Translating RNA into Protein

Jan Emmanuel Samson

2024-07-29

Problem

The 20 commonly occurring amino acids are abbreviated by using 20 letters from the English alphabet (all letters except B, J, O, U, X, and Z). Protein strings are constructed from these 20 symbols. Henceforth, the term genetic string will incorporate protein strings along with DNA strings and RNA strings.

The RNA codon table dictates the details regarding the encoding of specific codons into the amino acid alphabet.

- **Given**: An RNA string *s* corresponding to a strand of mRNA (of length at most 10 kbo).
- **Return**: The protein string encoded by *s*.

Sample Dataset

```
AUGGCCAUGGCGCCCAGAACUGAGAUCAAUAGUACCCGUAUUAACGGGUGA
```

Sample Output

```
MAMAPRTEINSTRING
```

Intuition

Translating an RNA string to a protein string is conceptually similar to transcribing a DNA string into an RNA string. The key difference is that we iterate over the RNA string three bases as a time, rather than one. Three consecutive bases represent codons that code for a specific amino acid. To map each codon to an amino acid, a dictionary containing codons as keys and amino acids as values is needed.

In python, we can iterate over fixed intervals by specifying the `step` argument in the `range` function. For example, `range(0, 10, 2)` iterates all indices starting from 0 to 10 (non-exclusive) at increments of 2.

Solution

We first create a dictionary that maps each codon to an amino acid:

```
codon_to_aa = {
    "UUU": "F", "UUC": "F", "UUA": "L", "UUG": "L", "UCU": "S", "UCC": "S",
    "UCA": "S", "UCG": "S", "UAU": "Y", "UAC": "Y", "UGU": "C", "UGC": "C",
    "UGG": "W", "CUU": "L", "CUC": "L", "CUA": "L", "CUG": "L", "CCU": "P",
    "CCC": "P", "CCA": "P", "CCG": "P", "CAU": "H", "CAC": "H", "CAA": "Q",
    "CAG": "Q", "CGU": "R", "CGC": "R", "CGA": "R", "CGG": "R", "AUU": "I",
    "AUC": "I", "AUA": "I", "AUG": "M", "ACU": "T", "ACC": "T", "ACA": "T",
    "ACG": "T", "AAU": "N", "AAC": "N", "AAA": "K", "AAG": "K", "AGU": "S",
    "AGC": "S", "AGA": "R", "AGG": "R", "GUU": "V", "GUC": "V", "GUA": "V",
    "GUG": "V", "GCU": "A", "GCC": "A", "GCA": "A", "GCG": "A", "GAU": "D",
    "GAC": "D", "GAA": "E", "GAG": "E", "GGU": "G", "GGC": "G", "GGA": "G",
```

```
    "GGG": "G", "UAA": "*", "UAG": "*", "UGA": "*"
}
```

Then create a function that will iterate over the RNA string three bases at a time. For each iteration, check if the current triplet is a stop codon. If it is a stop codon, terminate the loop and return the peptide sequence. Otherwise, continue to the next iteration.

```
def translate(seq: str, alphabet: str='ACGT') -> str:
    """Convert the input RNA string to an protein string."""
    peptide = ""
    stop_codons = ["UAA", "UAG", "UGA"]
    for i in range(0, len(seq), 3):
        codon = seq[i:i+3]
        if codon in stop_codons:
            break
        peptide += codon_to_aa[codon]
    # Return peptide even if no stop codon is encountered.
    return peptide
```

```
seq = "AUGGCCAUGGCGCCCAGAACUGAGAUCAAUAGUACCCGUAUUAACGGGUGA"
result = translate(seq)
print(result)
```

```
MAMAPRTEINSTRING
```