Finding a Motif in DNA

Jan Emmanuel Samson

2024-07-30

Problem

Given two strings s and t, t is a substring of s if t is contained as a contiguous collection of symbols in s (as a result, t must be no longer than s).

The position of a symbol in a string is the total number of symbols found to its left, including itself (e.g., the positions of all occurrences of 'U' in "AUGCUUCAGAAAGGUCUUACG" are 2, 5, 6, 15, 17, and 18). The symbol at position i of s is denoted by $s[i]$.

A substring of s can be represented as $s[j:k]$, where j and k represent the starting and ending positions of the substring in s; for example, if $s =$ "AUGCUUCAGAAAGGUCUUACG", then $s[2:5] =$ "UGCU".

- **Given**: A two DNA strings s and t (each of length at most 1 kbp).
- **Return**: All locations of t as a substring of s.

Sample Dataset

```
GATATATGCATATACTT
ATAT
```

Sample Output

```
2 4 10
```

Intuition

The idea is to create a window of length $|s|$ and slide that window across sequence t to get a slice of t. If the current slice is the same as sequence s, we append the starting index of the window to our growing list of indices. This results in $(st - t^2 + t)$ comparisons which is equivalent to $O(nm)$ complexity.

Solution

```python
def motif_search(s: str, t: str) -> list[int]:
    """Return a list of indices where `t` occurs as a substring of `s`."""
    pos = []
    k = len(t)
    for i in range(len(s)-k+1):
        if t == s[i:i+k]:
            pos.append(i+1)
    return pos
```

```python
s = "GATATATGCATATACTT"
t = "ATAT"
```

```
result = motif_search(s, t)
print(*result)
```

```
2 4 10
```