

1a. As subsequências com uma mutação de distancia da subsequência "CAATTGAATAATTG" foram geradas. Dentre elas, as seguintes ocorrem apenas uma vez no cromossomo 7:

CAATAGAATAATTG
CAATTAAATAATTG
CAATTCAATAATTG
CAATTTAATAATTG
CAATTGAATGATTG
CAATTGAATATTTG
CAATTGAATAACTG
CAATTGAATAATTT

Código fonte:

```
# quando nao mutada esta subseguencia e unica no Cromossomo 7
mutated_subseq = "CAATTGAATAATTG"

subseq_occurences = fastaSeq.count(mutated_subseq)
#print("subseq_occurences " + str(subseq_occurences)) # DEBUG

possible_chars = ['A', 'C', 'G', 'T'] # characters that can appear in a DNA sequence
# test mutations to find original subsequence
for i in range(len(mutated_subseq)): # len(mutated_subseq) == 14
    # for each position in the subsequence
    for c in possible_chars:
        # mutation:
        s = list(mutated_subseq)
        s[i] = c
        subseq = ''.join(s)
        subseq_occurences = fastaSeq.count(subseq)
        print(subseq + " ... " + str(subseq_occurences) + " occurences")
```

1b.

Número de diferentes palíndromos de tamanho 9: 1025

Número de diferentes palíndromos de tamanho 11: 3713

Código fonte:

```

ocorrenciasPalindromos9 = {}
ocorrenciasPalindromos11 = {}

for i in range(len(fastaSeq)-9):
    subseq = fastaSeq[i:i+9]
    if(subseq == subseq[::-1]): # check if it is a palindrome
        if(subseq in ocorrenciasPalindromos9.keys()):
            ocorrenciasPalindromos9[subseq] += 1
        else:
            ocorrenciasPalindromos9[subseq] = 1

#print(ocorrenciasPalindromos9) # DEBUG

for i in range(len(fastaSeq)-11):
    subseq = fastaSeq[i:i+11]
    if(subseq == subseq[::-1]): # check if it is a palindrome
        if(subseq in ocorrenciasPalindromos11.keys()):
            ocorrenciasPalindromos11[subseq] += 1
        else:
            ocorrenciasPalindromos11[subseq] = 1

#print(ocorrenciasPalindromos11) # DEBUG

# quantidade de palindromos de tamanho 9 e 11
numPalindromos9 = len(ocorrenciasPalindromos9.keys())
numPalindromos11 = len(ocorrenciasPalindromos11.keys())

print("num de palindromos de tamanho 9: " + str(numPalindromos9))
print("num de palindromos de tamanho 11: " + str(numPalindromos11))

# numero de ocorencias de cada palindromo:
# informacao contida nos dicionarios ocorrenciasPalindromos9 e ocorrenciasPalindromos11

```

1c. A execução do código estava demorando mais de 20min em um computador do laboratório para alunos de graduação da INF, então decidi cancelá-la. Mesmo assim, acredito que, com tempo suficiente, o número de ocorrências de cada subsequência de tamanho 37 do cromossomo 7 seria encontrado.

Código fonte:

```

ocorrencias_subseqs37 = {}

# iterate through chromosome 7
for i in range(len(fastaSeq)-37):
    # get subsequence:
    subseq = fastaSeq[i:i+37]
    # count occurrence
    if(subseq in ocorrencias_subseqs37.keys()):
        ocorrencias_subseqs37[subseq] += 1
    else:
        ocorrencias_subseqs37[subseq] = 1

# numero de subsequencias de tamanho 37
print(len(ocorrencias_subseq37.keys()))
# numero de ocorrencias para cada subsequencia de tamanho 37
#print(ocorrencias_subseq37)

```

1d. Sim, existe um caracter diferentes na sequência: N, que aparece 2396 vezes. Os caracteres A, C, T e G aparecem 17146584, 11918693, 17132531, e 11909449 vezes, respectivamente.

```
occurencesDict = {}

# iterate through chromosome 7
for i in range(len(fastaSeq)):
    # get char:
    c = fastaSeq[i]
    # count occurence
    if(c in occurencesDict.keys()):
        occurencesDict[c] += 1
    else:
        occurencesDict[c] = 1

print(occurencesDict)
```

[illegible]

```
# construct complementary DNA
complementaryDNA = ''

for i in range(len(fastaSeq)):
    if(fastaSeq[i] == 'G'):
        complementaryDNA += 'C'
    elif(fastaSeq[i] == 'C'):
        complementaryDNA += 'G'
    elif(fastaSeq[i] == 'T'):
        complementaryDNA += 'A'
    elif(fastaSeq[i] == 'A'):
        complementaryDNA += 'T'
    elif(fastaSeq[i] == 'N'):
        complementaryDNA += 'N'

print(complementaryDNA)
```