

Complementing a Strand of DNA

Jan Emmanuel Samson

2024-07-28

Problem

In DNA strings, symbols 'A' and 'T' are complements of each other, as are 'C' and 'G'.

The reverse complement of a DNA string s is the s^c formed by reversing the symbols of s , then taking the complement of each symbol (e.g., the reverse complement of "GTCA" is "TGAC").

- **Given:** A DNA string s of length at most 1000 nt.
- **Return:** The reverse complement s^c of s .

Sample Dataset

```
AAAACCCGGT
```

Sample Output

```
ACCGGGTTTT
```

Intuition

A straightforward approach is to use a dictionary that maps each base of the alphabet 'ACGT' to its complement. Create a new empty string rc and iterate over the sequence. For each base, **prepend** the complement of the base to string rc . The base complement is added at the beginning of the string since we are interested in the *reverse complement*. Return string rc once the loop has reached its end.

Solution

```
def reverse_complement(seq: str) -> str:
    """Compute the reverse complement of a DNA string."""
    rc = ""
    complement_map = {'A': 'T', 'T': 'A', 'G': 'C', 'C': 'G', 'N': 'N'}
    for base in seq.upper():
        rc = complement_map[base] + rc
    return rc
```

```
seq = "AAAACCCGGT"
result = reverse_complement(seq)
print(result)
```

```
ACCGGGTTTT
```