DNA and mRNA Base Pair Python Program

By: Mildred Monsivais

September 20, 2019



Table of Contents

DNA to mRNA Function	3
mRNA to DNA Function	4
User Input While Loop	5
Results	7

DNA to mRNA Function

```
1. def dna_to_mrna(sequence):
2.
        mrna_list = []
        sequence_list = list(sequence)
3.
        for dna nucleotide in sequence list:
4.
            if dna_nucleotide == 'G':
5.
                mrna list.append('C')
6.
            if dna nucleotide == 'T':
7.
                mrna list.append('A')
8.
9.
            if dna nucleotide == 'A':
                mrna list.append('U')
10.
            if dna_nucleotide == 'C':
11.
                mrna_list.append('G')
12.
13.
        return mrna_list
14.
```

The overall concept is to create a function that will return the complementary base pair during transcription. The user can enters a DNA or mRNA sequence to return the given base pair. In order to create a function you need the keyword def followed by the name of the function and an argument inside the parenthesis. The name of this function is dna_to_mrna and the argument is a sequence. A sequence is a list of any objects where the variables can be changed. You can define the list with any given name however, its easier for other programmers to read code that is relevant to the function. I named the first function that will transcribe DNA into mRNA as dna_to_mrna due to the output of mRNA nucleotides in the function. The second line of code indicates an empty list called mrna_list which allow the importation of any character. I used a for loop to insert the 4 DN nucleotide bases which include A,G,T,C. A for loop is necessary to use to allow the output of an item that is paired up in the list. For example, guanine always pairs up with cytosine so a for loop allows the output of either one of the pair. A for loop with an if statement lets the function return an object in that list that is paired with another object. Line 5 is defining the deoxyribose guanine in the sequence a simple way to add a variable would be to append the object in the list. The return statement is used when the function is ready to return a value.

mRNA to DNA Function

```
1. def mrna_to_dna(sequence):
2.3.
       dna_list = []
        sequence_list2 = list(sequence)
4.
        for mrna_nucleotide in sequence_list2:
5.
            if mrna_nucleotide == 'C':
6.
                dna_list.append('G')
7.
            if mrna_nucleotide =='U':
8.
                dna_list.append('A')
            if mrna_nucleotide == 'G':
9.
                dna_list.append('C')
10.
            if mrna_nucleotide == 'A':
11.
12.
                dna_list.append('T')
```

Another function is needed to reverse transcribe mRNA templates into DNA. The technical

skills would be exactly the same process as the function dna_to_mrna.

User Input While Loop

```
1. while True:
2.
3.
       choice = input("Do you want to transcribe or retrotranscribe, type t for transcribe
    and r for retrotranscribe?")
       sequence = input('Type in the sequence here:').upper()
4.
5.
6.
       if choice == 't':
7.
           print(dna to mrna(sequence))
           rerun = input("Would you like to type in another sequence?")
8.
9.
           if rerun == 'yes':
10.
                continue
11.
           if rerun == 'no':
12.
                print('Thank you, see you next time')
13.
                break
14.
        if choice == 'r':
15.
           print(mrna_to_dna(sequence))
16.
           rerun = input("Would you like to type in another sequence?")
17.
18.
           if rerun == 'yes':
                continue
19.
           if rerun == 'no':
20.
                print('Thank you, see you next time')
21.
22.
                break
```

A while loop takes in a "True" statement which means that the loop is being used to iterate the same block of code over and over again if the statement resigns true. A while loop is used for the first and second function to allow the user to utilize the function interchangeable. The first step towards writing this code was to ask the user to input whether to use the mrna_to_dna or the dna_to_mrna function. To make it easier for the user to input a function, I substituted T for transcribe and R as retrotranscribe. In addition, line 4 codes for the ability to input an upper or lower case nucleotide sequence. This allows the user to have more flexibility in the way they want to include their base series. On the technical side I created two different if choices, the first step one would be to code transcribing and the other retrotranscribing. How this works is that if the user inputs the substituted variable t then the function command is to prints dna_to_mrna. To make it more user friendly I printed the statement of "would you like to type in another sequence?" to allow the user to decide if they would like to rerun either of the two functions again. The continue function in python is used to continue using the function. Line 11 is

necessary for the function to stop and in addition I added the print statement of 'Thank you see you next time' to conclude the ending of this program. The break statement on line 12 is used to terminate the loop. For the second choice I coded the same structure as the first choice. In the sense I started with substituting the command of retro transcribing into 'r'. The print statement would print the mRNA sequence into the complementary DNA base pairing. After it returns the complementary base then I placed an input function to ask the user if they would like to continue using the function. Line 20 is being used if the user doesn't want reuse either function and end the program.

Results

Ex 1: Do you want to transcribe or retrotranscribe, type t for transcribe and r for retrotranscribe?t Type in the sequence here:gact ['C', 'U', 'G', 'A'] Would you like to type in another sequence?no Thank you, see you next time

This is a screenshot of the function dna_to_mrna ran on Jupyter Notebook.

Ex 2:

Do you want to transcribe or retrotranscribe, type t for transcribe and r for retrotranscribe?r Type in the sequence here:cuga ['G', 'A', 'C', 'T'] Would you like to type in another sequence?no Thank you, see you next time

This is a screenshot of the function mrna_to_dna ran on Jupyter Notebook.

Ex 3:

Do you want to transcribe or retrotranscribe, type t for transcribe and r for retrotranscribe?t Type in the sequence here:gact ['C', 'U', 'G', 'A'] Would you like to type in another sequence?yes Do you want to transcribe or retrotranscribe, type t for transcribe and r for retrotranscribe?r Type in the sequence here:cuga ['G', 'A', 'C', 'T'] Would you like to type in another sequence?no Thank you, see you next time

This is a screenshot of the both function being used as a while loop ran on Jupyter Notebook.