Counting Point Mutations

Jan Emmanuel Samson

2024-07-29

Problem

Given two strings *s* and *t* of equal length, the **Hamming distance** between *s* and *t*, denoted $d_H(s, t)$, is the number of corresponding symbols that differ in *s* and *t*.

- **Given**: Two DNA strings *s* and *t* of equal length (not exceeding 1 kbp).
- **Return**: The Hamming distance $d_H(s, t)$

Sample Dataset

```
GAGCCTACTAACGGGAT
CATCGTAATGACGGCCT
```

Sample Output

```
7
```

Intuition

The Hamming distance function requires two strings of equal length, thus we first assert that $|s| = |t|$ where $|s|$ denotes the length of string *s* and $|t|$ the length of string *t*. We initialize a variable `mismatches` to zero; this will be used for keeping track of the number of mismatches as we compare the base of each string at a specific index. As with other string algorithms, we iterate over the indices from 0 up to (but not including) $|s|$. If $s_i = t_i$, the value of `mismatches` is incremented by one. Otherwise we proceed to the following index for the next comparison.

Solution

```python
def hamming_distance(s: str, t: int) -> int:
    """Compute the number of index-specific mismatches between two strings."""
    assert len(s) == len(t), "Both strings must be of equal length"
    mismatches = 0
    for i in range(len(s)):
        if s[i] != t[i]:
            mismatches += 1
    return mismatches
```

```python
s = "GAGCCTACTAACGGGAT"
t = "CATCGTAATGACGGCCT"
result = hamming_distance(s, t)
print(result)
```

```
7
```