import sys

from Bio import SeqIO

def find\_promoter(dna\_sequence, promoter\_sequence):

"""Finds the promoter sequence in a DNA sequence using Biopython.

Args:

dna\_sequence: The DNA sequence to search.

promoter\_sequence: The promoter sequence to find.

Returns:

The start position of the promoter sequence in the DNA sequence, or -1 if the

promoter sequence is not found.

"""

promoter\_seq = Seq(promoter\_sequence)

for i in range(len(dna\_sequence) - len(promoter\_sequence) + 1):

if dna\_sequence[i:i + len(promoter\_sequence)] == promoter\_seq:

return i

return -1

def find\_start\_codon(dna\_sequence, promoter\_start\_position):

"""Finds the start codon in a DNA sequence using Biopython.

Args:

dna\_sequence: The DNA sequence to search.

promoter\_start\_position: The start position of the promoter sequence.

Returns:

The start position of the start codon in the DNA sequence, or -1 if the

start codon is not found.

"""

for i in range(promoter\_start\_position + 30, len(dna\_sequence)):

if dna\_sequence[i:i + 3] == "ATG":

return i

return -1

def find\_gene(dna\_sequence, promoter\_sequence):

"""Finds the gene in a DNA sequence using Biopython.

Args:

dna\_sequence: The DNA sequence to search.

promoter\_sequence: The promoter sequence to find.

Returns:

True if a gene is found, False otherwise.

"""

promoter\_start\_position = find\_promoter(dna\_sequence, promoter\_sequence)

if promoter\_start\_position == -1:

return False

start\_codon\_start\_position = find\_start\_codon(dna\_sequence, promoter\_start\_position)

if start\_codon\_start\_position == -1:

return False

for i in range(start\_codon\_start\_position + 3, len(dna\_sequence)):

if dna\_sequence[i:i + 3] == "ATG":

return False

return True

if \_\_name\_\_ == "\_\_main\_\_":

promoter\_sequence = sys.argv[1]

with open("dna\_sequence.fasta") as f:

dna\_sequence = SeqIO.read(f, "fasta")

gene\_found = find\_gene(dna\_sequence.seq, promoter\_sequence)

if gene\_found:

print("Gene Found")

else:

print("Gene Not Found")

**Code interpretation:**

**The sys and Bio modules from biopython are first loaded. The promoter sequence is obtained from the instruction line inputs using the sys module, and the DNA sequence is read from a FASTA file using the bio module.**

**The promoter sequence is initially converted into a Seq object via the "find\_promoter()" function. The DNA sequence is then iterated through in search of an identical match to the promoter sequence. The function provides the start location of the promoter sequence while a match has been identified. The function returns -1 in all other cases.**

**The DNA sequence is iterated over by the "find\_start\_codon()" function to locate the "start codon (ATG)". The function outputs the start position of the start codon if an identity is discovered. The function returns -1 in all other cases.**

**To determine the start location of the promoter sequence, the "find\_gene()" function first invokes the "find\_promoter()" function. The function returns False in the absence of the promoter sequence. If not, the method uses the "find\_start\_codon()" function to determine the start codon's position. The function returns False while the start codon cannot be identified. If not, the function determines if at least 50 subsequent amino acids must come after the following start codon. The function returns True if there exist. The function returns False if it is the case.**

**Once the program is executed as a script, the if \_\_name\_\_ == "\_\_main\_\_": block is used to run the function. The promoter\_sequ variable is assigned to the first command line parameter in this loop. Next, the a.fasta file's DNA sequence is read. The function "find\_gene()" is then used to locate the gene in the DNA sequence. The message "Gene Found" is printed to the console whenever a gene is discovered. If not, the console prints "Gene Not Found" on the screen.**