# Computational Identification of –1 Frameshift Signals

Sanghoon Moon, Yanga Byun, and Kyungsook Han*

School of Computer Science and Engineering, Inha University, Inchon 402-751, Korea
{jiap72,quaah}@hanmail.net, khan@inha.ac.kr

**Abstract.** Ribosomal frameshifts in the –1 direction are used frequently by RNA viruses to synthesize a single fusion protein from two or more overlapping open reading frames. The slippery heptamer sequence XXX YYY Z is the best recognized of the signals that promote –1 frameshifting. We have developed an algorithm that predicts plausible –1 frameshift signals in long DNA sequences. Our algorithm is implemented in a working program called FSFinder (Frameshift Signal Finder). We tested FSFinder on 72 genomic sequences from a number of organisms and found that FSFinder predicts –1 frameshift signals efficiently and with greater sensitivity and selectivity than existing approaches. Sensitivity is improved by considering all potentially relevant components of frameshift signals, and selectivity is increased by focusing on overlapping regions of open reading frames and by prioritizing candidate frameshift signals. FSFinder is useful for analyzing –1 frameshift signals as well as discovering unknown genes.

## 1 Introduction

Translation is the mechanism of protein synthesis in which RNA messages are transformed into the amino acid sequences of proteins. Two kinds of errors can alter the reading frame during translational elongation. One is spontaneous error that occurs at a frequency of less than $5 \times 10^{-5}$ per codon in all species. The other is non-standard error (also called *programmed translational frameshift*) that occurs in some genes with a frequency close to 100% [1, 2].

Programmed frameshift occurs in genes of organisms ranging from bacteria to lower eukaryotes, as well as in animal and plant viruses. The analysis of programmed frameshift is important because it plays a significant role in viral particle morphogenesis, and in the genetic control of alternative enzymatic activities [2]. In this process the ribosome shifts a reading frame by one or a few nucleotides at a specific site in a messenger RNA. The most common of these events requires the ribosome to shift to a codon that overlaps a codon in the existing frame. The shift of a single step backwards in effect reassigns a single nucleotide (-1 frameshift), whereas a slip forwards skips a single nucleotide (+1 frameshift) [3]. The most common type of frameshift is a -1 shift. The most common elements causing eukaryotic frameshifts consist of a slippery site that promotes frameshifting mechanically, and a stimulatory structure that probably induces the ribosome to pause [4]. The slippery site consists of

[*] To whom correspondence should be addressed. Email: khan@inha.ac.kr

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3036, pp. 334–341, 2004.

a heptameric sequence of the form X-XXY-YYZ (in the incoming 0-frame), where X, Y and Z can be the same nucleotide [4]. The downstream stimulatory structure is usually a pseudoknot in which certain bases in a loop pair with complementary bases outside the loop, or it is a simple stem-loop. The slippery heptamer is separated from the stimulatory structure by a short sequence of 5 to 9 nucleotides, the so-called spacer [5, 6]. The length of the spacer is known to influence the probability of frameshifting. Typically viral frameshifts produce fusion proteins in which the amino- and carboxy-terminal domains are encoded by overlapping open reading frames [7], as shown in Fig. 1.

Many existing approaches to identifying frameshift signals either depend on comparing DNA sequences with protein sequences in databases [11, 12], or focus on detecting experimental errors [13]. We have developed a set of algorithms that consider both downstream pseudoknots and simple stem-loops as downstream stimulatory structures in the overlaps between open reading frames. We have implemented these algorithms in a program called FSFinder (Frameshift Signal Finder).

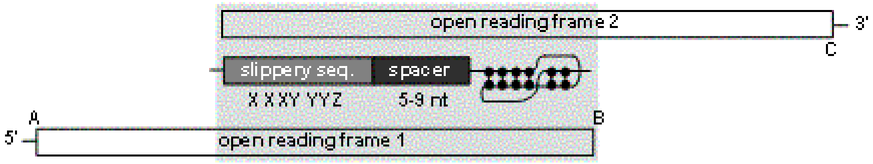


**Fig. 1.** Three components of –1 frameshift signals in the overlap between two open reading frames: slippery sequence, spacer, and pseudoknot (or stem-loop). When a frameshift takes place, protein synthesis terminates at C rather than at B

## 2 Computational Model

### 2.1 Components of Frameshift Signals

We extended the computational model for –1 frameshift signals of Hammell *et al.* [7] to improve its sensitivity and selectivity. Sequences of 3 codons (9 nucleotides) in a genomic sequence are first examined for possible slippery sequences X XXY YYZ. In X XXY YYZ, X and Z can be any nucleotide, and Y can be A or U (in Hammell's model, Z is either A, U, or C). If a slippery sequence is identified, FSFinder searches for a downstream structure by sliding along the spacer from one to 11 nucleotides. Fig. 2 (A) shows a programmed –1 frameshift signal with a pseudoknot as stimulatory structure. The pseudoknot is of the H-type, in which stem 1 has 13 base pairs, stem 2 has 6 base pairs, and both loops of the pseudoknot have 6 nucleotides. The first 4 base pairs of stem 1 must include at least 2 G-C pairs.

Some programmed –1 frameshift signals have a simple stem-loop as stimulatory structure. As explained in Fig. 2 (B), we examine the nucleotides in both directions from every pivot nucleotide for possible base pairing. The pivot nucleotide can be either included or excluded in the base pairing.

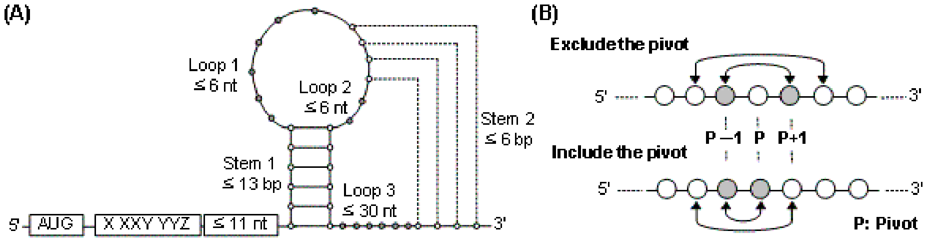


**Fig. 2.** **(A)** A programmed -1 ribosomal frameshift signal with an H-type pseudoknot. **(B)** The process of finding a simple stem-loop structure downstream from a slippery sequence. Nucleotides in both directions from each pivot nucleotide are examined for possible base pairing

## 2.2 Algorithms for Predicting Frameshift Signals

Algorithms 1 and 2 search for stem-loops and canonical base pairs, respectively. If a stem-loop crosses other stem-loops, they are considered to form a pseudoknot. Algorithm 3 finds an overlapping region of open reading frames (ORF). An overlapping region of ORFs is identified by first finding pairs of stop codons in frames –1 and 0. If the second stop codon of in frame –1 is to the left to the fist stop codon in frame 0, an overlapping region of the two frames is found. Overlapping frames with the largest ORF (light yellow) have the highest probability of containing frameshift signals, and overlapping frames with the second largest ORF (sky blue) have the second highest probability of having frameshift signals (see Fig. 3).

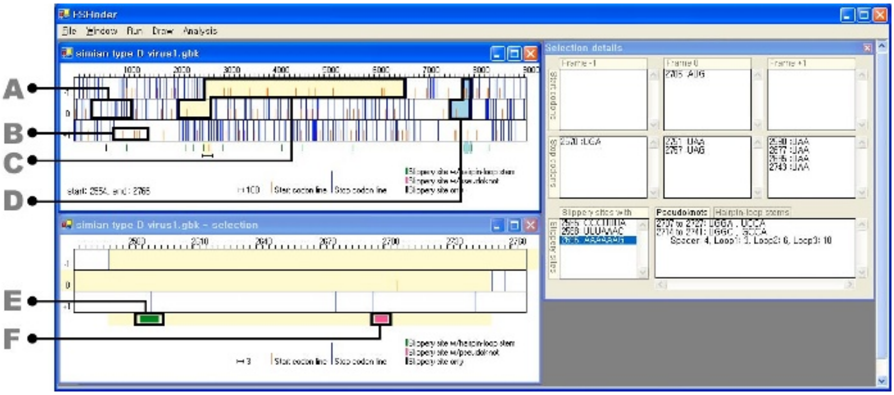
```
Algorithm 1 Find stems
Initialize boolean sequence array S[n,2], n is a length of the
sequence
  Set S according to each character of the sequence
   [false, true]: boolean type of 'A'
   [false, false]: boolean type of 'G'
   [true, true]: boolean type of 'C'
   [true, false]: boolean type of 'U'
  Find slippery sites
  for each slippery site do
    for i ← end index of a slippery site + 1 to n - 8 do
      Set ST be an empty list of stems
      checkCanonicalPair(S, false,
                         start index of a slippery site, i, ST)
    checkCanonicalPair(S, true,
                   start index of a slippery site, i, ST)
    end for
    Set stems of slippery site as ST
    findPseudoknots(slippery site)
  end for
```

**Algorithm 2**
checkCanonicalPair(seqArray, includePivot, startIndex,
pivotIndex, ST)

```
Set L be an empty list of base pairs
if startIndex = pivotIndex then
  return
end if
pairCount ← pivotIndex   2 + 1
insertLase ← false
if not checkMiddle then
  pivotIndex ← pivotIndex - 1
  pairCount ← pairCount - 1
end if
for i ← startIndex to pivotIndex do
  pairIndex ← pairCount - i
  if isCanonicalPair(seqArray[i], seqArray[pairIndex]) then
   if not insertLast then
     if size of L ≥ 4 then
        insert L to ST
     end if
    end if
   Set L be an empty list of pair
    insert pair(i, pairIndex) to L
     insertLast ← true
  else
  insertLast ← false
  end if
end for
if size of L ≥ 4 then
insert L to ST
end if
```

**Algorithm 3** FindOverlappingRegion

```
Set F0 be N + 1 element array of regions, N is count of frame 0 stop
codons
Set FM1 be M + 1 element array of regions, M is count of frame -1
stop codons
Set each element of F0 and FM1 as a region between front stop codon
and next stop codon, order of frame0 and frame -1 stop codon index
 Set O be an empty array of overlapping region
 j ← -1
 for i ← 0 to N do
   while j < M do
     j ← j + 1
    if end of F0[i] ≤ start of FM1[j] then
        break
      else if (start of F0[i] < start FM1[j]) and
         (end of F0[i] < end of FM1[j]) then
         Let R as a region (start of FM1[j] to end of F0[i])
         if any slippery site exists in R then
           insert R to O
       end if
      end if
  end while
 end for
```



2705 :AUG

2570 :UGA

2721 :UAA
2757 :UAG

2590 :UAA
2677 :UAA
2695 :UAA
2743 :UAA

2565 :CCCUUUA
2568 :UUUAAAC
2616 :AAAAAAG

2707 to 2727: UGGA . UCCA
2714 to 2741: UGGC . GCCA
Spacer: 4, Loop1: 3, Loop2: 6, Loop3: 10

**Fig. 3.** Graphical user interface of FSFinder. **A.** Stop codons (long, blue lines). **B.** Start codons (short, red lines). **C.** Frameshift signal with the highest probability (light yellow). **D.** Frameshift signal with the second highest probability (sky blue). **E.** Frameshift signal with a stem-loop (green bar). **F.** Frameshift signal with a pseudoknot (pink bar)

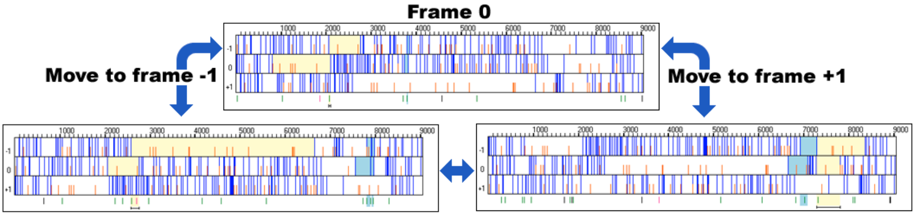


**Fig. 4.** Alternating frames

### 2.3 Implementation

FSFinder was implemented in Microsoft C#. It provides graphical views of -1, 0, and +1 frames, like DNA Strider [8]. The three frames (-1, 0 and +1 frames) are shown in the left upper window of Fig. 3. If a user specifies a region for detailed examination by the drag and drop operation in the left upper window, the specified region is enlarged in the lower left window. The right window displays the positions of start and stop codons, slippery sequences, pseudoknots and stem-loops found in the frames in the left window. Users can change the stem and loop sizes of a stem-loop or pseudoknot. They can also alternate frames to find frameshift signals in different overlapping frames. (see Fig. 4).

## 3 Results and Discussion

FSFinder was tested on 71 organisms with known programmed -1 frameshift mutations obtained from the databases PseudoBase [9] and RECODE [10]. PseudoBase contains 20 eukaryotic viruses and RECODE has 65 prokaryotes, eukaryotic viruses, bacteriophages, eukaryotic transposable elements and bacterial insertion sequences. The two databases share 14 frameshifts. Each of these organisms and elements has one or two authentic programmed -1 frameshift sites.

Hammell *et al.* [7] have attempted to identify frameshift signals in prokaryotic and eukaryotic DNA sequences [7], but the sensitivity of their approach is low. It misses many frameshift signals because it only considers pseudoknots as downstream stimulatory structures, the definition of pseudoknots is too restricting, and X XXY YYG is not considered a slippery sequence. For example, their approach does not locate the frameshift signals in Rous sarcoma virus (RSV), because loops 1 and 2 of the pseudoknots involved are larger than their approach permits. On the other hand, the selectivity of the computational model of Bekaert *et al.* [5] is low because it predicts too many false positives. Other computational models can identify potential frameshift signals only when they are given reference protein sequences along with DNA sequences [11, 12].

FSFinder identifies more frameshift signals than the approach of Hammell *et al.* because both pseudoknots and simple stem-loops are considered as downstream secondary structures and because conditions for slippery motifs and pseudoknots are relaxed. On the other hand, FSFinder finds less potential frameshift signals than the approach of Bekaert *et al.* because it searches for frameshift signals only in the overlapping regions of open reading frames, and prioritizes candidate frameshift signals.

A total of 26 frameshift signals in RECODE have simple stem-loops as downstream secondary structures, but 5 of these were excluded because PseudoBase assigns them different stimulatory structures. Seventeen of the remaining 21 frameshift signals were detected by FSFinder while 4 could not be found because their slippery sequences do not conform to the motif X XXY YYZ. It turns out that many frameshift signals have the slippery motif X XXY YYG. FSFinder identified 13 such sequences, and these can be classified into two types: A AAA AAG and G GGA AAG. The frameshift signals of RSV were also detected.

**Table 1.** Frameshift signals in RECODE with downstream stem-loops and X XXY YYG slippery sequences. * indicates a frameshift signal that was not identified by FSFinder because the slippery sequence does not conform to the motif X XXY YYZ.

| ID | frameshift signals with X XXY YYZ (Z≠G) and a downstream stem | ID | frameshift signals with X XXY YYG and a downstream stem | ID | frameshift signals with X XXY YYG and other downstream structures |
|---|---|---|---|---|---|
| **82** | HIV type 1 | **71** | Escherichia coli | **104** | Bacteriophage lambda |
| **83** | HIV type 2* | **238** | IS911 | **237** | IS2 |
| **84** | Human T-cell lympotrophic virus type 1 | **251** | IS150 | | |
| **85** | Human T-cell lympotrophic virus type 2 | **252** | IS1221A | | |
| **92** | RCNMV * | **360** | Salmonella typhi | | |
| **97** | Simian T-cell lymphosropic virus type 1 | **361** | Salmonella typhimurium | | |
| **106** | Drosophila buzzatii Ossvaldo retrotransposon | **362** | Vibrio cholerae | | |
| **257** | Carrot mottle mimic virus* | **363** | Neisseria meningtidis | | |
| **258** | Groundnut rosette virus | **364** | Neisseria gonorrhoeae | | |
| **260** | PEMV RNA 2* | **365** | Neisseria meningitides | | |
| | | **392** | Yersinia pestis | | |

Searching for frameshift signals in the overlapping region of ORFs is effective in predicting strong frameshift signal candidates. For example, a total of 157 potential frameshift signals were found in the sequences of the test cases in PseudoBase. Only 33 of these were in overlapping ORFs, and 19 of 33 proved to be the only genuine frameshift signals. FSFinder also identifies frameshift signals in alternative frames. For example, simian type D virus 1 has two slippery sequences G GGA AAC and A AAU UUU in different frames at positions 2058 and 2585, respectively. FSFinder detects two different signals in each of 6 organisms in RECODE: human T-cell lymphotropic virus type 2, mouse mammary tumor virus, simian type D virus 1, simian retrovirus type 2, simian T-cell lymphotropic virus type 1, and visna virus. There was only one alternative signal (in mouse mammary tumor virus) that could not be identified as it has a different motif (G GAU UUA). Table 2 summarizes the predicted frameshift signals in PseudoBase.

**Table 2.** Predicted frameshift signals in PseudoBase. * indicates a frameshift signal that was not detected by FSFinder because the slippery sequence does not conform to the motif X XXY YYZ

| PseudoBase numbers | Organisms | frameshift signals in the entire region | frameshift signals in the overlapping region |
|---|---|---|---|
| PKB1 | Bovine Leukemia Virus | 14 | 4 |
| PKB2 | Beet Western-Yellow Virus | 7 | 4 |
| PKB3 | Equine Infectious Anemic Virus | 12 | 2 |
| PKB4 | Feline Immunodeficiency Virus | 14 | 1 |
| PKB42 | Potato Leafroll Virus-W | 2 | 1 |
| PKB43 | Potato Leafroll Virus-S | 2 | 2 |
| PKB44 | CABYV | 4 | 1 |
| PKB45 | Pea Enation Mosaic Virus | 6 | 3 |
| PKB46 | Barley Yellow Dwarf Virus | 4 | 2 |
| PKB80 | Mouse Mammary Tumor Virus | 12 | 1 |
| PKB106 | Infectious Bronchitis Virus | 1 | 1 |
| PKB107 | Semian Retro Virus -1 | 9 | 2 |
| PKB127 | Equine Arteritis Virus* | - | - |
| PKB128 | Berne Virus | 11 | 1 |
| PKB171 | Human Corona Virus 229E | 12 | 1 |
| PKB174 | Rous Sarcoma Virus | 4 | 1 |
| PKB217 | LDV-C | 1 | 1 |
| PKB218 | PRRSV-16244B | 16 | 1 |
| PKB233 | PRRSV-LV | 17 | 1 |
| PKB240 | Beet Chlorosis Virus | 9 | 3 |
| Total number of true positives | | 19 | 19 |
| Total number of candidates | | 157 | 33 |

## 4 Conclusion

Identifying programmed -1 frameshifts is difficult because they are not uniform. However it is very important to achieve this identification in order to fully understand the underlying mechanisms and to discover new genes. Existing computational models predict too many false positives, or need reference protein sequences together with DNA sequence data from similar organisms.

We have developed an algorithm and a program called FSFinder for predicting plausible –1 frameshift signals in long DNA sequences. FSFinder was tested on 71 genomic sequences from different organisms and it predicted –1 frameshift signals more sensitively and selectivity than existing approaches. The procedure increases sensitivity by considering all potentially relevant components, and has increased selectivity because it focuses on the overlapping regions of open reading frames and prioritizes candidate signals. We believe FSFinder will be useful for analyzing –1 frameshift signals as well as for discovering novel genes.

**Acknowledgement.** This work was supported by the Korea Science and Engineering Foundation (KOSEF) under grant R01-2003-000-10461-0.

## References

1. Vimaladithan, A., Farabaugh, P.J.: Identification and analysis of frameshift sites. Methods in Molecular Biology 77 (1998) 399-411
2. Farabaugh, P.J.: Programmed translational frameshifting. Microbiological Reviews 60 (1996) 103-134
3. Farabaugh, P.J.: Programmed translational frameshifting. Annual Review of Genetics 30 (1996) 507-528
4. Jacks, T., Varmus, H.E.: Expression of the Rous sarcoma virus pol gene by ribosomal frameshifting. Science 230 (1985) 1237-1242
5. Bekaert, M., Bidou, L., Denise, A., Duchateau-Nguyen, G., Forest, J., Froidevaux, C., Hatin, I., Rousset, J., Termier, M.: Towards a computational model for -1 eukaryotic frameshifting sites. Bioinformatics 19 (2003) 327-335
6. Dinman, J.D., Icho, T., Wickner, R.B.: A -1 ribosomal frameshift in a double-stranded RNA virus of yeast forms a gag-pol fusion protein. Proc. Natl Acad. Sci. USA 88 (1991) 174-178
7. Hammell, A.B., Taylor, R.C., Peltz, S.W., Dinman, J.D.: Identification of putative programmed -1 ribosomal frameshift signals in large DNA databases. Genome Res. 9 (1999) 417-427
8. Marck, C.: DNA Strider: a C program for the fast analysis of DNA and protein sequence on the Apple Macintosh family of computers. Nucleic Acids Research 16 (1988) 1829-1836
9. van Batenburg, F.H.D., Gultyaev, A.P., Pleij, C.W.A., Ng, J., Oliehoek, J.: PseudoBase: a database with RNA pseudoknots. Nucleic Acids Research 28 (2000) 201-204
10. Baranov, P., Gurvich, O.L., Hammer, A.W., Gesteland, R.F., Atkins, J.F.: RECODE Nucleic Acids Research 31 (2003) 87-89
11. Birney, E., Thompson, J.D., Gibson, T.J.: PairWise and SearchWise: finding the optimal alignment in a simultaneous comparison of a protein profile against all DNA translation frames. Nucleic Acids Research 24 (1996) 2730-2739
12. Halperin, E., Faigler, S., Gill-More, R.: FramePlus: aligning DNA to protein sequences. Bioinformatics 15 (1999) 867-873
13. Fichant, G.A., Quentin, Y.: A frameshift error detection algorithm for DNA sequencing projects. Nucleic Acids Research 23 (1995) 2900-2908