GeneMark.hmm: new solutions for gene finding

Alexander V. Lukashin and Mark Borodovsky[1],*

School of Biology and [1]Schools of Biology and Mathematics, Georgia Institute of Technology, Atlanta, GA 30332-0230, USA

Received August 14, 1997; Revised and Accepted December 30, 1997

ABSTRACT

The number of completely sequenced bacterial genomes has been growing fast. There are computer methods available for finding genes but yet there is a need for more accurate algorithms. The GeneMark.hmm algorithm presented here was designed to improve the gene prediction quality in terms of finding exact gene boundaries. The idea was to embed the GeneMark models into naturally derived hidden Markov model framework with gene boundaries modeled as transitions between hidden states. We also used the specially derived ribosome binding site pattern to refine predictions of translation initiation codons. The algorithm was evaluated on several test sets including 10 complete bacterial genomes. It was shown that the new algorithm is significantly more accurate than GeneMark in exact gene prediction. Interestingly, the high gene finding accuracy was observed even in the case when Markov models of order zero, one and two were used. We present the analysis of false positive and false negative predictions with the caution that these categories are not precisely defined if the public database annotation is used as a control.

INTRODUCTION

For the 'post-genomic' molecular biology, a computer became the major tool for interpreting DNA and protein sequence information. By the end of 1997, 10 complete bacterial genomes were available from the GenBank database: *Haemophilus influenzae* (1), *Mycoplasma genitalium* (2), *Methanococcus jannaschii* (3), *Mycoplasma pneumoniae* (4), *Synechocystis* PCC6803 (5), *Escherichia coli* (6), *Helicobacter pylori* (7), *Methanobacterium thermoauthotrophicum* (8), *Bacillus subtilis* (9), *Archeoglobus fulgidus* (10). The majority of genes in these genomes were annotated using theoretical (computer derived) rather than experimental evidence. With many more genomes to come in the near future, the methods of highly accurate DNA sequence interpretation, particularly gene finding, become increasingly important. Here we present a new method, GeneMark.hmm, for gene finding in bacterial genomes. The previously developed GeneMark program (11), that has been used in practice (1–6,9–10), identified a gene mainly as the ORF (open reading frame) where the gene is residing. However, the 5′ boundary of the gene (the translation initiation codon associated with the protein N-terminus) might not be precisely predicted. The range of uncertainty for the initiation codon position is of the size of GeneMark sliding window, i.e. ~100 nucleotides (nt). As a palliative, GeneMark indicates several possible start codons and scores them (http://intron.biology.gatech.edu/GeneMark). However, the exact prediction of the N-terminus is important for further functional analysis of a putative protein, and, eventually, for correct annotation of thousands of genes in growing databases. Therefore we see our goal as developing an algorithm with a high accuracy of exact gene prediction.

Gene annotation in bacterial DNA defines a functional role of each nucleotide in the sequence. For a DNA sequence designated as $S = \{b_1, b_2, ..., b_L\}$, where the b_i stands for the nucleotide symbol, T, C, A or G, and L is the sequence length, the functional role of each nucleotide could be indicated by a 'functional' sequence $A = \{a_1, a_2, ..., a_L\}$. Here each a_i may take integer value '0' if nucleotide b_i is a part of non-coding region; value '1' if b_i is a part of a gene residing in the direct DNA strand; and a value of '2' if b_i is involved in encoding a protein in the complementary DNA strand. The aim of gene finding is to determine the 'true' functional sequence A for the anonymous DNA sequence S. Statistical patterns of nucleotide ordering specific for DNA sequences that carry (or do not carry) the genetic code have been used in gene finding algorithms since the 1980s (see ref. 12 for review). In GeneMark, for instance, these patterns were quantified and converted into parameters of Markov chain models (11). A general pattern recognition algorithm should be able to compute the probability that a particular functional sequence A underlies a given sequence S, $P(A|S) = P(a_1, a_2, ..., a_L | b_1, b_2, ..., b_L)$. The core GeneMark.hmm procedure computes the $P(A|S)$ value and, eventually, defines the functional sequence A^* having the largest value $P(A^*|S)$ among all possible A. The functional sequence A^*, the output of the algorithm, describes the most likely annotation of the DNA sequence S.

The problem of the $P(A|S)$ computation and maximization is considered in terms of hidden Markov models (HMM), the technique that was successfully applied in speech recognition (see ref. 13 for review). Applications of HHM theory to DNA and protein sequence analysis have also been described by several groups (14–21). The algorithm ECOPARSE developed by Krogh *et al.* (17) was the first HMM based gene-finding algorithm intended specifically for the *E.coli* genome. The GeneMark and GeneMark.hmm have been compared with the performance of ECOPARSE (see below).

*To whom correspondence should be addressed. Tel: +1 404 894 8432; Fax: +1 404 894 0519; Email: mark@amber.biology.gatech.edu

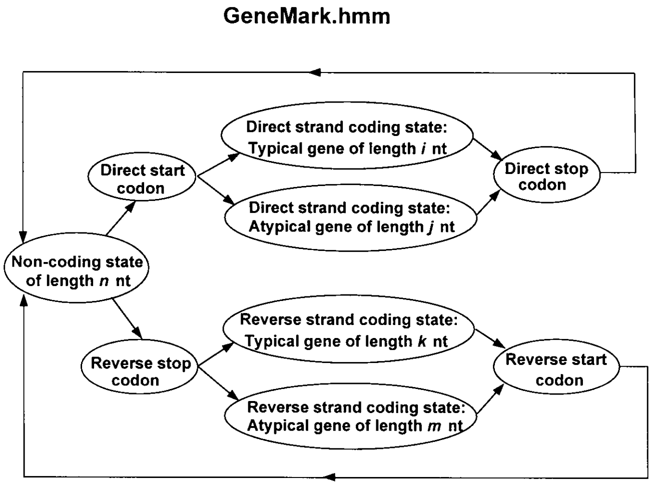
GeneMark.hmm
Direct strand coding state: Typical gene of length *i* nt
Direct start codon
Direct strand coding state: Atypical gene of length *j* nt
Direct stop codon
Non-coding state of length *n* nt
Reverse strand coding state: Typical gene of length *k* nt
Reverse stop codon
Reverse start codon
Reverse strand coding state: Atypical gene of length *m* nt

Figure 1. Hidden Markov model of a prokaryotic nucleotide sequence used in the GeneMark.hmm algorithm. The hidden states of the model are represented as ovals in the figure, and arrows correspond to allowed transitions between the states.

The HMM framework of GeneMark.hmm, the logic of transitions between hidden Markov states, followed the logic of the genetic structure of the bacterial genome (Fig. 1). The Markov models of coding and non-coding regions were incorporated into the HMM framework to generate stretches of DNA sequence with coding or non-coding statistical patterns. This type of HMM architecture is known as 'HMM with duration' (13). The sequence of hidden states associated with a given DNA sequence *S*, carries information on positions where coding function is switching into non-coding and vice versa. Thus, the previously introduced functional sequence *A* becomes equivalent to the sequence of hidden states, called the HMM trajectory. Since the nucleotide sequence *S* is given, every possible sequence *A* could be assessed by the value of $P(A|S)$, the conditional probability of *A* given *S*. This evaluation made use of the whole set of statistical models (see Materials and Methods). The core GeneMark.hmm procedure is the Viterbi algorithm (13) that finds the sequence A^*. However, this core procedure did not take into account the possibility of gene overlaps since the observed overlaps, though frequent, were not extensive enough to provide sufficient data for deriving statistical models of overlapping genes in several possible orientations. To further improve the prediction of the translation start position the model of the ribosome binding site (RBS) was derived. This model was used to refine translation initiation codon prediction at the post-processing step.

The GeneMark.hmm program was evaluated on several test sets including sequences of the 10 complete bacterial genomes mentioned above. The GeneMark.hmm predictions were compared with GeneBank annotations. It was shown that the frequency of exact gene predictions is much higher than that of GeneMark (the version which also used the RBS model). We understand that the evaluation of the algorithm performance by comparison with the database annotation may not be enough conclusive evidence, since only in a few cases is the precise position of the translation initiation codon known from an experiment. However, the database annotation of the initiation codon represents the expert decision summarizing much indirect evidence and is thought to be close to the real one. The GeneMark program, actually, was able to correctly identify ORFs where 98% of all genes predicted by GeneMark.hmm resided. Also there were genes missed by GeneMark.hmm, mainly due to overlaps, that were recovered by GeneMark. However, the GeneMark.hmm program made several new predictions and some of them were confirmed by similarity search. It seems that the GeneMark.hmm development brought us closer to the goal of accurate prediction of bacterial genes and further arguments in favor of this statement are presented below.

MATERIALS AND METHODS

Materials

We have used DNA sequences of the complete genomes of *H.influenzae* (GenBank accession no. L42023), *M.genitalium* (L43967), *M.jannaschii* (L77117), *M.pneumoniae* (U00089), *Synechocystis* PCC6803 (synecho), *E.coli* (U00096), *H.pylori* (AE000511), *M.thermoauthotrophicum* (AE000666), *B.subtilis* (AL009126), *Archeoglobus fulgidus* (AE000782). The data on annotated *E.coli* RBS were provided by W. Hayes (22). The data on experimentally verified N-terminal protein sequences were kindly provided by A. Link (23). The Markov models parameters were obtained from the GeneMark library (http://exon.biology.gatech.edu/ ~genmark/matrices/).

Model of prokaryotic sequence structure

The architecture of the hidden Markov model used in the GeneMark.hmm algorithm is shown in Figure 1. To deal simultaneously with direct and reverse DNA strands, as was done in the initial GeneMark algorithm (11), nine hidden states were defined. These states correspond to the functional units of bacterial genomes, namely: (i) a Typical gene in the direct strand, (ii) a Typical gene in the reverse strand, (iii) an Atypical gene in the direct strand, (iv) an Atypical gene in the reverse strand, (v) a non-coding (intergenic) region, (vi/vii) start/stop codons in the direct strand, and (viii/ix) start/stop codons in the reverse strand. It should be mentioned that this HMM does not account for gene overlap (see below). The models of Typical and Atypical genes were derived from the sets of protein-coding DNA sequence obtained by clusterization of the whole set of genes from the genome of a given species (22). The names 'Typical' and 'Atypical' were used for the following reason. For the *E.coli* genome it was shown that the majority of the *E.coli* genes mainly belong to the cluster of Typical genes, while many genes that are believed to have been horizontally transferred into the *E.coli* genome fall into the cluster of Atypical genes. Note, that the comprehensive accounts on the *E.coli* genes evolutionary classification have been presented earlier (24,25).

An important feature of the proposed HMM architecture is that any coding as well as non-coding hidden state is allowed to generate a nucleotide sequence, observed sequence, of the length of hidden state duration (13). Such an explicit state duration HMM was used previously in algorithms Genie and GENSCAN (18,20). The crucial point, however, is that an observed DNA sequence $S = \{b_1, b_2, ..., b_L\}$ is thought to be generated by an HMM such as depicted in Figure 1, in parallel with the HMM transitions from one hidden state to another. The hidden state trajectory *A*, one of a variety of allowed paths, can be concisely represented as a sequence of *M* hidden states a_i having duration d_i: $A = \{(a_1d_1)(a_2d_2) ... (a_Md_M)\}$, $\Sigma d_i = L$. For a given sequence of observed states (nucleotides) $S = \{b_1, b_2, ..., b_L\}$ the optimal trajectory of hidden (functional) states A^* is defined as the trajectory (functional sequence) A with the maximal value of conditional probability $P(A \mid S)$. Therefore, a computer optimization procedure is supposed to find the maximum likelihood sequence A^* that, according to its physical meaning, defines the predicted locations of protein coding regions in the nucleotide sequence S.

Viterbi algorithm for variable duration HMM

The problem formulated above is equivalent to a problem of finding the trajectory $A^* = \{(a_1{}^*d_1{}^*)(a_2{}^*d_2{}^*) \ldots (a_M{}^*d_M{}^*)\}$ that has the largest probability of occurring simultaneously with the sequence S in comparison with all other possible trajectories:

$$P_{\max} = P(A^*, S) = \max_{\substack{(a_1d_1)\ldots(a_Md_M)\\ \sum_{s=1}^{M} d_s = L}} Prob\left\{(a_1d_1)(a_2d_2)\ldots(a_Md_M), b_1b_2\ldots b_L\right\} \qquad \mathbf{1}$$

To describe the optimization algorithm we introduce the quantity (13):

$$z_1(a_m, d_m) = \max_{\substack{(a_1d_1)\ldots(a_{m-1}d_{m-1})\\ \sum_{s=1}^{m-1} d_s = l-d_m}} [Prob\{(a_1d_1)\ldots(a_{m-1}d_{m-1}), b_1\ldots b_{l-d_m}\}q_{a_{m-1}\,a_m}]\, p_{a_m}(d_m)P_{a_m}(b_{l-d_m+1}\ldots b_l) \qquad \mathbf{2}$$

where m is the number of hidden states visited during generation of the first l nucleotides, $q_{a_{m-1}\,a_m}$ is the probability of transition from hidden state a_{m-1} to state a_m, $p_{a_m}(d_m)$ is the probability of duration d_m for state a_m, and P_{am} $(b_{l-dm+1} \ldots b_l)$ is the probability of observing (generating) the nucleotide sequence, $b_{l-dm+1}, \ldots, b_l$, given the state a_m. By induction (m ≥ 2) we have

$$z_l(a_m, d_m) = \max_{(a_{m-1}d_{m-1})} [z_{l-d_m}(a_{m-1}, d_{m-1})q_{a_{m-1}\,a_m}]p_{a_m}(d_m)P_{a_m}(b_{l-d_m+1}\ldots b_l) \qquad \mathbf{3}$$

$$\{a^*_l(a), d^*_l(a)\} = \arg\max_{(a_md_m)} [z_l(a_m, d_m)q_{a_m\,a}] \quad 2 \le l \le L-1 \qquad \mathbf{4}$$

$$P_{\max} = \max_{(a_Md_M)} z_L(a_M, d_M) \qquad \mathbf{5}$$

$$\{a^*_L, d^*_L\} = \arg\max_{(a_Md_M)} [z_L(a_M, d_M)] \qquad \mathbf{6}$$

Equations **3–6** present the Viterbi algorithm which finds for the given (observed) nucleotide sequence S the maximum likely trajectory A^*. This algorithm is an extension of the Viterbi algorithm, described by Rabiner (13), for the case of HMM with variable duration of hidden states. The equations for straightforward initialization and backtracking procedures are not shown.

Parameters of the model

The described above mechanism of generating nucleotide sequence S by variable duration HMM could naturally use the Markov models of coding and non-coding DNA sequences. These models have been already defined the GeneMark algorithm (11). Therefore, the time-consuming and cumbersome procedure of HMM training was largely avoided. For instance, given a hidden state '1' corresponding to a coding region, the probability, $P_1(b_1, b_2, \ldots, b_d)$, of observing a particular DNA sequence $\{b_1, b_2, \ldots, b_d\}$ as a part of a coding region was calculated using the three-periodic inhomogeneous Markov chain model (11). For non-coding state '0' the probability of observing sequence $\{b_1, b_2, \ldots, b_d\}$ as a part of a non-coding region, $P_0(b_1, b_2, \ldots, b_d)$, was calculated using the homogeneous Markov model (11). The probability $p_a(d)$ that a state a has duration d was defined by analytical approximation of the frequency distribution of the lengths of coding (non-coding) regions in the *E.coli* genome (Fig. 2). As is seen in Figure 1, the only allowed transitions between hidden states were 'non-coding'→'direct start'→'direct coding'→'direct stop'→'non-coding', as well as 'non-coding'→'reverse stop'→'reverse coding'→'reverse start'→'non-coding'. Therefore, just a few additional parameters, such as the probabilities of possible start codons, initial and transition probabilities for hidden states had to be specified. Initial probabilities for four coding and one non-coding states were set to 0.2. Initial probabilities for start/stop states were set to zero. The probabilities of the start codons were defined in agreement with the *E.coli* genome statistics: P(ATG) = 0.905, P(GTG) = 0.090, P(TTG) = 0.005. The probability of transition from a non-coding state to a Typical (Atypical) coding state was set to 0.85 (0.15). These values are the estimates of frequencies of 'native' ('foreign') genes in the *E.coli* genome suggested by Medigue *et al*. (24) and Lawrence (25).

Post-processing: finding RBS

As follows from the described HMM architecture (Fig. 1) the optimal sequence A^* found by the Viterbi algorithm should have predicted genes separated from one another by at least a 1 nt long intergenic region. Therefore, the actual overlap of two genes will prevent finding the exact location of at least one gene. Initially, we considered an overlap of bacterial genes as an unlikely event. However, when the larger body of complete genomic sequences became available we found that at least short overlaps are quite common in bacterial genomes (see below). Obviously, the Viterbi algorithm tends to predict genes involved in overlaps shorter than they really are. Therefore, we used a post-processing procedure, searching for ribosome binding site (RBS), to refine initial Viterbi predictions. For a predicted gene, the RBS was searched in the interval from –19 to –4 nt upstream to each alternative start codons located between the position of start codon suggested by the Viterbi algorithm and the position of start codon producing the longest open reading frame (ORF) for the predicted gene. The initially predicted translation intiation position was redefined if the score of one of the RBS candidates associated with an admitted alternative start exceeded a certain threshold (see below). Otherwise, the position suggested by the Viterbi algorithm was accepted.

The probabilistic model for the RBS was derived as follows. First, the *E.coli* records in GenBank with annotated RBSs were analyzed, and 325 genes with known RBSs were selected from the complete *E.coli* genome (6). Second, from each of these 325 sequences, the 16 nt sequence preceding the annotated start (from –4 to –19) was collected. Third, these 325 short sequences were subjected to the multiple alignment procedure performed by the simulated annealing algorithm (26). Specifically, we have chosen a fixed size window, w, and searched for the best alignment by maximizing a matching score

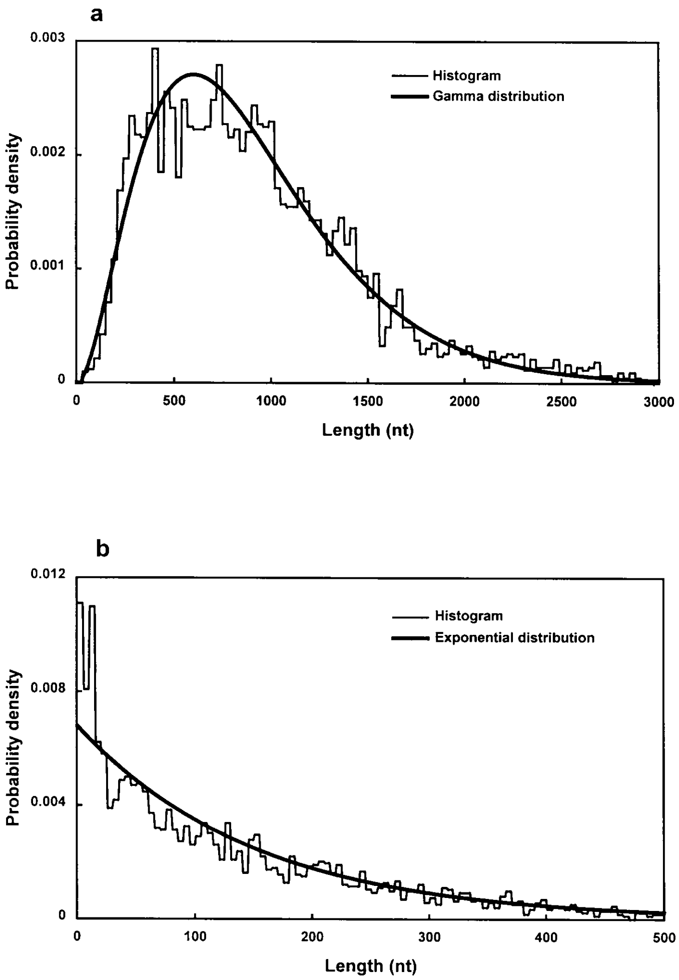

Figure 2. Length distribution probability densities of protein-coding and non-coding regions derived from the annotated *E.coli* genomic DNA (histograms). (**a**) Coding regions; the solid curve is the approximation by γ distribution $g(d) = N_c(d/D_c)^2 \exp(-d/D_c)$, where d is the length in nt, D_c = 300 nt, N_c is the coefficient chosen to normalize the distribution function on the interval from 30 nt (the minimal length of coding region) to 7155 nt (the maximal length). (**b**) Non-coding regions; the solid curve is the approximation by exponential distribution $f(d) = N_n\exp(-d/D_n)$, where D_n = 150 nt. The coefficient N_n normalizes the distribution function on the interval from 1 to 1000 nt.

$$R = \sum_{k=1}^{w} n_b^2(k) \qquad \mathbf{7}$$

Here $n_b(k)$ is the number of symbols b (b = T, C, A, G) in the position (column) k of the window alignment. In each step of the simulated annealing algorithm iterative procedure, one of the 325 sequences chosen at random was shifted to the right or to the left, relative to the fixed window, for a randomly chosen number of positions (with no gaps, deletions or insertions). The matching score R^* for the resulting alignment was calculated (equation **7**). If R^* was larger than R, the new alignment was unconditionally accepted and used as the starting point for the next iterative step. Otherwise, the new alignment was accepted with the probability $\exp[-R - R^*)/T]$, where the parameter T can be interpreted as the ‘temperature’ in the annealing procedure. We used the standard exponential cooling schedule $T_{n+1} = cT_n$, where c = 0.999999. The window size was chosen to be equal to w = 5.

Table 1. Nucleotide frequencies for the RBS model

Nucleotide	Position				
	1	2	3	4	5
T	0.161	0.050	0.012	0.071	0.115
C	0.077	0.037	0.012	0.025	0.046
A	**0.681**	0.105	0.015	**0.861**	0.164
G	0.077	**0.808**	**0.960**	0.043	**0.659**

The model was derived using the multiple sequence alignment of 325 annotated ribosomal binding sites (see text). Given the set of aligned sequences, the frequency of a given nucleotide was calculated as the number of occurrences of this nucleotide in a given position divided by the total number of sequences.

The finally obtained alignment of the 325 sequences has revealed the RBS sequence pattern in the form of a matrix of positional nucleotide frequencies (Table 1). It is seen that the matrix defines the strong consensus sequence: AGGAG, which is complementary to a pentamer located in the *E.coli* 16S rRNA near its 3′-end. This observation is in a good agreement with the generally accepted mechanism of ribosome-mRNA binding. Note that a similar result was obtained previously (27). To evaluate a putative RBS we calculated its probabilistic score as the product of corresponding elements of the matrix given in Table 1. The threshold value for RBS score was chosen as 0.00025. It can be shown that the log of this score is proportional to ribosome binding energy (with appropriate sign) under the assumption of independent formation of ribonucleotide pairs.

Algorithm modifications for genomes other than *E.coli*

The GeneMark.hmm predictions were obtained for nine other bacterial genomes. In these computations we used the species specific Markov models of coding and non-coding regions. All other parameters of the GeneMark.hmm algorithm stayed the same as defined for the *E.coli* genome. It is worth mentioning that for the gram-positive bacterium, *B.subtilis*, we have slightly modified the RBS prediction procedure. In species, such as *B.subtilis*, that do not have the ribosomal protein S1 involved in initiation of the ribosome–mRNA complex, the elevated strength of ribosome binding sites is thought to be a compensatory mechanism to facilitate ribosome binding. For the *B.subtilis* case the described above alignment procedure produced a highly biased frequency pattern with the strong RBS consensus. To obtain reasonable agreement between predicted initiation codons of *B.subtilis* genes and annotated ones we had to admit to competition the alternative start codons located not only upstream to the Viterbi prediction of translation start, but also those located downstream up the 66 nt distance. We think that this rule could be applicable to all other genomes, but presently, there is a tendency in genome annotation process to prefer longer ORFs to shorter ones provided there is no convincing evidence in favor of the shorter one. Statistically, this tendency is well justified since it is expected that in about 75% of cases actual genes occupy the longest ORFs. This figure can be obtained as follows. Consider the set of four codons: ATG, TAA, TAG, TGA and an intergenic region situated upstream to the true initiation codon of a gene *X*. Read codons in 5′ direction in the same reading frame as the initiation codon until the first codon from the above set is met. If this codon is ATG, then the gene *X* does not occupy the longest ORF. Otherwise gene *X* does occupy the longest ORF, which happens in 75% of cases assuming that the four codons specified above occur with equal frequencies and ATG is the only possible initiation codon. In *B.subtilis* the presence of a strong RBS site provided a good reason to override the 'longest ORF' annotation rule and shorter ORFs in *B.subtilis* were annotated more frequently than in other bacterial genomes.

RESULTS AND DISCUSSION

Gene prediction accuracy

The performance of the GeneMark.hmm program was tested using several control sets including 10 complete bacterial genomes. Our focus was on the *E.coli* genome. The complete genomic sequence of *E.coli* consists of 4 639 221 nt with 4288 genes annotated (6). When the GeneMark.hmm program was applied to the *E.coli* genomic sequence, as many as 4440 genes were identified. Each predicted gene was also characterized as Typical or Atypical (22) depending on the type of the underlying coding (hidden) state. Twenty percent of the predicted genes were identified as Atypical ones. The gene finding accuracy was evaluated using four control sets of genes annotated in the *E.coli* genome (Table 2). Control set #1 contained all annotated *E.coli* genes. Set #2 was compiled from non-overlapping *E.coli* genes. The *E.coli* genes whose RBS were annotated in GenBank constituted set #3. The genes coding for proteins with experimentally verified N-termini (23) were included in set #4.

Table 2. The GeneMark.hmm performance

Set #	Number of genes	Prediction method	Exact prediction	Only 3′-end prediction	Missing genes
1	4288	VA	2483 (58%)	1592 (37%)	213 (5%)
1	4288	PP	3233 (75%)	842 (20%)	213 (5%)
2	2821	VA	2017 (71%)	750 (27%)	54 (2%)
2	2821	PP	2268 (80%)	499 (18%)	54 (2%)
3	325	VA	255 (78%)	64 (20%)	6 (2%)
3	325	PP	289 (89%)	30 (9%)	6 (2%)
4	204	VA	156 (76.5%)	47 (23%)	1 (0.5%)
4	204	PP	177 (87.5%)	26 (12%)	1 (0.5%)

The four control sets of annotated genes selected for the comparison are described in the text. The numbers in the rows designated as VA correspond to predictions made by the GeneMark.hmm program with the Viterbi algorithm only. The rows designated as PP show the results of prediction with post-processing (the RBS identification procedure). The 'Exact prediction' column contains the numbers of genes with both 3′-end and 5′-end predicted exactly. The numbers of genes predicted with the 5′-end misplaced are shown in the column 'Only 3′-end prediction'. The genes annotated but not correctly predicted either at 5′- or at 3′-end fall into the category 'Missing genes'. The percentage shown in parentheses is the fraction relative to the total number of annotated genes.

The evaluation results (Table 2) show that the Viterbi algorithm alone (VA) was able to exactly predict 58% of the *E.coli* genes in Set #1. The gene overlap seems to be an important factor indeed, since the percentage of exact gene predictions jumped up to 71% when the overlapping genes were eliminated (Set #2). It is worth mentioning that both the 58% and the 71% figures may not be consistent estimates of the algorithm real performance since the majority of annotated translation initiation codons in control sets #1 and #2 were not verified in experiments. In control sets, #3 and #4, the Viterbi algorithm exactly predicted 78 and 76.5% of the genes respectively. These two close figures give a more realistic estimation of the Viterbi algorithm predictive power for genes with no overlaps.

The percentage of the *E.coli* genes predicted either exactly or with misplaced translation starts was 95, 98, 98 and 99.5% for the sets #1, #2, #3 and #4 respectively. These figures did not change when the RBS prediction was combined with the Viterbi prediction at the post-processing step (PP in Table 2). However, for many genes initially partially predicted by the Viterbi algorithm the correct position of the translation start was found. The fraction of exact predictions increased from 58 up to 75% for set #1, from 71 up to 80% for set #2, from 78 up to 89% for set #3, and from 76.5 up to 87.5% for set #4. One may conclude that RBS correction produces 10% increase in the percentage of exactly predicted genes under non-overlap conditions. Also, it appears from the results of program testing on set #1, that gene overlaps were responsible for ~10% of non-exact predictions.

'Missing' genes (false negatives)

A gene annotated in GenBank was counted as 'missing' in predictions if neither its 5′ nor 3′ boundary was precisely found by the algorithm (even if there was some overlap between annotated and predicted genes). The GeneMark.hmm algorithm missed 213 out of the 4288 annotated *E.coli* genes (set #1 in Table 2). Some of these genes, 113 out of 213, had a length exceeding 300 nt. In fact, the majority of these 113 genes overlapped with genes located in the opposite strand (the 'stop near stop' overlap). This fact, along with the observation that the percentage of missing genes in sets #2, #3 and #4 is lower than in test set #1, explains why these relatively long genes were missing. If an overlap occurs, the stop codons of the two genes fall into the region of overlap, and, consequently, at least one stop codon is overlooked by the algorithm. This means that a local 'mishap' such as just the four nucleotide overlap between two genes (i.e. TTAA, TTAG, CTAA, CTAG) makes the Viterbi algorithm lose the whole gene. Note that many overlapping genes are not likely to be missed by the GeneMark program. Its 'voting' mechanism accounts for detection of the coding potential within a number of windows covering a given ORF, thus suppressing the fluctuations that might affect just a few windows.

'Wrong' gene predictions (false positives)

Among 4440 genes predicted by the GeneMark.hmm program in the *E.coli* genome, there were 363 genes with neither the 5′-end nor the 3′-end matched to any annotated gene. Some of these predictions, 231 out of 363, were located in the regions annotated as non-coding and these 231 predictions might be classified as 'wrong' or 'new'. Thirteen of these predictions had a length larger than 300 nt. The protein products of these putative genes were searched for similarity against the non-redundant protein sequence database using the gapped BLAST (28). Four putative proteins were found to have significant similarity with hypothetical proteins previously identified in other species (Table 3). This analysis indicates once again that genome annotations in public databases are not perfect. Some real genes still may go unnoticed while some already annotated may not be functional. At any rate, 'false positive' gene predictions need much further analysis before they are sorted out as wrong ones. Therefore, the exact fractions of wrong predictions as well as the fractions of predicted new genes remain to be determined.

Table 3. The results of similarity search for four putative *E.coli* proteins

Gene #	Strand	5′-end	3′-end	Score	E-value	Subject
1	comp.	238736	238257	270	4e-72	gi|1552787; hypothetical protein
2	comp.	279586	279248	229	4e-60	pir||I41306; hypothetical protein (argF-lacZ region)
3	direct	1286288	1286854	122	1e-27	gi|1787481; 35 pct identical <3 gaps> to 54 residues of approx. 1040 aa protein BGAL_KLEPH
4	direct	2201992	2202309	217	2e-56	sp|P33347|YEHK; hypothetical 12.6 kDa protein

Locations of the genes are specified. The similarities were found by the gapped BLAST algorithm (28).

Comparison with the earlier programs

We have compared the performance of the GeneMark.hmm program with the GeneMark program (11) and with the ECOPARSE program (17). The ECOPARSE algorithm differs from GeneMark and GeneMark.hmm, particularly, in analyzing DNA strands in turn, one after another, while GeneMark and GeneMark.hmm deal with both strands simultaneously. The test set for this comparison included five *E.coli* DNA contigs of 30 000 nt length each (the maximum possible length for the ECOPARSE e-mail server input sequence as of June, 1997). The predictions for each DNA contig were obtained by each of the three algorithms (including post-processing cycles) and compared with the GenBank annotation (6). The results (Table 4) indicate that the GeneMark.hmm program was more accurate in exact predictions: 71 versus 62% by GeneMark and 53% by ECOPARSE. It is worth mentioning that the current versions of GeneMark and ECOPARSE use RBS models as well. The GeneMark.hmm program also had the least number of missing genes and the highest percentage of annotated genes found exactly or partially (Table 4). Particularly, the genes thrL, yacG, cspE and ydiE missed by GeneMark were detected by GeneMark.hmm.

Table 4. A comparison of the GeneMark.hmm program with the GeneMark program and with the ECOPARSE program

Number of genes	Prediction method	Exact prediction	Only 3′-end prediction	Missing genes
148	GeneMark.hmm	105 (71%)	28 (19%)	15 (10%)
148	GeneMark	92 (62%)	37 (25%)	19 (13%)
148	ECOPARSE	79 (53%)	33 (23%)	36 (24%)

All designations are the same as in Table 2. The data shown are the average results obtained by using five sequences of 30 000 nt in length each from the entire *E.coli* record (5). The left ends of the sequences have been chosen as $(i-1) \times 10^6 + 1$, where $i = 1,\ldots, 5$. Only those annotated genes have been taken for the comparison with predicted parses whose 5′- and 3′-ends are both inside the chosen sequences (148 genes).

Robustness of the algorithm

The GeneMark.hmm performance may depend on the choice of the algorithm parameters. The robustness of the algorithm was tested with regard to the values of the Markov models' transition probabilities. The GeneMark.hmm predictions for *E.coli* were recalculated using the transition probability matrices obtained by training on an alternative set of *E.coli* genes (22). The prediction versus annotation comparisons were close to those shown in Table 2. For example, the number of set #1 genes exactly predicted (with post-processing) was equal to 3088 compared to 3233 shown in Table 2. A 20% variation of other algorithm parameters had changed the overall performance even less noticeably (data not shown).

Other bacterial genomes

The GeneMark.hmm predictions obtained for nine other bacterial genomes were compared with the GenBank annotations and the results are shown Table 5. It is seen that the program, on average, found exact locations of 78.1% of annotated genes. For 94.6% of annotated genes the reading frames were predicted correctly but the initiation codons did not coincide with the annotated one. The average percentage of missing genes was 5.4%. For a particular genome the frequency of missed genes was strongly correlated with the frequency of gene overlaps. The largest frequencies of overlap were observed in *A.fulgidus* (61% of all annotated genes had overlaps), *M.genitalium* (59%) and *M.pneumonia* (51%), while the smallest were found in *B.subtilis* (24%), *H.influenzae* (27%) and *M.jannaschii* (29%). The average percentage of false positive predictions, 10%, is relatively high, but how many of these predictions are actually correct remains to be found by further analysis. We did not use any filters for false positives. Even the restriction on the minimum length of the gene prediction was not applied since the genomic sequence still may contain small pieces of frameshifted genes. Actually, from 382 gene predictions that did not find annotated analogs in *A.fulgidus* genome, 42 have already been confirmed as real genes and their protein products were included in protein sequence database prior to our study. By using the gapped BLAST significant similarities of predicted protein products to known proteins from species other than *A.fulgidus* were found for 18 more predictions. In total, 291 of the GeneMark.hmm 'false positive' predictions for the 10 species were already confirmed to some extent by other researches and were included in protein databases. Another 71 predictions, as the current study shows, have good additional evidence (from the gapped BLAST) to be real genes. Many from the remaining 2068 predictions could be genes encoding so called 'pioneer proteins' (29).

Table 5. Results of GeneMark.hmm predictions for 10 complete bacterial genomes

Genome	Genes annotated	Genes predicted	Exact prediction (%)	Missing genes (%)	Wrong genes (%)
A.fulgidus	2407	2530	73.1	10.8 (2.0)	15.1
B.subtilis	4101	4384	77.5	3.6 (2.8)	9.8
E.coli	4288	4440	75.4	5.0 (2.7)	8.2
H.influenzae	1718	1840	86.7	3.8 (3.2)	10.2
H.pylori	1566	1612	79.7	6.0 (4.4)	8.7
M.genitalium	467	509	78.4	9.9 (1.7)	17.3
M.jannaschii	1680	1841	72.7	4.6 (0.8)	12.9
M.pneumoniae	678	734	70.1	7.8 (4.1)	13.6
M.thermoauthotrophicum	1869	1944	70.9	5.0 (3.5)	8.6
Synechocystis	3169	3360	89.6	4.0 (1.5)	9.4
Averaged	21 943	23 194	78.1	5.4 (2.7)	10.4

The second and third columns show the number of genes annotated in GenBank and the corresponding number of genes predicted, respectively. ‘Exact prediction’ is a fraction of annotated genes for which both the 5′-end and the 3′-end were predicted exactly. ‘Missing genes’ is a fraction of annotated genes for which neither the 5′-end nor the 3′-end was predicted exactly; in this column the numbers in brackets show the missing genes after using the combined program (GeneMark.hmm + GeneMark). ‘Wrong genes’ is a fraction of predicted genes for which no annotated analog was found. All measures are expressed as percentages. The data shown are the results obtained after post-processing procedure (RBS recognition).

Higher order models and models of Typical and Atypical genes

The results presented in Table 2 were obtained by GeneMark.hmm employing second order Markov models of coding and non-coding regions. The graphs in Figure 3a show the percentage of exact predictions as a function of the model order. Surprisingly, even the zero order models yield high enough accuracy. The reason for this is that GeneMark.hmm accumulates detectable signal within the rather long bacterial gene even if the relatively weak zero order model is used. This does not happen with the GeneMark algorithm where the length of an analyzed DNA sequence is restricted by the short window, and, as a consequence, the higher order models are known to be more accurate in coding potential detection (29). The later corresponds, however, to the observation (Fig. 3b) that the number of missing genes, presumably short genes, decreases as the model’s order increases. Note that the slight accuracy improvement observed for higher order models was achieved at the price of a non-linear increase in computer memory requirements. For analysis of eukaryotic DNA with coding regions (exons) being, in average, much shorter than bacterial coding regions this is a well justified price.

The role of Atypical gene model is illustrated in Figure 3. Switching off the Atypical model produced a decrease in the number of exact predictions (Fig. 3a) and an increase in the number of missing genes (Fig. 3b).

Gene overlaps

In spite of casual opinion that gene overlaps are likely to happen only in phage and virus genomes where requirements for tight gene packing are ‘vitally’ important, the complete bacterial genomes demonstrate quite a few gene overlaps. The overlap regions are of special interest because of their double genetic code load. The distributions of length of gene overlaps observed in *E.coli* genome are shown in Figure 4. These length distributions are different for overlapping genes residing in the same strand (Fig. 4a) and for genes residing in opposite strands (Fig. 4b). The overlaps in the same strand are more common, with the trivial overlaps of the length 1 (TGA/ATG) or 4 (ATGA) constituting the majority (406 out of 695 same strand overlaps). An overlap length larger than 48 nt was observed in 45 cases. As expected, there were no observed overlaps in the same strand with a length equal to a multiple of three.

Note that at least one verified example, the *E.coli* gene *infB*, presents a special case that defies the normal rules. The *infB* gene was shown to have two translation initiation codons situated at a significant distance from each other. Two different proteins are encoded within one and the same ORF (30) that can be considered as the exceptional case of same strand overlap.

More than one third of the number of overlaps between genes residing in opposite strands (39 out of 113 overlaps) are the trivial overlaps of length 4 (TTAA, TTAG, CTAA, CTAG). In 23 instances, the overlap length was larger than 48 nt. We observed similar distributions of overlaps in other complete genomes (data not shown). The gene overlaps cause several difficulties for a high accuracy prediction. First, some overlapping genes could be missed (see above). Second, it might be hard to exactly predict the 5′-end of the gene whose translation initiation codon and ribosome binding site fall into the overlap region where oligonucleotide statistics may not fit to regularly used models.

In the extreme case, the overlap may contain a whole gene. For example, in the *E.coli* genome the 714 nt long coding region located near the origin of replication (10 643…11 356) overlaps the 591 nt gene residing in the opposite strand (10 725…11 315). The 714 nt gene was exactly predicted by GeneMark and by GeneMark.hmm (and predicted by ECOPARSE in the region 10 643…11 293). However, all three methods missed the 591 nt gene completely. The existence of the 591 nt gene was experimentally confirmed (31). ‘It is the direct strand 603 nt ORF from which the *E.coli* heat shock protein HtpY is expressed’ (this

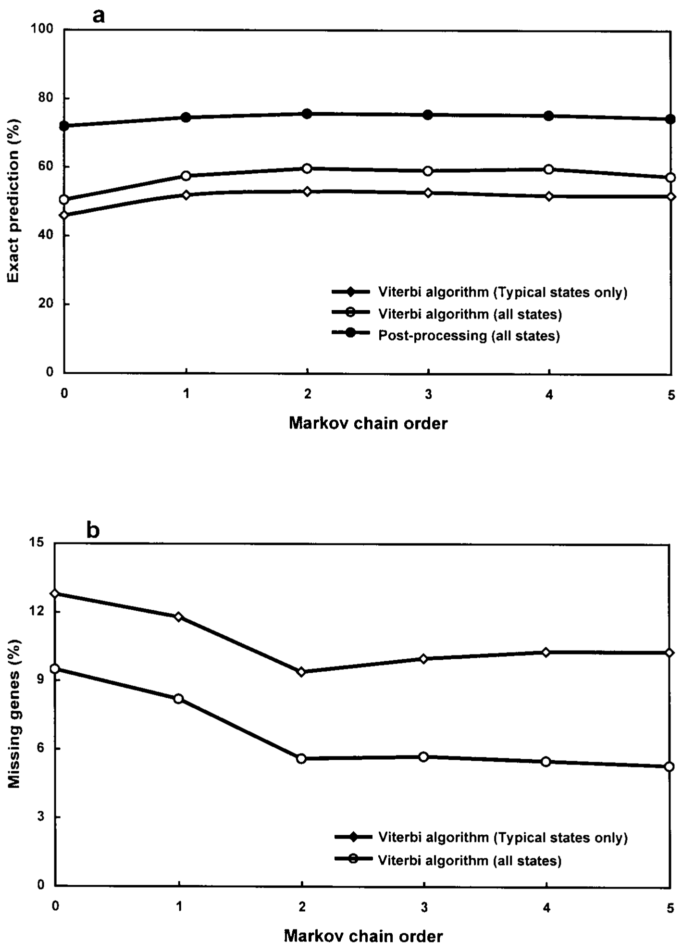

Figure 3. GeneMark.hmm performance as a function of the Markov chain order used to calculate the probability of observed nucleotide sequence. The results of comparison between the annotated and predicted parses are shown for the sequence of the first 500 000 nt taken from the entire *E.coli* genomic sequence. This contig contains 468 annotated genes. (**a**) Exact prediction: the fraction of annotated genes for which both the 5′ -and 3′-ends have been predicted exactly; diamonds: the predicted parse was generated by the Viterbi algorithm using the Markov models for Typical genes only; open circles: the Markov model for Atypical genes were included into the GeneMark.hmm algorithm; filled circles: the parse was corrected by the post-processing with the use of the RBS model. (**b**) Missing genes: the fraction of annotated genes for which neither their 5′- nor 3′-ends were predicted exactly (the post-processing procedure does not change the number of missing genes (see Table 2). The data legend is the same as in (a).

603 nt ORF contains the 591 nt gene mentioned above). With regard to the longer 714 nt ORF in the complementary strand, predicted by the computer methods, all attempts to demonstrate the expression of this ORF remained unsuccessful (31). This led to the conclusion that this ORF 'may be transcribed *in vivo,* albeit at very low levels'. Note that among 197 residues of HtpY protein there is an unusually high abundance of serines (42 residues) and cysteines (17 residues). This highly biased amino acid composition makes the HtpY gene a difficult target for any statistical gene finding method.

GeneMark.hmm and GeneMark combination

The results presented above demonstrate that GeneMark.hmm provides an improved tool for exact prediction of bacterial genes.

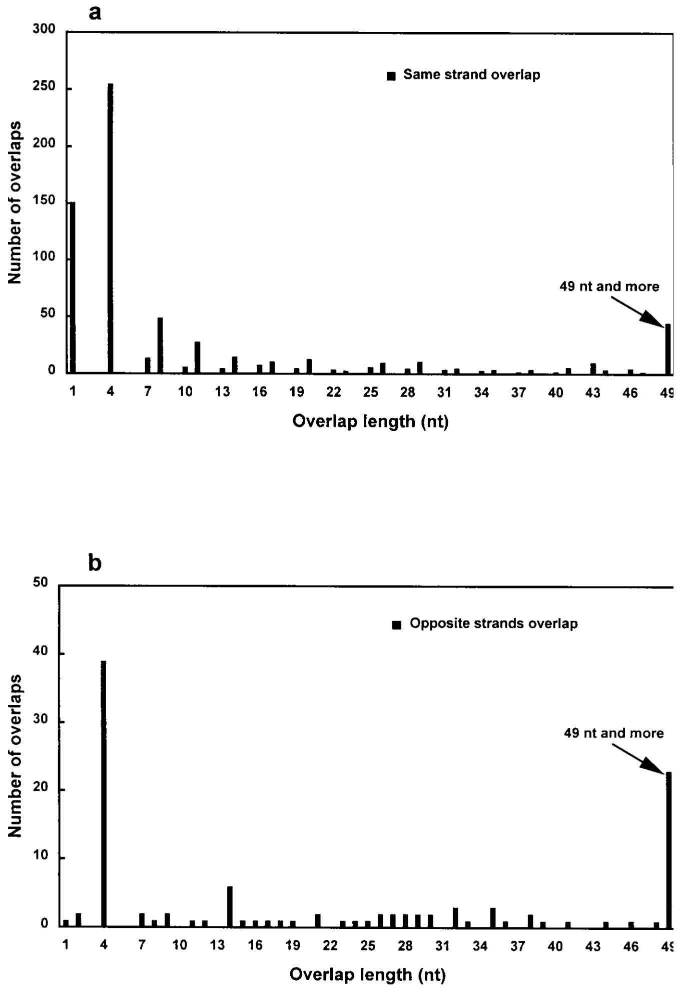

Figure 4. The distribution of the *E.coli* genes overlaps over their length. (**a**) Same strand overlap; (**b**) opposite strands overlap.

One drawback is the tendency to underpredict genes with overlaps. Nevertheless, it is worth mentioning that GeneMark.hmm and GeneMark have complementary properties in the sense that the genes missed by GeneMark.hmm may be recovered by GeneMark and the partial gene predictions made by GeneMark may be corrected by GeneMark.hmm. A combination of the two programs could be, therefore, an even better tool for gene prediction. Note, though, that we do not mean such a combination that would decrease the number of false negative predictions at the mere price of an increase of the number of false positive ones. By selecting those GeneMark predictions that are clear patches to the GeneMark.hmm prediction list we indeed avoided an increase in the number of false positives. The evaluation of the combined program for the 10 genomes has shown that the fraction of missing genes significantly decreased (Table 5). As is seen, one of the largest figures of missing genes, 4.4%, was observed for *H.pylori*. It is worth mentioning that of 956 genes of *H.pylori* that have verified protein database matches, the combined program missed only seven genes. The combined GeneMark.hmm and GeneMark program with about a 1 min run time for a sequence of

100 kb, is available through Internet: http://genemark.biology.gatech.edu/GeneMark

ACKNOWLEDGEMENTS

We thank William Hayes for excellent software programming help and, particularly, for setting up the WWW facilities, James McIninch for valuable assistance, Jim Fickett for useful discussions, David Dusenbery for making valuable remarks on the previous draft of the manuscript. Anders Krogh kindly helped with using the ECOPARSE e-mail server, Andy Link provided the data on proteins with verified N-terminals, Steven Salzberg helped in the *H.pylori* gene prediction comparisons. All this help is greatly appreciated. The project was supported in part by the National Institutes of Health.

REFERENCES

1 Fleischmann, R.D., Adams, M.D., White, O., Clayton, R.A., Kirkness, E.F., Kerlavage, A.R., Bult, C.J., Tomb, J.-F., Dougherty, B.A., Merrick, J.M. *et al.* (1995) *Science*, **269**, 496–512.

2 Fraser, C.M., Gocayne, J.D., White, O., Adams, M.D., Clayton, R.A., Fleischmann, R.D., Bult, C.J., Kerlavage, A.R., Sutton, G., Kelley, J.M. *et al.* (1995) *Science*, **270**, 397–403.

3 Bult, C.J., White, O., Olsen, G.J., Zhou, L., Fleischmann, R.D., Sutton, G.G., Blake, J.A., Fitzgerald, L.M., Clayton, R.A., Gocayne, J.D. *et al.* (1996) *Science*, **273**, 1058–1073.

4 Himmelreich, R., Hilbert, H., Plagens, H., Pirkl, E., Li, B.-C. and Herrmann, R. (1996) *Nucleic Acids Res*., **24**, 4420–4449.

5 Tabata, S. (1996) GenBank accession no. synecho.

6 Blattner, F.R., Plunkett, G., III, Bloch, C.A., Perna, N.T., Burland, V., Riley, M., Collado-Vides, J., Glasner, J.D., Rode, C.K., Mayhew, G.F., Gregor, J., Davis, N.W., Kirkpatrick, H.A., Goeden, M.A., Rose, D.J., Mau, B. and Shao, Y. (1997) *Science*, **277**, 1453–1462.

7 Tomb, J.-F., White, O., Kerlavage, A.R., Clayton, R.A., Sutton, G.G., Fleischmann, R.D., Ketchum, K., Klenk, H.P., Gill, S., Dougherty, B.A. *et al.* (1997) *Nature*, **388**, 539–548.

8 Smith, D.R., Doucette-Stamm, L.A., Deloughery, C., Lee, H.-M., Dubois, J., Aldrego, T., Bashizadeh, R., Blakely, D., Cook, R., Gilbert, K. *et al.* (1997) *J. Bacteriol*., **179**, 7135–7155.

9 Kunst, F., Ogasawa, N., Moszer, I., Albertini, A.M., Alloni, G., Azevedo, V., Bertero, M.G., Bessieres P., Bolotin A., Borchert, S. *et al.* (1997) *Nature*, **390**, 249–256.

10 Klenk, H.P., Clayton R.A., Tomb, J., White O., Nelson K.E., Ketchum K.A., Dodson R.J., Gwinn M., Hickey E.K., Peterson, J.D. *et al.* (1997) *Nature*, **390**, 364–370.

11 Borodovsky, M. and McIninch, J. (1993) *Comput. Chem*., **17**, 123–133.

12 Gelfand, M.S. (1995) *J. Comp. Biol*., **2**, 87–115.

13 Rabiner, L.R. (1989) *Proc. IEEE*, **77**, 257–286.

14 Churchill, G.A. (1989) *Bullet. Math. Biol*., **51**, 79–94.

15 Krogh, A., Brown, M., Mian, I.S., Sjolander, K. and Haussler, D. (1994) *J. Mol. Biol*., **235**, 1501–1531.

16 Baldi, P., Chauvin, Y., Hunkapiller, T. and McClure, M.A. (1994) *Proc. Natl. Acad. Sci. USA*, **91**, 1059–1063.

17 Krogh, A., Mian, I.S. and Haussler, D. (1994) *Nucleic Acids Res*., **22**, 4768–4778.

18 Kulp, D., Haussler, D., Reese, M.G. and Eeckman, F.H. (1996) In States, D.J., Agarwal, P., Gaasterland, T., Hunter, L. and Smith, R.F. (eds), *Proceedings Fourth International Conference on Intelligent Systems for Molecular Biology (ISMB-96).* AAAI Press, Menlo Park, CA, pp. 134–142.

19 Yada, T. and Hirosawa, M. (1996) In States, D.J., Agarwal, P., Gaasterland, T., Hunter, L. and Smith, R.F. (eds), *Proceedings Fourth International Conference on Intelligent Systems for Molecular Biology (ISMB-96).* AAAI Press, Menlo Park, CA, pp. 252–260.

20 Burge, C. and Karlin, S. (1997) *J. Mol. Biol*., **268**, 78–94.

21 Henderson, J., Salzberg, S. and Fasman, K.H. (1997) *J. Comp. Biol*., **4**, 127–141.

22 Hayes, W. and Borodovsky, M. (1998) *Genome Res*., in press.

23 Link, A.J., Robison, K. and Church, G.M. (1997) *Electrophoresis*, **18**, 1259–1313.

24 Medigue, C., Rouxel, T., Vigier, P., Henaut, A. and Danchin, A. (1991) *J. Mol. Biol*., **222**, 851–856.

25 Lawrence, J.G. (1977) *Trends Microbiol*., **5**, 355–359.

26 Lukashin, A.V., Engelbrecht, J. and Brunak, S. (1992) *Nucleic Acids Res*., **20**, 2511–2516.

27 Hayes, W.S. and Borodovsky, M. (1998) *Pacific Symp. Biocomput*., **3**, 279–290.

28 Altshul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J. (1997) *Nucleic Acids Res*., **25**, 3389–3402.

29 Borodovsky, M., McIninch, J.D., Koonin, E.V., Rudd, K.E., Medigue, C. and Danchin, A. (1995) *Nucleic Acids Res*., **23**, 3554–3562.

30 Sacerdot, C., Dessen, P., Hershey, J.W.B., Plumbridge, J.A. and Grunberg-Manago, M. (1984) *Proc. Natl. Acad. Sci. USA*, **81**, 7787–7791.

31 Missiakas, D., Georgopoulos, C. and Raina, S. (1993) *J. Bacteriol*., **175**, 2613–2623.