BC205: Algorithms for Bioinformatics.

Exercise 1. Introduction and Sequence Analysis

Mourouzidou Eleni

1. Finding palindromes in a sequence using recursion

The function **palindrome_subsequence** takes a sequence string as an input.

First, a list comprehension is created which generates all the possible 3-mer substrings that are represented in the input sequence. Inside the list comprehension, the conditional if, checks if each of these substrings generated in the loop, are equal to its reverse.

If a palindrome substring with length > 3 is found successfully, it is added to a list called **pal_list**.

If **pal_list** is not empty, which means that at least one palindrome substring is found, then the function returns **True**, and the palindrome with the maximum length.

Else, if **pal_list** is empty, no palindromes of length > 3 were found in the input sequence, and the function will return **False**.

In order to cross check the results, we can try different inputs of small sequences and check whether the results are the expected.

```
# create a function that will take a sequence string as input and return
# True or False depending on whether it contains a palindrome substring and if
so, will also return the longest substring

def palindrome_subsequence(sequence):
  #scan the sequence string with step = 2 to exclude 2-mers
  pal_list = [sequence[i:j+1] for i in range(len(sequence))
       for j in range(i+2, len(sequence))

       if sequence[i:j+1] == sequence[i:j+1][::-1]]

  # if palindrome substrings were found, return True and the biggest one
  if pal_list:
      p = max(pal_list)
      return True, p
  else:
  # if pal_list is empty, sequence has no palindrome substrings, return False

      return False


# check for palindromes in the sequence 'TTCAGGTGG'

print(palindrome_subsequence('TTCAGGTGG'))        # output True, GGTGG

#check for palindromes in the sequence 'CTGTTATTAATTATTGCAT'

print(palindrome_subsequence('CTGTTATTAATTATTGCAT')) # ouput TRUE TTATTAATTATT

#check for palindromes in the sequence 'GGCATCGGATTCGT'
print(palindrome_subsequence('GGCATCGGATTCGT'))     # output False
```

The output of the given examples, is as we would expect:

- TTCAGGTGG : As we can see, this sequence contains these palindromes > 3 nt GGTGG', 'GTG'

- TTCACGTAAGTGG : Contains the following substrings
  TGT', 'TAAT', 'ATTA', 'GTTATTAATTATTG', 'TTATTAATTATT', 'TATTAATTAT', 'TAT', 'TTAATT', 'TTATT', 'ATTAATTA' .
  In this sequences there are also many 2nt long palindrome substrings but they were excluded.

- GGCATCGGATTCGT : This sequence does not contain any palindrome subsequence with length bigger than 2, thus the output is False, as expected.