New tools to analyze overlapping coding regions

A.H. Bayegan, J.A. Garcia-Martin[*], P. Clote[†]

Department of Biology, Boston College, Chestnut Hill, MA 02467
a.h.bayegan@gmail.com, j.antonio.garciamartin@gmail.com, clote@bc.edu

Abstract

Retroviruses transcribe messenger RNA for the overlapping Gag and Gag-Pol polyproteins, by using a programmed -1 ribosomal frameshift which requires a slippery sequence and an immediate downstream stem-loop secondary structure, together called frameshift stimulating signal (FSS). It follows that the molecular evolution of this genomic region of HIV-1 is highly constrained, since the retroviral genome must contain a slippery sequence (sequence constraint), code appropriate peptides in reading frames 0 and 1 (coding requirements), and form a thermodynamically stable stem-loop secondary structure (structure requirement).

In this paper, we describe a unique computational tool, `RNAsampleCDS`, designed to compute the number of RNA sequences that code two (or more) peptides p, q in overlapping reading frames, that are identical (or have BLOSUM/PAM similarity that exceeds a user-specified value) to the input peptides p, q. `RNAsampleCDS` then samples a user-specified number of messenger RNAs that code such peptides; alternatively, `RNAsampleCDS` can exactly compute the position-specific scoring matrix and codon usage bias for all such RNA sequences. Our software allows the user to stipulate overlapping coding requirements for all 6 possible reading frames simultaneously, even allowing IUPAC constraints on RNA sequences and fixing GC-content.

We generalize the notion of *codon preference index* (CPI) to overlapping reading frames, and use `RNAsampleCDS` to generate control sequences required in the computation of CPI. Moreover, by applying `RNAsampleCDS`, we are able to quantify the extent to which the overlapping coding requirement in HIV-1 [resp. HCV] contribute to the formation of the stem-loop [resp. double stem-loop] secondary structure known as the frameshift stimulating signal. Using our software, we confirm that certain experimentally determined deleterious HCV mutations occur in positions for which our software `RNAsampleCDS` and `RNAiFold` both indicate a single possible nucleotide. We generalize the notion of codon preference index (CPI) to overlapping coding regions, and use `RNAsampleCDS` to generate control sequences required in the computation of CPI for the Gag-Pol overlapping coding region of HIV-1. These applications show that `RNAsampleCDS` constitutes a unique tool in the software arsenal now available to evolutionary biologists.

Source code for the programs and additional data are available at `http://bioinformatics.bc.edu/clotelab/RNAsampleCDS/`.

1 Introduction

Programmed ribosomal frameshift (PRF) is a curious phenonenon, exploited especially by certain viruses, in order to translate two different protein products from the same messenger RNA. The frameshift is caused by particular sequence and structural elements of the mRNA which sometimes cause the ribosome to slip and readjust the reading frame, thus allowing viruses to pack more information into their genomes. Since the ratio of the protein products coded in overlapping reading frames depends on the PRF efficiency, which has been finely tuned by evolution, any chemical that can modify this efficiency could prove to be a useful anti-viral agent. Though partcularly important for the life cycle of certain viruses, such as HIV-1 and HCV, programmed ribosomal frameshift can be found in all kingdoms of life [4].

[*]New address: Systems Biology Program Centro Nacional de Biotecnología Consejo Superior de Investigaciones Cientficas (CSIC) C/ Darwin 3, Madrid, 28049, Spain.

[†]Corresponding author: `clote@bc.edu`

In HIV-1, Pol is obtained from a fused Gag-Pol polyprotein via a programmed -1 ribosomal frameshift, which naturally occurs with a frequency of 5-10%; moreover, an increase of ribosomal frameshift frequency is associated with a decrease in viral infectivity [15]. The -1 ribosomal frameshift is caused by two *cis*-acting RNA elements, together known as *frameshift stimulating signal* (FSS): (1) a heptameric *slippery sequence* (U UUU UUA), where the Gag reading frame is indicated, and (2) a downstream stem-loop secondary structure, often with either internal loop or right bulge. The FSS from HIV-1 genome (AF033819.3/1631-1682) is shown in Figure 1a, where the minimum free energy (MFE) secondary structure was determined by `RNAfold` from *Vienna RNA Package* 2.1.9 [12]. The Pol reading frame is -1 with respect to the Gag reading frame, or equivalently, the Gag reading frame is +1 with respect to the Pol reading frame (convention adopted throughout this paper) – Figure 1b depicts the six reading frames considered in this paper. While the entire Gag-Pol overlap region in HIV-1 AF033819.3 is from position 1631 to 1838 (Pr55 Gag polyprotein is coded at AF033819.3/336-1838), the 17-mer Pol [resp. Gag] peptide coded in the 52 nt FSS region 1631-1682 is FFREDLAFLQGKAREFS [resp. FLGKIWPSYKGRPGNFL]. Moreover, we found the secondary structure from Figure 1a to be the most common MFE structure for 52 nt segments of the Pol coding region, which begin by UUUUUUA, taken from the HIV Sequence Database in Los Alamos National Laboratory (LANL) available at `www.hiv.lanl.gov`. Due to its importance, a collection of 145 HIV-1 ribosomal frameshift elements is given in the family RF00480 in Rfam 12.0 [14]. Figure 1c displays the sequence logo obtained from the 145 sequences in the seed alignment of RF00480, while Figures 1d and 1e respectively display the sequence logos for the 17-mer Pol and Gag peptides coded in RF00480.

For decades, research in evolutionary biology has focused mostly on protein-coding regions, leading to the development of sophisticated computational tools, such as `PAML` [26] and `HYPHY` [19], to compute the ratio dN/dS of non-synonomous mutation rate dN to the synonomous mutation rate dS [8, 9, 27]. Pedersen and Jenson [16] extended the codon substitution model of Goldman and Yang [9] to overlapping genes in a site-specific manner, where evolutionary constraints of both genes are taken into account. However, estimation of evolutionary parameters in this model required computationally expensive Markov chain Monte Carlo simulations. By dropping the condition of site specificity, Sabath et al. [22] were able to apply a maximum likelihood method to estimate parameters in a more efficient manner. The resulting tool has been used to predict functionality of overlapping reading frames [21]. An evolutionary model has been developed for coding regions with conserved RNA secondary structures [17] as well. This approach was used to determine the effects of structural elements on nucleotide substitution in hepatitis C virus.

Several methods have been developed to sample sequences using an evolutionary model derived from a given phylogeny [20, 11, 6]. To the best of our knowledge, however, there is no previously published method for sampling sequences in overlapping coding regions. The program `SISSI` [6] incorporates a user-defined system of dependencies between the nucleotides; however, it is not possible using `SISSI` to sample sequences that code in overlapping reading frames, since `SISSI` requires that any position in an RNA sequence must belong to a single codon. Moreover, `SISSI` does not allow sequence and structural dependencies to be specified simultaneously. Our work in this paper is orthogonal to the foregoing computational models and tools of mathematical evolution theory and does not rely on phylogeny information. In full generality, the new software `RNAsampleCDS` supports the following. For each reading frame $r \in \{+0, +1, +2, -0, -1, -2\}$ illustrated in Figure 1b, let p_r be a length n sequence in the 22-letter alphabet consisting of IUPAC codes for each amino acid, together with symbol X (any residue) and O (any residue or STOP). `RNAsampleCDS` computes the number of RNA sequences $a_0, \ldots, a_{3n+2}$ which simultaneously code protein p'_r in reading frame r, such that either p'_r is identical to p_r, or (optionally) whose BLOSUM/PAM similarity to p_r exceeds a user-specified value. (Throughout the article, we say that the peptide p is *BLOSUM[PAM] θ similar* to another peptide p', if each amino acid of p has BLOSUM[PAM resp.] similarity of *at least* θ with the corresponding amino acid of p'.) `RNAsampleCDS` can then compute the PSSM and codon usage frequency for such proteins, as well as sample a user-specified number of such sequences. `RNAsampleCDS` runs in linear time and space, although if GC-content is optionally controlled, then time and space requirements are quadratic. For expository reasons, we describe the algorithms for only two proteins p, q respectively in reading frame 0 and 1; however, our code is general as just described – see the supplementary information for details on the general algorithm. Using `RNAsampleCDS`, we undertake a preliminary analysis of the Gag-Pol overlapping reading frame in human immunodeficiency virus (HIV-1) and of the triple overlapping reading frame of hepatitis C virus (HCV).

2 Methods

2.1 RNAsampleCDS

Let $p = p_1, \ldots, p_n$ and $q = q_1, \ldots, q_n$ be two peptides of equal length. In this section, we are interested in the following questions.

1. Which sequences $a_0, \ldots, a_{3n}$ of messenger RNA translate the peptide p in reading frame 0 and also translate the peptide q in reading frame +1?
2. Which sequences $a_0, \ldots, a_{3n}$ of messenger RNA translate peptides $p' = p'_1, \ldots, p'_n$ in reading frame 0 and peptide $q' = q'_1, \ldots, q'_n$ in reading frame +1, where the BLOSUM/PAM similarity of p with p' and q with q' is greater than or equal to a user-specified threshold θ?
3. What is the profile, or PSSM, for the collection of mRNAs from (1) and (2)?
4. What is the total number of sequences satisfying (1) and (2), and how can we sample sequences $a_0, \ldots, a_{3n}$ of messenger RNA in an unbiased manner, in order to satisfy either (1) or (2)?

By developing software to sample mRNA sequences that code user-specified proteins in different reading frames, we can then analyze the samples with other tools to provide an estimate of the probability of satisfying a given property of interest, hence give approximate answers for questions like the following: What is the expected stem size in the minimum free energy (MFE) structure of RNAs that translate peptides p', q' in reading frames 0,1, where the BLOSUM/PAM similarity of p, p' and of q, q' is at least a user-specified threshold value of θ? As we show, it is not difficult to see that questions (1,2) are easily answered using breadth first search (BFS); however, for large values of n, it can happen that BFS in not practical, since the number of messenger RNAs can be of size exponential in n. For that reason, we describe a novel dynamic programming (DP) algorithm to answer questions (3) and (4).

We first need a few definitions. If xyz is a trinucleotide, then let $tr(xyz)$ denote the amino acid whose codon is xyz in the genetic code; i.e. $tr(xyz)$ is the amino acid translated from codon xyz, unless xyz is a stop codon. If $xyzu$ is a tetranucleotide, then let $tr_0(xyzu)$ [resp. $tr_1(xyzu)$] denote the amino acid whose codon is xyz [resp. yzu]; i.e. $tr_0(xyzu) = tr(xyz)$ and $tr_1(xyzu) = tr(yzu)$. For each $k = 1, \ldots, n$, define the collection L_k of 4-tuples $s = s_0, s_1, s_2, s_3$ such that $tr_0(s) = tr(s_0, s_1, s_2) = p_k$ and $tr_1(s) = tr(s_1, s_2, s_3) = q_k$. Define two 4-tuples $s = s_0 s_1 s_2 s_3$ and $t = t_0 t_1 t_2 t_3$ to be *compatible* if $s_3 = t_0$ – i.e. the tail of s equals the head of t. Note that if 4-tuples s, t are compatible, then the *merge* $s_0, s_1, s_2, t_0, t_1, t_2, t_3$ of s, t has the property that amino acids are translated by each of the four codons $s_0 s_1 s_2$, $s_1 s_2 s_3$, $t_0 t_1 t_2$, and $t_1 t_2 t_3$.

ALGORITHM 1: (BFS computation of sequences that code in reading frames 0 and 1) Define the tree T by induction on depth as follows.

- **Base case:** The root of T is $\emptyset$; the children of the root are those 4-tuples s, such that $tr_0(s) = p_1$, $tr_1(s) = q_1$. The depth of the root is 0, and the depth of each child of the root is 1.
- **Inductive case:** If s is a 4-tuple in T of depth k, then the children of s are those 4-tuples t, such that $s_3 = t_0$ (compatibility requirement) and $tr_0(t) = p_{k+1}$, $tr_1(t) = q_{k+1}$ (coding requirement). The depth of each child of s is $k + 1$.

Suppose that $\sigma_1, \sigma_2, \ldots, \sigma_k$ is a *path* from root to level k; i.e. $\sigma_1, \sigma_2, \ldots, \sigma_k$ is a sequence of 4-tuples belonging to T, where for each $i = 1, \ldots, k$, the level of σ_i is equal to i, and for each $i = 1, \ldots, k-1$, σ_{i+1} is a child of σ_i. Define the *merge* of $\sigma_1, \sigma_2, \ldots, \sigma_k$ to be the RNA sequence $a_0, a_1, \ldots, a_{3k}$, where $\sigma_1 = a_0 a_1 a_2 a_3$, $\sigma_2 = a_3 a_4 a_5 a_6$, $\sigma_3 = a_6 a_7 a_8 a_9$, $\ldots$, $\sigma_k = a_{3(k-1)} a_{3k-2} a_{3k-1} a_{3k}$. By induction, it is easy to establish that in this case $tr_0(\sigma_i) = p_i$, $tr_1(\sigma_i) = q_i$ for each $i = 1, \ldots, k$. An easy application of breadth first search then allows one to generate the collection of level n nodes of T. It follows that the answer to question (1) is the set of RNAs obtained by merging the paths from root to level n nodes of T. ■

Using our implementation of the BFS approach in Algorithm 1, we can easily determine that there are exactly 32 52-nt RNAs that translate the 17-residue Pol peptide FFREDLAFLQGKAREFS in reading frame 0, and the 17-residue Gag peptide FLGKIWPSYKGRPGNFL in reading frame +1. These 17-mer

peptides are those which constitute the beginning of the Gag-Pol overlap in the HIV-1 genome (nucleotides 1631-1682 in GenBank AF033819.3). The entire Gag-Pol overlap region is from 1631-1835, whereby the 68-mer Pol [resp. Gag] peptide is coded in the region 1631-1834 [resp. 1632-1835 with a Gag STOP codon at 1836-1838]. Our implementation of the BFS method returns exactly 256 205-nt RNAs that code the Pol [resp. Gag] 68-mers from HIV-1 (GenBank AF033819.3). Figure 2 displays the centroid secondary structure, `RNAalifold` [1] consensus structure, and the corresponding mountain plot for the alignment of all 256 205-nt RNA sequences that code the Pol and Gag 68-mer peptides from HIV-1 (Pol 1631-1835, Gag 1632-1836 in GenBank AF033819.3), *not* necessarily containing the slippery sequence UUUUUUA. Further analysis (data not shown) indicates that there is considerable variation in the low energy structures of RNAs that exactly code the same 68-mer Pol and Gag peptides as those coded by AF033819.3/1631-1836. Question (2) is an obvious generalization of (1), and is easy to answer by generalizing the collection L_k of 4-tuples $s = s_0, s_1, s_2, s_3$ such that $tr_0(s) = tr(s_0, s_1, s_2) = p'_k$ and $tr_1(s) = tr(s_1, s_2, s_3) = q'_k$, where the BLOSUM/PAM similarity of p_k, p'_k and of q_k, q'_k is at least a user-specified threshold θ.

It is more interesting to turn to question (3), which requires a different strategy, since the number of RNAs returned by BFS may be exponentially large. Indeed, if RNA sequences are required to code peptides p [resp. q] whose amino acids have BLOSUM62 similarity of at least θ to those of the Pol [resp. Gag] 17-mer peptide coded in reading frame 0 [resp. 1] in AF033819.3/1631-1682, then the number of solution sequences is 256 ($\theta = 4$), 34,560 ($\theta = 3$), 90,596,966,400 ($\theta = 2$), 2.14285987145e+32 ($\theta = 1$), 3.61150917928e+56 ($\theta = 0$), 1.20555937201e+81 ($\theta = -1$), 1.17643153215e+106 ($\theta = -2$)! To address question (3), define the forward and backwards partition function ZF, ZB as follows.

- **Forward partition function:** For integer $k = 1, \ldots, n$ and nucleotide $ch \in \{A, C, G, U\}$, define $ZF(k, ch)$ to be the number of RNAs $\mathbf{a} = a_0, \ldots, a_{3k}$ such that a_{3k} is the nucleotide ch, and $\mathbf{a}$ translates the peptide $p_1, \ldots, p_k$ resp. $q_1, \ldots, q_k$ in reading frame 0 resp. 1; i.e. $tr_0(\mathbf{a}) = p_1, \ldots, p_k$ and $tr_1(\mathbf{a}) = q_1, \ldots, q_k$.

- **Backward partition function:** For integer $k = 1, \ldots, n$ and nucleotide $ch \in \{A, C, G, U\}$, define $ZB(k, ch)$ to be the number of RNAs $\mathbf{a} = a_{3k}, a_{3k+1}, \ldots, a_{3n}$ such that a_{3k} is the nucleotide ch, and $\mathbf{a}$ translates the peptide $p_k, \ldots, p_n$ resp. $q_k, \ldots, q_n$ in reading frame 0 resp. 1; i.e. $tr_0(\mathbf{a}) = p_k, \ldots, p_n$ and $tr_1(\mathbf{a}) = q_k, \ldots, q_n$.

By dynamic programming, it is straightforward to compute the forward and backward partition functions in linear time and space.

Recall that the *indicator function* I[boolean condition] returns the value 1 if the boolean condition within its scope is true, and otherwise the value returned is 0.

ALGORITHM 2: (DP partition function for sequences that code in reading frames 0 and 1) Given n-mer peptides p_0, q_0, for $k = 1, \ldots, n$ and $ch \in \{A, C, G, U\}$ define the *forward partition function* $ZF(k, ch)$ inductively as follows:

- Case 1: $k = 1$
  $ZF(k, ch) = \sum_{s_0 s_1 s_2 s_3 \in L_k} I[s_3 = ch]$

- Case 2: $k = 2, \ldots, n$
  $ZF(k, ch) = \sum_{s_0 s_1 s_2 s_3 \in L_k} I[s_3 = ch] \cdot ZF(k-1, s_0)$

For $k = n, \ldots, 1$ and $ch \in \{A, C, G, U\}$, define the *backward partition function* ZB inductively as follows:

- Case 1: $k = n$
  $ZB(k, ch) = \sum_{s_0 s_1 s_2 s_3 \in L_k} I[s_0 = ch]$

- Case 2: $k = n-1, \ldots, 1$
  $ZB(k, ch) = \sum_{s_0 s_1 s_2 s_3 \in L_k} I[s_0 = ch] \cdot ZB(k+1, s_3)$

Note the use of the boolean valued indicator function $I[\ldots]$, which has the value 1 if the expression within the brackets is true, and otherwise has the value 0. It follows that

$$Z = \sum_{ch \in \{A,C,G,U\}} ZF(n, ch) = \sum_{ch \in \{A,C,G,U\}} ZB(1, ch)$$

is the total number of RNA sequences that translate p in reading frame 0 and q in reading frame +1. ■

By appropriately redefining L_k, the recursions of Algorithm 2 can easily be modified to instead count the number of sequences coding $p'_1, \ldots, p'_n$ in reading frame 0 and $q'_1, \ldots, q'_n$ in reading frame +1, such that for each i, the BLOSUM/PAM similarity of p_i, p'_i and of q_i, q'_i exceeds a user-specified threshold θ, or for which the Kyte-Doolittle hydrobicity of p_i, p'_i and q_i, q'_i differ by at most a user-specified upper bound, etc. The same remark applies to *all* algorithms of this section, although for reasons of space, we do not explicitly mention such extensions. Nevertheless, such extensions are supported by the software `RNAsampleCDS`.
By refining the definition of forward and backward partition function, Algorithms 1 and 2 can be modified to keep track of the GC-content, albeit at an overhead for the space required. For an arbitrary RNA sequence $\mathbf{a}$, let $gccount(\mathbf{a})$ denote the number of Gs or Cs occurring in $\mathbf{a}$.

- **Forward partition function accounting for GC-content:** For integer $k = 1, \ldots, n$ and nucleotide $ch \in \{A, C, G, U\}$, define $ZF_{GC}(k, x, ch)$ to be the number of RNAs $\mathbf{a} = a_0, \ldots, a_{3k}$ such that a_{3k} is the nucleotide ch, $gccount(\mathbf{a}) = x$, and $\mathbf{a}$ translates the peptide $p_1, \ldots, p_k$ resp. $q_1, \ldots, q_k$ in reading frame 0 resp. 1; i.e. $tr_0(\mathbf{a}) = p_1, \ldots, p_k$ and $tr_1(\mathbf{a}) = q_1, \ldots, q_k$.
- **Backward partition function accounting for GC-content:** For integer $k = 1, \ldots, n$ and nucleotide $ch \in \{A, C, G, U\}$, define $ZB_{GC}(k, x, ch)$ to be the number of RNAs $\mathbf{a} = a_{3k}, a_{3k+1}, \ldots, a_{3n}$ such that a_{3k} is the nucleotide ch, $gccount(\mathbf{a}) = x$, and $\mathbf{a}$ translates the peptide $p_k, \ldots, p_n$ resp. $q_k, \ldots, q_n$ in reading frame 0 resp. 1; i.e. $tr_0(\mathbf{a}) = p_k, \ldots, p_n$ and $tr_1(\mathbf{a}) = q_k, \ldots, q_n$.

Though not explicitly described, *all* the following algorithms (PSSM computation and sampling) can be modified to account for GC-content. Our program, `RNAsampleCDS`, implements all the algorithms described in this section, including versions that account for GC-content. Moreover, our program supports any *two or more* overlapping coding regions in any of the 6 reading frames – i.e. reading frame 0,1,2 on the plus-strand and 0,1,2 on the minus-strand, as shown in Figure 1b.

Note that an easy modification of the above algorithm allows one to compute the total number of RNAs of length $3n + 1$, which code n-mer peptides p [resp. q] in reading frames 0 [resp. 1], i.e. for which neither reading frame contains a stop codon. This modification is later used to compute the probability that a random RNA of length $3n + 1$ will code in both reading frames 0 and 1. The following algorithm applies Algorithm 2 in order to compute the exact value of the position specific scoring matrix (PSSM).

ALGORITHM 3: (PSSM computation of sequences that code in reading frames 0 and 1) Given n-mer peptides p_0, q_0, for $i = 0, \ldots, 3n$ and $ch \in \{A, C, G, U\}$, define the profile or PSSM of nucleotides at positions $0, \ldots, 3n$ as follows:

- Case 1: $i = 0$. Then $PSSM(i, ch)$ equals $\sum_{s \in L_1} I[s_0 = ch] \cdot ZB(1, ch)/Z$
- Case 2: $i \equiv 0 \bmod 3$. Then $PSSM(i, ch)$ equals $ZF(i/3, ch) \cdot ZB(i/3, ch)/Z$
- Case 3: $i \equiv 1 \bmod 3$. Then $PSSM(i, ch)$ equals $\sum_{s \in L_{\lfloor i/3 \rfloor}} I[s_1 = ch] \cdot ZF(\lfloor i/3 \rfloor, s_0) \cdot ZB(\lceil i/3 \rceil, s_3)/Z$
- Case 4: $i \equiv 2 \bmod 3$. Then $PSSM(i, ch)$ equals $\sum_{s \in L_{\lfloor i/3 \rfloor}} I[s_2 = ch] \cdot ZF(\lfloor i/3 \rfloor, s_0) \cdot ZB(\lceil i/3 \rceil, s_3)/Z$

The recursions can be easily modified, if the RNA sequence is instead required to code $p'_1, \ldots, p'_n$ in reading frame 0 and $q'_1, \ldots, q'_n$ in reading frame +1, such that for each i, the BLOSUM/PAM similarity of p_i, p'_i and of q_i, q'_i exceeds a user-specified threshold θ. This answers question (3). The resulting DP program is very fast, since the run time is linear in n, while the BFS program has run time that is exponential in n.

Given a gapless alignment S of mRNA sequences of length $3n+1$, each of which codes a protein in reading frame 0 and 1, define the *positional codon frequency* $PCF(w,k,r)$ to be the number of occurrences of w in the kth codon position in reading frame $r \in \{0,1\}$ of a sequence in S. If S is the collection of all mRNAs that code proteins p, q respectively in reading frame $0, 1$, which are identical to (or alternatively have BLOSUM/PAM similarity that exceeds threshold θ), then the positional codon frequency can be defined from the partition functions ZF, ZB as follows.

ALGORITHM 4: (Positional codon frequency) Given n-mer peptides p_0, q_0, integer $k = 1, \ldots, n$, codon $w = w_0w_1w_2 \in (\{A,C,G,U\})^3$, and reading frame $r \in \{0,1\}$, the positional codon frequency $PCF(w,k,r)$ for the set of all mRNAs that code p_0, q_0 respectively in reading frame $0, 1$ can be computed as follows.

- CASE 1: $r = 0$. Then $PCF(w,k,0)$ equals $ZF(k-1, w_0) \cdot \sum_{ch \in \{A,C,G,U\}} ZB(k, ch)$.
- CASE 2: $r = 1$. Then $PCF(w,k,1)$ equals $\sum_{ch \in \{A,C,G,U\}} ZF(k-1, ch) \cdot ZB(k, w_2)$

Next, in order to sample RNA sequences that code peptides $p = p_1, \ldots, p_n$ resp. $q = q_1, \ldots, q_n$ in reading frames 0 resp. 1, we construct the sampled sequence from last to first character, each time ensuring that $ZF(k, ch) > 0$ where ch is the leading character of the current sample $a_{3k-1}, a_{3k}, \ldots, a_{3n}$. This is described as follows, where we recall that L_k denotes the collection of 4-tuples $s = s_0, s_1, s_2, s_3$ such that $tr_0(s) = tr(s_0, s_1, s_2) = p'_k$ and $tr_1(s) = tr(s_1, s_2, s_3) = q'_k$, and the BLOSUM/PAM similarity of p_k, p'_k and of q_k, q'_k is at least a user-specified threshold θ.

ALGORITHM 5: (Uniform sampling of RNAs that code in reading frames 0 and 1)

```
1.  k   = n  //initialize to the common length of peptides p,q
2.  rna = "" //initialize to empty sequence
3.  ch  = random nucleotide in { A,C,G,U } satisfying ZF(k, ch) > 0
4.  while k>0
5.      choose random 4-tuple s = s0, s1, s2, s3 such that s3 = ch
6.      rna = s1, s2, s3 + rna
7.      ch  = s0
8.      k   = k-1
9.  rna = ch + rna  //prepend the remaining initial nucleotide
```

It is straightforward to modify the previous algorithm to sample in a *weighted* fashion. First, recall that L_k denotes the collection of 4-tuples $s = s_0, s_1, s_2, s_3$ such that $tr_0(s) = tr(s_0, s_1, s_2) = p'_k$ and $tr_1(s) = tr(s_1, s_2, s_3) = q'_k$, and the BLOSUM/PAM similarity of p_k, p'_k and of q_k, q'_k is at least a user-specified threshold θ. Additionally, if $ch \in \{A,C,G,U\}$ then let $L_{k,ch}$ denote the set of tuples t in L_k, whose last element t_3 is ch.

ALGORITHM 6: (Weighted sampling of RNAs that code in reading frames 0 and 1)

```
1.  k   = n  //initialize to the common length of peptides p,q
2.  rna = "" //initialize to empty sequence
3.      a = ZF(k,A); c = ZF(k,C); g = ZF(k,G); u = ZF(k,U);
4.      z = a+c+g+u
5.      a = a/z; c = c/z; g = g/z; u = u/z
6.      select ch from A,C,G,U with prob a,c,g,u using roulette wheel
7.  while k>0
8.      sum = 0; r = random(0,1) · ZF(k-1,ch))
9.      for t in L_{k-1,ch}  //note that t = t0t1t2t3 and t3 = ch
10.         sum = sum + ZF(k − 1, t0)
11.         if r < sum
12.             rna = t + rna; ch = t0; k = k-1; break
13. return rna
```

Our implementation of the algorithms described in this section allows the user to stipulate *sequence constraints* using any IUPAC nucleotide codes, for instance, designating the first 7 nucleotides to be the slippery sequence `UUUUUUA`, or to consist of an alternation of purines and pyrimidines `RYRYRYR`, etc.

Finally, we note that all the previous algorithms in this section can be extended to handle *multiple* overlapping reading frames in all six reading frames, i.e. reading frames +0,+1,+2 on the plus strand and reading frames -0,-1,-2 on the minus strand, as illustrated in Figure 1b. For instance, in order to compute the forward partition function for reading frames 0,1,2 we define $ZF(k, ch1, ch2)$ to be the number of RNA sequences $\mathbf{a}$ of length $3k+2$ whose last two nucleotides are $ch1, ch2$, such that $tr_0(\mathbf{a}) = p_1, \ldots, p_k$, $tr_1(\mathbf{a}) = q_1, \ldots, q_k$, $tr_2(\mathbf{a}) = r_1, \ldots, r_k$, for user-specified peptides $\mathbf{p} = p_1, \ldots, p_n$, $\mathbf{q} = q_1, \ldots, q_n$, $\mathbf{r} = r_1, \ldots, r_n$. Now we define L_k to be the set of 5-tuples $s = s_0, \ldots, s_4$ such that $s_0 s_1 s_2$ codes residue p_k, $s_1 s_2 s_3$ codes residue q_k, and $s_2 s_3 s_4$ codes residue r_k. The definition of the generalization of the forward partition function $ZF(k, ch1, ch2)$, analogous to that defined in Algorithm 2, is as follows:

- Case 1: $k = 1$. Then $ZF(k, ch1, ch2)$ equals
$$\sum_{s_0 s_1 s_2 s_3 s_4 \in L_k} I[s_3 = ch1, s_4 = ch2]$$
- Case 2: $k = 2, \ldots, n2, \ldots, n$. Then $ZF(k, ch1, ch2)$ equals
$$\sum_{s_0 s_1 s_2 s_3 s_4 \in L_k} I[s_3 = ch1, s_4 = ch2] \cdot ZF(k-1, s_0, s_1)$$

Our publicly available code `RNAsampleCDS` supports all the above described variants of Algorithms 1-6 with possible IUPAC sequence constraints, stipulation of GC-content, and where the user may stipulate that particular peptides are coded in any or all of the six reading frames displayed in Figure 1b. See supplementary information for details of how we determine the run time estimate of $\approx 0.58831373 \cdot L + 0.00550239 \cdot N$ to generate compute the partition function and generate N samples of RNA sequences of length L that code any peptide in each of the six possible reading frames.

3 Results and Discussion

In this section, we use `RNAsampleCDS` to study novel aspects of human immunideficiency virus HIV-1 and hepatitis C virus HCV, that cannot be determined using methods other than those described in this paper.

3.1 HIV-1 programmed -1 frameshift

Analysis of HIV-1 overlap: Since HIV-1 and other retroviruses have a -1 ribosomal frameshift in the initial portion of the Gag-Pol overlap, this can be detected by the software `FRESCo` [23], which predicts regions of excess synonymous constraint in short, deep alignments. Figure 3a displays the dN/dS ratio we obtained for HIV-1 AF033819.3 with respect to the Gag reading frame, when aligned with other HIV-1 genomes from the Los Alamos HIV Database – see also Figure S1 from supplementary information. This figure indicates that there is *positive selection* in the Gag region before the Gag-Pol overlap. In contrast, starting with the beginning of the Gag-Pol overlap (nucleotide 1631), there is *purifying selection*; i.e. Figure 3a suggests the presence of an important signal starting around position 1631. Figure 3b displays the dN/dS ratio of the 52 nt Gag-Pol overlap region, for both the Gag and Pol reading frames, using the method of [22] which computes a rate matrix for overlapping reading frames – an aspect ignored by PAML and other software. Since Sabath's program computes dN/dS from a pairwise alignment, which is wholly inappropriate for the short 52 nt sequences considered here, we modified the approach by first producing multiple alignments of 52 nt Gag-Pol overlap regions, and then computed the number of (observed) synonomous and nonsynonomous mutations within the Gag [resp. Pol] reading frame, taking account for all codon pairs in the same column. We then modified Sabath's Matlab program to compute dN/dS by maximum likelihood using counts obtained from the multiple alignments. The multiple alignments considered in Figure 3b are from Rfam family RF00480 and from 52 nt RNA sequences generated by the programs `RNAsampleCDS` and `RNAiFold 2.0`. `RNAsampleCDS` generates 52 nt sequences, that translate peptides in the Gag [resp. Pol] reading frame, each of whose amino acids has BLOSUM62 similarity of either 0 or 1 to the corresponding amino acids in the Gag [resp. Pol] reading frame of the peptides translated by the 52 nt HIV-1 overlap region of AF033819.3/1631-1682. `RNAiFold 2.0` generates 52 nt sequences, that not only satisfy the same coding requirements as

RNAsampleCDS, but which also fold into the minimum free energy secondary structure shown in Figure 1a. In each case, RNAiFold 2.0 generates *all* sequences that satisfy both the coding and structure requirements, their number being substantially less than the 100,000 sequences generated by RNAsampleCDS. Note the presence of purifying selection for the Gag reading frame, as indicated by dN/dS values less than 1.

Codon preference index: In this section, we generalize the notion of *codon preference index* (CPI) [10] to the context of overlapping coding regions. For RNA sequence $\mathbf{a} = a_0, \ldots, a_{3n}$ which codes n-mer peptides in reading frames $0, 1$, for codon $w \in (\{A, C, G, U\})^3$ and reading frame $r \in \{0, 1\}$, define $f_{(w,\mathbf{a},r)}$ to be the number of occurrences of codon w in reading frame r of $\mathbf{a}$, and for amino acid AA, define $f_{(AA,\mathbf{a},r)}$ to be the number of occurrences of codons coding AA in reading frame r of $\mathbf{a}$. Define the *observed codon preference* in $\mathbf{a}$ by $p_{obs}(w, \mathbf{a}) = \sum_{r=0}^{1} f_{(w,\mathbf{a},r)} / \sum_{r=0}^{1} f_{(AA,\mathbf{a},r)}$. If S is a set of mRNAs of length $3n + 1$, each of which codes n-mer peptides in both reading frames $0, 1$, then define the *observed codon preference* in S by $p_{obs}(w, S) = \sum_{r=0}^{1} \sum_{\mathbf{a}\in S} f_{(w,\mathbf{a},r)} / \sum_{r=0}^{1} \sum_{\mathbf{a}\in S} f_{(AA,\mathbf{a},r)}$. Note that $p_{obs}(w, S)$ is the *probability* that codon w will be used for amino acid AA in the collection S of overlapping coding sequences. Finally, define the *codon preference index* $I(w)$ of codon w in S by $I(w) = p_{obs}(w, S)/p_{obs}(w, S')$, where S' is a *control* set of mRNAs of length $3n + 1$.

With these notations, Figure 4 depicts a heat map for the codon preference index $I(w)$, computed over 5,125 entire Gag-Pol overlap regions of average length 205 ± 10 (Gag and Pol peptide size ≈ 68) extracted from LANL HIV-1 database, each starting with the slippery sequence UUUUUUA and terminating with the last Gag codon; additionally the heat map includes Gag-only and Pol-only values for the same overlap region. For this figure, the control set S' is defined differently for each column $1 - 5$, although in all cases, each sequence in S' contains the initial slippery sequence UUUUUUA. For column 1 [resp. 2] S' is the set of all mRNAs that code proteins in the Gag [resp. Pol] reading frame that are coded by some sequence of S. For column 3, S' is the set of all mRNAs that code proteins p and q that are identical to proteins coded in the Gag and Pol reading frames of some sequence $\mathbf{a}$ of S. For column 4, S' is defined as in the case for column 3, except that 'identical to' is replaced by 'BLOSUM62 +1 similar to'. For column 5, S' is the set of all mRNAs that code proteins p and q that are BLOSUM62 +1 similar to proteins coded in the Gag and Pol reading frames of a sequence $\mathbf{a}$ of S, and whose GC-content lies in the range of GC-content of $\mathbf{a} \pm 5$. The heat map of Figure 4 shows that for serine, $I(AGU, Gag) < I(AGU, Pol) < I(AGU, Gag/Pol) \approx 1$; for valine, $I(GUG, Gag) < 1 < I(GUU, Gag)$ but $I(GUG, Gag/Pol) > 1 > I(GUU, Gag/Pol)$; for proline, $I(CAU, Gag) < I(CAU, Pol) < I(CAU, Gag/Pol) \approx 1$, but when the control set is taken to be BLOSUM62 +1 similar peptides to Gag and Pol, then $I(CAU, Gag/Pol + 1) \gg 1$. See Figures S2 and S3 and the text from supplementary information for more detailed explanation. These figures show that the codon usage bias observed at the Gag-Pol junction is not due to natural selection [18] or to the underlying mutational bias, but rather imposed by the overlapping coding constraints.

Overlapping coding and stem-loop formation: Here we describe how to quantify the extent to which coding HIV-1 17-mer peptides in overlapping reading frames induces a stem-loop structure. In particular, we consider the following questions.

1. What is the probability that random RNA forms a stem-loop structure?
2. What is the probability that RNA forms a stem-loop structure, if it is required to code (any arbitrary) peptides in reading frames 0 and 1?
3. What is the probability that RNA forms a stem-loop structure, if it is required to code peptides in reading frames 0 and 1, which are *similar* to peptides coded in the HIV-1 frameshift stimulating signal (FSS)?
4. To what extent do HIV-1 coding requirements in the Pol-Gag overlap region alone induce stem-loop formation?
5. What is the (conditional) probability of coding peptides in reading frames 0 and 1 if the RNA forms a secondary structure similar to the FSS stem-loop structure of HIV-1?

To answer question 1, we generated 200,000 52-nt RNAs, where the first seven nucleotides constituted the slippery sequence UUUUUUA, and each nucleotide in position 8 through 52 was randomly selected with probability 0.25 for each of A,C,G,U. Using RNAshapes, cf. [24], we determined the Boltzmann probability

that each RNA sequence has shape `[ ]` [7], i.e. $P(\texttt{[ ]}) = \sum_s \exp(-E(s)/RT)$, where the sum is taken over all *stem-loop* secondary structures, which may contain internal loops and bulges, but no multiloops or multiple stem-loops. Throughout the sequel of the paper, the probability that a given RNA sequence will form a *stem-loop* structure is identified with $P(\texttt{[ ]})$. A finer analysis could consider type 1 shapes of the form `_[_[ ]_]` or `_[ [ ]_]`, corresponding to a stem loop with internal loop or right bulge, with left flanking unpaired region, but in this paper we consider only the type 5 stem loop shape `[ ]`. By *MFE stem-loop structure*, we mean the stem-loop secondary structure which has the minimum free energy, taken over all stem-loop structures. Similarly, *stem-loop MFE* means the minimum free energy of all stem-loop structures. Note that the stem-loop MFE is not necessarily equal to the MFE, since it is possible that a structure having two or more external loops, or containing a multiloop, could have lower energy than that of any stem-loop structure. By uniformly sampling 200,000 52 nt RNAs with no coding requirements, we estimate an average probability of stem-loop formation of 60.7% with standard deviation of 36.2%, and average stem-loop MFE was −7.65 kcal/mol with standard deviation 3.42 kcal/mol – again, this is for 52 nt RNA with no constraints.

Before answering question 2, we first note that the conditional probability is 45.32% that a 52-nt RNA codes in both reading frames 0,1 assuming that it begins by the slippery heptamer UUUUUUA is 23.14%, and that the conditional probability that a 52-nt RNA codes in reading frame 1, given that it begins by the slippery heptamer UUUUUUA *and* that it already codes in reading frame 0 45.32% – i.e. $P(A|B,C) = 0.4532$, where event A is that a 52-nt RNA codes in reading frame 0, event B is that the 52-nt RNA contains slippery heptamer UUUUUUA, and event C is that reading frame 0 of the 52-nt RNA contains no stop codon. In contrast, the conditional probability that a 52-nt RNA codes in reading frame 0 assuming that it begins by the slippery heptamer UUUUUUA is 51.06%.

Indeed, using `RNAsampleCDS`, we determine that the number x_1 of 52-nt RNAs beginning by UUUUUUA and which code in both reading frames 0,1 is $2.86451 \cdot 10^{26}$. In contrast, the number x_2 of 52-nt RNAs beginning by UUUUUUA and which code in reading frame 0 is $x_2 = 16 \cdot 61^{14} \cdot 4 = 6.32117 \cdot 10^{26}$, since there are 16 codons that begin by A, a choice of 61 coding codons for the remaining 14 residues (since the first two residues must be FF and the third residue have a codon beginning by A), times 4 for the last nucleotide to ensure the RNA length is 52. The number x_3 of all 52-nt RNAs that begin by UUUUUUA is clearly $4^{45} = 1.23794 \cdot 10^{27}$. These computations justify the previous probabilities, and suggest the potential utility of `RNAsampleCDS` when speculating about molecular evolution.

To answer question 2, we used `RNAsampleCDS` to generate 200,000 52-nt RNA sequences, each of which contains the slippery sequence UUUUUUA and codes 17-mer peptides in both reading frames 0 and 1. Executing `RNAshapes` as previously described yielded an average probability of stem-loop formation of 59.8% with standard deviation of 36.7%, and average stem-loop MFE of −8.06 kcal/mol with standard deviation 3.58 kcal/mol.

To answer question 3, we extracted 145 52-nt Pol-Gag overlapping FSS sequences in family RF00480 from the Rfam 12.0, of which 133 sequences remained after disambiguation and removal of sequences containing gaps or stop codons. For each of the 133 sequences, we generated 100,000 sequences using `RNAsampleCDS`, each of which begins by the same initial 7 nucleotides of the Rfam sequence constituting a slippery sequence (since most but not all RF00480 sequences begin with UUUUUUA), and which code peptides p [resp. q] having BLOSUM62 similarity of at least +1 with the corresponding amino acids of the 17-mer peptide coded by the Rfam sequence in frame 0 [resp. 1].

After removing two outliers (discussed shortly), we have the following statistics for the remaining 131 sequences from RF00480. Average probability of stem-loop formation for RF00480 is $99.3 \pm 2.2\%$, and average stem-loop MFE is -24.43 ± 3.91 kcal/mol. For the collection of 100,000 sequences generated by `RNAsampleCDS` for each sequence from Rfam family RF00480, coding BLOSUM62 +1 similar peptides to those coded by the Rfam sequence, the average stem-loop formation probability is is $69 \pm 12\%$, and average stem-loop MFE is -13.43 ± 2.32 kcal/mol. Figures 5a and 5b depict respectively the stem-loop formation probabilities and stem-loop minimum free energies. In contrast, a similar computational experiment using `RNAsampleCDS` shows that the average probability of stem-loop formation is $98.1\% \pm 8.1$ if each sampled sequence is required to code *exactly* the same peptides as those from HIV-1 in RF00480. This answers question 4.

The previous analysis was performed for 131 Rfam sequences, obtained after removal of the sequences AF442567.1/1455-1506 and L11798.1/1290-1341, from the set of 133 Rfam sequences obtained from 145

sequences in RF00480, after disambiguation and removal of sequences containing gaps or stop codons. These two sequence were removed as outliers, since their stem-loop formation probabilities were respectively 53.1% and 55.5% – far removed from the average of $99.3 \pm 2.2\%$ of the remaining sequences. GenBank annotations indicate that AF442567.1 is highly G to A hypermutated with very many, mostly in-frame, stop codons throughout the genome, and that the Gag gene of L11798.1 has a premature termination at position residue 46.

Together, these results show that stem-loop formation is a consequence of the *precise* HIV-1 Gag and Pol 17-mer peptides, but not of BLOSUM62 +1 similar peptides. As well, stem-loop formation probability is not statistically different (T-test) between random sequences, sequences that have no stop codon in reading frame 0 or 1, and sequences that code peptides having BLOSUM62 similarity of at least +1 to HIV-1 peptides. To determine particular nucleotide positions in the 52-nt FSS that appear to be critical in stem-loop formation, we computed the position-dependent nucleotide frequency (PSSM), denoted by π_1, for 200,000 sequences generated by `RNAsampleCDS` that begin by the slippery sequence UUUUUUA, and code peptides p [resp. q], each of whose amino acids has BLOSUM62 similarity greater than or equal to 1 with the corresponding amino acids of the Pol [resp. Gag] 17-mer peptides FFREDLAFPQGKAREFS [resp. FLGKIWPSHKGRPGNFL] coded in AF033819.3/1631-1682. Using `RNAiFold 2.0`, we also computed the PSSM, denoted by π_2, for all possible sequences that begin by slippery heptamer UUUUUUA, and fold into the MFE structure of AF033819.3/1629-1682 shown in Figure 1a, and which code peptides that are BLOSUM62 +1 similar to the peptides coded by AF033819.3/1631-1682. We then computed the position-dependent total variation distance between π_1 and π_2, defined by $\delta(\pi_{1,i}, \pi_{2,i}) = 1/2 \cdot \sum_{x \in \{A,C,G,U\}} |\pi_{1,i}(x) - \pi_{2,i}(x)|$, where $\pi_{1,i}$ resp. $\pi_{2,i}$ denotes the mononucleotide frequency at position i of the PSSM for sequences generated by `RNAsampleCDS` resp. `RNAiFold 2.0`. With the exception of specific regions, the total variation distance is close to zero, thus pinpointing critical nucleotides necessary for stem-loop formation of the FSS. Figures 6a, 6b display the sequence logo for the PSSM π_1 and π_2, and Figures 6c and 6d respectively depict the position-dependent entropy and total variation distance.

To answer question 5, we used `RNAiFold 2.0` with target structure as depicted in Figure 1a, in order to generate 200,000 52-nt RNA sequences, each containing the slippery sequence UUUUUUA and each folding into the target structure. We determined that 61.91% of these sequences have no stop codon in reading frames 0 or 1. The percentage of sequences that have no stop codon in reading frame 0 [resp. 1] alone is somewhat higher, with value 78.7% [resp. 79.59%]. We additionally determined that the average base pair distance between the MFE structure of the sampled sequences and the target FSS secondary structure is 2.04 and average ensemble defect is 3.58.

The probability of stem-loop formation for frameshift stimulating signal (FSS) regions of HIV-1 is close to 1, with average value of $99\% \pm 2$ for RF00480 as shown in Figure 5a. This value is much larger than that of random 52-nt RNAs ($\approx 61\%$), or 52-nt RNA having no stop codons in reading frames 0 or 1 ($\approx 60\%$), or even 52-nt RNA coding peptides in reading frames 0,1 with BLOSUM62 similarity of at least +1 to HIV-1 peptides ($\approx 69\%$). It follows that coding BLOSUM62 +1 similar peptides to those of HIV-1 at most slightly induces stem-loop formation. Yet the probability that stem-loop structures do not have a stop codon in either reading frame 0 or 1 is only about 62%, without requiring that the peptides be similar to those of HIV-1. It follows that BLOSUM62 +1 similarity to HIV-1 peptides cannot induce the required stem-loop FSS structure, nor can the target FSS structure from Figure 1a induce BLOSUM62 +1 similarity to HIV-1 peptides. We speculate that starting from a genomic region that codes a polyprotein similar to that of Gag, a series of pointwise mutations could slowly induce a stem-loop FSS structure and at the same time slowly create a Pol-like reading frame. Although speculative, it is possible to create an adaptive walk or Monte Carlo program to test the likelihood of this hypothesis, using intermediate sequences generated by `RNAsampleCDS` and `RNAiFold2.0`.

As shown in Figure 6a, the average pairwise Hamming distance of sequences generated by `RNAsampleCDS` with the overlapping coding constraint and the slippery sequence constraint is 10.92 ± 4.32 (length-normalized value of 0.21 ± 0.083), when computed with a random sample of 1000, 5000, and 10,000. As shown in Figure 6b, the average pairwise Hamming distance of sequences generated by `RNAiFold` with the frameshift stimulating structure (FSS) constraint, overlapping coding constraint and the slippery sequence constraint is 5.80 ± 1.84 (length-normalized value of 0.11 ± 0.035). Essentially, this means that approximately 11% of the positions (pairwise) are different for `RNAiFold` sampled sequences, compared with approximately 21% of the positions (pairwise) for `RNAsampleCDS`, compared with 81% of the positions (pairwise) for random RNA

in positions 8-52 (i.e. after the fixed 7 nt slippery sequence). The greatest reduction in pairwise Hamming distance appears to be due to overlapping coding constraints, with an additional small reduction due to the FSS structural constraint.

3.2 HCV programmed -1 and +1 frameshifts

There is both *in vitro* and *in vivo* experimental evidence for a -2/+1 (hereafter designated as +1) and -1/+2 (hereafter designated as +2) programmed ribosomal frameshift in the core protein of the hepatitis C virus (HCV) [2]. The +1 frameshift produces a 17 kDa protein called protein F (Frameshift), also designated as ARFP (Alternative Reading Frame Protein). In addition, the +2 frameshift produces a 1.5 kDa protein. As measured by *in vitro* assays, the +1 ribosomal frameshift efficiency is $\sim 12-15\%$, while the +2 ribosoma frameshift efficiency is $\sim 30-45\%$ [2]. Figure 7 depicts the organization of the overlapping coding region for the HCV genome (GenBank M62321.1), including a double stem-loop RNA structure designated as *frameshift stimulating signal* (FSS) depicted in Figure 8. According to [2], the frameshift is caused by a poly-A slippery sequence (A AAA AAA AAC) in the triple coding region, although a mutated slippery sequence (A AGA AAA ACC) has also been shown to cause a frameshift, but with a lower efficiency. Out of 6,589 sequence hits for the HCV1 frameshift signal for the LANL HCV database (`www.hcv.lanl.gov`), we found that 94% of the sequences started with (A AGA AAA ACC). Furthermore, downstream of the slippery sequence a double stem-loop structure facilitates translational frameshifting (Figure 8). For this analysis, we took nucleotides 344-500 from the 9401 nt HCV subtype 1a genome (GenBank M62321.1) [2], corresponding to the region starting at the triple coding region and extending to the end of double-stem loop. Using `RNAsampleCDS` we computed the logo plot for all sequences that code BLOSUM62 +1 similar peptides to those coded by the reference genome (Figure 9a). Using `RNAiFold 2.0` [5], we generated more than 11 million sequences that fold into the double-stem loop structure indicated in Figure 8 and which have BLOSUM62 similarity of at least +1 to the reference genome peptides (Figure 9b). Although `RNAiFold 2.0` does not support pseudoknot structures, by providing structural compatibility constraints, we ensured that every sequence returned by `RNAiFold 2.0` has the property that the nucleotides, which participate in the "kissing hairpin" model of Figure 1A of [2], can indeed form a base pair together. Note that the set of all sequences returned by `RNAiFold 2.0`, which satisfy both the coding and structural requirements, forms a proper subset of the set of all sequences returned by `RNAsampleCDS`, which are required to satisfy only the coding requirements. Figure 9c depicts the total variation distance between these sequence two profiles. At positions where the total variation distance is zero, the secondary structure is likely to be *induced* by the overlapping coding constraints. Indeed, a mutation in such positions could lead to a disruption of the double stem-loop or to a modification of the amino acid in one of the overlapping reading frames. Our results from Figure 9c agree with experimental evidence showing that modifications of nucleotides at positions 64, 91, 130 and 137 lead to *detrimental mutations* for the hepatitis C virus [13]. Mutations at these positions resulted in an attenuated HCV infection in chimpanzee. According to our analysis, an introduction of mutations at positions whose variation distance is much greater than zero, should allow the disruption of the double-stem loop with minimal effects on the protein function. This hypothesis could be tested experimentally.

To further investigate whether the overlapping coding requirement of HCV possibly induces the FSS double stem-loop structure, we proceeded in a manner analogous to that for our HIV-1 analysis. We sampled 100,000 RNA sequences using `RNAsampleCDS` with BLOSUM62 similarity of +1 and 0 to the reference peptides in each reading frame. Using `RNAshapes`, we computed the average Boltzmann probability of formation of a double-stem loop with shape `[ ] [ ]` , in the sampled RNA sequences as well as 6,589 sequences from LANL database (Figure S5 from supplementary information). Average Boltzmann probability of the double stem-loop shape `[ ] [ ]` is 19% [resp. 9%] for BLOSUM62 similarity of +1 [resp. 0], compared with 98% probability for the sequences from LANL HCV database. In contrast, dinucleotide shuffles of sequences generated by `RNAsampleCDS` having BLOSUM62 +1 similarity to the reference peptides have average probability of 5% of double stem-loop formation, while the probability double stem-loop formation is 6% for random RNA sequences generated with probability of $\frac{1}{4}$ for each nucleotide. Figure S5 displays average double stem-loop probability and free energy results for the HCV overlapping coding region, which are analogous to results for HIV-1 presented in Figure 5.

4 Conclusion

In this paper, we have developed the novel program `RNAsampleCDS`, the only existent program which computes the number of RNA sequences that code user-specified peptides in one to six overlapping reading frames, as depicted in Figure 1b. More importantly, `RNAsampleCDS` can compute (exact) PSSMs and sample, in an unweighted or weighted fashion, a user-specified number of RNA sequences that code the specified proteins (or code proteins having BLOSUM/PAM similarity that exceeds a user-specified threshold to the given proteins). With extensions to `RNAiFold2.0` made in this paper, `RNAsampleCDS` and `RNAiFold2.0` complement each other and together allow one to analyze the HIV-1 Gag-Pol overlapping reading frame and the HCV triple overlapping reading frame in a manner that cannot be supported by any other software, thus augmenting the software arsenal available to evolutionary biologists.

Acknowledgements

Research of the Clote Lab was supported by National Science Foundation grant DBI-1262439. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] S. H. Bernhart, I. L. Hofacker, S. Will, A. R. Gruber, and P. F. Stadler. RNAalifold: improved consensus structure prediction for RNA alignments. *BMC. Bioinformatics*, 9:474, 2008.

[2] J Choi, Z Xu, and J H Ou. Triple decoding of hepatitis C virus RNA by programmed translational frameshifting. *Mol Cell Biol*, 23(5):1489–1497, 2003.

[3] G. E. Crooks, G. Hon, J. M. Chandonia, and S. E. Brenner. Weblogo: a sequence logo generator. *Genome Res.*, 14(6):1188–1190, June 2004.

[4] J. D. Dinman. Programmed Ribosomal Frameshifting Goes Beyond Viruses: Organisms from all three kingdoms use frameshifting to regulate gene expression, perhaps signaling a paradigm shift. *Microbe. Wash. DC.*, 1(11):521–527, November 2006.

[5] J. A. Garcia-Martin, I. Dotu, and P. Clote. RNAiFold 2.0: a web server and software to design custom and Rfam-based RNA molecules. *Nucleic. Acids. Res.*, 43(W1):W513–W521, July 2015.

[6] Tanja Gesell and Arndt von Haeseler. In silico sequence evolution with site-specific interactions along phylogenetic trees. *Bioinformatics*, 22(6):716–722, 2006.

[7] R. Giegerich, B. Voss, and M. Rehmsmeier. Abstract shapes of RNA. *Nucleic Acids Res.*, 32(16):4843–4851, 2004.

[8] T. Gojobori, K. Ishii, and M. Nei. Estimation of average number of nucleotide substitutions when the rate of substitution varies with nucleotide. *J. Mol. Evol.*, 18(6):414–423, 1982.

[9] N. Goldman and Z. Yang. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Mol. Biol. Evol.*, 11(5):725–736, September 1994.

[10] M. Gribskov, J. Devereux, and R. R. Burgess. The codon preference plot: graphic analysis of protein coding sequences and prediction of gene expression. *Nucleic. Acids. Res.*, 12(1):539–549, January 1984.

[11] Cendrine Hudelot, Vivek Gowri-Shankar, Howsun Jow, Magnus Rattray, and Paul G Higgs. RNA-based phylogenetic methods: application to mammalian mitochondrial RNA sequences. *Mol Phylogenet Evol*, 28(2):241–252, 2003.

[12] R. Lorenz, S. H. Bernhart, C. Höner zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker. Viennarna Package 2.0. *Algorithms. Mol. Biol.*, 6:26, 2011.

[13] Laura K McMullan, Arash Grakoui, Matthew J Evans, Kathleen Mihalik, Montserrat Puig, Andrea D Branch, Stephen M Feinstone, and Charles M Rice. Evidence for a functional RNA element in the hepatitis C virus core gene. *Proceedings of the National Academy of Sciences of the United States of America*, 104(8):2879–2884, 2007.

[14] E. P. Nawrocki, S. W. Burge, A. Bateman, J. Daub, R. Y. Eberhardt, S. R. Eddy, E. W. Floden, P. P. Gardner, T. A. Jones, J. Tate, and R. D. Finn. Rfam 12.0: updates to the RNA families database. *Nucleic Acids Res.*, 0(O):O, November 2014.

[15] L. O. Ofori, T. A. Hilimire, R. P. Bennett, N. W. Brown, Jr, H. C. Smith, and B. L. Miller. High-affinity recognition of HIV-1 frameshift-stimulating RNA alters frameshifting in vitro and interferes with HIV-1 infectivity. *J. Med. Chem.*, 57(3):723–732, February 2014.

[16] a M Pedersen and J L Jensen. A dependent-rates model and an MCMC-based methodology for the maximum-likelihood analysis of sequences with overlapping reading frames. *Molecular biology and evolution*, 18(5):763–76, 2001.

[17] Jakob Skou Pedersen, Roald Forsberg, Irmtraud Margret Meyer, and Jotun Hein. An evolutionary model for protein-coding regions with conserved RNA structure. *Molecular Biology and Evolution*, 21(10):1913–1922, 2004.

[18] J. B. Plotkin and G. Kudla. Synonymous but not the same: the causes and consequences of codon bias. *Nat. Rev. Genet.*, 12(1):32–42, January 2011.

[19] S. L. Pond, S. D. Frost, and S. V. Muse. Hyphy: hypothesis testing using phylogenies. *Bioinformatics*, 21(5):676–679, March 2005.

[20] A Rambaut and N C Grassly. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees, 1997.

[21] N. Sabath and D. Graur. Detection of functional overlapping genes: simulation and case studies. *J. Mol. Evol.*, 71(4):308–316, October 2010.

[22] N. Sabath, G. Landan, and D. Graur. A method for the simultaneous estimation of selection intensities in overlapping genes. *PLoS. One.*, 3(12):e3996, 2008.

[23] R. S. Sealfon, M. F. Lin, I. Jungreis, M. Y. Wolf, M. Kellis, and P. C. Sabeti. FRESCo: finding regions of excess synonymous constraint in diverse viruses. *Genome Biol.*, 16:38, 2015.

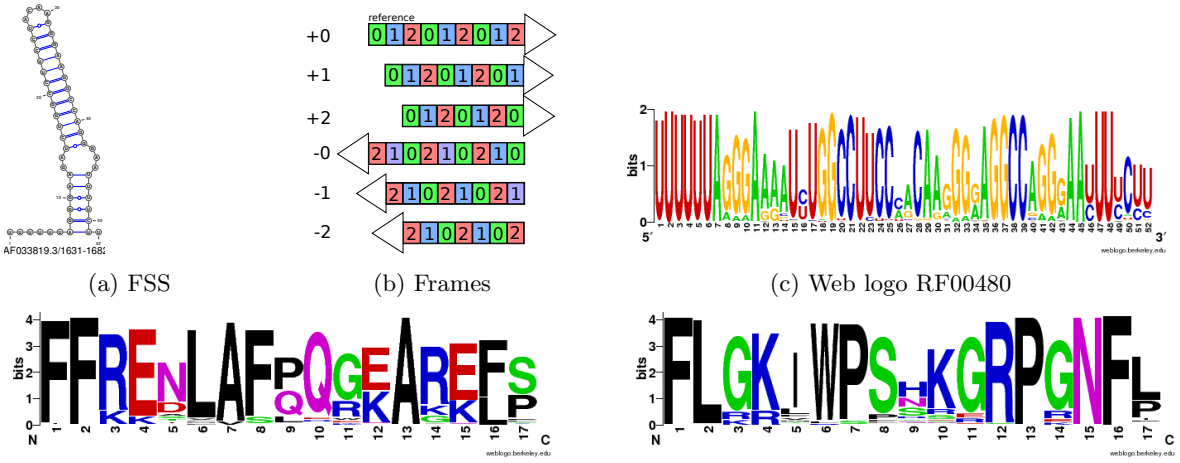
[24] P. Steffen, B. Voss, M. Rehmsmeier, J. Reeder, and R. Giegerich. RNAshapes: an integrated RNA analysis package based on abstract shapes. *Bioinformatics*, 22(4):500–503, 2006.

[25] K. C. Wiese, E. Glen, and A. Vasudevan. JViz.Rna–a Java tool for RNA secondary structure visualization. *IEEE. Trans. Nanobioscience.*, 4(3):212–218, September 2005.

[26] Z. Yang. PAML: a program package for phylogenetic analysis by maximum likelihood. *Comput. Appl. Biosci.*, 13(5):555–556, October 1997.

[27] Z. Yang, R. Nielsen, N. Goldman, and A. M. Pedersen. Codon-substitution models for heterogeneous selection pressure at amino acid sites. *Genetics.*, 155(1):431–449, May 2000.

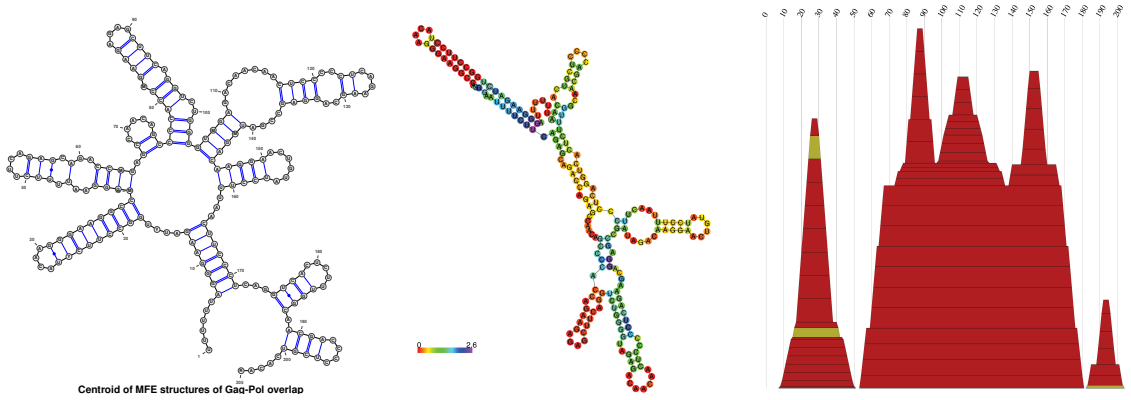
Figures



(a) FSS (b) Frames (c) Web logo RF00480

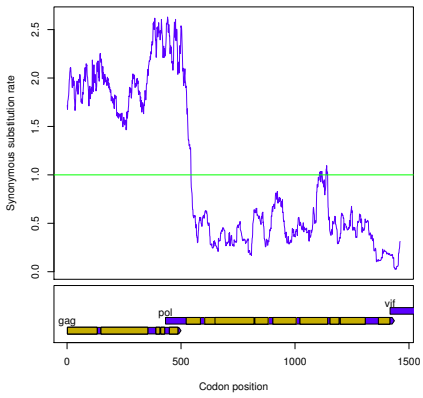
(d) Pol 17-mer peptides (e) Gag 17-mer peptides

Figure 1: *(a)* Minimum free energy (MFE) structure of the initial 52-nt Gag-Pol overlapping reading frame in positions 1631-1682 of the HIV-1 complete genome (GenBank AF033819.3). This frameshift stimulating signal (FSS) contains the initial slippery sequence heptamer, given by `U UUU UUA` in the Gag reading frame, as well as the displayed stem-loop secondary structure, which together promote a programmed -1 frameshift `UUU UUU A` in the Pol reading frame. *(b)* Depiction of all 6 possible reading frames – `RNAsampleCDS` samples RNA sequences that code in all possible reading frames, allowing IUPAC sequence constraints *(c)* Sequence logo for 145 RNA HIV-1 frameshift signal sequences from the RF00480 seed alignment from Rfam 12.0 [14]. *(d)* Sequence logo for the Pol peptide coded by 138 RNA HIV-1 frameshift signal sequences from the RF00480 seed alignment from Rfam 12.0; Pol peptide translated from nucleotide positions 1-51. *(e)* Sequence logo for the Gag peptide coded by 138 RNA HIV-1 frameshift signal sequences from the RF00480 seed alignment from Rfam 12.0; Gag peptide translated from nucleotide positions 2-52. Since some sequences from RF00480 contained IUPAC codes for uncertain data, the data were disambiguated–for instance, the code B (not A) was disambiguated by randomly assigning either C,G or U with probability 1/3. Seven sequences were removed from the seed alignment of 145 RNAs due to gaps in the alignment, and another five sequences were removed since either the Pol or Gag peptide contained a stop codon–resulting in 133 sequences for nucleotide analysis. Peptide sequence logos for the 138 Pol and Gag peptides were created using `WebLogo` [3].



(a) Centroid (b) Consensus (c) Mountain plot

Figure 2: (A) The centroid secondary structure,(B) `RNAalifold` consensus structure, and (C) the corresponding mountain plot for the alignment of all 256 205-nt RNA sequences that code the Pol and Gag 68-mer peptides from HIV-1 (Pol 1631-1835, Gag 1632-1836 in GenBank AF033819.3).



(a) FRESCo

	dN/dS Pol	*dN/dS Gag*	*branch len t*	*transition/transversion*	*num seq*
Lanl-B0-CDS	1.03	0.24	2.93	1.54	100,000
Lanl-B0-ifold	0.66	0.38	1.23	2.78	42,534
Lanl-B1-CDS	0.18	0.13	1.13	0.97	100,000
Lanl-B1-ifold	0.16	0.17	0.63	4.73	1,196
Ofori-B0-CDS	1.13	0.33	2.45	1.94	100,000
Ofori-B0-ifold	0.66	0.44	1.08	3.09	26,640
Ofori-B1-CDS	0.25	0.16	0.89	1.48	100,000
Ofori-B1-ifold	0.25	0.19	0.45	15.15	276

(b) dN/dS

Figure 3: *(a)* Output from the program `FRESCo` [23], when run on the Gag reading frame of an alignment of 200 sequences from the LANL HIV-1 database using 50 nt windows. Note the precipitous drop in dN/dS value at the beginning of Gag-Pol overlap region. *(b)* Values of dN/dS, branch length, and transition/transversion rate (see [9] for definitions) for the 52 nt Gag-Pol overlap regions within a multiple alignment from Rfam family RF00480 as well as from 52 nt RNA sequences generated by the programs `RNAsampleCDS` and `RNAiFold`. These programs generate sequences that code peptides, each of whose amino acids has BLOSUM62 similarity of either 0 or 1 to the corresponding amino acids in the Gag [resp. Pol] reading frame of the peptide translated by the 52 nt HIV-1 overlap region of [15] or by GenBank accession code AF033819.3/1631-1681. The program `RNAsampleCDS` ensures only coding requirements, while `RNAiFold` ensures both coding requirements and that the 52 nt RNAs fold into the minimum free energy structure of the Gag-Pol overlap region of HIV-1 from [15] and GenBank accession code AF033819.3/1631-1682.

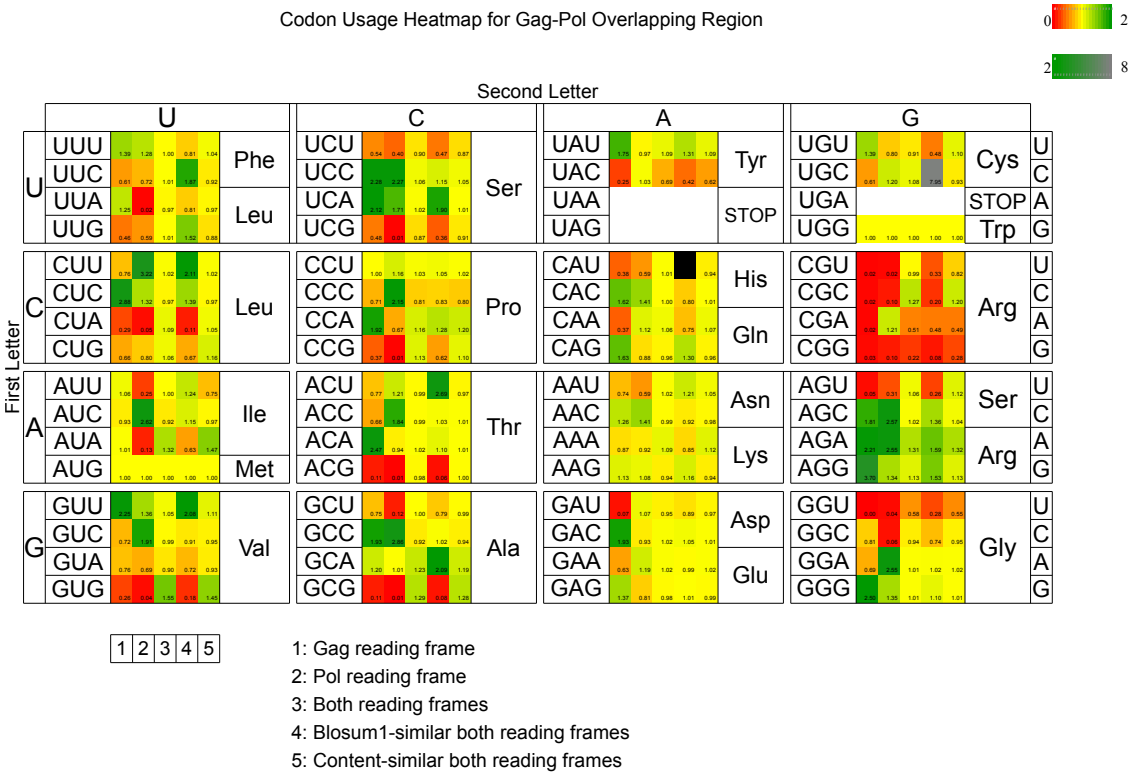
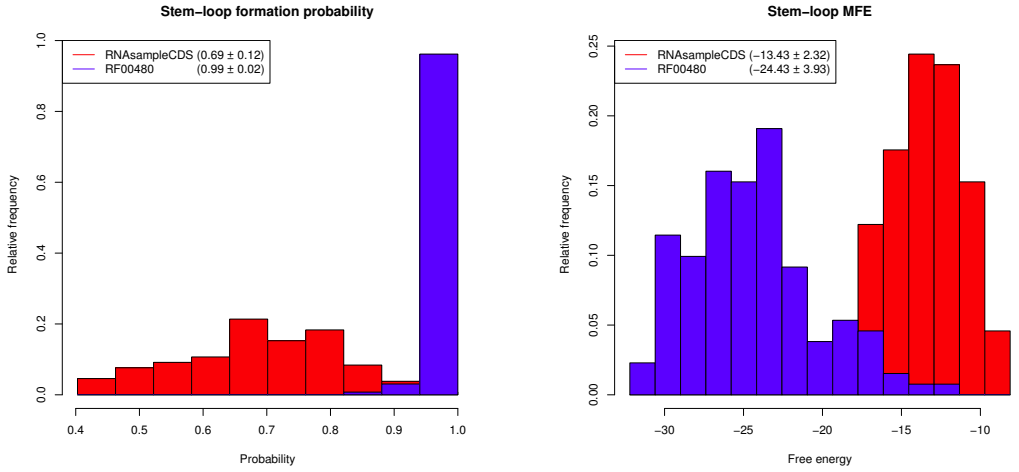


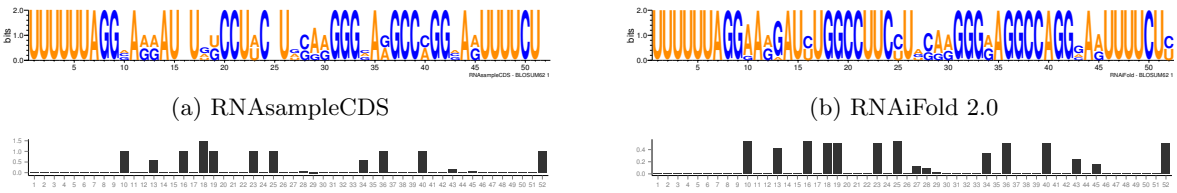
Figure 4: Heat map of the *codon preference index* (CPI) for a collection of 5125 entire Gag-Pol overlap regions of average length 205 ± 10 extracted from LANL HIV-1 database. CPI values shown at bottom right of each square. See text for additional explanation.



(a) Stem-loop formation probability

(b) Stem-loop MFE

Figure 5: For each of 131 52 nt frameshift stimulating signals (FSS) from family RF00480 from the Rfam 12.0, `RNAsampleCDS` generated 100,000 RNAs that have the same slippery sequence as the Rfam sequence, and code 17-mer peptides p [resp. q] in reading frame 0 [resp. 1] each of whose amino acids has BLOSUM62 similarity of at least $+1$ with the corresponding amino acid in the Pol [resp. Gag] peptide coded by the Rfam sequence. Stem-loop formation probability, $P(\texttt{[ ]})$, and stem-loop minimum free energy (MFE) were computed by `RNAshapes` [24] with the command `RNAshapes -q -m '[]'`. *(a)* Average stem-loop formation probability for 100,000 sequences sampled from `RNAsampleCDS` for each RF00480 sequence (red); stem-loop formation probability of HIV-1 frameshift stimulating signals from RF00480 (blue). Overall mean `RNAsampleCDS` samples is $69\% \pm 12$ (red), while that for the RF00480 sequences is 99.3 ± 2.2 (blue). *(b)* Average stem-loop MFE for 100,000 sequences sampled by `RNAsampleCDS` for each RF00480 sequence (red); stem-loop minimum free energy for HIV-1 frameshift stimulating signals from RF00480 (blue). Overall mean for `RNAsampleCDS` samples is 13.43 ± 2.32 kcal/mol (red), while that for RF00480 sequences is -24.43 ± 3.91 kcal/mol (blue).



(a) RNAsampleCDS

(b) RNAiFold 2.0

(c) Entropy1 - Entropy2

(d) Total variation dist

Figure 6: *(a)* Sequence logo from `RNAsampleCDS` for *all* 8,819,712 sequences that code peptides p [resp. q], each of whose amino acids has BLOSUM62 similarity $\geq +1$ with the corresponding amino acids of the Pol [resp. Gag] 17-mer peptides FFREDLAFPQGKAREFS [resp. FLGKIWPSHKGRPGNFL] in AF033819.3/1631-1682. The PSSM is (exactly) computed by `RNAsampleCDS` with flag `-pssm`, and the logo plot was produced using `WebLogo` [3]. The average pairwise Hamming distance is 10.92 ± 4.32 (length-normalized value of 0.21 ± 0.083), when computed with a random sample of 1000, 5000, and 10,000. *(b)* Sequence logo for all 1196 sequences determined by `RNAiFold 2.0` to fold into the frameshift stimulating signal (FSS) given by the MFE structure from AF033819.3/1629-1682 and code peptides P,Q, each of whose BLOSUM62 similarity with the Gag,Pol peptides in the overlap region is greater than or equal to $+1$. The average pairwise Hamming distance is 5.80 ± 1.84 (length-normalized value of 0.11 ± 0.035). *(c)* The position-dependent entropy is defined by $H_i = -p_A \ln p_A - p_C \ln p_C - p_G \ln p_G - p_U \ln p_U$ for each nucleotide position $i = 1, \ldots, 52$. Subfigure (c) shows the position-dependent *difference* $H_i^a - H_i^b$ in entropies of (a) minus (b). *(d)* Position-dependent total variation distance $\delta(\pi_{1,i}, \pi_{2,i}) = 1/2 \cdot \sum_{x \in \{A,C,G,U\}} |\pi_{1,i}(x) - \pi_{2,i}(x)|$ in the 52 nt region of the Gag-Pol overlap in the HIV-1 genome (GenBank AF033819.3/1631-1682) that contains the frameshift stimulating signal (FSS). Here $\pi_{1,i}$ resp. $\pi_{2,i}$ is the mononucleotide frequency at position i of the PSSM in the left resp. right panel. If total variation distance is zero, then it is suggestive that the coding constraint automatically may already entail the FSS secondary structure constraint.

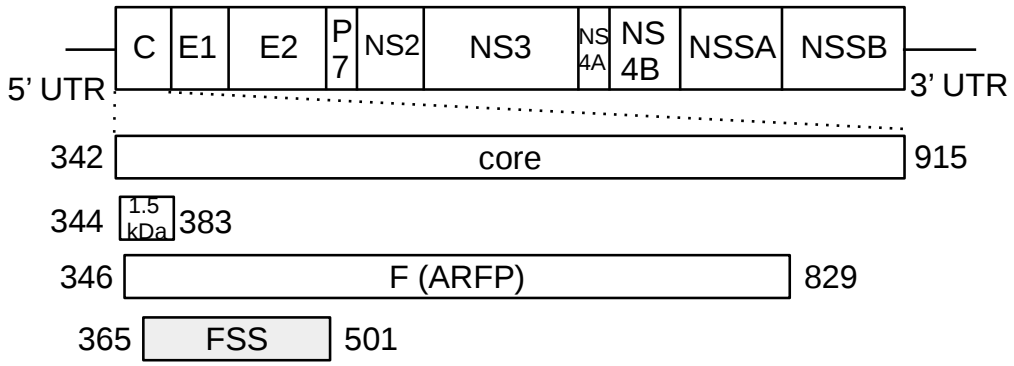
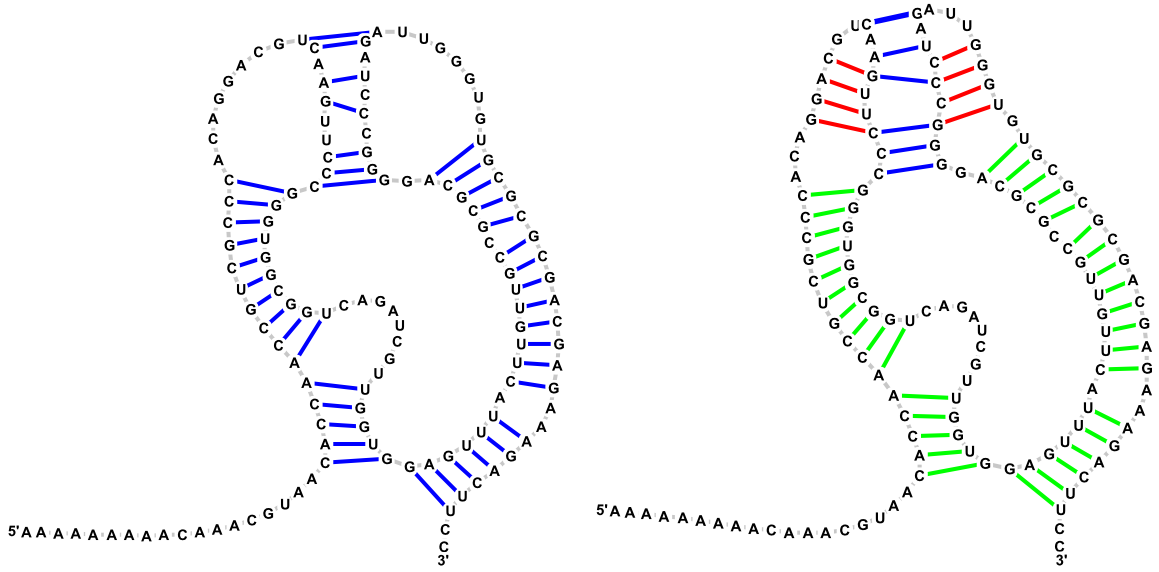


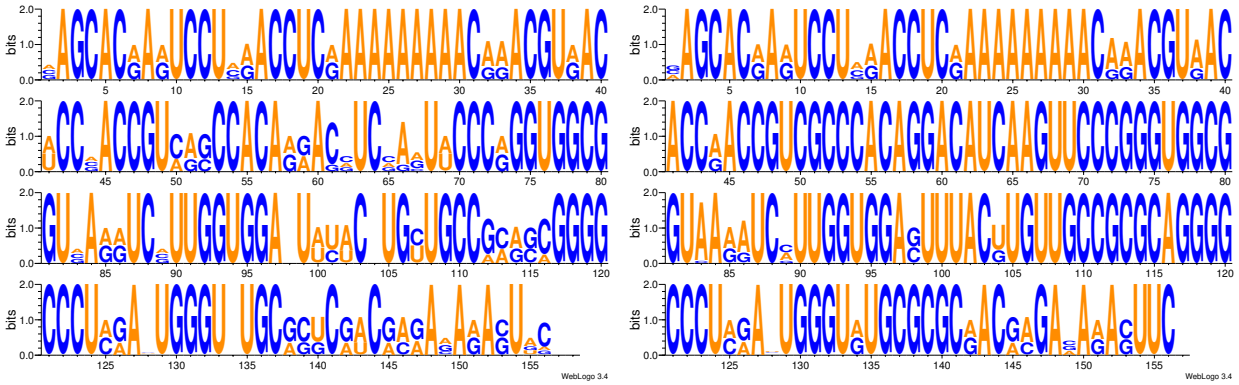
Figure 7: Organization of the initially triple, then double overlapping reading frame region of hepatitis C virus (HCV) (GenBank M62321.1). The top gene organization map is adapted from Figure 1A of [2]. All coding regions mentioned in the following include a terminal stop codon. The second line depicts the core in-frame protein, coded in nucleotides 342–915. Next, a 1.5 kDa protein is coded in nucleotides 344–383, while protein F is coded in nucleotides 346–829. The double stem-loop frameshift stimulating signal (FSS) is found at nucleotides 365-501; the FSS structure is depicted in Figure 8.



(a) Double stem-loop FSS from [2]

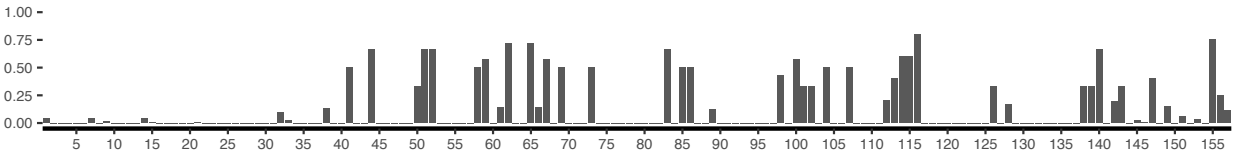
(b) MFE structure from `RNAfold`

Figure 8: HCV ribosomal frameshift stimulating signal (FSS). (a) Proposed pseudoknotted structure from [2]. (b) Minimum free eneergy (MFE) structure computed by `RNAfold 2.1.9` (green, red), with added pseudoknot (blue). Green arcs indicate common base pairs; red arcs indicate base pairs predicted by RNAfold but not present in the structure from [2]; blue arcs indicate pseudoknot base pairs from the model proposed by [2] that are absent from the `RNAfold` MFE structure. Figures produced using `jViz` [25].



(a) RNAsampleCDS

(b) RNAiFold



(c) Total variation distance

Figure 9: (A) Exact sequence logo determined by `RNAsampleCDS` for all 2.55×10^{17} sequences, whose initial 39 nucleotides code amino acids having BLOSUM62 +1 similarity to the corresponding amino acids from each of the three reading frames in the triple overlapping coding region 344-383 of the reference HCV genome, and whose remaining nucleotides code amino acids having BLOSUM62 +1 similarity to the corresponding amino acids from each of the two reading frames in the double overlapping coding region 383-501 of the reference HCV genome. (B) Sequence logo determined by `RNAiFold 2.0` for the more than 11 million sequences that fold into the HCV FSS structure depicted in Figure 8, whose initial 39 nucleotides code BLOSUM62 +1 amino acids having BLOSUM62 +1 similarity to the corresponding amino acids from each of the three reading frames in the triple overlapping coding region 344-383 of the reference HCV genome, and whose remaining nucleotides code amino acids having BLOSUM62 +1 similarity to the corresponding amino acids from each of the two reading frames in the double overlapping coding region 383-501 of the reference HCV genome. (C) Total variation distance shown for each nucleotide position, determined by computing the total variation distance between the position-specific profiles of (A) and (B).

SUPPLEMENTARY INFORMATION FOR "NEW TOOLS TO ANALYZE OVERLAPPING CODING REGIONS"

A.H. BAYEGAN, J.A. GARCIA-MARTIN, PETER CLOTE
BIOLOGY DEPARTMENT, BOSTON COLLEGE

In this supplement, we discuss the following topics in greater detail: (1) How to use the software `RNAiFold 2.0` [2] to investigate a secondary structure frameshift stimulating signal within an overlapping coding region, as in the case of the HIV-1 Gag-Pol overlap region and the HCV triple overlapping coding region. (2) Computation of the *codon preference index* (CPI), using different control data sets. (3) Analysis of the HIV-1 Gag-Pol synonymous substitution rate with respect to different reading frames. (4) Run time analysis of `RNAsampleCDS`.

1. `RNAiFold 2.0`

In this section, we provide examples for the syntax used to run `RNAiFold 2.0` [2], in order to generate RNA sequences that fold into a user-specified target secondary structure and which code user-specified peptides in possibly overlapping coding regions. For more details, please consult [2]. The command file in Example 1 below generates all RNAs that code the 17-mer peptide FFREDLAFLQGKAREFS [resp. FLGKIWPSYKGRPGNFL] in the Pol [resp. Gag] reading frame 0 [resp. 1]. Although not shown here, `RNAiFold 2.0` allows one to stipulate simultaneous coding requirements in reading frames 0,+1,+2 (but not -0,-1,-2).

```
1. > Example 1: overlapping amino acid constraints; NO secstr
constraint
2. ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
3. NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
4. #AAseqcon
5. FFREDLAFLQGKAREFS,FLGKIWPSYKGRPGNFL
6. #AAstartPos
7. 1,2
8. #AAsimilCstr
9. 1
10. #MAXsol
11. 0
```

Line 1 is a comment, while lines 2,3 indicate structure resp. sequence constaints, neither of which apply in the current Example 1. Indeed, a comma in a given position indicates that the nucleotide may be either paired or unpaired (no constraint), and 'N' in a given position indicates any RNA nucleotide can be listed. Lines 4-7 indicate that coding constraint is imposed in positions 1 and 2 (i.e. reading frames 0 and 1), and that the translated peptides are FFKKNGAGSCDPAPDFF resp. FLKKTEPGRVTRLRTFL. Note that '#' is not a comment, but rather part of a 'hashtag' the precedes the value of a parameter. Lines 8,9

Corresponding author: `clote@bc.edu`.

indicate that all returned RNA sequences must code peptides p [resp. q] in reading frame 0 [resp. 1], whose amino acids have BLOSUM62 similarity is at least 1 with the corresponding amino acids of the peptides FFREDLAFLQGKAREFS [resp. FLGKIWPSYKGRPGNFL], in the Pol [resp. Gag] reading frame, as shown in line 5. Lines 10,11 indicate that there is no upper bound on the number of solutions, hence all 6,614,784 solutions are to be generated. See [2] for more detailed explanation of the syntax for `RNAiFold 2.0` command files.

If the parameter for `#AAsimilCstr` had been 0, instead of 1, then `RNAiFold 2.0` could *in theory* generate all 55,552,444,416 solutions – this is due to the fact that memory usage does not depend on the number of solutions in the search space. Examples 3 and 4 show that `RNAiFold 2.0` can generate enormous numbers of sequences, whose MFE structure is identical to a given target structure, and which satisfy possible additional constraints.

Executing `RNAiFold 2.0` on the command file from Example 1 returns all 6,614,784 solutions in user+sys time 3.27 hours (11766.06 sec). In contrast, within 12.96 min (777.87 sec), `RNAsampleCDS` sampled 10 million sequences that code peptides p [resp. q] in the Pol [resp. Gag] reading frame 0 [resp. 1], each of whose amino acids has BLOSUM62 similarity of at least +1 with those of the Pol peptide FFREDLAFLQGKAREFS [resp. Gag peptide FLGKIWPSYKGRPGNFL]. After sorting and removing duplicates, which required an additional 43.0 sec, we obtained 4,758,267 unique solutions. Clearly, `RNAsampleCDS` is much faster tool to sample sequences that code in overlapping reading frames, provided that the sequences returned are not required to fold into a specified target structure, and provided that *all* solutions need not be returned; otherwise `RNAiFold 2.0` is the appropriate tool to apply.

The following `RNAiFold 2.0` command file generates all frameshift stimulating signals, that include the `UUU UUU A` slippery sequence in the Pol reading frame, have (exactly) the minimum free energy structure corresponding to that of GenBank AF033819.3/1631-1681, and code peptides in reading frames 0,1 that have at least BLOSUM62 similarity of +1 with corresponding peptides translated in AF033819.3/1631-1681. There are exactly 1196 solutions for BLOSUM62 threshold of +1, 42,534 solutions for threshold 0, and more than 230,261,152 solutions for threshold -1.

```
1. > Example 2: overlapping amino acid (AA) constraints with target
sec str
2. ......((((((..(((((((((((....)))))))))))...)))))).
3. UUUUUUANNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
4. #AAseqcon
5. FFREDLAFLQGKAREFS,FLGKIWPSYKGRPGNFL
6. #AAstartPos
7. 1,2
8. #MAXsol
9. 0
10. #AAsimilCstr
11. 1
```

Since the publication of [2], `RNAiFold 2.0` allows the desired peptides to be entered using PROSITE pattern syntax, given in Example 3 below. The PROSITE patterns below for Pol and Gag peptides were obtained by analyzing those 665 sequences from LANL HIV-1 database which contain slippery sequence UUUUUUA and whose MFE FSS structure is identical to that of Figure 1a of the main paper, the most common structure found in

LANL. The wild card '*' (representing a stop codon) can be used in `RNAiFold 2.0` as shown in the following command file.

```
1. > Example 3: overlapping AA constraints, FSS, PROSITE pattern
syntax
2. ......((((((..(((((((((((....))))))))))))...)))))).
3. UUUUUUANNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
4. #AAseqcon
5. [F][F][KR][KE][ASTNKHDIG][FWMPRELV][ATPV][SAFCL][HSQPL]\
[SKEHQCRGL][QWDRGE][AQRKGE][SATRKPE][ASWRGEV][FNRKGE]\
[SFPKEL][FSP*YHCL],\
[F][L][KG][NRK][AFNKPVMIL][WCRGL][SHQPL][AFSCPV]\
[STNKPYHQCRL][ASQTNKRG][TDRKGE][SPRG][QRPG][ANP*GE]\
[FNIG][SFNPL][ASFQCIPLV]
6. #AAstartPos
7. 1,2
8. #MAXsol
9. 0
```

Line 5 is a *single* line, where continuation is indicated by a backslash (shown as displayed in order to fit column dimensions). Notice as well the presence of the comma in line 5, which separates the amino acid ccoding constraints for reading frame 0 from those for reading frame 1. We could have stipulated amino acid constraints in three reading frames by replacing line 5 by three PROSITE patterns separated by commas, and by replacing line 7 by '1,2,3'. Running `RNAiFold 2.0` in the background with the command file of Example 3 for a few weeks, we obtained more than 273,926,421 solutions before we chose to terminate the computation.

The following example returns sequences containing the slippery heptamer UUUUUUA, that fold into the most common secondary structure found in the LANL HIV database and having GC-content between 20-40%. Note that no amino acid constraints are given.

```
1. > Example 4: FSS
2. ......((((((..(((((((((((....))))))))))))...)))))).
3. UUUUUUANNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
4. #minGCcont
5. 20
6. #maxGCcont
7. 40
8. #MAXsol
9. 0
```

By adding two additional lines, respectively containing '#LNS' and '1', `RNAiFold 2.0` applies Large Neighborhood Search (LNS) instead of default Constraint Programming (CP) – see [1] for explanation. In contrast to CP, LNS may perform restarts, leading to multiple occurrences of the same solution. LNS can be faster than CP, especially when the target structure is large; when using LNS for Example 3, we obtained 559,877,906 solutions, of which 440,389,701 were unique, before terminating execution. By running `RNAiFold 2.0` on this command file, along with two other command files where the values 20,40 in lines 5,7 are replaced by 40,60 and by 60,80, a total of 713,134,134 solutions were obtained before we decided to terminate the computations. These examples show that `RNAiFold 2.0` can efficiently generate a very large number of sequences, all of which are guaranteed to fold into the target structure and comply with any additional constraints that may be imposed.

2. Codon usage analysis

We begin by recalling the following definitions. For RNA sequence $\mathbf{a} = a_0, \ldots, a_{3n}$ which codes n-mer peptides in reading frames $0, 1$, for codon $w \in (\{A, C, G, U\})^3$ and reading frame $r \in \{0, 1\}$, define $f_{(w,\mathbf{a},r)}$ to be the number of occurrences of codon w in reading frame r of $\mathbf{a}$, and for amino acid AA, define $f_{(AA,\mathbf{a},r)}$ to be the number of occurrences of codons coding AA in reading frame r of $\mathbf{a}$. Define the *observed codon preference* in reading frame r in sequence $\mathbf{a}$ by $p_{obs}(w, \mathbf{a}) = \sum_{r=0}^{1} f_{(w,\mathbf{a},r)} / \sum_{r=0}^{1} f_{(AA,\mathbf{a},r)}$. If S is a set of mRNAs of length $3n + 1$, each of which codes n-mer peptides in both reading frames $0, 1$, then define the *observed codon preference* in S by $p_{obs}(w, S) = \sum_{r=0}^{1} \sum_{\mathbf{a} \in S} f_{(w,\mathbf{a},r)} / \sum_{r=0}^{1} \sum_{\mathbf{a} \in S} f_{(AA,\mathbf{a},r)}$. Define the *codon preference index* $I(w)$ of codon w in S by $I(w) = p_{obs}(w, S)/p_{obs}(w, S')$, where S' is a *control* set of mRNAs of length $3n+1$. In some cases below, we consider only one reading frame, as when analyzing the Gag only and Pol only reading frames, in which case we define the observed codon preference with respect to the appropriate reading frame r alone: $p_{obs}(w, \mathbf{a}) = f_{(w,\mathbf{a},r)}/f_{(AA,\mathbf{a},r)}$ and $p_{obs}(w, S) = \sum_{\mathbf{a} \in S} f_{(w,\mathbf{a},r)} / \sum_{\mathbf{a} \in S} f_{(AA,\mathbf{a},r)}$. By definition, codon preference index $I(w)$ values less than 1 (greater than 1) indicate that codon w is avoided (preferred).

In the following, we consider the *general* formulation of the forward and backward partition function, defined to account for all six reading frames $+0, +1, +2, -0, -1, -2$. As mentioned in the paper, this requires the consideration of 5-tuples $s = s_0 s_1 s_2 s_3 s_4$. For simplicity of exposition in the main paper, the forward and backward partition function were defined only for reading frames 0 and 1, for which reason, we considered 4-tuples $s = s_0 s_1 s_2 s_3$. In the sequel, $p_{obs}(w, S)$ can be calculated by counting codons in S, and $p_{obs}(w, S)$ is computed utilizing the forward and backward partition functions as follows:

$$\begin{aligned} f_{(w,\mathbf{a},r)} &= \sum_{k=1}^{n} \sum_{s \in L_k} I[w \in \text{reading frame } r \text{ of } \mathbf{a}] \cdot ZF(k-1, s_0, s_1) \cdot ZB(k, s_3, s_4) \\ f_{(AA,\mathbf{a},r)} &= \sum_{w \in \text{all codons translating AA}} f_{(w,\mathbf{a},r)} \end{aligned}$$

where $ZF[0, ch1, ch2] = 1$ and $ZB[n, ch1, ch2] = 1$ for $ch1, ch2 \in \{A, C, G, U\}$.
Similarly, $p_{obs}(w, S', gc)$ can be defined as the probability of observing w in sequences of S' with GC-content in range $gc = [gcl, gcu]$:

$$\begin{aligned} f_{(w,\mathbf{a},r,gc)} &= \sum_{k=1}^{n} \sum_{s \in L_k} \sum_{x_1=1}^{3k+2} \sum_{x_2=1}^{3(n-k)+2} I[w \in \text{reading frame } r \text{ of } \mathbf{a} \text{ and gcl} \le x_1 + x_2 + \text{GCcount}(s_2) \le \text{gcu}] \\ &\quad \cdot \; ZF(k-1, x_1, s_0, s_1) \cdot ZB(k, x_2, s_3, s_4) \\ f_{(AA,\mathbf{a},r,gc)} &= \sum_{w \in \text{all codons translating AA}} f_{(w,\mathbf{a},r,gc)} \end{aligned}$$

where $ZF[0, 0, ch1, ch2] = 1$ and $ZB[n, 0, ch1, ch2] = 1$ for $ch1, ch2 \in \{A, C, G, U\}$.

With this notation, Figures S2, S3, and S4 depict heat maps for the codon preference index $I(w)$, computed over 5,125 sequences from the LANL HIV-1 database. Figure S2 shows the heat map of $I(w)$ computed for 5,125 Gag and Pol sequences obtained from the LANL HIV-1 database. Five columns are indicated for each codon:

Column 1: S is the set of Gag sequences from the LANL HIV-1 database without the overlapping region. S' is the collection of $\sim 8 \times 10^{232}$ sequences that code proteins in S in the Gag reading frame.

Column 2: S is the set of Pol sequences from the LANL HIV-1 database without the overlapping region. S' is the collection of sequences that code proteins in S in the Pol reading frame. The number of sequences in S' is so huge that is could not be exactly defined.

Column 3: S is the set of Gag-Pol overlapping sequences from the LANL HIV-1 database. S' is the collection of $1,204,620$ sequences that code proteins in S in both Gag and Pol reading frames.

Column 4: S is the set of Gag-Pol overlapping sequences from the LANL HIV-1 database. S' is the collection of $\sim 10^{34}$ sequences that code proteins in S having BLOSUM62 similarity of at least +1 in both Gag and Pol reading frames.

Column 5: S is the set of Gag-Pol overlapping sequences from the LANL HIV-1 database. S' is the collection of $1,022,784$ sequences that have GC-content of ± 5 to at least one sequence in S and code the same proteins as S in both Gag and Pol reading frames.

The heat map in Figure S3 depicts values of $I(w)$ computed for the same dataset as above. In all columns of Figure S3, S is the set of Gag-Pol overlapping sequences from the LANL HIV-1 database. Note that this is the same S collection used for Figure 3 of the main text. The control set S' (see main text for explanation) in columns 1 and 2 is the collection of sequences that code any protein of length 68 in a single reading frame. However, in columns 3-5, S' is the collection of sequences that code any protein of length 68 in both +0 and +1 reading frames. Mean peptide length in the overlapping region of the dataset is 68. Note that the codon preference index (CPI) computed in Figure S3 is with respect to all possible coding sequences regardless of amino acid coded, and so is natural generalization of the method of [3] to the case of overlapping reading frames.

Figure S4 shows the standard deviation of $I(w)$ for the codons of each amino acid. Here, $I(w)$ is computed as in Figure 4 of the main text. Arginine is the most varied and thus the most optimized amino acid in the Gag-Pol overlapping region.

3. Synonymous substitution rate analysis

To further clarify that Gag-Pol overlapping region is under high evolutionary constraint we used FRESCo, a phylogenetic codon model-based to find regions in excess synonymous constraint to analyse 200 Gag-Pol sequences from LANL HIV-1 database. The phylogenetic tree expected as an input to FRESCo was built by `RAxML v.8` [5]. As Figure S1a indicates, in the starting and ending regions of Pol where it has overlap with Gag and Vif genes, synonymous substitution rate is low. Figure S1b also indicates a sudden drop in the

the synonymous substitution rate for 200 artificial Gag-Pol sequences in which an extra nucleotide 'U' is inserted at the end of Gag to coordinate the reading frames.

4. Run time analysis of RNAsampleCDS

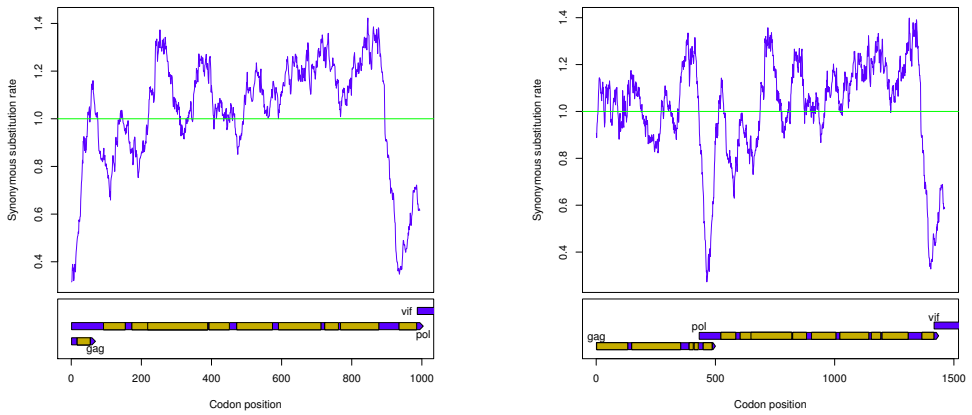
With the exception of Algorithm 1, which uses breadth first search (BFS), all algorithms run in linear space and time. For the benefit of readers unfamiliar with algorithmic complexity, we provide a brief discussion of the linear run time, and then use least squares fitting to give an estimate of the run time constant in computational experiments.

In Algorithm 1, our method explicitly constructs a *prefix tree* (also called *trie* in computer science), whose root is the empty string, such that nodes at depth k are 4-nt RNAs $\mathbf{t} = t_0t_1t_2t_3$ with the property that the first nucleotide of $\mathbf{t}$ is identical with the last nucleotide of the parent $\mathbf{s} = s_0s_1s_2s_3$ of $\mathbf{t}$ – and of course, that the merge of all 4-tuples from the root to $\mathbf{t}$ satisfies the (overlapping) coding requirement in both reading frames. It follows that every mRNA that satisfies the coding requirement appears as the merge of a unique path from root to leaf, hence the run time and memory requirements are $O(N)$, where there are N possible solution mRNA sequences.

Given overlapping n-mer peptides, Algorithm 2 uses dynamic programming to compute the forward partition function $ZF(k, ch)$ and the backward partition function $ZB(k, ch)$ for $k = 0, \ldots, n$ and each nucleotide $ch \in \{A, C, G, U\}$. When inductively computing the value of $ZF(k, ch)$ [resp. $ZB(k, ch)$], finitely many arithmetic operations are applied to the previously computed values $ZF(k-1, A)$,ZF(k-1,C), $ZF(k-1, G)$,ZF(k-1,U) [resp. $ZB(k+1, A)$,ZB(k+1,C), $ZB(k+1, G)$,ZB(k+1,U)] are performed. It follows that there are $O(n)$ many inductive steps, each of which requires constant time, hence the run time is $O(n)$, as well as the memory requirements. Similarly, the computation of PSSM (Algorithm 3), of positional codon frequency (Algorithm 4), and both unweighted sampling (Algorithm 5) and weighted sampling (Algorithm 6) require linear time and space. It should be noted that in both sampling algorithms, the run time is $O(n)$ to first compute the forward and backward partition functions ZF, ZB, and then for *each* sequence that is sampled, the run time is $O(n)$. Ultimately, the run time for the sampling algorithms depends on the number N of desired samples, so the overall run time is $O(n + N)$, where n is the length of the peptides that must be coded, and N is the number of samples. An estimate of the actual run time constants for n and N are given next.

The run time for RNAsampleCDS is ostensibly linear in RNA sequence length and number of samples to be generated. Using least squares fitting, we can compute the run time as follows. For each sample size N equal to 10^4, 2×10^4, 3×10^4, we generated N samples using RNAsampleCDS, which code peptides having $n = 20, 30, 40, \ldots, 160$ many amino acids in all 6 overlapping reading frames (i.e. the only requirement is absence of a stop codon). It follows that sequence length $L = 3 \cdot n + 2$ takes values $62, 92, 122, \ldots, 482$ thus providing 45 data points. Now define M to be the 45×2 matrix, for which $M_{i,1}$ is the sequence length $L \in \{62, 92, \ldots, 482\}$ and $M_{i,2}$ is the number of samples $N \in \{10^4, 2 \times 10^4, 3 \times 10^4\}$ for the ith data point. Define B to be the 45×1 column vector, where B_i is the run time for RNAsampleCDS to compute the partition function and generate N samples for the

ith data point. Using the Python function `numpy.linalg.lstsq`, we solved $MX = B$ by least squares to determine that `RNAsampleCDS` computes the partition function in time $\approx 0.58831373 \cdot L$, and samples N RNA sequences of length L in time $\approx 0.00550239 \cdot N$. See Figure S6 for a plot of the run time of `RNAsampleCDS` for this data.



(A) Pol reading frame (B) Artificial

FIGURE S1. Synonymous substitution rate analysis using `FRESCo` [4], with window size 50 nt, for 200 Pol *(Left)* and 200 modified Gag-Pol *(Right)* sequences from the LANL HIV-1 database. Gag-Pol sequences were modified by inserting one additional nucleotide at the beginning of the overlapping coding region, thus causing the Pol reading frame to be in-frame, rather than -1. Codon positions in the lower panel are based on HXB2 reference sequence. Mature peptides are shown in yellow.

REFERENCES

[1] J. A. Garcia-Martin, P. Clote, and I. Dotu. RNAiFOLD: a constraint programming algorithm for RNA inverse folding and molecular design. *J. Bioinform. Comput. Biol.*, 11(2):1350001, April 2013.

[2] J. A. Garcia-Martin, I. Dotu, and P. Clote. RNAiFold 2.0: a web server and software to design custom and Rfam-based RNA molecules. *Nucleic. Acids. Res.*, 43(W1):W513–W521, July 2015.

[3] M. Gribskov, J. Devereux, and R. R. Burgess. The codon preference plot: graphic analysis of protein coding sequences and prediction of gene expression. *Nucleic. Acids. Res.*, 12(1):539–549, January 1984.

[4] R. S. Sealfon, M. F. Lin, I. Jungreis, M. Y. Wolf, M. Kellis, and P. C. Sabeti. FRESCo: finding regions of excess synonymous constraint in diverse viruses. *Genome Biol.*, 16:38, 2015.

[5] Alexandros Stamatakis. RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688–2690, 2006.

[6] P. Steffen, B. Voss, M. Rehmsmeier, J. Reeder, and R. Giegerich. RNAshapes: an integrated RNA analysis package based on abstract shapes. *Bioinformatics*, 22(4):500–503, 2006.

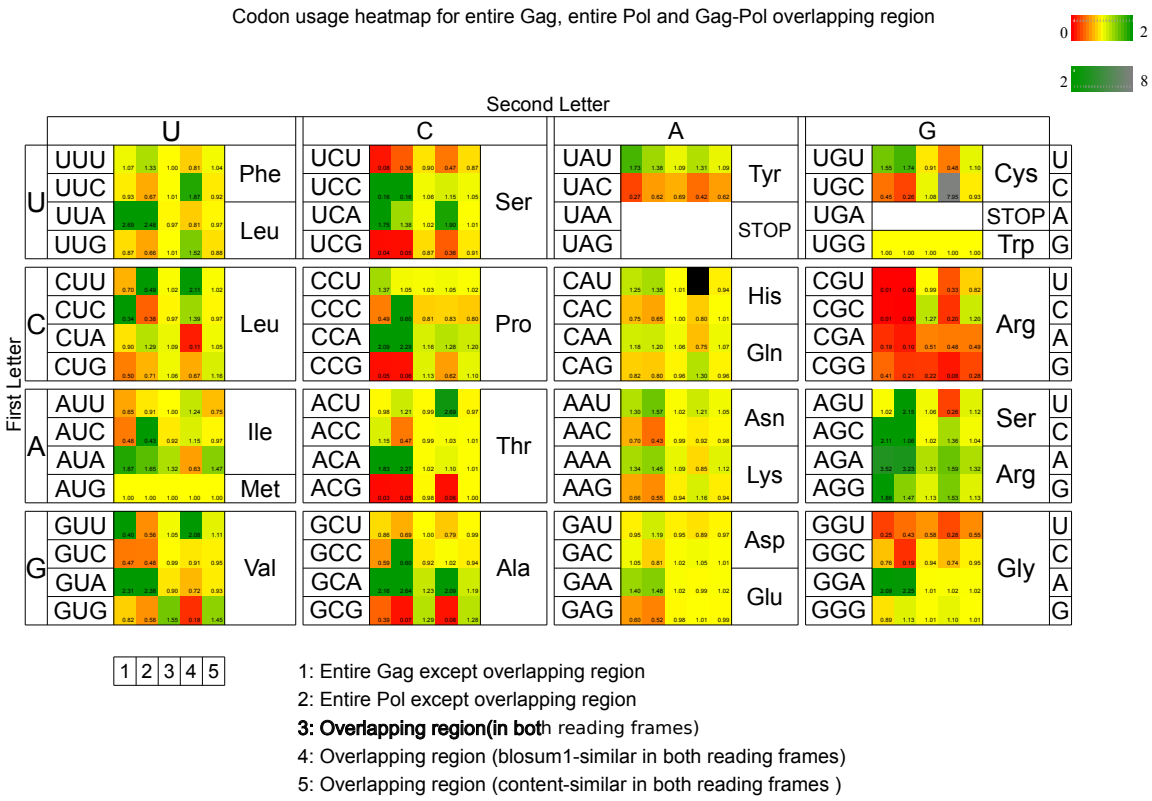


FIGURE S2. Heat map of the codon preference index (CPI) for a collection of 5,125 Gag, Pol and Gag-Pol overlapping sequences obtained from the LANL HIV-1 database.

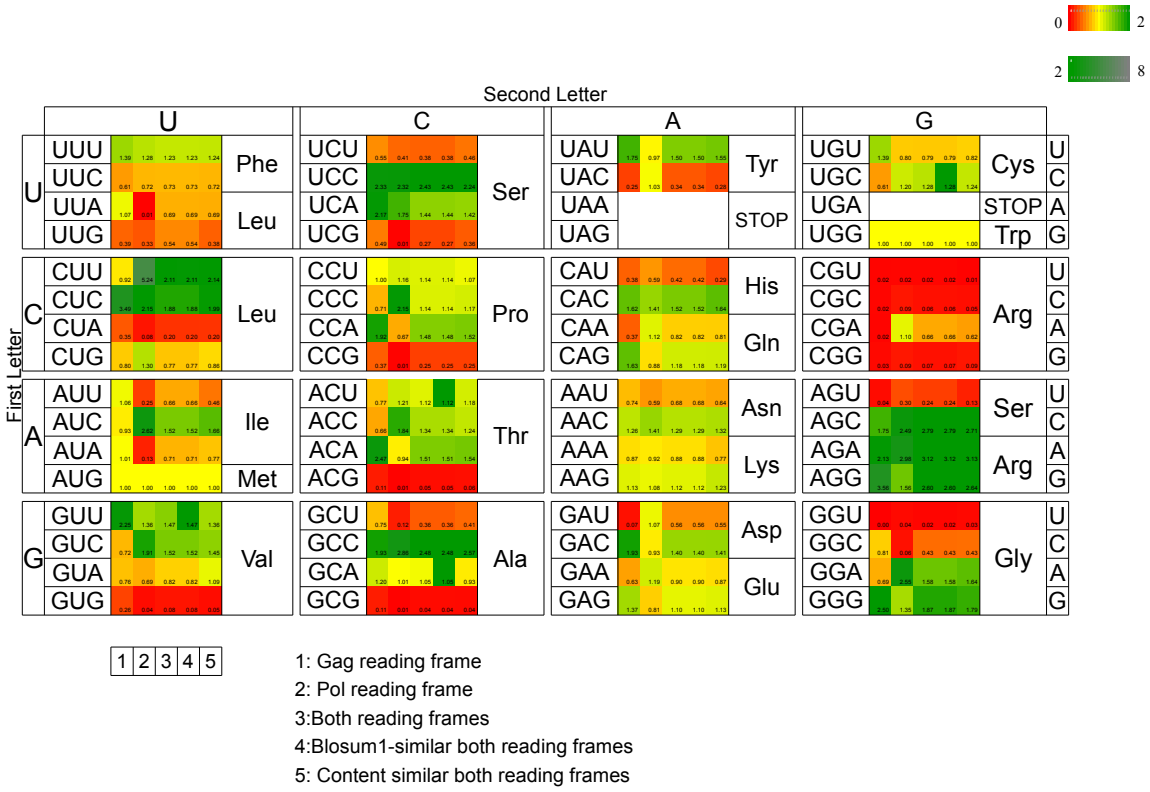


FIGURE S3. Heat map of the codon preference index (CPI) for a collection of 5,125 Gag-Pol overlapping sequences obtained from the LANL HIV-1 database where S' is the collection of sequences coding any amino acid (i.e. not containing a stop codon) in the corresponding reading frames.

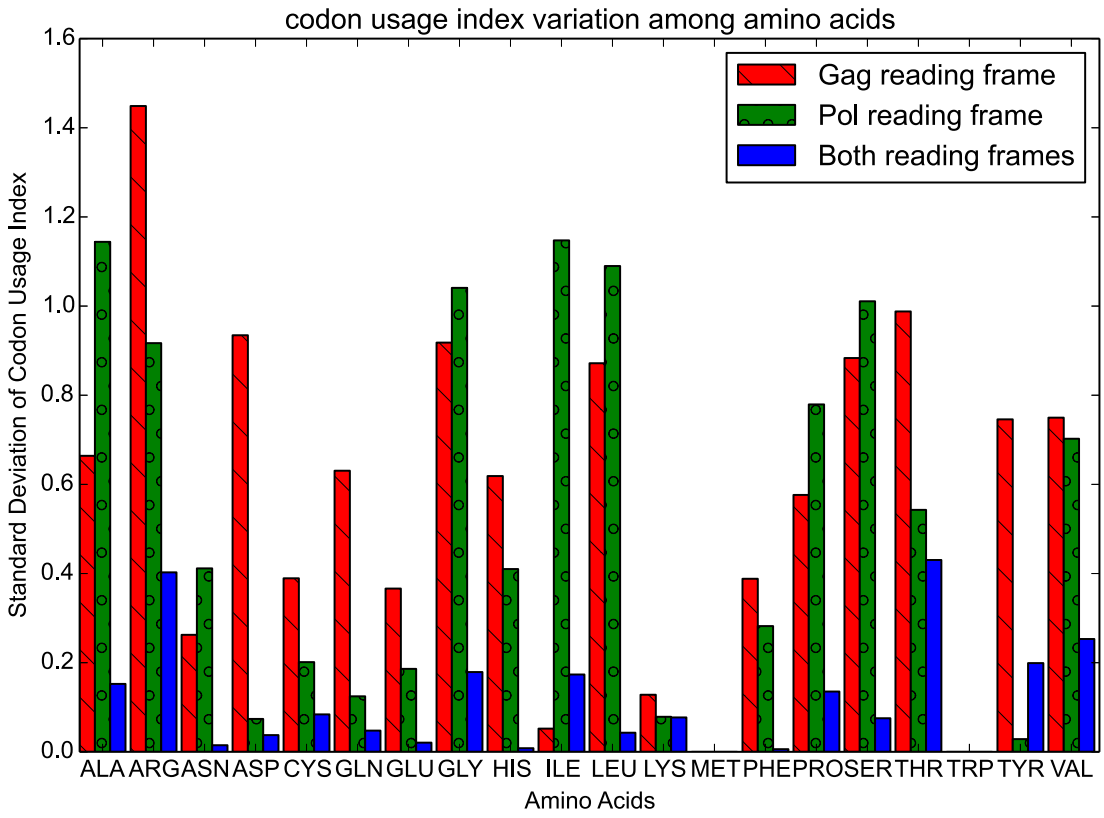
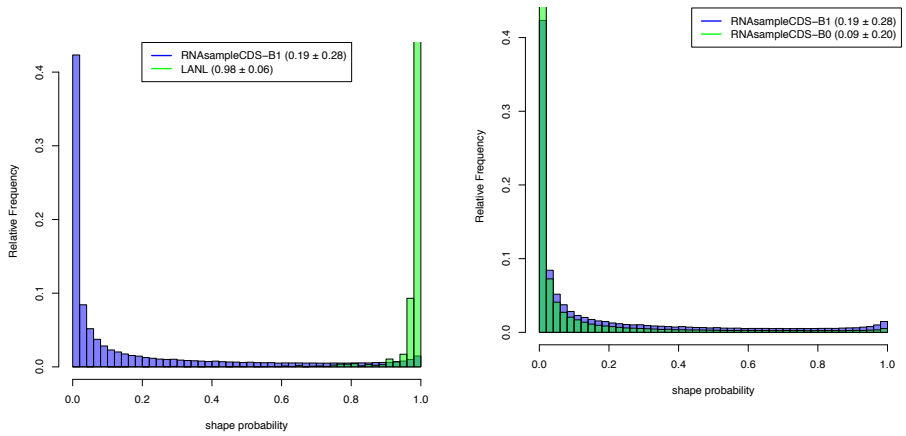
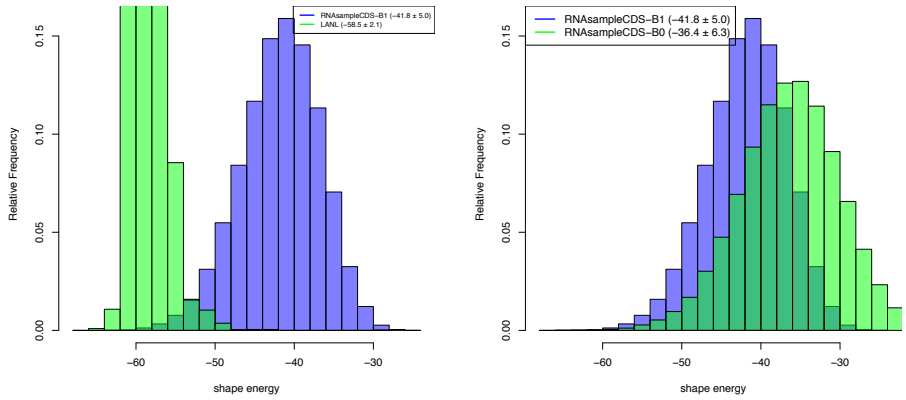


FIGURE S4. Standard deviation of CPI for synonymous codons computed from the Gag-Pol overlapping sequence of 5,125 sequences from the LANL HIV-1 database.



(A) BLOSUM62 +1 vs LANL HCV (B) BLOSUM62 +1 vs BLOSUM62 0



(C) BLOSUM62 +1 vs LANL HCV (D) BLOSUM62 +1 vs BLOSUM62 0

FIGURE S5. Using `RNAsampleCDS`, we sampled 100,000 sequences coding peptides having BLOSUM62 +1/0 similarity to the peptides in each overlapping reading frame of the reference HCV1a genome (GenBank M62321.1). Using `RNAshapes` [6], we determined the Boltzmann probability of having a double stem-loop shape `[ ] [ ]`. We also determined the Boltzmann probability of double stem-loop shape `[ ] [ ]` in 6,589 sequences from the LANL HCV database. (A) Average double stem-loop probability of BLOSUM62 +1 sequences compared with that of the LANL HCV sequences. (B) Average double stem-loop probability of BLOSUM62 +1 sequences compared with Blosum 0 sequences. (C) Average double stem-loop free energy of BLOSUM62 +1 sequences compared with that of the LANL HCV sequences. (D) Average double stem-loop free energy of BLOSUM62 +1 sequences compared with that of BLOSUM62 0 similar sequences.

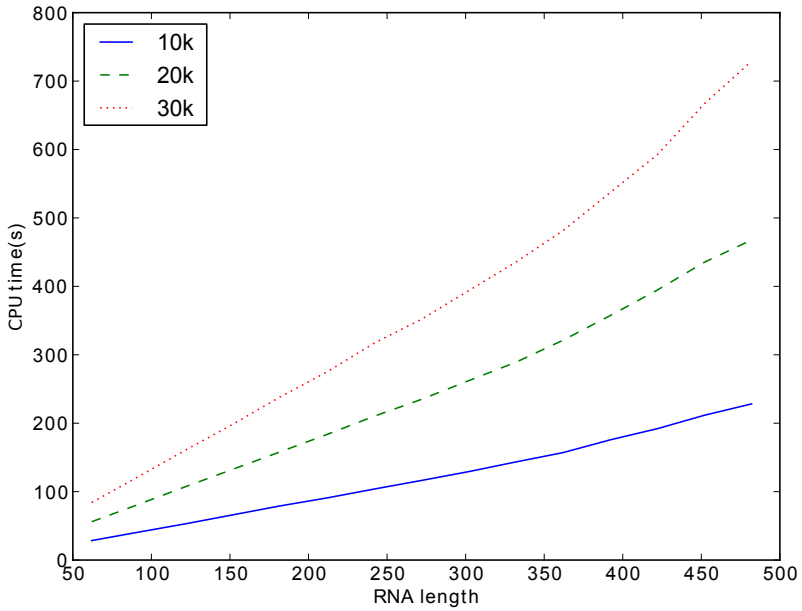


FIGURE S6. Run time for `RNAsampleCDS` to generate RNA sequences of length L that code peptides in all six reading frames – i.e. a stop codon does not appear in any of the six reading frames. For each sample size N equal to 10^4, 2×10^4, 3×10^4, `RNAsampleCDS` generated N samples that code peptides having $n = 20, 30, 40, \ldots, 160$ many amino acids. Thus sequence length $L = 3 \cdot n + 2$ takes values $62, 92, 122, \ldots, 482$ thus providing 45 data points. Using least squares fitting, we determine that `RNAsampleCDS` computes the partition function in time $\approx 0.58831373 \cdot L$, and samples N sequences each of length L in time $\approx 0.00550239 \cdot N$.