BIOINFORMATICS

Vol. 20 no. 7 2004, pages 993–1005
DOI: 10.1093/bioinformatics/bth028

Effects of choice of DNA sequence model structure on gene identification accuracy

Rajeev K. Azad[1] and Mark Borodovsky[1,2,*]

[1]School of Biology and [2]School of Biomedical Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0230, USA

Received on August 2, 2003; accepted on November 10, 2003
Advance Access publication January 29, 2004

ABSTRACT

Motivation: Markov chain models of DNA sequences have frequently been used in gene finding algorithms. Performance of the algorithm critically depends on the model structure and parameters. Still, the issue of choosing the model structure has not been studied with sufficient attention.

Results: We have assessed performance of several types of Markov chain models, both fixed order (FO) models and models with interpolation, within the framework of the GeneMark algorithm. The performance was measured in two ways: (i) the accuracy of detection of protein-coding potential in artificial DNA sequences and (ii) the accuracy of identifying genes in real prokaryotic genomes. We observed that the models built by deleted interpolation (DI) slightly outperformed other models in detecting protein-coding potential in artificial DNA sequences with GC content in the medium range and also in detecting genes in real genomes with medium GC content. For artificial and real genomic DNA with high or low GC content, we observed that the models built by DI were in some cases slightly outperformed by FO models.

Contact: mark@amber.biology.gatech.edu

INTRODUCTION

As the number of completely sequenced genomes continues to rise, it is very important to use the best possible statistical models in computer algorithms and pipelines of genome annotation. Since the fraction of experimentally validated genes arguably makes up less than 0.1% of the number of genes currently annotated in known genomic DNA, computer methods are poised to remain the dominant tools for gene finding. Extrinsic (similarity search) and intrinsic (*ab initio* statistical) methods are frequently used in concert as the former are more specific and the latter are more sensitive gene identification tools. While changing the parameters of a statistical method may bring its sensitivity close to 100%, concurrently the specificity will drop dramatically. Further improvement of statistical methods in terms of decreasing the rate of false positive predictions remains a challenging issue. A notorious challenge is to develop an *ab initio* method with both sensitivity and specificity above 95–97%. Several frequently used statistical methods of gene identification rely on Markov chain models and, particularly, on three-periodic Markov models (Borodovsky and McIninch, 1993; Salzberg *et al*., 1998; Lukashin and Borodovsky, 1998; Delcher *et al*., 1999). These models quantify the nucleotide frequency patterns observed in DNA sequences in a form of three-periodic inhomogeneous Markov chains suitable for protein-coding sequences (Borodovsky *et al*., 1986a,b,c; Tavare and Song, 1989) and homogeneous Markov chains suitable for non-coding sequences.

A DNA sequence represented as a string of nucleotide symbols over alphabet $\mathcal{A} = \{A, T, C, G\}$, could be modeled by a Markov chain of an order m with 4^m initial and 4^{m+1} transition probabilities. Here a transition probability defines the probability of a nucleotide type in a particular position given a sequence of m preceding nucleotides, called ‘pretext’ or ‘history’. The GeneMark algorithm (Borodovsky and McIninch, 1993) utilizes fixed order (FO) Markov chain models of coding and non-coding sequence in a Bayesian formalism to calculate the a posteriori probability of a DNA sequence segment (and, in parallel, its complement) to be coding or non-coding.

In the ideal case of having a training sequence of unlimited length, a higher order model is either a better predictor of a nucleotide type in a given position or just equivalent to a lower order model. In a real case, dealing with the model of order m, if the length of the training sequence is N, there are only $N - m$ strings of size $m + 1$ available to estimate $4^m + 4^{m+1}$ initial and transition probability parameters related to 4^m types of oligonucleotides (pretexts). Frequencies of some of these oligonucleotides become too small (or even reach zero) as m grows. Therefore, an attempt to derive a model of too high an order will end up in overfitting. Still, even if the training sequence becomes insufficient for accurate estimation of each model parameter, a certain subset of this whole set will be estimated from large enough samples and, as such, should be used by an appropriate modeling procedure. The problem of most efficient extraction of information and estimation of parameters of Markov models from sparse training data has been discussed in the literature on statistical

[*] To whom correspondence should be addressed.

language modeling. Techniques such as ‘interpolation’ and ‘backoff’ have been implemented to account for the sparsity in the training datasets (Jelinek and Mercer, 1980; Bahl *et al*., 1983; Katz, 1987; Weinberger *et al*., 1995; Ron *et al*., 1996). The idea is to use a Markov chain model with a new structure. In this model a transition probability is a combination of lower and higher order models transition probabilities related with the aid of weight factors (interpolation parameters) to a particular pretext. Such an approach has the potential to produce a more precise model by extracting more information from sparse data than a FO model. Several heuristic techniques for choosing interpolation parameters have been described earlier (Jelinek and Mercer, 1980; Bahl *et al*., 1983; Salzberg *et al*., 1998) and one of them was implemented in the GLIMMER gene finding algorithm (Salzberg *et al*., 1998).

We have incorporated several interpolation techniques into the GeneMark algorithm. The performance of each version of the program was assessed in terms of accuracy of recognition of protein-coding sequence pattern in artificial DNA sequences and recognition of genes in real DNA. These results were compared with results obtained for FO models.

MATERIALS AND METHODS

The genomic sequences of *Bacillus subtilis*, *Escherichia coli*, *Helicobacter pylori*, *Ralstonia solanacearum* and *Ureaplasma urealyticum* were downloaded from NCBI ftp site: ftp://ftp.ncbi.nih.gov/genomes/Bacteria. These genomes span the range of GC composition from 25.5 to 67%.

Use of Markov chain models for gene identification

The use of FO Markov chain models for gene identification was described earlier (Borodovsky *et al*., 1986c; Borodovsky and McIninch, 1993). As the GeneMark algorithm was used for the assessment of performance of each model type, we briefly discuss the algorithm here. A DNA sequence fragment of length n (assuming n a multiple of 3), $S = \{s_1, s_2, \ldots, s_n\}$, may fall into one of the three categories: (a) protein-coding, (b) gene shadow (complement of protein-coding sequence) or (c) non-coding. A priori probabilities of these events are assumed to be $P(\mathrm{cod})$, $P(\mathrm{shadow})$ and $P(\mathrm{non})$. Given a DNA sequence, S, GeneMark calculates the *a posteriori* probabilities of the events $P(\mathrm{cod}\,|\,S)$, $P(\mathrm{shadow}\,|\,S)$ and $P(\mathrm{non}\,|\,S)$ using the three-periodic and ordinary Markov models of DNA sequences. Given a three-periodic Markov model of order m, the model of a protein-coding sequence, the probability of a sequence S to be generated by this model in phase 1 is defined as

$$P(S\,|\,\mathrm{cod}_1) = P^1(s_1^m) * P^1(s_{m+1}\,|\,s_1^m) * P^2(s_{m+2}\,|\,s_2^{m+1}) * P^3(s_{m+3}\,|\,s_3^{m+2}) * \cdots * P^t(s_n\,|\,s_{n-m}^{n-1}), \tag{1}$$

where $P^i(s_1^m)$ is the initial probability of a string of bases or pretext s_1^m (the subscript and superscript indicates the pretext start and end position in S, respectively) situated in phase i and $P^i(s_k\,|\,s_{k-m}^{k-1})$ is the transition probability of base s_k to follow the pretext s_{k-m}^{k-1} situated in phase $i, t = 2$, 1 and 3 if $\mathrm{mod}(m, 3) = 1, 2$ and 0, respectively. Here we say that the sequence is generated in phase $i, i = 1, 2, 3$ (or the pretext s is situated in phase i) if the first nucleotide of a sequence (or pretext) is located in the position i of a codon. Expressions for $P(S\,|\,\mathrm{cod}_2)$ and $P(S\,|\,\mathrm{cod}_3)$ can be obtained by the cyclic permutation of the superscripts of P in Equation (1).

Note that the initial probability estimate $P^i(s_1^m)$ is determined as

$$P^i(s_1^m) = \frac{N^i(s_1^m)}{[(N - m + 1)/3]}. \tag{2}$$

Here $N^i(s_1^m)$ is the number of occurrences of pretext s_1^m in phase i and $(N - m + 1)$ is the count of all possible pretexts of size m in the training data.

The transition probability estimate $P^i(s_k\,|\,s_{k-m}^{k-1})$ is defined as

$$P^i(s_k\,|\,s_{k-m}^{k-1}) = \frac{N^i(s_{k-m}^{k-1}, s_k)}{N^i(s_{k-m}^{k-1})}. \tag{3}$$

Here $N^i(s_{k-m}^{k-1}, s_k)$ and $N^i(s_{k-m}^{k-1})$ define the numbers of occurrences of the string s_{k-m}^{k} and s_{k-m}^{k-1} in phase i, respectively.

The probability of a sequence S to be generated by an order m three-periodic Markov model of a shadow of a coding region, $P(S\,|\,\mathrm{shadow}_i)$, is defined in three possible phases by equations similar to (1)–(3). The calculation of probability of a sequence S to be generated by a order m ordinary Markov model of a non-coding region, $P(S\,|\,\mathrm{non})$ is even more straightforward, since the phase consideration is not involved (Borodovsky and McIninch, 1993).

Finally, an *a posteriori* probability of S to be a part of a protein-coding region in phase $i, P(\mathrm{cod}_i\,|\,S), (i = 1$–$3)$ is defined as

$$\begin{aligned} P(\mathrm{cod}_i\,|\,S) = {} & \Big[P(S\,|\,\mathrm{cod}_i) * P(\mathrm{cod}_i)\Big] \\ & \Big/ \Bigg[\sum_{j=1}^{3} P(S\,|\,\mathrm{cod}_j) * P(\mathrm{cod}_j) \\ & + \sum_{j=1}^{3} P(S\,|\,\mathrm{shadow}_j) * P(\mathrm{shadow}_j) \\ & + P(S\,|\,\mathrm{non}) * P(\mathrm{non})\Bigg], \end{aligned} \tag{4}$$

where $P(\mathrm{cod}_i)$, $P(\mathrm{shadow}_i)$ and $P(\mathrm{non})$ are the mentioned above a priori probabilities of the respective events. *A posteriori* probabilities of S to be a gene-shadow sequence (in three phases) or a non-coding sequence are defined by equations similar to (4). As the seven *a posteriori* probabilities have been computed, the sequence S type is predicted to be of a category which probability exceeds a chosen threshold value (the unambiguous default is 0.5). This step completes the pattern recognition procedure. Note that since we consider a rather short DNA sequence segment, the assumption that the entire segment belongs to a single category is a reasonable one.

A long genomic sequence can be analyzed by a sliding window technique, with sequence fragments selected by moving a window (with default length of 96 nt) and the vectors of *a posteriori* probabilities calculated for each fragment. The *a posteriori* probabilities $P(\text{cod}_i \mid S)$ and $P(\text{shadow}_i \mid S)$ are used to score open reading frames (ORFs) observed on the direct or complimentary DNA strands, respectively. The ORF score is the mean value of the *a posteriori* probabilities of the windows that fall inside the ORF and have the same reading frame (phase). The ORF is predicted as a gene if the score exceeds the established threshold.

Markov models with variable pretext length (variable order models)

As a Markov model of order m predicts a type of nucleotide $b \in \mathcal{A}$ following an oligonucleotide (pretext) c_m, if the pretext occurs in the training data with sufficient frequency, the estimate of the transition probability (3) is robust. As was mentioned before, some pretexts may have low or zero counts due to the finite size of the training data. The frequency of such events increases with the increase of the model order. The simplest recourse is to use a Laplace rule and assign one extra count to each of the 4^m oligonucleotides at the initialization step. This approach is, however, not precise enough. Another approach is to allow in formulas of Markov chain theory, such as (1), the use of pretexts of variable length thus allowing transition probabilities of different orders. The rule of choosing a pretext length could be as follows. For a base b_i, take the longest pretext for which the frequency exceeds a certain threshold. This rule allows the definition of the transition and initial probabilities of an order m Markov chain model with pretext variability (the variable order Markov chain model). For a pretext c_m, whose frequency exceeds the threshold, the transition probability $P(b \mid c_m)$ is defined as for a regular Markov chain. Otherwise, the value of $P(b \mid c_m)$ is equal to $P(b \mid c_{m-1})$, where $P(b \mid c_{m-1})$ is determined recursively with regard to the just defined rule. The initial probabilities are defined in the same manner.

Interpolated Markov models

This class of Markov chain models defines transition probabilities $P(b \mid c_m)$ as linear combinations of transition probabilities associated with pretexts c_m of smaller lengths contained in c_m. The recursive equation,

$$P^{\text{IMM}}(b \mid c_k) = \lambda(c_k) * P(b \mid c_k) + [1 - \lambda(c_k)] * P^{\text{IMM}}(b \mid c_{k-1}), \tag{5}$$

defines $P^{\text{IMM}}(b \mid c_k)$, the value of interpolated transition probability, as a function of the transition probability $P(b \mid c_k)$, defined by Equation (3); the interpolated transition probability of a lower order $P^{\text{IMM}}(b \mid c_{k-1})$; and the interpolation parameter or weight factor $\lambda(c_k), 0 \leq \lambda(c_k) \leq 1$.

Note that for pretexts not observed in the training data $N(c_k) = 0$, the value of the transition probability is formally defined as

$$P(b \mid c_k) = P(b \mid c_{k-n}), \quad \text{where } n = \min_{1 \leq n < k} \{n\colon N(c_{k-n}) > 0\}. \tag{6}$$

For a pretext c_k, the value of parameter $\lambda(c_k)$ can be interpreted as a measure of strength of the statistical association of a particular pretext with the subsequent base. Generally, if the frequency of a pretext c_k is sufficiently high, the value of $\lambda(c_k)$ is close to 1; for pretexts c_k with low frequency, the value of $\lambda(c_k)$ is close to 0 and the interpolated probability $P^{\text{IMM}}(b \mid c_k)$ gains most of its value from $P^{\text{IMM}}(b \mid c_{k-1})$. The issue of how to define interpolation parameters is not simple. In the following sections, we give a brief description of two possible approaches.

χ^2-confidence based interpolation

For a pretext c_j, $\{j = 1, \ldots, m\}$ such that $N(c_j) \geq T$, $\lambda(c_j)$ is equal to 1. For c_j such that $N(c_j) < T$, the value of $\lambda(c_j)$ is defined as follows.

$$\lambda(c_j) = 0, \quad \text{if } q < 0.5, \quad \text{otherwise } \lambda(c_j) = q * \frac{N(c_j)}{T}. \tag{7}$$

Here $q = 1 - p$ where p is the p-value of the χ^2-test of the hypothesis H_0 that the observed frequencies $N(c_j, b)$, $b \in \mathcal{A}$ fit the predicted frequency distribution $P^{\text{IMM}}(b/c_{j-1}) * N(c_j)$. Thus, the value $\lambda(c_j)$ is estimated based on χ^2 confidence and the frequency of pretext c_j. This rule has been used to define the parameters of interpolated Markov models utilized in the GLIMMER gene finding algorithm (Salzberg *et al.*, 1998; Delcher *et al.*, 1999). Note that the variable order Markov model becomes a special instance of this model if the value of $\lambda(c_j)$ is defined by the rule: if $N(c_j) \geq T$ then $\lambda(c_j) = 1$, otherwise $\lambda(c_j) = 0$.

Deleted interpolation (DI) Deleted interpolation has a history of use in speech recognition algorithms (Jelinek and Mercer, 1980; Bahl *et al.*, 1983; Potamianos and Jelinek, 1998). The training dataset is divided into two sets: a development set and a held-out set, denoted as D set (sequence) and H set (sequence), respectively. The estimates of initial and transition probabilities are obtained from the D set. The H sequence is used to define the interpolation parameters λs by maximizing the likelihood of the H sequence to be generated by the D-interpolated model (Bahl *et al.*, 1991) or equivalently by maximizing the log-probability defined by D-interpolated model per character of the H set (Potamianos and Jelinek, 1998). This maximization can be done by forward–backward algorithm (Jelinek and Mercer, 1980; Bahl *et al.*, 1983) or by simpler algorithms like Newton–Raphson and bisection methods (Bahl *et al*., 1991; Potamianos and Jelinek, 1998).

The DI method for obtaining λ values for Equation (5) was implemented as follows (for details, see Potamianos and Jelinek, 1998). The pretexts were grouped into several, M, order specific 'frequency buckets', according to the pretext frequency of occurrence in the D set. Each bucket was associated with a single interpolation parameter. A bucket should contain sufficient number of observation points within the H set, in order to get reliable estimate of interpolation parameters. The frequency bucket boundaries were defined by the equation

$$R_{k,i} = \delta * R_{k,i-1}, \quad i = 1, \ldots, M, \tag{8}$$

where $R_{k,i}$ denotes the right boundary of a bucket i. The left boundary of first bucket was defined as $R_{k,0} = \min\{N_D(c_k)\}$. Parameter $\delta > 1$ defines the ratio of the widths of adjacent frequency buckets. Note that

$$M = \left\lceil \frac{\ln\{\max[N_D(c_k)]/\min[N_D(c_k)]\}}{\ln(\delta)} + 1 \right\rceil,$$

$\lceil\,\rceil$ is the minimum integer operator.

If the right boundary of the last bucket is less than $\max\{N_D(c_k)\}$, the last two buckets are merged together.

A set of pretexts of order k that falls into a particular bucket is denoted as

$$B_{k,i} = \{c_k\colon R_{k,i-1} \le N_D(c_k) < R_{k,i}\}. \tag{9}$$

The interpolation parameter $\lambda_{k,i}$ for all pretexts from bucket $B_{k,i}$ is defined as

$$\lambda_{k,i} = \arg\max_{\lambda} \left\{ \left[\sum_{c_k \in B_{k,i}} \sum_{b \in \mathrm{A}} N_H(c_k, b) * \log_2(\lambda * P(b \mid c_k) + (1-\lambda) * P^{\mathrm{IMM}}(b \mid c_{k-1})) \right] \Big/ \left[\sum_{c_k \in B_{k,i}} \sum_{b \in \mathrm{A}} N_H(c_k, b) \right] \right\}. \tag{10}$$

Here $N_H(c_k, b)$ are counts of oligonucleotides (c_k, b) observed in the H sequence. It was proved (Bahl *et al*., 1991; Potamianos and Jelinek, 1998) that the function to maximize in (10) is concave with respect to λ, i.e. there is a unique maximum in the range $0 \le \lambda \le 1$. The argmax value of λ corresponds to the root of the following equation

$$\sum_{c_k \in B_{k,i}} \sum_{b \in \mathrm{A}} N_H(c_k, b) * [P(b \mid c_k) - P^{\mathrm{IMM}}(b \mid c_{k-1})] / [\lambda * [P(b \mid c_k) - P^{\mathrm{IMM}}(b \mid c_{k-1})] + P^{\mathrm{IMM}}(b \mid c_{k-1})] = 0, \tag{11}$$

and can be found numerically by bisection method, Newton–Raphson method (Bahl *et al*., 1991; Press *et al*., 1992) or other efficient optimization algorithm.

Computation of transition probabilities in the DI method started with the initialization step,

$$P^{\mathrm{IMM}}(b \mid c_{-1}) = P(b \mid c_{-1}) = \frac{1}{|A|}. \tag{12}$$

For a DI Markov model of order m, we determined successively $P^{\mathrm{IMM}}(b \mid c_0), P^{\mathrm{IMM}}(b \mid c_1), \ldots, P^{\mathrm{IMM}}(b \mid c_m)$, with λ obtained from Equations (10) and (11), and then substituted in (5) to get the set of D-interpolated probabilities.

To further refine estimates of the initial and transition probabilities $(P(c_k), P(b \mid c_k),\ k = 1, \ldots, m)$ the D and H sets were combined in the final step. These updated estimates were used in Equation (5) with the previously derived set of λ to get the final set of interpolated probabilities $P^{\mathrm{IMM}}(b \mid c_m)$.

Error rate determination

The performance of the described models for gene identification was assessed by two methods. First, we used artificial (or model) DNA sequences to simulate the processes of training and testing the program implementing GeneMark algorithm with different types of models. Artificial protein-coding and non-coding sequences were generated by the order m inhomogeneous Markov model and homogeneous Markov model, respectively. Parameters of these models were determined for each genome in our test set. Then we applied an n-fold cross-validation procedure ($n = 3, 6, 9$) to assess the performance of the algorithm versions either using Markov models built by the different interpolation techniques or FO models. Note that the size of a training sequence was also a free parameter. The 'coding' and 'non-coding' sequences of the test set were divided into 96 nt long non-overlapping segments. The error rates Fc and Fn were determined as the percentage of incorrectly identified coding and non-coding segments by the algorithm, respectively.

Second, to assess the performance of the different types of models for gene finding in genomic sequences we used real DNA sequences obtained from GenBank. Using a 6-fold cross-validation procedure, we built models from the training set and then used the models to predict genes in the test set. For comparison, we also implemented 3- and 9-fold cross-validation. The error rates Rn and Rp were defined as follows: Rn = 100% − Sn and Rp = 100% − Sp, where Sn and Sp stand for sensitivity and specificity values, respectively. Sensitivity is defined as the percentage of annotated genes in the test set that were correctly identified by an algorithm. Specificity is defined as the percentage of total number of predictions made by the algorithm that match genes annotated in the test set. The average value of Rn and Rp was also used as a single error rate parameter.

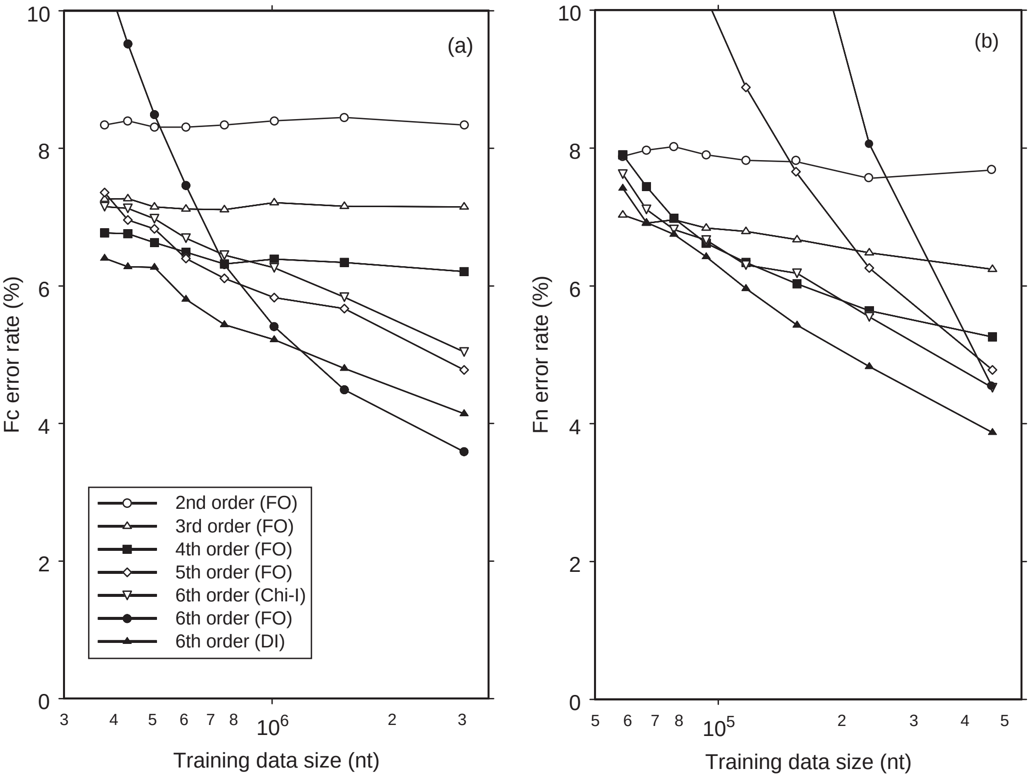

Fig. 1. Error rates of identification of model sequences of *B.subtilis* genome (GP case) as functions of model order and training set size: (a) Error rate, Fc, for protein coding sequence and (b) Error rate, Fn, for non-coding sequence. The coding and non-coding sequences were generated by 6th order inhomogeneous and homogeneous Markov models, respectively. Error rates Fc and Fn are shown for algorithm versions using models built by DI, by Chi-I and also FO models.

RESULTS AND DISCUSSION

Assessment of error rates using artificial DNA sequences

The artificial sequences were generated by the Markov models with parameters derived from genomic sequences of *B.subtilis* (4.2 Mbp, 43.5% GC), *R.solanacearum* (3.7 Mbp, 67.1% GC) and *U.urealyticum* (0.75 Mbp, 25.5% GC). These genomes were chosen to represent the medium, high and low GC content ranges. The orders of the Markov models were 6, 6 and 5 respectively. Two possibilities were considered. In one case, we generated coding and non-coding sequences of the same size as in the actual genome (we call this the genomic proportion, or the GP case). In the second case, the non-coding sequence was made equal in length to one-third of the coding sequence length (three-periodic proportion, or the TP case). In the TP case, the Markov matrix for the non-coding model and each of the three Markov matrices for the protein-coding model were built from the same size training sets. Along with the whole training dataset, we considered reduced datasets where sizes of both coding and non-coding sequence in the training set were decreased by a factor of $1/n$ ($n = 2, 3, \ldots, 8$).

For χ^2-confidence based interpolation (Chi-I) the threshold T was set to 400. Changing the T value did not produce any noticeable decrease of Fc and Fn. For building the D-interpolated models, one-fifth of the training data was used for the H set and the remaining part for the D set. In our experiments with varying the sizes of D and H set, one-fifth proved to be the best case. Note that the case when $H = D$, i.e. λs were estimated from the same dataset used for estimation of transition probabilities, was also considered in these experiments. The value of δ (parameter defining a geometric progression of frequency bucket widths) was set to 2.0. Changing the value of δ within a range 1.5–3 did not produce significant variations in Fc and Fn. Note that using other bucketing schemes, e.g. making buckets of equal volume in terms of number of pretexts did not reduce error rates either.

The *B.subtilis* model sequences (medium GC content)

The error rates Fc and Fn as functions of a size of training data and model orders for the GP and TP cases are shown in Figures 1 and 2, respectively. In the GP case, the D-interpolated model made less Fc errors than the FO models

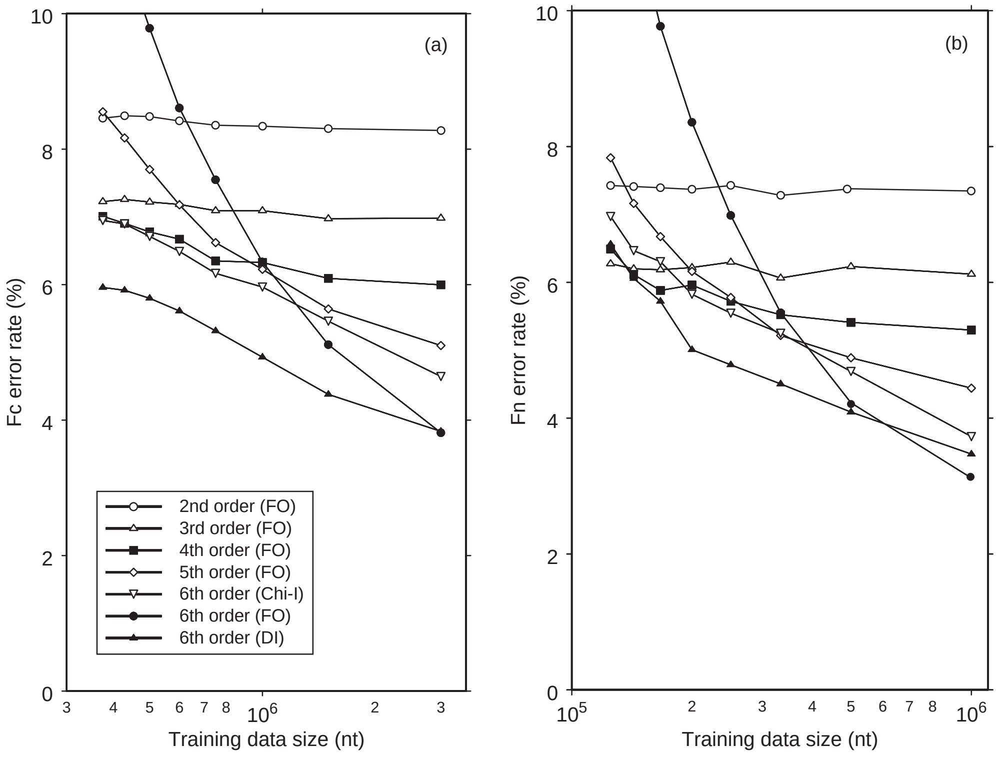

Fig. 2. As in Figure 1, for the TP case (the length of model non-coding sequence is equal to one-third of coding sequence length).

as the length of the training coding sequence in training set decreased (Fig. 1a). The D-interpolated model always performed better in terms of Fn errors (Fig. 1b) (for very small training sizes, however, lower order DI model may perform better than the FO model). Note also that the Chi-I model was outperformed by the FO models in terms of Fc error rate. In the TP case, the DI model showed the best results in comparison with FO and Chi-I models (Fig. 2a and b). The DI model also outperformed both the FO models and the Chi-I model by a larger margin.

The *R.solanacearum* model sequences (high GC content)

In the GP case, Figure 3a and b, the FO model performed better than the DI model in terms of Fc error rate (coding sequence recognition) whereas the DI model slightly outperformed the FO model in terms of Fn error rate (non-coding sequence recognition). The Chi-I model performed better than others in terms of Fn for some of the training data sizes. In the TP case, Figure 4a and b, the DI model produced lower Fc error rate than the FO model. The Chi-I model did not perform better than the FO model either in terms of Fc or Fn.

The *U.urealyticum* model sequences (low GC content)

The observed results were similar to those cited for *R.solanacearum* (Figs 5 and 6). Note that in the TP case, the D-interpolated model outperformed both the Chi-I and FO models by a noticeable margin (Fig. 6a and b).

To compare performance of the FO models (of various orders) with the performance of the DI model (of the same order as for the model for sequence generation), we plotted in Figure 7 the minimum difference between Fc error rates made by FO and DI models as a function of the size of coding (artificial) training set. We show the GP and the TP case plots for each genome. Figure 7 demonstrates that the DI model was more efficient than the FO model in the TP case. This is also true for the *B.subtilis* model sequences in the GP case as the size of training set was reduced. Use of the DI model for smaller and more realistic non-coding datasets in the GP case (Fig. 1) led to higher Fc and Fn error rates as compared to the TP case (Fig. 2) since λs for high order pretexts were estimated from too small *H* sets. As the size of the *H* set increased, better estimates of λs resulted in better parameters of interpolated models and faster improvement in coding potential detection as compared to other models. Similar variation in error rate (ΔFn) was seen for artificial non-coding sequence, though it was not as prominent as for coding sequence (data not shown).

The DI and Chi-I approaches define parameters λ differently. To illustrate these differences we present the following example. As just described, the interpolated models showed better performance as the size of training data was reduced. The *B.subtilis* model sequences, derived for the TP case, were

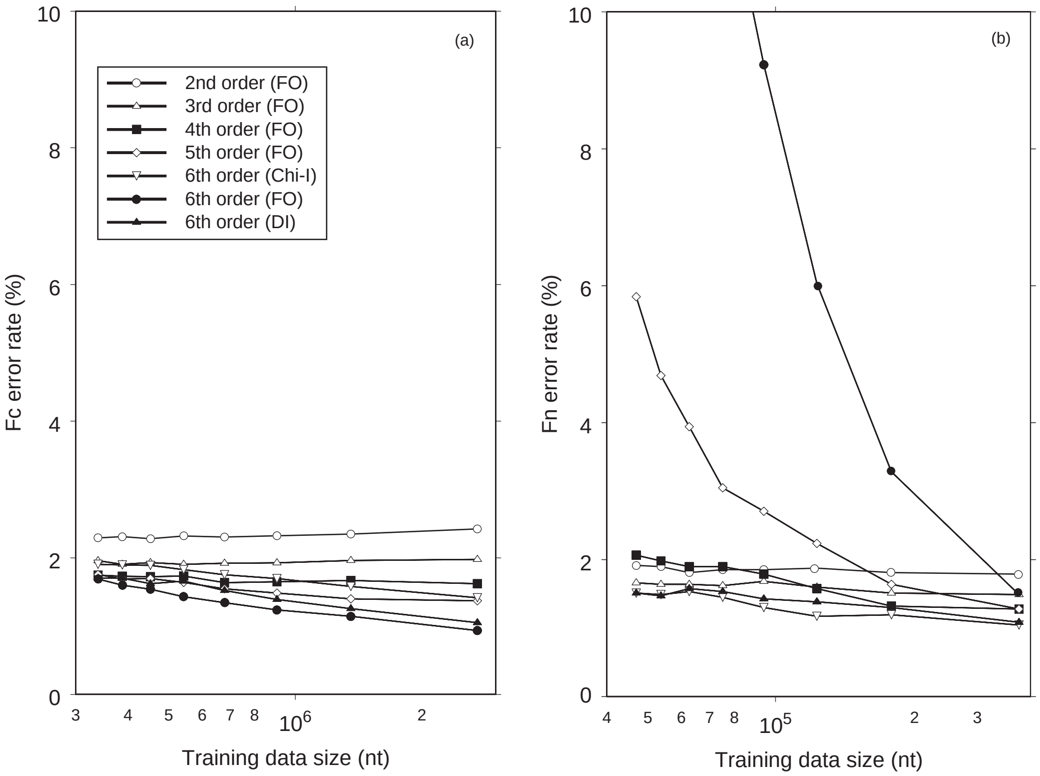

Fig. 3. As in Figure 1, for *R.solanacearum* genome.

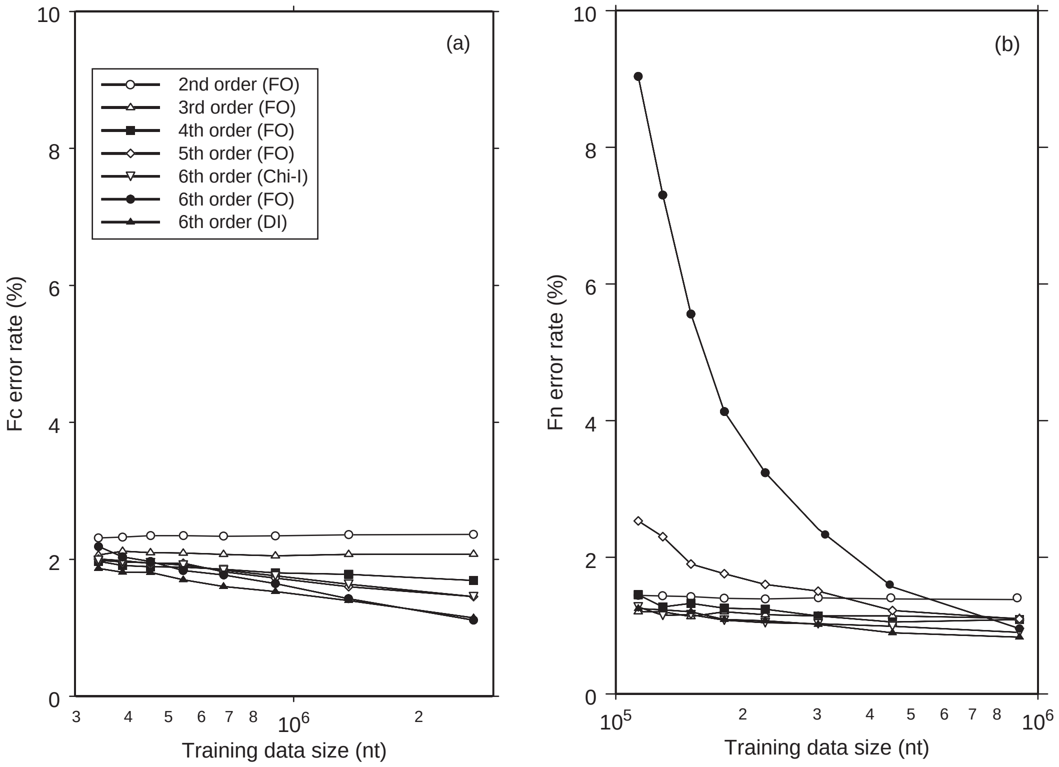

Fig. 4. As in Figure 3, for the TP case.

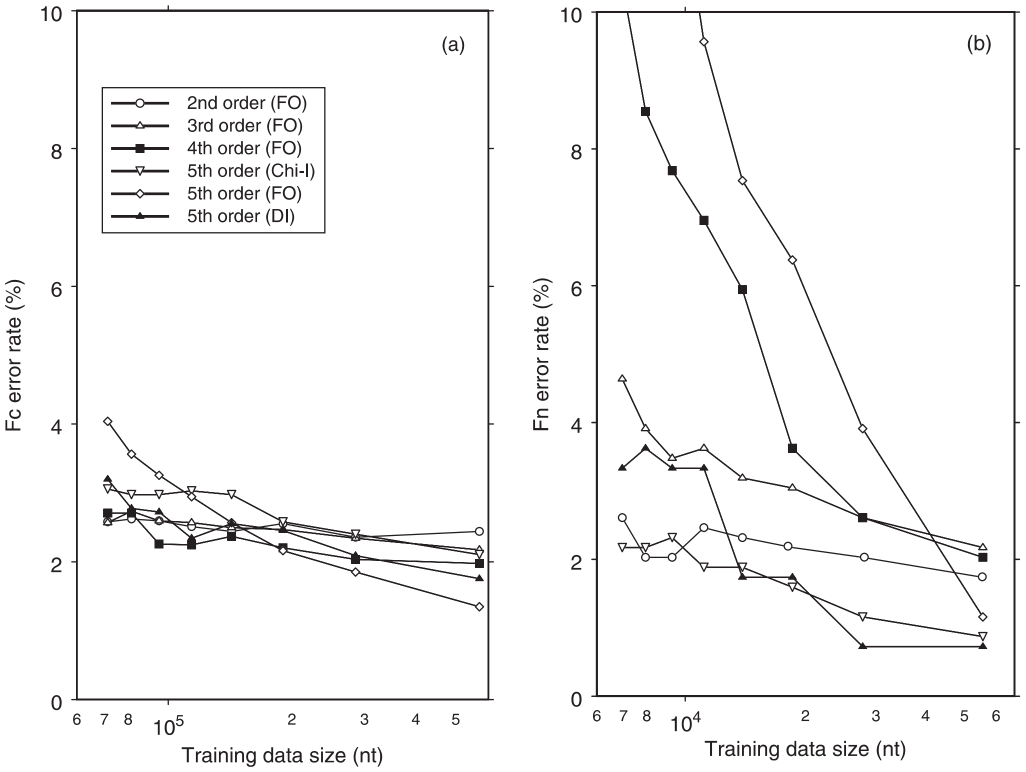

Fig. 5. As in Figure 1, for *U.urealyticum* genome. The protein-coding and non-coding model sequences were generated by 5th order inhomogeneous and homogeneous Markov models, respectively.

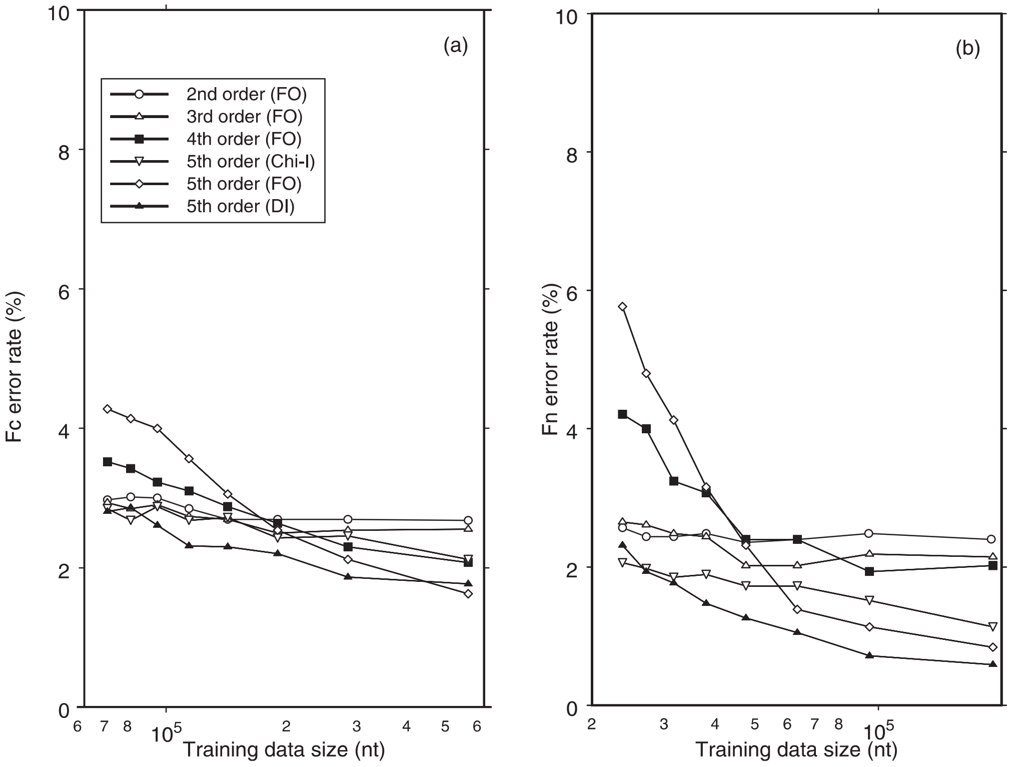

Fig. 6. As in Figure 5, for the TP case.

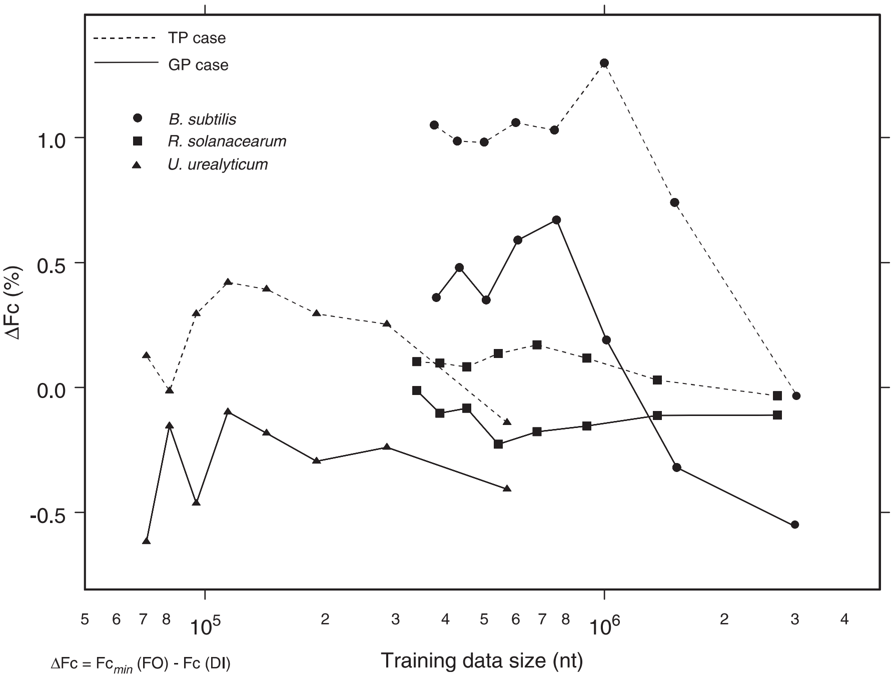

Fig. 7. The performance of the FO models (of various orders) is compared with the performance of the DI model (with the same order as for the model for sequence generation) by computing the minimum difference in the error rate observed for the two types of models as a function of the size of coding sequence training data. The error rate differences are shown for the GP and TP cases as explained in the text.

reduced by half and parameters λ were estimated by both methods. In the DI case (Fig. 8), the distributions of 4^6 values of λ for pretexts $c_k[k = 6, N(c_k) > 0]$ are distinct for the codon positions associated with the last base of a pretext. The distributions of λ values determined by the Chi-I are of similar shape in case of all three codon positions, an exponential distribution with a hump at the tail. For DI, the λ distributions peaked in the range 0.3–0.4 (more than half of all pretexts were associated with λ lying in this range), while the Chi-I method generated many with values close to zero. Thus in the Chi-I case, transition probability values frequently got significant contribution from the lower order pretexts ($\lambda < 0.1$). In the DI case, we observed that pretexts of all orders contributed in the estimation of a transition probability. In general (for any choice of k), we observed that in the Chi-I case, the λ distribution peaked near $= 0$ or 1 or both. It shows that for the prediction of a base, contribution was taken only from some of its pretexts. However, in the DI case, the contributions of pretexts of almost all orders were more evenly distributed.

A characteristic feature of an interpolated model is the type of dependence of λ on a pretext frequency. For instance, such dependence for pretexts $c_k[k = 6, N(c_k) > 0]$ starting in the second codon position of the generated *B.subtilis* coding sequence is shown in Figure 9a and b. Parameters λ of the DI model were defined by a step function. Each step width corresponds to the frequency range of the pretext bucket (Fig. 9a). This type of function indicates that the model relies more on the high-frequency pretexts. Subsequently, associated with them transition probabilities make greater contribution to the interpolated model parameters. Note that first bucket width corresponds to a bucket with frequencies less than or equal to 64 (this amounts to merger of first few buckets in order to eliminate fluctuations observed in λ values for low frequency regions, however, we do not observe any change in error rate values by doing this). In the Chi-I case, the non-zero λs (for pretexts with frequency less than T) were observed to be scattered in a region bounded by the lines with slope $1/T$ and $1/2T$ (Fig. 9b). Also, some λs took zero value irrespective of pretext frequency and due to the variation of p-values taking any value from 0 to 1. For example, if the frequency of a pretext c_k is close to T, and probabilities $P(b \mid c_k)$ and $P^{\mathrm{IMM}}(b \mid c_{k-1})$ are not significantly different as determined by χ^2-test, then, the λ value associated with c_k will be assigned a value 0. Note, however, that for pretext with slightly different frequency the value of λ might be equal to 1. Overall the observations described above showed that use of λs defined by the Chi-I did not bring in a noticeable reduction of error rates Fc and Fn as compared to the FO model, while the DI method of λ assignment was able to decrease error rates Fc and Fn to a small but noticeable margin. To test whether the decrease in error rates obtained by the use of DI model is statistically significant, we carried out the Student's t-test. The mean value of the error (number of incorrectly identified

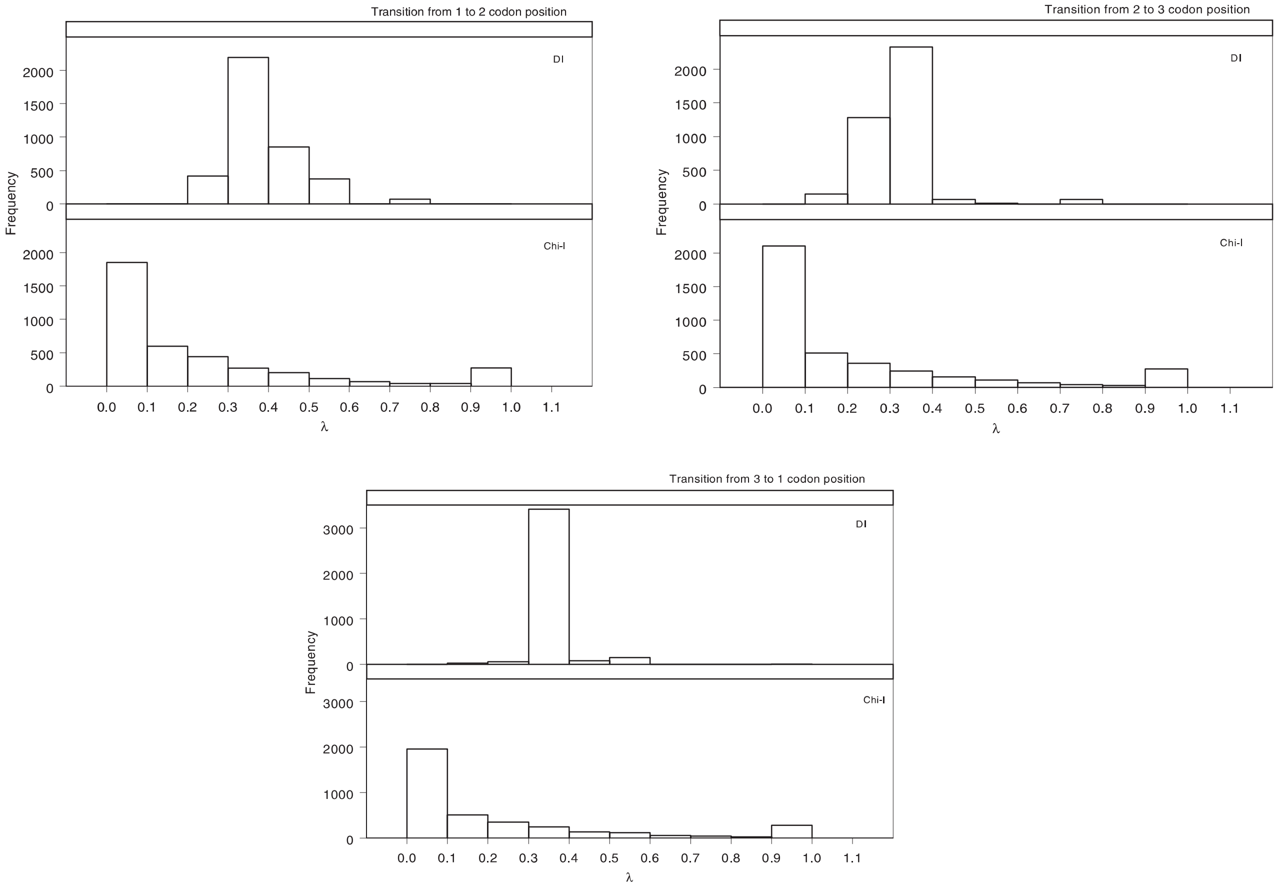

Fig. 8. The frequency distribution of interpolation parameter values, λ, associated with the pretexts of 6th order Markov model of artificial protein-coding sequence (*B.subtilis* genome) for Chi-I and DI cases. Note that the 'Transition from i to j codon position' means that the pretext ending with base in i codon position is succeeded by a base in j codon position (see text for details).

segments in each of the six test sets in 6-fold cross-validation procedure) in coding segment identification was obtained for the FO model. Note that we considered the case when the training data size was reduced by a factor of 2 (TP case, Fig. 2). The total number of segments in each of the test sets was 6250. The mean value, 319 was then used to define the H_0 and H_1 hypothesis as follows. H_0: $\mu = 319$, there was no reduction in error rate; H_1: $\mu < 319$, there was reduction in error rate. The test significance level was set to 0.05. Thus, for one tail test, the following decision rule was adopted. Accept H_0 if t was less than $-t_{0.95}$ (for ν degrees of freedom), otherwise reject H_0. For a sample of three test sets chosen randomly in the DI case, the mean and standard deviation of error values were 282 and 14.35, respectively. Under H_0 the value of t was -3.64. From the Student's t-distribution, for $\nu = 2$, $-t_{0.95}$ is -2.92. Thus H_0 was rejected and we conclude that the decrease in error rate is significant at 0.05 level.

Gene prediction error rates

We discuss below the results obtained for each of the five genomes *B.subtilis*, *E.coli*, *H.pylori*, *R.solanacearum* and *U.urealyticum*. The model order (best among models of a particular category) is shown in parentheses in the second column of Table 1. Since the Chi-I models and variable order models did not show any noticeable improvement over the FO models, we present here the comparison of the results produced by the FO and DI models.

B.subtilis

Application of DI model of order 7 reduced Rn by about 3% in comparison with best FO model (of order 5) though Rp increased by nearly 1.5% . Thus, the DI model predicted 197 more genes than the FO model with 117 out of 197 being annotated in original GenBank record. The error rate [(Rn + Rp)/2] produced by DI model was lower by about 0.5% as compared to FO error rate.

E.coli

Application of the DI model of order 8 reduced Rn error rate by about 1.5% in comparison with the best FO model (of order 5) with Rp remaining nearly the same. The DI model predicted 75 more genes than the FO model with 66 out of 75 being

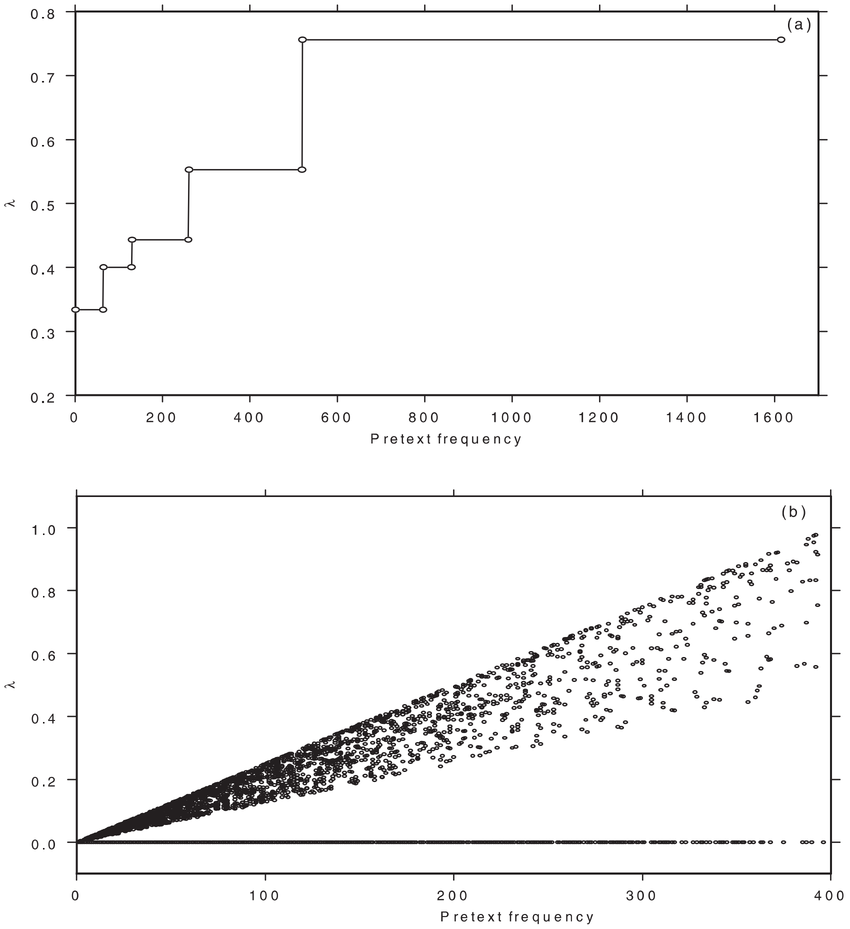

Fig. 9. The plot of interpolation parameter λ as a function of pretext frequency in the case of **(a)** DI and **(b)** Chi-I. The plots are shown for the case of ‘Transition from 1 to 2 codon position’ as explained in the caption of Figure 8.

annotated in original GenBank record. This result showed that that identification of additional annotated genes does not come at the cost of adding false positives. The DI model showed a lower error rate, 0.7% less than the FO error rate.

H.pylori

The results of assessment of the method accuracy by 6-fold cross-validation showed that the error rate remained the same for both DI and FO models. The same results were obtained for 3- and 9-fold cross-validation (data not shown). However, as the size of training dataset was significantly decreased, the DI model outperformed the FO models. For example, for a 1/36 factor of training set size reduction, the error rate of the DI model was lower by nearly 0.75%.

R.solanacearum

The FO model of order 6 performed better than the DI model with the error rates being smaller by nearly a quarter percent. This result is consistent with observations of better performance of FO models for this genome in the GP case.

U.urealyticum

Application of the DI model of order 3 produced slightly better results in the 6- and 9-fold cross-validation tests (but not in 3-fold cross-validation test). Note that this genome uses a non-standard genetic code, with TGA encoding the amino acid Tryptophan.

Short gene prediction

We compared the performance of the models in identifying short genes. We used three sets of annotated *B.subtilis* genes shorter than 300 nt with at least one, two and ten significant similarities to known proteins determined by BLAST analysis (Besemer *et al.*, 2001, the data are available at http://opal.biology.gatech.edu/GeneMark/GeneMarkS/index.html). In Table 2 we show these results along with the results from the web-versions of GeneMark and GeneMarkS. According

Table 1. Error rate of gene prediction observed for different types of Markov models used in the GeneMark algorithm. The 6-fold cross-validation procedure was used

Organism	Prediction method (model order)	Genes annotated	Genes predicted	Annotated genes detected	Rn (%)	Rp (%)	Error rate = (Rn + Rp)/2 (%)
B.subtilis	FO (5)	4105	3934	3751	8.63	4.66	6.64
	Variable-order (8)		3715	3595	12.43	3.24	7.83
	Chi-I (8)		3904	3742	8.85	4.15	6.50
	DI (7)		4131	3868	5.78	6.37	6.07
E.coli	FO (5)	4255	3954	3832	9.95	3.09	6.52
	Variable-order (8)		3817	3726	12.44	2.39	7.41
	Chi-I (8)		3956	3837	9.83	3.01	6.42
	DI (8)		4029	3898	8.40	3.26	5.83
H.pylori	FO (4)	1572	1504	1451	7.70	3.53	5.61
	Variable-order (8)		1468	1422	9.55	3.14	6.34
	Chi-I (8)		1500	1446	8.02	3.61	5.81
	DI (4)		1488	1443	8.21	3.03	5.62
R.solanacearum	FO (6)	3442	3238	3177	7.70	1.89	4.79
	Variable-order (8)		3109	3075	10.67	1.10	5.88
	Chi-I (8)		3176	3139	8.81	1.17	4.99
	DI (7)		3160	3126	9.19	1.08	5.13
U.urealyticum	FO (2)	614	603	595	3.10	1.33	2.21
	Variable-order (8)		586	579	5.71	1.20	3.45
	Chi-I (8)		605	593	3.43	1.99	2.71
	DI (3)		602	596	2.94	1.00	1.97

Table 2. Three sets of annotated *B.subtilis* genes shorter than 300 nt with at least one (Set 1), two (Set 2) and ten (Set 3) significant homologies as determined by BLAST analysis were used to compare the performance of GeneMark (web-version) and versions of GeneMark using the FO models, the models built by Chi-I and the models built by DI. Results are also shown for GeneMarkS algorithm which uses hidden Markov models to predict genes. The number of genes identified by an algorithm in each of the test sets is shown

Test set	GeneMark (web-version)	The FO model			The Chi-I model	The DI model		GeneMarkS
	4th order	4th order	5th order	6th order	8th order	7th order	8th order	2nd order
Set 1 (123 genes)	84	88	93	94	88	96	104	113
Set 2 (72 genes)	54	55	59	59	55	60	63	68
Set 3 (51 genes)	39	40	41	41	38	42	44	48

to these results, the DI model performed better in detecting short genes than the FO models which in turn outperformed the Chi-I model. As GeneMarkS performed the best, we should emphasize that this is a different algorithm using hidden Markov models.

Having done the comparison of the performance of Markov models with different structure, we should state the following. Experiments with artificial DNA sequences showed that when the training data was sparse, the DI model outperformed other models in accuracy of protein-coding potential detection. However, the degree of improvement varied depending on the GC content of the genome. For *B.subtilis* which has a moderate GC content, as the training data became sparse the DI model performed better than other models and showed a consistently lower error rate as the size of training set was further decreased. However, for genomes with high or low GC content, the picture of relative performance depended on the size of the training set. The FO model performed better than the DI model in coding potential detection in the GP case. However, in the TP case, the DI model again performed better than the others. As the size of training set becomes smaller, the lower order DI models perform better. This happens because the interpolations parameters λ of a higher order DI model no longer remain reliable. Interestingly, the DI models performed better for genomes of moderate GC content than for genomes with low or high GC content.

In comparison of predicted and annotated genes, we found that the DI model performed better than other models for the *E.coli* and *B.subtilis* genomes (medium GC content). For *H.pylori*, the FO and DI models gave similar results, however, when the training set was significantly reduced, the DI model outperformed the FO model. The results for *R.solanacearum* showed that the FO model of order 6 performed better than other models. This was consistent with the results on artificial sequences where the FO model detected the coding potential better than other model in the GP case. In the case of *U.urealyticum*, the DI model performed better than the FO model in 6- and 9-fold cross-validation test, while the FO model showed lower error rate in 3-fold cross-validation test.

In our experiments, the variable order model and Chi-I model did not yield any noticeable improvement over the FO models. On the contrary, we found the FO models frequently outperformed the Chi-I models and variable order models. Introducing the DI model allows in many cases to identify more accurately protein-coding patterns in artificial DNA sequences and genes in real genomes in comparison with the FO models. This gain comes though at a significant cost of additional computations. However, this is an acceptable and affordable option given the power of current computers and a growing concern with the accuracy of annotation of genes in GenBank (see, e.g. Skovgaard *et al*., 2001). The conclusion that we can make from the results presented above is that the optimal choice of the model structure and the model order is species specific. The models built by DI may produce superior results for some genomes, while the FO models remain the best choice for others. Other interpolation methods could be considered to get further insights into the yet impenetrable depth of complexity of genome organization.

ACKNOWLEDGEMENTS

We are grateful to Dr John Besemer for valuable comments on the manuscript. This work was supported in part by research grants from the US Department of Energy and the US National Institutes of Health.

REFERENCES

Bahl,L.R., Jelinek,F. and Mercer,R.L. (1983) A maximum likelihood approach to continuous speech recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, **5**, 179–190.

Bahl,L.R., Brown,P.F., de Souza,P.V., Mercer,R.L. and Nahamoo,D. (1991) A fast algorithm for deleted interpolation. In *Proceedings of the EUROSPEECH' 91*, Genova, pp. 1209–1212.

Besemer,J., Lomsadze,A. and Borodovsky,M. (2001) GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Res.*, **29**, 2607–2618.

Borodovsky,M., Sprizhitsky,Yu.A., Golovanov,E.I. and Alexandrov,A.A. (1986a) Statistical patterns in the primary structures of functional regions of the genome in *Escherichia coli*: I. Frequency characteristics. *Mol. Biol.*, **20**, 826–833.

Borodovsky,M., Sprizhitsky,Yu.A., Golovanov,E.I. and Alexandrov,A.A. (1986b) Statistical patterns in the primary structures of functional regions of the genome in *Escherichia coli*: II. Nonuniform Markov models. *Mol. Biol.*, **20**, 833–840.

Borodovsky,M., Sprizhitsky,Yu.A., Golovanov,E.I. and Alexandrov,A.A. (1986c) Statistical patterns in the primary structures of functional regions of the genome in *Escherichia coli*: III. Computer recognition of coding regions. *Mol. Biol.*, **20**, 1144–1150.

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

Delcher,A.L., Harmon,D., Kasif,S., White,O. and Salzberg,S.L. (1999) Improved microbial gene identification with GLIMMER. *Nucleic Acids Res.*, **27**, 4636–4641.

Jelinek,F. and Mercer,R.L. (1980) Interpolated estimation of Markov source parameters from sparse data. In Gelsema,E.S. and Kanal,L.N. (eds), *Pattern Recognition in Practice*. North-Holland Publishing Company, Amsterdam, pp. 381–397.

Katz,S. (1987) Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. Acoust. Speech Signal Process.*, **35**, 400–401.

Lukashin,A.V. and Borodovsky,M. (1998) GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Res.*, **26**, 1107–1115.

Potamianos,G. and Jelinek,F. (1998) A study of N-gram and decision tree letter language modeling methods. *Speech Commun.*, **24**, 171–192.

Press,W.H., Teukolsky,S.A., Vetterling,W.T. and Flannery,B.P. (1992) *Numerical recipes in FORTRAN: The Art of Scientific Computing*. Cambridge University Press, Cambridge.

Ron,D., Singer,Y. and Tishby,N. (1996) The power of amnesia: learning probabilistic automata with variable memory length. *Mach. learn.*, **25**, 117–149.

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

Skovgaard,M., Jensen,L.J., Brunak,S., Ussery,D. and Krogh,A. (2001) On the total number of genes and their length distribution in complete microbial genomes. *Trends Genet.*, **17**, 425–428.

Tavare,S. and Song,B. (1989) Codon preference and primary sequence structure in protein-coding regions. *Bull. Math. Biol.*, **51**, 95–115.

Weinberger,M.J., Rissanen,J.J. and Feder,M. (1995) A universal finite memory source. *IEEE Trans. Inform. Theory*, **41**, 643–652.