2nd Report (Exercise 2)

Vouzina Olympia Dialekti

Discovering and Evaluating hidden motifs in Sequences

To solve the problem, I used the functions you gave us with some changes to make the script a bit more simple.

The following code creates a list “seq_list” that contains every sequence as an item, and a string “seq” that contains all the sequences concatenated.

```
fn = "motifs_in_sequence.fa"

   #Make a list with all the 50 sequences
   seq_list = []
   with open (fn, "r") as file:
       for line in file:
           seq_list.append(line)

   #Make a string that contains all the sequences
   seq = ""
   for string in seq_list:
       seq+=string[:-1] #until -1 to ignore the '\n'
```

The next step was to calculate the frequency of each nucleotide (A, C, G, T) in the concatenated sequence (”seq” variable). I am not sure if this is the right thing to do, or if I should calculate the frequency of each nucleotide in each sequence, individually.

```
#Find the frequences of each nucleotide in whole concatenated sequence
nucfreqs = kmers(fn,1)
```

After that, I formed a for loop for the kmers with lengths from 3 to 7. The commands included in the for loop are explained bellow.

```
for k in range(3,8):
```

Then, I created a list with a random kmer from each sequence (the length of the kmers is given by the k in the for loop above).

```
#Make a list with random kmers of length k
random_kmers=[]
for i in seq_list[:50]:
  seqs_kmers=[]
  for j in range(len(i)-k):
    seqs_kmers.append((i[j:j+k]))
  random_kmers.append(random.choice(seqs_kmers))
```

Lastly, a while loop was formed to find the motifs that had Information Content ≥ 1.8. If I≥1.8, then the length of the kmer, the motif, and the PWM were printed. The I is printed every time so that we know how many times of the calculation in the while loop.

```
I=0
while I < 1.8:

  mypwm=pwm(random_kmers)
  mypssm=pssm(mypwm, nucfreqs)
  pssmsearches = pssmSearch(mypssm, seq, 0.95)
  I=pwmEntropyInformation(pwm(pssmsearches))
  print(I)

  if I >= 1.8:
    print (f"k =",k)
    print(pssmsearches[0])
    print(mypwm)
```

Discussion:

While trying to solve the exercise I ran into some issues:

- In some calculations, the pssmSearch gave as a result an empty list, so the calculations could not continue further. I couldn’t find the reason why this happened to fix the problem.

- Also, every time I ran the code, it resulted in a different motif, and they all had somewhat high Information Content, so I think that something went wrong.
- I think that the Information Content should increase in every calculation, but this did not happen, probably because every time a new list of random kmers was created, and not a better/more suitable one with higher Information Content.

I still don’t know what went wrong and I have some questions.