Algorithm 1 SRI Generator

Given: We define $\mathcal{B} = (A,T,C,G) = (\mathcal{B}_i)_{i=1}^4$ to be the sequence of canonical bases. Furthermore, let $\mathcal{O}_i^K = (\mathcal{O}_i^K)_{i=1}^K$ be the sequence of **oligomers** of length $K$, e.g., $\mathcal{O}_i^3 = (AAA, AAT, \ldots, CGG, GGG)$. (Notice that $\mathcal{O}_i^3$ has $4^3 = 64$ elements.) Similarly, let $\mathbb{F}_i^K = (\mathbb{F}_i^K)_{i=1}^4$ be the sequence of oligomer **frequencies** of length $K$, such that $\sum_{i=1}^4 \mathbb{F}_i^K = 1$ and $\mathbb{F}_i^K \in [0,1]$ $\forall i$. For each fixed $i$, the oligomer $\mathcal{O}_i^K$ has one corresponding frequency value $\mathbb{F}_i^K$. These frequencies are computed from a sample input sequence using SRI Analyzer. Let $N \geq 1$ be the user-chosen oligomer length for which the frequency composition is being approximated. Finally, we assume the input sequence is composed purely of A, T, C, and G. For a modified version of this algorithm which handles impure samples, please see our source code.

Ensure: The output sequence(s) approximates the short-range inhomogeneity of the input sequence. In other words, the input and output sequence(s) will share a similar $N$-mer frequency composition. The output sequence(s) will be randomly constructed to satisfy this constraint.

1: $(\mathcal{O}_i^{N}, \mathbb{F}_i^{N}) = \text{ReadCompositionTable()}$ #Example for $N = 1 : ( (A,T,C,G), (0.25, 0.40, 0.20, 0.15) )$
   #Calculate the demarcation table for the random selection of the first oligomer.
2: sum $\leftarrow 0$
3: for $i = 1$ to $4^N$ do
4:   sum $\leftarrow$ sum $+$ $\mathbb{F}_i^{N}$
   #Let partialSum be an array such that partialSum$^{(i)} \in [0,1]$, for $1 \leq i \leq 4^N$.
5:   partialSum$^{(i)} \leftarrow$ sum
6: end for
7: for each FASTA sequence, “seq” do
8:   length $= \text{GetSequenceLength(seq)}$
9:   if length $\geq N$ then
10:      $R_1 \leftarrow \text{GenerateRandomNumber}(0,1)$ #Generate some random $R_1 \in [0,1]$.
11:         for $i = 1$ to $4^N$ do
12:            if $R_1 < \text{partialSum}^{(i)}$ then
13:               randomSeq $\leftarrow \mathcal{O}_i^{N}$
14:               Exit Loop.
15:         end if
16:      end for
17:      for $i = 1$ to length $-$ $N$ do
18:         sum $\leftarrow 0$
19:         #The next base is chosen randomly using the frequencies of the 4 possible overlapping $N$-mer sequence tails.
20:        $R_2 \leftarrow \text{GenerateRandomNumber}(0,1)$
21:        tail $\leftarrow \text{Suffix(seq,$N$) - 1}$ #Example: Suffix(“hotdog”, 3) = “dog”.
22:        for $i = 1$ to $4$ do
23:           oligo $\leftarrow \text{Concatenate(tail, $\mathcal{B}_i$)}$ #Example: Concatenate(“hot”, “dog”) = “hotdog”.
24:           $f \leftarrow \text{GetOligoFrequency(oligo)}$ #GetOligoFrequency(oligo) = $\mathbb{F}_j^N$ for one $j$ such that $\mathcal{O}_j^{N} = \text{oligo}$.
25:           sum $\leftarrow$ sum $+$ $f$
16:         #Let demarcation be an array such that demarcation$^{(i)} \in [0,1]$, for $1 \leq i \leq 4$.
26:        demarcation$^{(i)} \leftarrow$ sum
27:        if $R_2 < \text{demarcation}^{(i)}$ then
28:           randSeq $\leftarrow \text{Concatenate(randSeq, $\mathcal{B}_i$)}$
29:           Exit Loop.
30:      end if
31:      end for
32:      if $R_2 \geq$ sum then
33:         randomBase $\leftarrow \text{PickRandomBase()}$ #Randomly choose a base from {A,T,C,G}.
34:         randSeq $\leftarrow \text{Concatenate(randSeq, randomBase)}$
35:      end if
36:   end for
37: end for
38: \text{WriteOutputFile(randSeq)}