Journal of Bioinformatics and Computational Biology
Vol. 8, No. 3 (2010) 535–551

DOI: 10.1142/S0219720010004847

Imperial College Press
www.icpress.co.uk

GENETACK: FRAMESHIFT IDENTIFICATION IN PROTEIN-CODING SEQUENCES BY THE VITERBI ALGORITHM

IVAN ANTONOV

Division of Computational Science and Engineering
Georgia Institute of Technology
801 Atlantic Drive, Atlanta
Georgia, USA 30332-0280
ivan.antonov@gatech.edu

MARK BORODOVSKY*

Department of Biomedical Engineering and Division
of Computational Science and Engineering
Georgia Institute of Technology, 313 Ferst Drive
Atlanta, Georgia, USA 30332-0535
borodovsky@gatech.edu

Received 25 October 2009
Revised 12 February 2010
Accepted 13 February 2010

We describe a new program for *ab initio* frameshift detection in protein-coding nucleotide sequences. The task is to distinguish the same strand overlapping ORFs that occur in the sequence due to a presence of a frameshifted gene from the same strand overlapping ORFs that encompass true overlapping or adjacent genes. The GeneTack program uses a hidden Markov model (HMM) of genomic sequence with possibly frameshifted protein-coding regions. The Viterbi algorithm finds the maximum likelihood path that discriminates between true adjacent genes and those adjacent protein-coding regions that just appear to be separate entities due to frameshifts. Therefore, the program can identify spurious predictions made by a conventional gene-finding program misled by a frameshift. We tested GeneTack as well as two earlier developed programs FrameD and FSFind on 17 prokaryotic genomes with frameshifts introduced randomly into known genes. We observed that the average frameshift prediction accuracy of GeneTack, in terms of (Sn+Sp)/2 values, was higher by a significant margin than the accuracy of two other programs. In addition, we observed that the average accuracy of GeneTack is favorably compared with the accuracy of the FSFind-BLAST program that uses protein database search to verify predicted frameshifts, even though GeneTack does not use external evidence. GeneTack is freely available at http://topaz.gatech.edu/GeneTack/.

Keywords: Frameshifts; hidden Markov models; the Viterbi algorithm.

*Corresponding author.

1. Introduction

Analysis of the complete genome sequences available in public databases (such as NCBI) revealed that some protein-coding regions contain frameshifts, i.e. changing frame of reading the genetic code. The sequence behind the frameshifted genes could be either entirely correct, thus, the frameshift appears to be a consequence of mutation, or the sequence may contain a sequencing error.

Frameshifts due to mutations are of special interest. Programmed ribosomal frameshifts[1] belong to a category of translational recoding events annotated in the RECODE database.[2] With tRNA slippage (relocation) from codons in the initial frame into overlapping codons in the alternative frame, the ribosome changes the reading frame, producing a protein encoded by overlapping ORFs.[1] Such frameshifts play regulatory roles in gene expression and are usually flanked by signaling sequences. Sequencing error-induced frameshifts are of significant interest as well. A volume of genomic data is increasing dramatically with advent of next-generation sequencing technologies (454,[3] Illumina,[4] \ SOLiD[5]). Still, the assembly of a huge mass of short sequence reads may result in less homogeneous sequence coverage and higher rate of sequence errors than at a time of "slow sequencing". Errors of insertion or deletion type that occur inside protein-coding regions lead to frameshifts (unless the indel size is a multiple of three) and to erroneous gene prediction. It is highly desirable to detect frameshifts early and to correct predicted errors before genome sequence release.

Programs[6–8] developed to identify programmed frameshifts rely on the signaling sequences and do not accurately identify frameshifts related to sequence errors. Several programs have been developed to detect frameshifts of both kinds: natural ones as well as frameshifts due to sequencing errors. These programs can be divided into two groups with respect to the approach they use: (i) those based on comparative genomics (similarity search), and (ii) those based on single sequence (*ab initio*).

Similarity search-based programs[9–14] use translation of concatenated ORFs located in the same DNA strand as a query for a protein database search. The search may identify a database protein with statistically significant similarity region (a hit) that overlaps the junction in the "chimeric" query. Such an outcome indicates either a frameshift or naturally occurred events of gene fusion or gene fission occurred in evolution. To discriminate between these events, further analysis of conservation of the protein primary structures in multiple species is required. The similarity-based methods have a clear limitation: it is impossible to detect frameshifts in genes of orphan proteins that do not have known homologs. The *Ab initio* (single sequence-based) approach does not have this limitation; it was implemented in the programs ProFED,[15] FrameD[16] and FSFind[17] A comprehensive review of existing methods can be found in Kislyuk *et al.* (2009).[17]

Here, we present a new algorithm and the program for *ab initio* frameshift detection in nucleotide sequences containing intronless genes (prokaryotic genomes,

metagenomes, phage genomes, EST sequences). The GeneTack program (tack – a zigzag movement) is designed to run on a DNA fragment with all genes located in the same strand. To analyze the whole genome, we use a combination program, GeneTack-GM, a wrapper around GeneTack utilizing earlier developed program GeneMarkS[18] (GM) which makes a parse of the whole new genome into fragments with collinear genes.

It should be noted that GeneTack predicts both types of frameshifts: natural and error-related. However, in the case of a natural frameshift, the GeneTack predictions may not be as effective as ones made by a specialized program since GeneTack does not use information about signaling sequences.

2. GeneTack Algorithm

The problem of predicting protein-coding regions has been successfully solved by the algorithms employing hidden Markov models (HMMs)[18–20]. Some of these algorithms include provisions for finding frameshifts (EcoParse,[21] EasyGene,[22] FrameD[16]). The accuracy of frameshift finding by these programs was not systematically assessed.

The logic of the GeneTack algorithm is as follows. The program takes as an input a fragment of a genomic sequence containing collinear genes in the direct strand. Such fragments could be selected based on gene predictions by GeneMarkS. Assuming that the frameshift may result in prediction of two (overlapping or not) adjacent genes located in the same strand, we designed GeneTack to discriminate between correctly predicted ingenious adjacent genes and those adjacent genes that are predicted due to a sequence error and a split of a single gene by a frameshift.

The algorithm uses a probabilistic model (HMM) that allows for three alternative scenarios: presence of true overlapping genes, true non-overlapping adjacent genes, and adjacent genes (overlapping or not) predicted due to the presence of a frameshift (Fig. 1). The HMM consists of 28 states divided into four groups (Table 1): (i) states $1, 2$ and 3 emit protein-coding sequence and correspond to the three possible "global" reading frames; (ii) the state denoted as "n/c" emits a non-coding sequence; (iii) six states designated as "$i - j$", where $i, j = \{1, 2, 3\}$ and $i \neq j$, emit sequences where two adjacent genes overlap (here numbers i and j indicate the global frames of the upstream and downstream genes respectively); and (iv) 18 states emitting nucleotides of start and stop codons (shown as triangles and squares on the diagram).

Each hidden state emits a single nucleotide. The emission probability of a nucleotide X depends on the type of hidden state as well as the nucleotides emitted earlier. If this probability depends on s previous nucleotides then the probability of emission of a nucleotide X from a hidden state K, $\mathrm{P}_k(\mathrm{X})$ is numerically equal to the value of transition probability for the order s Markov chain model: $P_k(X_i) = P_k(X_i|X_{i-s}, X_{i-s+1}, \ldots, X_{i-1})$ defined for the state K. In the computations described below we used 4^{th}-order three-periodic Markov chains[23] as emission

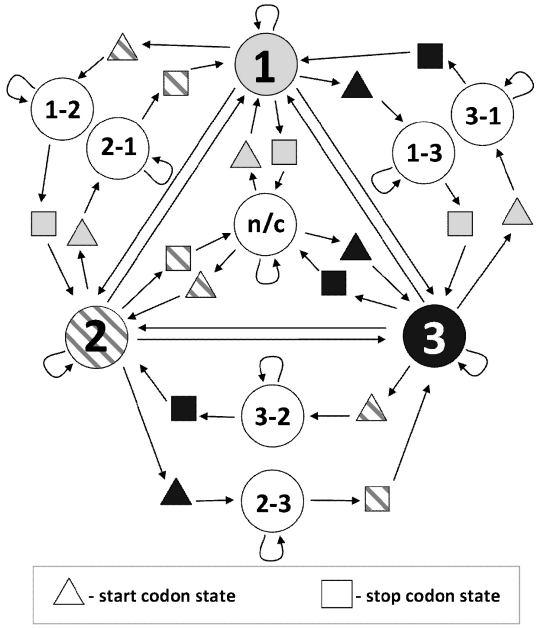


Fig. 1. Hidden Markov model used in GeneTack. States 1, 2 and 3 correspond to the three global frames of reading the genetic code. The type of shading of a state reflects its frame. There is no single frame related to n/c and overlap states, thus they have no shading.

Table 1. Types of states used in the GeneTack HMM and the properties of the emission probabilities.

Type	State(s)	Periodicity	Order
Coding states	1, 2, 3	3	4
Non-coding state	n/c	1	4
Start/stop states	9 start and 9 stop states	3	2
Overlapping states	"$i-j$", $i,j=1,2,3$; $(i \neq j)$	3	4

probabilities for the states $1, 2$ and 3 (see Table 1). The maximum likelihood path through the HMM with respect to the given DNA sequence is determined by the Viterbi algorithm[24] This path makes a decoding of the DNA in terms of which nucleotide corresponds to which hidden state. Notably, the transitions between hidden states in the determined maximum likelihood path carry important information. Direct transitions between states $1, 2$ and 3 correspond to frameshifts; transition between states $1, 2$ and 3 through the "n/c" state(s) indicates a presence of non-overlapping adjacent genes; transition between states $1, 2$ and 3 through "$i-j$" states indicates a presence of overlapping adjacent genes.

The initial (terminal) hidden state in the analysis of the DNA fragments with collinear genes is supposed to be either "n/c" or "start" or "stop" state. Input for GeneTack program includes a file with genome-specific parameters of the HMM in Fig. 1.

3. GeneTack-GM Algorithm

For rather long genomic sequences where genes may reside in both strands, we need an initial run of the self-training GeneMarkS program[18] to estimate the GeneTack-GM parameters and to make a parse of the whole sequence into fragments with collinear genes. The logic of operations of GeneTack-GM running on a new genomic sequence is shown in Fig. 2.

The GeneMarkS program runs in several iterations to determine parameters of Markov chain models for protein-coding and non-coding regions[18] that will be used also in the GeneTack run. At the end of the training process, GeneMarkS defines the final set of predicted genes. Since GeneMarkS is not designed to recognize frameshifts, instead of a gene with a frameshift, a pair of adjacent genes in the same strand (overlapping or not) will be predicted. All sequence fragments that contain collinear genes predicted by GeneMarkS may contain a frameshift. The output of GeneMarkS is used to split the genomic sequence into genomic fragments (Fig 2(b)) that carry collinear genes augmented by non-coding flanks on both sides (500 nt or less). It is convenient to analyze sequences with gene in direct strand, therefore, reverse complements are used if the original fragments contain genes in complementary strand. Further, the GeneTack program is applied to each fragment to identify possible frameshifts. Finally, to reduce the number of false positive predictions, we apply several filters (Fig. 2(c)). The decision rules and parameters

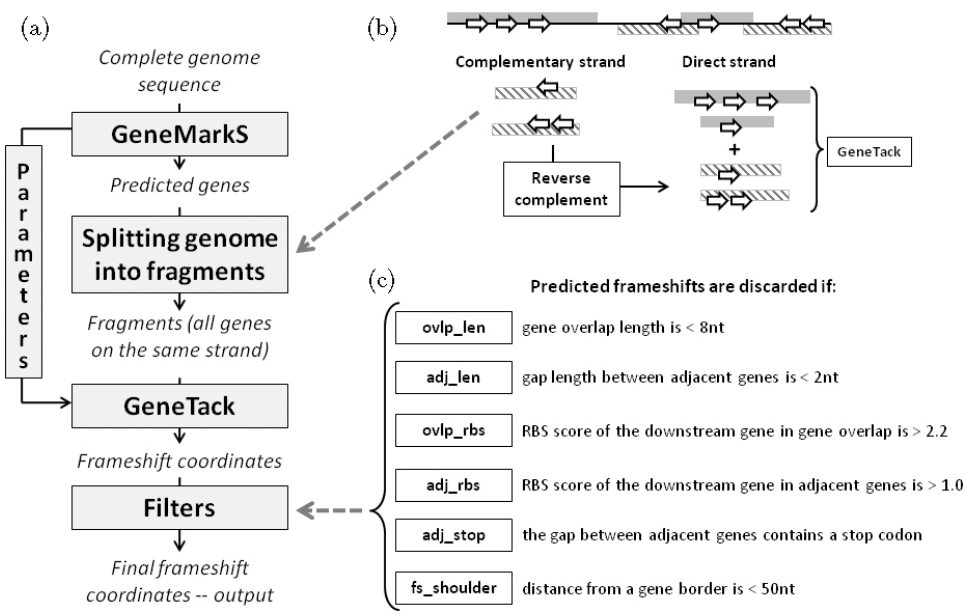


Fig. 2. (a) GeneTack-GM algorithm overview; (b) splitting genome into fragments; (c) description of filters used to reduce number of false positive predictions (filters listed in the order they applied).

of the filters were determined from the results of the analysis of GeneTack-GM predictions for the *E. coli* genome with artificially introduced frameshifts. The predicted frameshifts possessing the following features were discarded:

(A) in the sequences carrying predicted non-overlapping gene pairs:

- if the gap (an intergenic region) between genes is less than 2 nt;
- if there is a stop codon in the upstream region (the gap length + 20 nt) to the start codon of the downstream gene (in the same frame);
- if the score of the RBS for the downstream gene is larger than 1.0;

(B) in the sequences carrying predicted overlapping gene pairs:

- if the overlap region length is shorter than 8 nt;
- if the score of the RBS for the downstream gene is larger than 2.2;

(C) in the sequences from both A/ and B/ classes — if the predicted frameshift location is closer than 50 nt to a border of the coding region.

Our analysis has shown that the indicated values of parameters produce sufficiently accurate results for genomes with various GC content (see below).

3.1. *Parameter estimation*

Emission probabilities for the states $1, 2, 3$ and "n/c", the coding and non-coding states, are defined in the run of GeneMarkS. To compile a standard training set for estimation of emission probabilities for overlapping regions is difficult since overlaps longer than 1 nt and 4 nt are rare in real genomic sequences. To overcome this difficulty we used the following heuristic model that uses emission probabilities of nucleotides defined for non-overlapping coding states. Presence of two overlapping genetic codes reduces probability of accumulating so-called neutral mutations (usually mutations in the third position of codon) because the mutation would also touch either the 1st or the 2nd position of a codon in another gene. In the model for the gene overlapping states, the first and the second positions of a codon are considered as the "strong" ones and the third position as the "weak" one. We assume that in an overlapping region strong codon positions dominate weak positions, i.e. if the first (strong) position of the upstream gene overlaps the third (weak) position of the downstream gene, the emission probabilities typical for the first position will be used. Note that two weak positions never overlap. If two strong positions overlap (for example, the 2nd position of upstream gene and 1st position of downstream gene) then an emission probability, F_{12}, is calculated as an average between two "strong" probability values:

$$F_{12}(\alpha|\mathit{prefix}) = \frac{1}{2}(F_1(\alpha|\mathit{prefix}) + F_2(\alpha|\mathit{prefix})), \tag{1}$$

where α is a nucleotide, *prefix* is a string of upstream letters with the length equal to the order of the Markov chain model of the coding region, F_1 and F_2

Table 2. An example of emission probabilities calculation for overlap of genes carrying the genetic code in frames 1 and 2, (for the 1-2 hidden state). The pattern of frequencies (F_1, F_{12}, F_2) repeated for the whole sequence carrying overlapping genes is shown in bold font.

Position	0	1	2	3	4	5	6	7	8	9
Position % 3	0	1	2	0	1	2	**0**	**1**	**2**	0
Gene in frame 1	F_1	F_2	F_3	F_1	F_2	F_3	F_1	F_2	F_3	F_1
Gene in frame 2					F_1	F_2	F_3	F_1	F_2	F_3
State 1–2					F_{12}	F_2	$\mathbf{F_1}$	$\mathbf{F_{12}}$	$\mathbf{F_2}$	F_1

are probabilities of nucleotide emission from a coding state for the first and second codon positions, respectively. This heuristic approach is illustrated in Table 2.

Another important group of the HMM parameters are transition probabilities between hidden states. The sum of probabilities of all outgoing transitions for each state must be equal to one. Since the GeneTack HMM is symmetrical, only two transition probabilities are needed to be defined: probability of transition between coding states (p), and probability of transition to the start codon (Fig. 3). As mentioned above, a frameshift is predicted if a *direct* transition between coding states does occur. Therefore, the value of this probability can be interpreted as a probability of a frameshift. The value of p should be different for different genomes because frequency of sequence errors depends on the sequencing method. The default value of parameter $p = 0.0006$ was chosen to minimize the frameshift prediction errors in experiments with the *E. coli* genomic sequences. Although we expect this value to be different for other genomes, the difference is apparently very small, given the comparable to the *E. coli* case figures of frameshift prediction accuracy in other genomes where we have used the same value p (see below).

Probability of return (i.e. transition from a state to itself) for a coding state is $1 - 2p$ (Fig. 3). All around, there are ten circular transition probabilities defined for the three coding states, the n/c state and the overlap states. In the current implementation all of them have the same values.

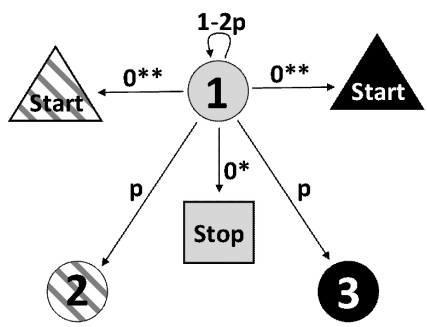


Fig. 3. Calculation of transition probabilities for GeneTack HMM. *Transition probability to stop codon (upon approaching TAA, TAG, TGA) is 1. **Transition probabilities to start codon in frames 2 and 3 (upon approaching ATG, GTG, TTG) are 0.0001 (0.001 for high GC genomes).

Notably, each coding state has two more probabilities, the ones that control transition to the stop state of the same frame and the start state of the overlapping downstream gene. These transition probabilities are sequence-dependent; the transition probabilities are equal to zero in each position in a sequence which does not complete a start or stop codon triplet. For example, with a sequence NNTGA and T in the first position of a codon, emission of the A is made from a "stop" state upon transition from preceding coding state with probability one. Upon approaching a possible start codon, transition probability 0.0001 to start codon (0.001 for high GC genomes, see below) is used.

3.2. *High-GC genomes*

To analyze genomes with high GC content, we have made two modifications. First, for genomes with GC content higher than 65%, instead of 0.0001, we use 0.001 as the sequence-dependent transition probability to a start codon state. This choice reduces the number of false positive predictions in high-GC genome where the frequency of AT reach triplets such as start (as well as stop) codons is lower than in low- and mid-GC genomes. The lower (0.0001) value of transition probability to start codon makes less likely prediction of gene overlap and forces the program to make frameshift predictions more frequently. Second, we have observed that for high-GC genomes the parse of a genome into segments with collinear genes, as predicted by GeneMarkS, does not deliver all the candidates for frameshift detection. In some cases, a gene split by a frameshift is interpreted by the GeneMarkS program not as a pair of genes in the same strand but, surprisingly, as a pair of genes in different strands (Fig. 4).

This misinterpretation is explained as follows. First, in a gene in a high-GC genome the third position of a codon is occupied by C or G nucleotides in 80–90% of cases. Thus, the reading frame in the complementary strand which mirrors the reading from of a true gene has a strong three-periodicity of C and G as well. Second, with diminished frequency of stop codons, we observe long ORFs that occur by chance; an appearance of such ORF in the mirror frame in the complementary strand makes it a candidate for false positive gene prediction. Third, interruption of the true gene by a frameshift does not preclude an initial reading frame from continuation to a significant distance (100–200 nucleotides) until the stop codon type triplet would occur at random. Since a coding potential exists only in a true coding section of this elongated ORF it can be omitted by the Viterbi algorithm in favor of the shorter but actually non-coding ORF with low coding potential located in the complementary strand. Thus, the pair of genes predicted in the place of a gene with frameshift turns out to be not a collinear gene pair; moreover, one of the predicted genes has no coding region at all. This frameshift-related prediction of a gene in a wrong strand poses a problem. Now, the parse of a sequence will split the pair of coding regions originated from a gene with the frameshift into separate fragments and make the frameshift detection impossible (Fig. 2(b)). Such outcomes result in a drop in Sensitivity value observed in the computational modeling.

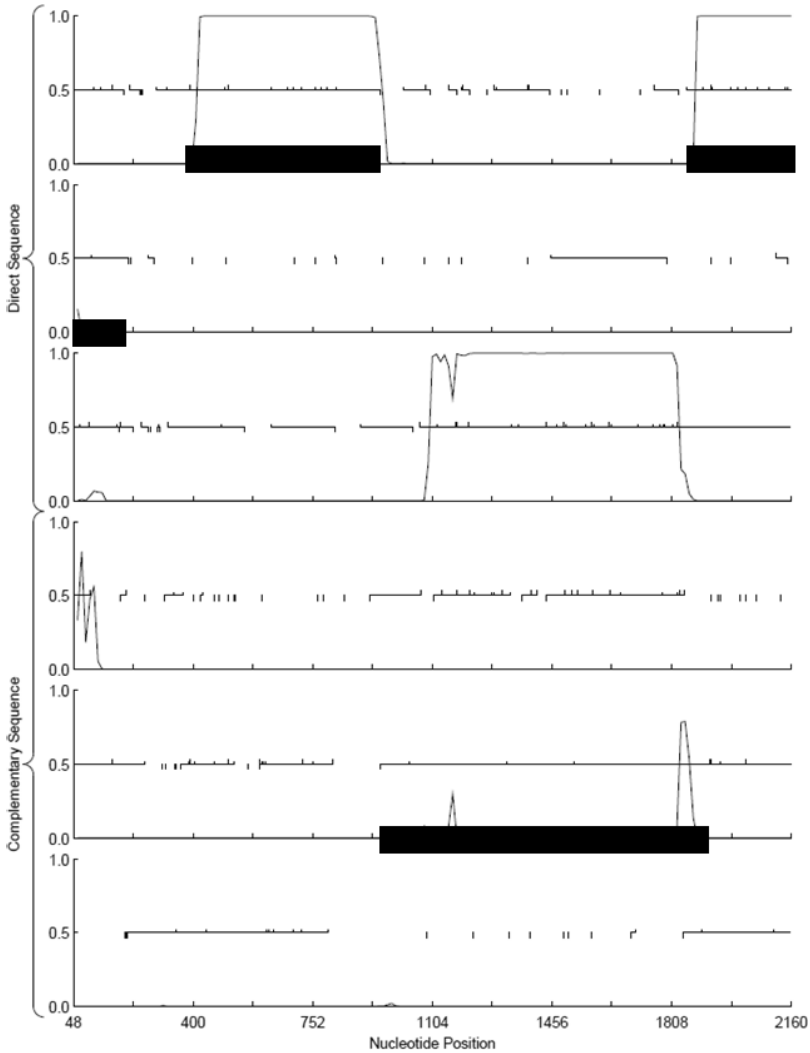


Fig. 4. An example of the GeneMarkS gene prediction for a gene with simulated frameshift in a high-GC genome. The figure shows the coding potentials in all six frames as determined by the GeneMark program[23] Black bars on the horizontal axis indicate predicted genes. A frameshift was introduced at position 1848 in the gene on direct strand. There is a clear jump of coding potential from frame 3 to frame 1 at the location where the frameshift was introduced. However, there is a gene predicted in frame 2 on the opposite strand. Such artifacts are corrected by the modification in GeneTack-GM for high GC genomes as described in the text.

To deal with the problem we have modified the parsing procedure for high-GC content genomes. We use the output of the GeneMark program[23] to calculate an average coding potential for each gene predicted by GeneMarkS. If an average coding potential for a predicted gene is less than 0.4 while it is larger than 0.6 for an ORF in the opposite strand, we reassign the predicted strand of a gene. This

reassignment effectively elongates the upstream part of the fragment with collinear genes and thus includes the earlier missed position of a potential frameshift into the sequence fragment for the GeneTack analysis.

4. Results

4.1. *Datasets*

The accuracy of the GeneTack-GM predictions was assessed on 17 prokaryotic genomes with GC content ranging from 28% to 75% (*E. coli* genome used to estimate program parameters is not included in the dataset in order keep training and test datasets separate). From this set, we generated datasets to test program performance at different gene length ranges.

Dataset_1000 included 17 genomes with frameshifts simulated in 400 genes longer 1000 nt. Dataset_600_1000 included 17 genomes with frameshifts simulated in 200 genes with length ranges from 600 to 1000 nt.

In both datasets frameshifts were simulated by insertion of a single nucleotide into a randomly selected gene at a random position located at a distance of at least 180 nt (dataset_1000) or 100 nt (dataset_600_1000) from either gene end. The accuracy of the frameshift detection for the case when a frameshift was located closer than 100 nt to one of the gene borders was studied separately (see below).

4.2. *GeneTack-GM performance; comparison with other programs*

GeneTack-GM as well as two earlier developed frameshift prediction programs, FrameD[16] and FSFind,[17] were applied to both dataset_1000 and dataset_600_1000. Coordinates of predicted frameshifts were compared with precisely known coordinates of simulated frameshifts. A predicted frameshift was considered as a true positive (TP) if it was located not farther away than 50 nt from the real frameshift, otherwise prediction was considered as false positive (FP). A simulated frameshift was classified as "not found", a false negative (FN) prediction case, if no predicted frameshifts were reported in the 50 nt vicinity of the simulated frameshift.

Program performance was characterized by conventional characteristics Sensitivity (Sn) and Specificity (Sp). The value of Sn is defined with respect to the actual number of simulated frameshifts and the value of Sp is defined with respect to the number of predictions made:

$$Sn = \frac{TP}{TP + FN}, \tag{2}$$

$$Sp = \frac{TP}{TP + FP}. \tag{3}$$

To compare the GeneTack performance with the performance of the FrameD program genomic sequences with artificial errors were submitted to the FrameD web server (http://bioinfo.genotoul.fr/apps/FrameD/FDM.pl) for model generation. The models were used in a local copy of the FrameD program. The FSFind program was installed and run on the local server.

Table 3. Frameshift prediction accuracy estimation for 17 prokaryotic genomes (sorted by GC content). The Sn and Sp values were calculated for GeneTack-GM, FrameD, FSFind and FSFind-BLAST programs. The programs were compared based on average sensitivity and specificity (Sn+Sp)/2. Bold numbers indicate the best performance. *FSFind-BLAST results for *R. solanacearum* were not available because of a runtime error, thus the average values were computed for 17 genomes.

	GC %		GeneTack-GM		FrameD		FSFind		FSFind-BLAST	
Methanosphaera stadtmanae	28	Sn	71.3	77.2	62.5	72.5	65.5	72.4	64.5	**77.5**
		Sp	83.1		82.5		79.2		90.5	
Campylobacter jejuni	31	Sn	81.7	**73.3**	60.2	60.1	64.9	63.2	63.4	71.9
		Sp	64.9		60.0		61.5		80.3	
Staphylococcus aureus Mu50	33	Sn	79.8	**80.1**	49.5	68.4	63.0	69.7	60.5	75.6
		Sp	80.4		87.2		76.4		90.6	
Picrophilus torridus	36	Sn	83.8	75.1	68.0	64.4	84.8	74.8	85.3	**85.5**
		Sp	66.3		60.7		64.7		85.7	
Streptococcus pyogenes	39	Sn	77.3	**75.8**	42.0	61.4	58.8	67.0	56.8	72.2
		Sp	74.3		80.8		75.1		87.6	
Pasteurella multocida	40	Sn	83.8	**81.9**	54.8	71.6	73.5	77.8	70.8	81.5
		Sp	80.0		88.3		82.1		92.2	
Bacillus subtilis	44	Sn	79.5	**71.4**	40.5	52.3	62.0	58.2	60.3	66.1
		Sp	63.2		64.0		54.4		71.9	
Thermotoga maritima	46	Sn	82.8	76.9	77.5	72.8	76.0	67.1	73.3	**78.3**
		Sp	71.0		68.1		58.1		83.2	
Archaeoglobus fulgidus	49	Sn	89.3	68.3	70.0	60.0	82.5	65.4	81.0	**77.5**
		Sp	47.2		50.0		48.3		74.0	
Pyrobaculum aerophilum	51	Sn	85.2	64.7	60.3	46.8	61.4	55.4	54.6	**65.7**
		Sp	44.2		33.2		49.3		76.8	
Thermococcus kodakaraensis	52	Sn	86.0	81.2	77.8	73.8	78.5	75.0	76.8	**83.0**
		Sp	76.3		69.7		71.4		89.2	
Salmonella typhimurium	52	Sn	85.3	71.8	64.5	66.5	75.5	70.3	74.0	**79.3**
		Sp	58.2		68.4		65.1		84.6	
Methanopyrus kandleri	61	Sn	87.0	**73.3**	74.2	66.6	72.9	55.8	70.7	58.1
		Sp	59.5		59.0		38.6		45.4	
Ralstonia solanacearum	67	Sn	93.0	**88.5**	95.0	86.5	79.8	70.1	*n/a**	*n/a**
		Sp	84.0		78.0		60.4		*n/a**	
Caulobacter crescentus	67	Sn	96.0	**87.3**	95.5	82.8	86.3	65.6	83.5	70.1
		Sp	78.5		70.1		44.9		56.7	
Clavibacter michiganensis	73	Sn	98.5	**81.0**	98.3	78.6	66.5	59.2	61.0	61.3
		Sp	63.4		58.9		51.8		61.6	
Anaeromyxobacter dehalogenans	75	Sn	97.3	**82.5**	98.0	77.3	59.8	53.0	52.0	56.3
		Sp	67.7		56.6		46.1		60.5	
AVERAGE:		Sn	**85.8**	**77.0**	69.5	68.2	71.5	66.1	68.3	72.8*
		Sp	68.2		66.9		60.6		**77.3**	

For all 17 genomes in dataset_1000, GeneTack-GM has shown better average values of (Sn+Sp)/2 than FrameD and FSFind (run in *ab initio* mode) with margins of 9.4% and 9.1%, respectively (Table 3). Notably, every genome in the test set could contain additional inherent frameshifts. For instance, the *E. coli* genome contains

33 annotated programmed frameshifts. Since we could not know the locations of additional frameshifts, we considered the frameshifts predicted in the locations not coinciding with the artificial frameshifts as false positive for all programs. Therefore, the actual performance of each tested program could be even better in terms of Specificity than it appears in Table 3. Additional comparison was done with the FSFind program running in the mode of verification of predicted frameshifts via BLAST analysis by search for similarity to tentative translations of frameshifted genes[17] in the nr database. This step improves the Sp value. Still, the overall average (Sn+Sp)/2 on 17 genomes is not as high as we have observed for GeneTack run in the purely *ab initio* mode (Table 3).

The data on program performances on dataset_600_1000 are shown in Suppl. Table 1. It indicates that performance of the same set of program, though reduced, is ranked in the same way. Specifically, the observed (Sn+Sp)/2 values are 66.2%, 60.3%, 56.3% and 64.1% for GeneTack, FrameD, FSFind and FsFind-BLAST, respectively.

5. Discussion

5.1. *Can GeneTack predict programmed frameshifts?*

We applied GeneTack to the 23 DNA sequences, from 19 different species, retrieved from the RECODE database[2] containing +1 and −1 annotated programmed frameshifts. GeneTack successfully predicted annotated frameshifts in 18 sequences. The five sequences where GeneTack did not predict frameshifts had in fact no frameshifts on DNA level; in all five cases the coding region lengths were multiples of three. Notably, the notion of a frameshift was used by the authors[2] in a general sense: shifting the reading frame in the process of translation. In these five cases the ribosome could either translate a gene from start to end or, under certain conditions, the ribosome could change the frame at a certain point and quickly get to a stop codon. This type of translation regulation has been experimentally observed and was documented in RECODE. This case study indicates that GeneTack can be used in a pipeline for prediction of programmed frameshifts. The pipeline could also contain filters to decrease the number of false positives by checking for presence of signal sequences in the vicinity of programmed frameshifts.

5.2. *Insensitivity zones*

It is difficult to detect frameshifts located close to the gene start or end. Thus, we have defined two *insensitivity* zones for GeneTack at the borders of a gene. To determine the characteristic length of the insensitivity zone (expected be of about the same size at both ends), we conducted the test on individual genes flanked by 500 nt of non-coding sequence and with frameshifts introduced at a distance from the gene border ranging with step 5 nt from 1 to 200 nt. The analysis was done for 400 genes from the *E. coli* genome longer than 1000 nt (Fig. 5).

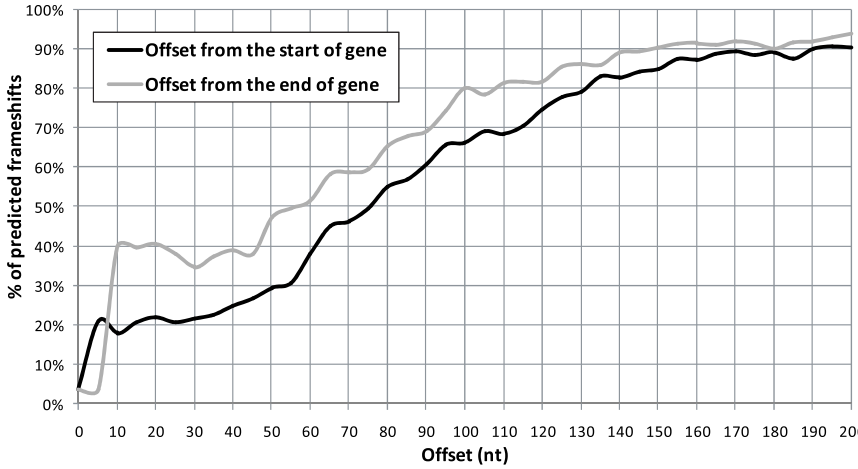


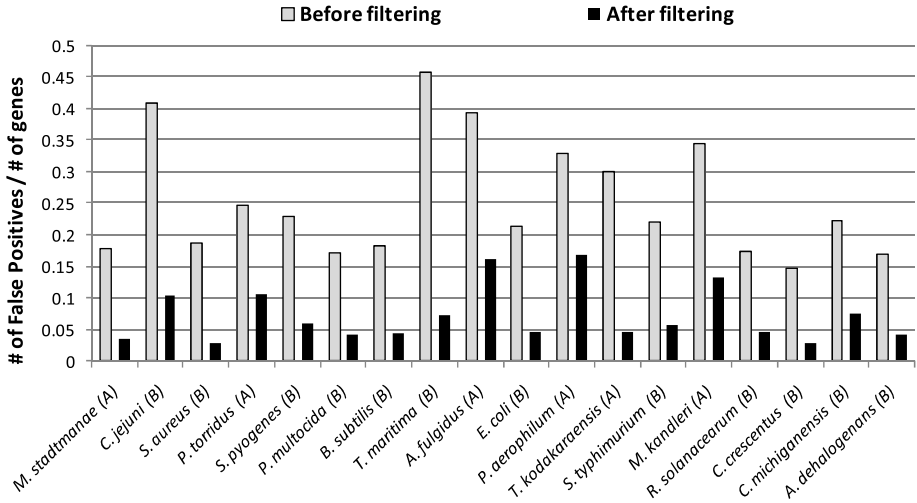
Fig. 5. Dependence of the number of correctly predicted frameshifts on the distance from the artificially made frameshift to the gene border (either start or end).

It is seen that GeneTack correctly detects frameshifts with offset 60 nt from the gene *end* in ∼50% of genes and frameshifts with offset 75 nt from the gene *start* also in ∼50% of genes. The accuracy increases steadily as the offset grows and at 180 nt the performance reaches saturation (>90%). The length of 180 nt was chosen as the minimal distance for a simulated frameshift from the gene borders in the accuracy tests described above.

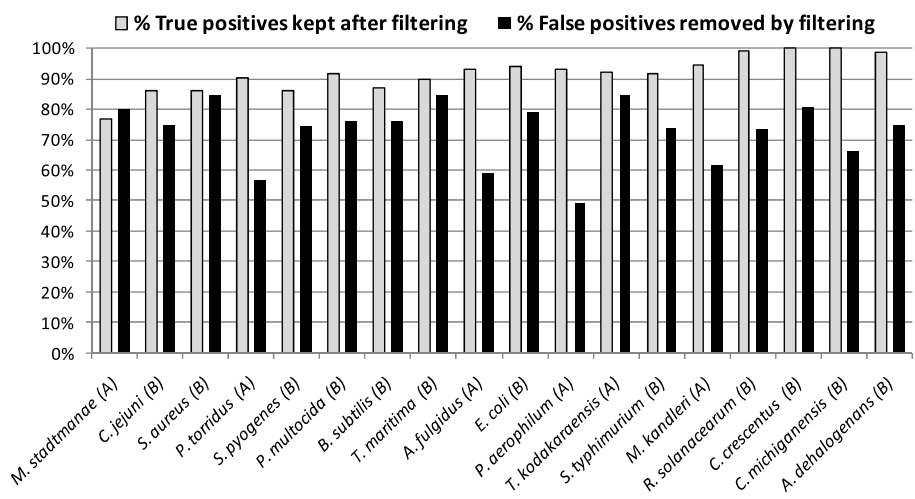
We observed (Fig. 5) that GeneTack is able to detect frameshifts located close to the gene end better than the ones simulated in the beginning of a gene. The observation can be explained as follows. We need to show that it is easier to predict adjacent genes (overlapping or not) if a frameshift is located at a given distance downstream from a true start of a gene than if a frameshift is located at the same distance upstream to the gene end. We have to consider the expected distance L_s from a frameshift down to the *random* stop codon forming the short upstream gene in the adjacent gene pair in the "start" case. On the other hand, we have to consider the expected distance L_e from a frameshift up to the *random* start codon forming the downstream gene in the gene pair in the "end" case. Obviously, with three stop codons and one (two) start(s) in the genetic code, L_s is smaller than L_e. Therefore, in the "start" case a larger part of a short gene in the gene pair will be occupied by the true coding region. Thus, the chance of predicting the short gene making the gene pair (hence no frameshift) is larger in the "start" case.

5.3. *Filters effectiveness*

We have assessed a filter's performance by the percentage of eliminated false positive predictions and the percentage of true positive predictions it keeps in the list. These values were calculated for each of the 17 genomes from the dataset_1000 plus *E. coli* genome (Fig. 6).



(a)



(b)

Fig. 6. Performance of the filters for 18 prokaryotic genomes (genomes are shown along the X axis, sorted by GC content). A domain of life is indicated in parenthesis ("A" stands for Archaea and "B" for Bacteria) (a) Filtering false positive predictions. The fraction of false positives with respect to the total number of genes in a genome, before (gray bars) and after (black bars) filtering are shown for each species. (b) Relative impact of filtering on true positives and false positives. For each genome percentages of removed false positives (with respect to false positives before filtering) and kept true positives (with respect to the number of true positives before filtering) are shown. The filters are supposed to remove as many false positives and as few true positives as possible. Thus, the sum of heights of two bars reflects the filters performance for a given genome. The best performance was observed for *Caulobacter crescentus*, the worst performance was for *Pyrobaculum aerophilu*.

On average, the filters remove 72% of false positives and keep 91% of true positives initially predicted by GeneTack (Fig. 6(b)). At the same time filters have different effectiveness for different genomes.

One of the reasons for the variability in effectiveness is that the same filter parameters, optimized for *Escherichia coli* genome, are used for genomes with different GC content. In the future improvement of GeneTack we have to make filter parameters dependent on genome GC content.

Also, the level of conservation of the RBS site is variable between genomes. There are two filters, *ovlp_rbs* and *ajd_rbs* (Fig. 2(c)), that rely on the RBS score determined by GeneMarkS; these two filters do not work efficiently for genomes with weak RBS. For example, for *Pyrobaculum aerophilum*, the species that has a weak RBS for genes inside operons and no RBS at all for the first genes in operons due to the use of leaderless transcripts, GeneTack-GM predicts the largest number of false positive frameshifts, with only 49% of false positives filtered out. Similarly, these two filters work poorly for *Archaeoglobus fulgidus* with *adj_rbs* filtering out 26 FP and 10 TP. In contrast, for *Thermococcus kodakaraensis* and *Thermotoga maritime,* the *ovlp_rbs* and *ajd_rbs* filters remove more than 84% of false positive predictions. Together, these two filters eliminate 217 false positives and 16 true positives frameshifts for *Thermococcus kodakaraensis* and 286 false positives and 19 true positives for *Thermotoga maritime*.

The GeneTack-GM program can be adapted for analysis of other genomic sequences with intronless genes, such as metagenomic sequences as well as EST sequences. For metagenomic sequences GeneTack-GM can use the heuristic models[19] that allow for quite accurate gene prediction in short sequences, i.e. without a knowledge of full genomic context for estimating parameters of the three-periodic Markov chain model of the coding region. For the EST sequences that belong to one and the same species, the training procedure of GeneMarkS has been modified to account for the Kozak pattern at the gene start (Ter-Hovhannisyan & Lomsadze, unpublished). The models thus derived in the training on the sequenced transcripts (EST) can be immediately used to run GeneTack to detect the frameshifts. Note that alignment of EST sequences to genomic sequence helps to correct majority of frameshifts. Still, the genome projects that focus on sequencing EST only will benefit from using GeneTack-GM to correct the gene and protein predictions.

Acknowledgments

Authors would like to thank Andrey Kislyuk, Alex Lomsadze and Wenhan Zhu for valuable technical support, and Pavel Baranov for useful discussions on programmed frameshifts. The work of IA and MB was supported in part by the NIH grant HG00783 to MB.

References

1. Hansen TM, Baranov PV, Ivanov IP, Gesteland RF, Atkins JF, Maintenance of the correct open reading frame by the ribosome, *EMBO Rep* **4**(5):499–504, 2003.
2. Bekaert M, Firth AE, Zhang Y, Gladyshev VN, Atkins JF, Baranov PV, Recode-2: new design, new search tools, and many more genes, *Nucleic Acids Res*, 2009.
3. Ronaghi M, Pyrosequencing sheds light on DNA sequencing, *Genome Res* **11**(1):3–11 (2001).
4. Johnson DS, Mortazavi A, Myers RM, Wold B, Genome-wide mapping of in vivo protein–DNA interactions, *Science* **316**(5830):1497–1502, 2007.
5. Shendure J, Porreca GJ, Reppas NB, Lin X, McCutcheon JP, Rosenbaum AM, Wang MD, Zhang K, Mitra RD, Church GM, Accurate multiplex polony sequencing of an evolved bacterial genome, *Science* **309**(5741):1728–1732, 2005.
6. Theis C, Reeder J, Giegerich R, KnotInFrame: prediction of -1 ribosomal frameshift events, *Nucleic Acids Res* **36**(18):6013–6020, 2008.
7. Bekaert M, Ivanov IP, Atkins JF, Baranov PV, Ornithine decarboxylase antizyme finder (OAF): fast and reliable detection of antizymes with frameshifts in mRNAs, *BMC Bioinformatics* **9**:178, 2008.
8. Moon S, Byun Y, Kim H-J, Jeong S, Han K, Predicting genes expressed via −1 and +1 frameshifts, *Nucleic Acids Res* **32**(16):4884–4892, 2004.
9. Birney E, Thompson JD, Gibson TJ, PairWise and SearchWise: finding the optimal alignment in a simultaneous comparison of a protein profile against all DNA translation frames, *Nucleic Acids Res* **24**(14):2730–2739, 1996.
10. Claverie JM, Detecting frame shifts by amino acid sequence comparison, *J Mol Biol* **234**(4):1140–1157, 1993.
11. Guan X, Uberbacher EC, Alignments of DNA and protein sequences containing frameshift errors, *Comput Appl Biosci* **12**(1):31–40, 1996.
12. Pearson WR, Wood T, Zhang Z, Miller W, Comparison of DNA sequences with protein sequences, *Genomics* **46**(1):24–36, 1997.
13. Posfai J, Roberts RJ, Finding errors in DNA sequences, *Proc Natl Acad Sci USA* **89**(10):4698–4702, 1992.
14. States B, Harris D, Segal S, Differences between OK and LLC-PK1 cells: cystine handling, *Am J Physiol* **261**(1 Pt 1):C8–16, 1991.
15. Médigue C, Rose M, Viari A, Danchin A, Detecting and analyzing DNA sequencing errors: toward a higher quality of the Bacillus subtilis genome sequence, *Genome Res* **9**(11):1116–1127, 1999.
16. Schiex T, Gouzy J, Moisan A, de Oliveira Y, FrameD: A flexible program for quality check and gene prediction in prokaryotic genomes and noisy matured eukaryotic sequences, *Nucleic Acids Res* **31**(13):3738–3741, 2003.
17. Kislyuk A, Lomsadze A, Lapidus AL, Borodovsky M, Frameshift detection in prokaryotic genomic sequences, *Int J Bioinform Res Appl* **5**(4):458–477, 2009.
18. Besemer J, Lomsadze A, Borodovsky M, 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**(12):2607–2618, 2001.
19. Besemer J, Borodovsky M, Heuristic approach to deriving models for gene finding, *Nucleic Acids Res* **27**(19):3911–3920, 1999.
20. Lukashin AV, Borodovsky M, GeneMark.hmm: new solutions for gene finding, *Nucleic Acids Res* **26**(4):1107–1115, 1998.
21. Shmatkov AM, Melikyan AA, Chernousko FL, Borodovsky M, Finding prokaryotic genes by the 'frame-by-frame' algorithm: targeting gene starts and overlapping genes, *Bioinformatics* **15**(11):874–886, 1999.

22. Larsen TS, Krogh A, EasyGene – A prokaryotic gene finder that ranks ORFs by statistical significance, *BMC Bioinformatics* **4**:21, 2003.
23. Borodovsky M, McIninch J, GeneMark: parallel gene recognition for both DNA strands, *Computers & Chemistry* **17**(19):123–133, 1993.
24. Durbin R, Eddy S, Krogh A, Mitchison G, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*, Cambridge University Press, 1998.

Ivan Antonov received his M.Sc. in Bioengineering and Bioinformatics from Moscow State University, Russia in 2008. From February 2005 to June 2008 he worked at GeneGo Inc. developing software for systems biology and drug discovery.

He is currently a Ph.D. student in Bioinformatics at the College of Computing at Georgia Tech, Atlanta, Georgia, USA under supervision of Dr. Mark Borodovsky.

Mark Borodovsky received his M.Sc. in Physics and Operations Research and Ph.D. in Applied Mathematics from Moscow Institute of Physics and Technology.

He is currently Director of the Center for Bioinformatics and Computational Genomics as well as Regents' Professor of Biomedical Engineering and Computational Science and Engineering at Georgia Tech and Emory University, Atlanta, Georgia, USA.