Counting DNA Nucleotides

Jan Emmanuel Samson

2024-07-28

Problem

A string is simply an ordered collection of symbols selected from some **alphabet** and formed into a word; the **length** of a string is the number of symbols that it contains. The DNA alphabet can be represented as $\{A, C, G, T\}$. A string is a valid DNA sequence if each element of the string is a member of the DNA alphabet set.

An example of a length 21 DNA string is `ATGCTTCAGAAAGGTCTTACG`.

- **Given**: A DNA string s of length at most 100 nt.
- **Return**: Four integers (separated by spaces) counting the respective number of times that the symbol 'A', 'C', 'G', and 'T' occurs in s.

Sample Dataset

```
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC
```

Sample Output

```
20 12 17 21
```

Intuition

Represent the DNA sequence as a string and initialize an empty dictionary with unique bases as keys with a value of zero. Iterate over the sequence one base at a time. If the current base is not a dictionary key, then it is a non-canonical base which should be ignored. Otherwise, index the dictionary by the current base and increment its value by 1.

Solution

```python
def count_nucleotides(seq: str, alphabet: set={'A','C','G','T'}) -> dict[str, int]:
    """Return the counts of A, C, G, and T in a DNA sequence."""
    counts = {base: 0 for base in alphabet}
    for base in seq.upper():
        if base in counts:
            counts[base] += 1
    return counts.values()
```

```python
seq = "AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC"
result = count_nucleotides(seq)
print(*result)
```

```
20 21 17 12
```

Bibliography