-shift detection. We also artificially mutate bases at random positions because low quality sequence data contains not only frameshifts but also base substitutions, which can make unexpected stop codons in CDS. We expected about one sequencing error (one frame-shift error or one substitution error) in 200-500 nucleotides. Four types of data were prepared according to a ratio of errors (i.e. Data1, Data2, Data3, Data4 are just the same data with a different error occurrence). Table 3 presents details of those data.

Table 2. Details of original RefSeq data.

Download date Jan, 2005	Taxonomy Homo Sapiens		
Num of Seqs 28,671	Num of bases 74,953,258	Num of bases in CDS 43,781,010	
Num of Seqs of $l < 1000$ 4,756	$1000 \leq l < 3000$ 15,260	$3000 \leq l$ 8,655	
Num of Seqs of $p < 43\%$ 6,289	$43\% \leq p < 51\%$ 8,706	$51\% \leq p < 57\%$ 6,007	$57\% \leq p$ 7,669

l: length of sequence. p: GC pressure of sequence.

Table 3. Details of experimental data.

	Ins	Del	Sub	FR	SR	Rstart	Rstop	Astop
Data1	21,941	21,842	43,879	1.00	1.00	168	171	1,429
Data2	27,043	27,493	54,719	1.25	1.25	218	241	1,711
Data3	36,278	36,743	72,986	1.68	1.68	261	287	2,256
Data4	54,615	54,649	109,526	2.50	2.50	449	429	3,519

Ins: the number of insertions in CDS. Del: the number of deletions in CDS. Sub: the number of substitutions in CDS. FR: the number of frameshifts per 1000 nucleotides. SR: the number of substitutions per 1000 nucleotides. Rstart: the number of start codons which is replaced by other codon because of artificial errors in CDS. Rstop: the number of stop codons which is replaced by other codon because of artificial errors in CDS. Astop: the number of stop codons which appear in CDS because of artificial substitutions.

4. Training

As we described in Section 3, ANGLE is based on hybrid algorithm of boosting and Markov model. A classifier (boosting) determines coding potential of each codon, and Markov model traces a final path. While the classifier requires training data with no error, Markov model requires data with errors for calculating parameters for transition probability. Therefore classifier was trained and related parameters were calculated on training data, which is extracted from original RefSeq. Parameters of Markov chains were calculated on training data, which is the same source as the data for training classifier, but has errors. We used training data which has one frameshift and one substitution per 500 nucleotides.

5. Evaluation methods

We used 5-fold cross validation for experimental evaluation. Experimental data is randomly divided into training data and evaluating data. The following criteria were used to evaluate ANGLE.

- Sensitivity per nucleotide
- Specificity per nucleotide
- Correlation coefficient per nucleotide

Since sensitivity and specificity are trade-off criteria, we needed a balancing criterion, the Matthews' correlation coefficient[26], which was calculated as follows.

$$\frac{(tn * tp) - (fn * fp)}{\sqrt{(tp + fp) * (tn + fn) * (tp + fn) * (tn + fp)}}.$$

(tp: true positive, tn: true negative, fp:false positive, fn: false negative).

For comparison, the predictions were computed not only with ANGLE but also with ESTSCAN2.0[17]. Both tools were executed with default parameters. We did not try DECODER[19] because it requires sequencing quality of every bases of all data, which we cannot acquire.

6. Result

The results of computing for sensitivity, specificity, and the correlation coefficient of coding regions predictions are given in Table 4. It is obvious that performance of both ANGLE and ESTSCAN2.0 are better for low error rate data, since high quality data is easy to evaluate coding potential. For original data, Data1, Data2 and Data3, ANGLE got better result than ESTSCN2.0, while its performance was almost the same for Data 4. Since ESTSCAN2.0 is design mainly for ESTs, it is tuned up to suit lower quality data.

The most remarkable result is the predictions of short sequences. Table 5 shows each correlation coefficient separately computed according to the length of sequences. ANGLE performs about 9.26% better than ESTSCAN2.0 on sequences

Table 4. Comparing prediction accuracy of ANGLE and ESTSCAN2.0. All scores are computed per nucleotide.

	Sensitivity	Specificity	Correlation coefficient
Original Data (no sequencing error.)			
ANGLE	96.65%	97.97%	92.96%
ESTSCAN2.0	93.12%	96.88%	88.28%
Data1(1 frameshift or 1 substitution per 500 nucleotides.)			
ANGLE	92.74%	97.21%	88.74%
ESTSCAN2.0	91.32%	96.45%	86.18%
Data2(1 frameshift or 1 substitution per 400 nucleotides.)			
ANGLE	91.72%	97.05%	87.72%
ESTSCAN2.0	90.64%	96.31%	85.46%
Data3(1 frameshift or 1 substitution per 300 nucleotides.)			
ANGLE	89.90%	96.71%	85.82%
ESTSCAN2.0	89.29%	95.98%	83.89%
Data4(1 frameshift or 1 substitution per 200 nucleotides.)			
ANGLE	86.84%	96.12%	82.57%
ESTSCAN2.0	87.67%	95.64%	82.14%

shorter than 1000. For examples, ANGLE scored 89.53% while ESTSCAN2.0 scored 75.84% on original data. Detail result of short sequences is shown in Table 6.

The scores of ANGLE are balanced, while those of ESTSCAN2.0 are rather spread against sequence length. On Table 5, a difference between the score of short sequences and that of longer sequences shows that the performance of ANGLE is more independent than that of ESTSCAN2.0. For example, ANGLE scored 1.69% while ESTSCAN2.0 scored 7.18% for $L2 - L1$ on Data1.

ANGLE does not depend on the length of a target sequence, because it predicts coding regions with only information from short windows. On the other hand, HMM is a relatively length-dependent method. In Viterbi process of HMM, the scores of each state are piled with the result that longer sequences have a larger gap between the score of the correct path and that of the wrong path, while shorter sequences have smaller gaps. HMM normalizes involved scores to avoid length-dependency, but still It does because choosing optimal parameters are difficult task especially for short sequences.

7. Discussion

In this article, we have described a new program ANGLE that predicts coding sequences in low quality cDNA. We proposed the hybrid method of machine learning and Markov model, which can model protein structure. We also described some practical techniques to implementation, such as pseudocounts, which makes up for shortage of information of a short segment, and so on.

Table 5. Comparing prediction accuracy of ANGLE and ESTSCAN2.0 on three types of data that are divided according to the length (l) of input sequences. All scores are computed per nucleotide.

	Correlation coeficient score				
	$L1$	$L2$	$L3$		
	$l < 1000$	$1000 \leq l < 3000$	$3000 \leq l$	$L2 - L1$	$L3 - L1$
Original Data (no sequencing error.)					
ANGLE	89.53%	90.12%	95.25%	0.59%	5.72%
ESTSCAN2.0	75.84%	84.65%	91.98%	8.81%	16.14%
Data1(1 frameshift or 1 substitution per 500 nucleotides.)					
ANGLE	84.03%	85.73%	91.21%	1.69%	7.18%
ESTSCAN2.0	73.72%	82.60%	89.83%	8.87%	16.10%
Data2(1 frameshift or 1 substitution per 400 nucleotides.)					
ANGLE	82.00%	84.49%	90.46%	2.48%	8.45%
ESTSCAN2.0	73.03%	81.97%	89.08%	8.94%	16.04%
Data3(1 frameshift or 1 substitution per 300 nucleotides.)					
ANGLE	80.05%	82.45%	88.61%	2.40%	8.56%
ESTSCAN2.0	71.95%	81.25%	86.86%	9.30%	14.91%
Data4(1 frameshift or 1 substitution per 200 nucleotides.)					
ANGLE	75.40%	79.25%	85.42%	3.85%	10.02%
ESTSCAN2.0	70.17%	79.70%	84.93%	9.54%	14.77%

ANGLE has a remarkable advantage of less dependency on sequence length compared with HMM based algorithms, because our algorithm evaluate optimal coding potential independently from a small window of an input sequence. Since predicting shorter sequences is much harder than predicting longer sequences, this advantage facilitates practical analysis of mRNA. We have shown in our evaluation that ANGLE achieved higher performance than the conventional tool. When input sequences are shorter than 1000 nucleotides, the average performance is about 9.26% better than that of ESTSCAN2.0 in computing Matthews' correlation coefficients. ANGLE has an average sensitivity of 91.57% and an average specificity of 97.01% on total dataset (Four datasets with one frame-shift error or one substitution error per 200-500 nucleotides and one dataset with no error). This result is 2.38% better than the performance of ESTSCAN2.0 in computing a correlation coefficient.

In addition to high performance, ANGLE provides utility. When a mRNA is analyzed, coding potential of each codons helps annotation tasks. ANGLE also provides detailed boosting score as well as classification result, which can be used for concrete coding potential. On the process of classification, ANGLE does not require large training dataset comparing to HMM which require thousands of parameters for training. Some kinds of genomes like virus are difficult to train HMM because those genomes are of small size[27]. ANGLE may help this problem.

Our method can be applied to ESTs with small modifications of the Markov

Table 6. Details of comparing prediction accuracy of ANGLE and ESTSCAN2.0 on short cDNA (<1000). All socres are computed per nucleotide.

	Sensitivity	Specificity	Correlation coefficient
Original Data (no sequencing error.)			
ANGLE	94.22%	96.69%	89.53%
ESTSCAN2.0	80.88%	93.87%	75.84%
Data1(1 frameshift or 1 substitution per 500 nucleotides.)			
ANGLE	89.03%	95.72%	84.03%
ESTSCAN2.0	79.12%	93.42%	73.72%
Data2(1 frameshift or 1 substitution per 400 nucleotides.)			
ANGLE	87.23%	95.33%	82.00%
ESTSCAN2.0	77.84%	93.26%	73.03%
Data3(1 frameshift or 1 substitution per 300 nucleotides.)			
ANGLE	85.42%	94.98%	80.05%
ESTSCAN2.0	77.14%	92.94%	71.95%
Data4(1 frameshift or 1 substitution per 200 nucleotides.)			
ANGLE	81.02%	94.21%	75.40%
ESTSCAN2.0	75.36%	92.70%	70.17%

model and a window size. Since ESTs are a part of full-length cDNA, a model without a start codon and a stop codon are required, and shorter window size will perform well.

We have two suggestions for further improvement. The first is adding specific models of boundary sites. In this study, our system did not have models of start and stop sites. However, those sites have some consensus[28,29], and modeling them may increase the accuracy of coding region boundaries. The second suggestion is using a dynamic penalty for frame-shift errors.

From the viewpoint of machine learning, using Support Vector Machines as a comparative classifier of boosting algorithms is an interesting trial.

8. Availability

We implemented our method as a web application which can be accessed from: http://angle.muraoka.info.waseda.ac.jp. The server is freely available to both academic and commercial users. A stand alon software and source code are available upon request.

9. Acknowledgements

This study was supported by special funds for Waseda University. We would like to thank Taishin Kin from the University of Tokyo and Hisanori Kiryu from Computational Biology Research Center for helpful comments and discussions.

References

1. Okazaki, Y., Kawai, J., Carninci, P., Bono, H., and Y. Hayashizaki, *et al.* (2002) Analysis of the mouse transcriptome based on functional annotation of 60,770 full-length cdna. *Nature,* **420**, 563–573.
2. Strausberg, R. L., Feingold, E. A., Klausner, R. D., and Collins, F. S. (1999) The mammalian gene collection. *Sicience,* **286**, 455–457.
3. Kikuno, R., Nagase, T., Waki, M., and Ohara, O. (2002) Huge: a database for human large proteins identified in the kazusa cdna sequencing project. *Nucleic Acids Res.,* **30**, 166–168.
4. Kawai, J. and Saito, T. (2001) Functional annotation of a full-length mouse cdna collection. *Nature,* **409**, 685–690.
5. Mathé, C., Sagot, M. F., Schiex, T., and Rouzé, P. (2002) Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Res.,* **30**, 4103–4117.
6. Mironov, A. A., Novichkov, P. S., and Gelfand, M. S. (2001) Pro-frame: similarity-based gene recognition in eukaryotic dna sequences with errors. *Bioinformatics,* **17**, 13–15.
7. Fichant, G. and Quentin, Y. (1995) A frameshift error detection algorithm for dna sequencing projects.. *Nucleic Acids Res.,* **23**, 2900–2908.
8. Xu, Y., Mural, R. J., and Uberbacher, E. C. (1995) Correcting sequencing errors in dna coding regions using a dynamic programming approach. *Comput. Appl. Biosci.,* **11**, 117–124.
9. Schiex, T., Gouzy, J., Moisan, A., and Oliveira, Y. (2003) Framed: a flexible program for quality check and gene prediction in prokaryotic genomes and noisy matured eukaryotic sequences. *Nucleic Acids Res.,* **31**, 3738–3741.
10. Borodovsky, M. and McIninch, J. (1993) Genemark: parallel gene recognition for both dna strands. *Comput. Chem.,* **17**, 123–133.
11. Salzberg, S. L., Delcher, A., Kasif, S., and White, O. (1998) Microbial gene identification using interpolated markov models. *Nucleic Acids Res.,* **26**, 544–548.
12. Burge, C. and Karlin, S. (1997) Prediction of complete gene structures in human genomic dna. *J.Mol. Biol.,* **268**, 78–94.
13. Salzberg, S. L., Pertea, M., Delcher, A. L., Gardner, M. J., and Tettelin, H. (1999) Interpolated markov models for eukaryotic gene finding. *Genomics,* **59**, 24–31.
14. Badger, J. H. and Olsen, G. J. (1999) Critica: coding region identification tool invoking comparative analysis. *Mol. Biol. Evol.,* **16**, 512–524.
15. Castrignanò, T., Canali, A., Grillo, G., Liuni, S., Mignone, F., and Pesole, G. (2004) Cstminer: a web tool for the identification of coding and noncoding conserved sequence tags through cross-species genome comparison. *Nucleic Acids Res.,* **32**, W624–W627.
16. 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.
17. Lottaz, C., Iseli, C., Jongeneel, C. V., and Bucher, P. (2003) Modeling sequencing errors by combining hidden markov models. *Bioinformatics,* **19**, 103–112.
18. Adams, M. D., Kelley, J. M., Gocayne, J. D., Dubnick, M., Polymeropoulos, M. H., Xiao, H., Merril, C. R., Wu, A., Olde, B., and R. F. Moreno, *et al.* (1991) Complementary dna sequencing: expressed sequence tags and human genome project. *Science,* **252**, 1651–1656.
19. Fukunishi, Y. and Hayashizaki, Y. (2001) Amino-acid translation program for full-length cdna sequences with frame-shift error. *Physiol. Genomics.,* **5**, 81–87.
20. Rajeev, K. A. and Borodovsky, M. (2004) Effects of choice of dna sequence model structure on gene identification accuracy. *Bioinformatics,* **20**, 993–1005.

21. Freund, Y. (1995) Boosting a weak learning algorithm by majority. *Information and Computation,* **121**, 256–285.
22. Jones, D. T. (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol.,* **292**, 195–202.
23. Wu, K.-P., Lina, H.-N., Chang, J.-M., Sung, T.-Y., and Hsu, W.-L. (2004) Hyprosp: a hybrid protein secondary structure prediction algorithm–a knowledge-based approach. *Nucleic Acids Res.,* **32**, 5059–5065.
24. Freund, Y. and Schapire, R. (1997) A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences,* **55**, 119–139.
25. Pruitt, K. D., Tatusova, T., and Maglott, D. R. (2001) Refseq and locuslink: Ncbi gene-centered resources. *Nucleic Acids Res.,* **29**, 137–140.
26. Matthews, B. W. (1975) Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochim. Biophys Acta,* **405**, 442–451.
27. Sharma, R., Maheshwari, J. K., Prakash, T., and andS.K. Brahmachari, D. D. (2004) Recognition and analysis of protein-coding genes in severe acute respiratory syndrome associated coronavirus. *Bioinformatics,* **20**, 1074–1080.
28. Kozak, M. (1999) Initiation of translation in prokaryotes and eukaryotes. *Gene,* **234**, 187–208.
29. Rogozin, I. B., Kochetov, A. V., Kondrashov, F. A., Koonin, E. V., and Milanesi, L. (2001) Presence of atg triplets in 5' untranslated regions of eukaryotic cdnas correlates with a 'weak' context of the start codon. *Bioinformatics,* **17**, 890–900.