

Luiza Schmidt 281954 Ciência da Computação

1a) A subsequência ACCTGGTGTCCCA contém uma mutação. Identifique o(s) potencial(is) candidatos a sequência original, informando a localização da subsequência, bem como a mutação identificada.

Entrada: sequência do cromossomo 21

```
if __name__ == '__main__':  
    f = open(sys.argv[1])  
  
    s = f.read().replace("\n", "") #s contém seq do cromossomo 21
```

Saída: sequência não mutada, região de localização da subsequência, mutação existente.

Na saída do programa são impressas 13 strings, que são palíndromos, e, consecutivamente, a quantidade de vezes que eles aparecem na string de entrada, no formato “s : n” sendo s string do palíndromo e n número de vezes.

```
ACCTGGTTTTCCCA posição inicial: 5082790  
T posição caracter original: 5082797  
ACCTGGTGTCCGA posição inicial: 6467967  
G posição caracter original: 6467979  
AGCTGGTGTCCCA posição inicial: 10038319  
G posição caracter original: 10038320  
ACATGGTGTCCCA posição inicial: 10399167  
A posição caracter original: 10399169  
ACCTGGTGATCCCA posição inicial: 20966057  
A posição caracter original: 20966065  
ACCTAGTGTCCCA posição inicial: 35162825  
A posição caracter original: 35162829  
ACTTGGTGTCCCA posição inicial: 35745840  
T posição caracter original: 35745842  
ACCTGGTGTCCCTA posição inicial: 36705952  
T posição caracter original: 36705964  
ACCTGGTGTTCCTA posição inicial: 37453236  
T posição caracter original: 37453246  
ACCTGGTGCTCCCA posição inicial: 42746544  
C posição caracter original: 42746552  
ACCTGGTGTCCGA posição inicial: 43076289  
G posição caracter original: 43076301  
ACCTGCTGTCCCA posição inicial: 43703887  
C posição caracter original: 43703892  
ACCTGGAGTCCCA posição inicial: 44397996  
A posição caracter original: 44398002
```

Para encontrar os palíndromos e suas quantidades, primeiramente foi criada uma variável onde a string da sequência que continha uma mutação foi salva, para que então fosse criado um laço *for* comparando pedaços de tamanho 14 da string de entrada (a sequência do cromossomo 21), utilizando a função “compara” para tal.

A função *compara* compara a string da mutação com uma string de tamanho 14.

Se o conteúdo das posições for igual então o contador “eq” é incrementado, se forem diferentes o caracter da string que não é fixa é salvo na variável “carac”.

Após terminado o laço *for* retorna o valor *verdadeiro*, se o contador tiver valor igual a 13, e o caracter divergente, se o contador tiver valor diferente de 13 retorna o valor *falso* e uma string vazia.

```
subs = 'ACCTGGTGTCCCA' # sequência que contém uma mutação

for i in range(len(s)-14):
    ts = s[i:i+14]
    torf, carac = compara(ts, subs)
    if torf:
        print(ts + ' posição inicial: ' + str(i)) #imprime todos os candidatos à seq original
        print(ts[carac] + ' posição caracter original: ' + str(i + carac))
```

Pseudocódigo:

S = sequência do cromossomo 21

Subs = 'ACCTGGTGTCCCA'

Do início de **S** até o final de **S**

Divide **S** em janelas de 14 em 14, avançando de uma em uma unidade

Compara com **subs**, caractere a caractere

Se o caractere da vez for igual

Incrementa o contador

Senão

Salva o caractere diferente

Verifica se o contador é igual a 13

Se sim

Retorna Verdadeiro e o caractere diferente

Senão

Retorna Falso e uma string vázia

Imprime a sequência original e a posição inicial

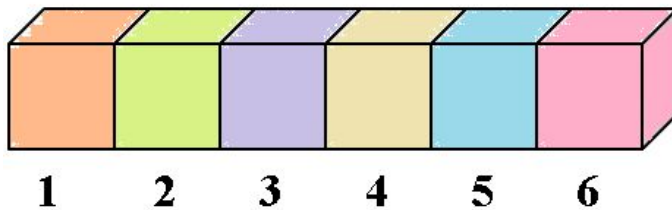
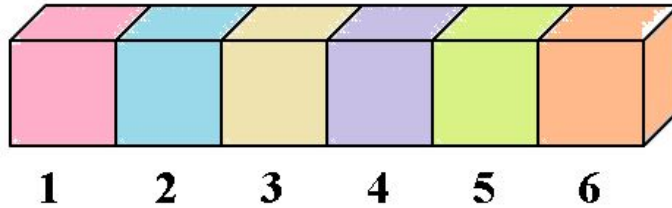
Entrada: sequência do cromossomo 21

Saída: quantidade de palíndromos de tamanho 9, número de ocorrências de cada palíndromo.

GAAGAGGAAG:	553	TCCTCTCTCT:	913	AGATAATGAT:	873	AATGAGTAA:	307	TTGTGTGTT:	405
TCAGAGACT:	269	ACCCCTCCCA:	303	ACTATAATCA:	185	ACTACTACCA:	397	GTGACAGTG:	250
CGAAAAAGG:	22	AAAGTGAAAA:	736	AGATAAATGA:	387	ACATGTACA:	352	AACTATCAAC:	215
CAAAAAAAG:	1143	GGTCCCTGG:	248	TTGTGTGTT:	405	GTGAGAGTG:	212	CACTATGAA:	421
GTCCGGGCTG:	41	GGTGTCTGG:	206	GTGACAGTG:	250	AGGTTGTGA:	197	CACCTCAAG:	428
TTGAGAGTG:	170	TTGAGAGTG:	206	AACTATAACA:	215	GTTTATGTA:	529	GGTAAATG:	143
CGAAGAACG:	41	CGCTGTCCG:	41	AGAAATAGAA:	421	CGACCCACG:	183	CATGTGTAC:	193
CCTGAGTCC:	202	CAGGAGGAC:	234	CAACTCAAC:	138	TCACCTACT:	394	CAGGGGGAC:	116
TTGTTTTGT:	2066	GTAGCGGAT:	19	AGGAAATGG:	138	TCACCCACT:	226	ACCGGGCCA:	34
GTTTTGTGT:	1172	CAAGAAAGC:	292	CATGTGTAG:	193	ATCCACTTA:	314	CGCGCCGAC:	99
CTGAGTCTCT:	165	GGTGGGTTG:	237	CAGGGGGAC:	116	TATCTACT:	154	GACCTTTCAG:	190
ACCATACAC:	113	GGTGGGTTG:	190	ACCGGGCCA:	34	ATCTCCACT:	154	AACCCCCAA:	198
CAATTTTAA:	211	AACATACAA:	473	ACCCGGCAC:	79	CTTTTCTTC:	1085	AATAGATAA:	454
CAGCCACAG:	126	GAGGGCGAG:	1230	AACCCGCCA:	198	ATTCTCTTA:	564	AACTCTCAA:	300
AGTCCCTGA:	200	GGTGGGTTA:	194	AATAGTAAA:	454	CCAGCGACC:	37	CGTGTGTGC:	48
CAGAGAGCA:	339	TAATATAAT:	3013	AATGTCTAA:	300	TGGGGGGTG:	133	TGGACAGTG:	168
GTGTGTGAG:	20	AATAAAATA:	3677	GGATTATAG:	163	ATGGAGGTA:	170	CACCTCCAC:	282
TTCCACCTT:	307	CACCCACAC:	1389	GTGATCTGC:	188	CAGGTGTGA:	200	AGACACAGA:	599
ACCGAGACCA:	127	TAAAAAAAT:	1832	TCACAGAGT:	282	TTGACAGTT:	447	TCATACATC:	657
GAACCCAAAG:	163	AAAAAGAAA:	4965	CACCTCCAC:	282	TTGCTCTGT:	257	CCCAAACCC:	260
TTGTGGGAG:	98	AGACAGAGA:	3488	AGACACAGA:	599	TAATCACTA:	254	TGACTCAGT:	230
CGGCGGCG:	1244	GTGGCGGTG:	74	CTCTCTCTC:	447	CGGGCGGGC:	392	GGGTATGGT:	257
CGCGCGCG:	126	CAAAAAAAC:	757	CTCTCTCTC:	653	CGCGGGCGG:	392	GGGTATGGT:	257
CGCGCGCG:	132	CGGGTGGGC:	69	CTCTCTCTC:	239	CGCGGGCGG:	392	GGGTATGGT:	257
CATCCCTAC:	106	TTGTGAGTG:	355	CTCTCTCTC:	230	CGCGGGCGG:	392	GGGTATGGT:	257
CAGACAGAC:	251	GTGCTCTGT:	35	TGGTGTGTT:	257	CGCGGGCGG:	392	GGGTATGGT:	257
CTTATCTGT:	35	GAGACAGAC:	1745	GGGTATGGT:	76	CGCGGGCGG:	392	GGGTATGGT:	257
AAATATAAA:	1854	GTACACACT:	207	GGGTATGGT:	76	CGCGGGCGG:	392	GGGTATGGT:	257
TCGCCCACT:	70	CAGAGAGAC:	294	CTTTTCTGC:	239	CGCGGGCGG:	392	GGGTATGGT:	257
GTGGGGGGTG:	382	GAGCCACAG:	284	CTTTTCTGC:	174	CGCGGGCGG:	392	GGGTATGGT:	257
TGCTCTCGT:	58	CGGAAAGGG:	416	CTTTTCTGC:	10454	CGCGGGCGG:	392	GGGTATGGT:	257
CTTATCTGT:	35	CTCTCTCTC:	4285	CTTTTCTGC:	37	CGCGGGCGG:	392	GGGTATGGT:	257
ACGAGAGGA:	460	AGACCCAGA:	258	CTTTTCTGC:	10454	CGCGGGCGG:	392	GGGTATGGT:	257
ATAAGAATA:	436	GGAGTGAG:	247	CTTTTCTGC:	37	CGCGGGCGG:	392	GGGTATGGT:	257
CCAAGAAC:	180	GGGCCCGGG:	144	CTTTTCTGC:	37	CGCGGGCGG:	392	GGGTATGGT:	257
TACCCCCAT:	109	GGGGGAGAG:	426	CTTTTCTGC:	37	CGCGGGCGG:	392	GGGTATGGT:	257
TTCTTTTCT:	109	CTCTCTCTC:	909	CTTTTCTGC:	37	CGCGGGCGG:	392	GGGTATGGT:	257
CTCTTTTCT:	109	GTGCTCTGT:	118	CTTTTCTGC:	37	CGCGGGCGG:	392	GGGTATGGT:	257
CGGTTGGGA:	332	AGCCCTTGA:	192	CTTTTCTGC:	37	CGCGGGCGG:	392	GGGTATGGT:	257
CATCTTTCAC:	248	CATCTGTCA:	263	CTTTTCTGC:	37	CGCGGGCGG:	392	GGGTATGGT:	257
CGAGAGAGC:	112	CGCTCTCTC:	377	CTTTTCTGC:	37	CGCGGGCGG:	392	GGGTATGGT:	257
CTTATCTGT:	35	CGCTCTCTC:	377	CTTTTCTGC:	37	CGCGGGCGG:	392	GGGTATGGT:	257
GAGCGCGAG:	13	CCAGGGAC:	260	CTTTTCTGC:	37	CGCGGGCGG:	392	GGGTATGGT:	257
TCCTTATCT:	409	TCAGACACT:	191	CTTTTCTGC:	37	CGCGGGCGG:	392	GGGTATGGT:	257

A estratégia utilizada para montar esse programa foi salvar a sequência do cromossomo em uma string, dividi-la em pedaços de tamanho 9, invertê-la/espelhá-la e comparar a invertida com sua forma original verificando se o conteúdo de suas posições inversas são iguais e criar um dicionário de palavras para salvar os palíndromos e seu número de aparições

Posições opostas equivalentes seriam comparadas de acordo com as cores na imagem abaixo, 1 com 6, 2 com 5, 3 com 4.



Pseudocódigo:

Cria dicionário de palíndromos

Enquanto a string de entrada não tiver terminado

Divide ela em pedaço de tamanho 9, avançando de 1 em 1

Se a string já estiver registrada no dicionário

Incrementa sua quantidade

Senão

Adiciona string ao dicionário, com valor 1

Imprime todas os palíndromos do dicionário junto com seu valor (quantidade de vezes que aparece)

Imprime a quantidade total de palíndromos de tamanho 9 contidas na sequência

1c) Identifique as diferentes subsequências de tamanho 37, contabilizando o número de ocorrências de cada uma delas.

Entrada: sequência do cromossomo 21.

```
if __name__ == '__main__':  
    f = open(sys.argv[1])  
    s = f.read().replace("\n", "") #s contém seq do cromossomo 21
```

Saída: lista das diferentes subsequências e o número de ocorrências de cada uma delas.

```
TACGATGCAGGTTACTGTTTCTCACGTGCTTCAATC : 1  
AAGGATGCAGGTTACTGTTTCTCACGTGCTTCAATCT : 1  
AGGATGCAGGTTACTGTTTCTCACGTGCTTCAATCTT : 1  
GGATGCAGGTTACTGTTTCTCACGTGCTTCAATCTTT : 1  
GATGCAGGTTACTGTTTCTCACGTGCTTCAATCTTTA : 1  
ATGCAGGTTACTGTTTCTCACGTGCTTCAATCTTTAA : 1  
TGCAGGTTACTGTTTCTCACGTGCTTCAATCTTTAAA : 1  
GCAGGTTACTGTTTCTCACGTGCTTCAATCTTTAAAT : 1  
CAGGTTACTGTTTCTCACGTGCTTCAATCTTTAAATC : 1  
AGGTTACTGTTTCTCACGTGCTTCAATCTTTAAATCA : 1  
GGTTACTGTTTCTCACGTGCTTCAATCTTTAAATCAG : 1  
GTTACTGTTTCTCACGTGCTTCAATCTTTAAATCAGA : 1  
TTACTGTTTCTCACGTGCTTCAATCTTTAAATCAGAA : 1  
TACTGTTTCTCACGTGCTTCAATCTTTAAATCAGAAA : 1  
ACTGTTTCTCACGTGCTTCAATCTTTAAATCAGAAAA : 1  
CTGTTTCTCACGTGCTTCAATCTTTAAATCAGAAAAAT : 1  
TGTTCCTCACGTGCTTCAATCTTTAAATCAGAAAAATA : 1  
GTTTCTCACGTGCTTCAATCTTTAAATCAGAAAAATAA : 1  
TTTCTCACGTGCTTCAATCTTTAAATCAGAAAAATAAA : 1  
TTCTCACGTGCTTCAATCTTTAAATCAGAAAAATAAAC : 1  
TCTCACGTGCTTCAATCTTTAAATCAGAAAAATAAAT : 1  
CTCACGTGCTTCAATCTTTAAATCAGAAAAATAAATC : 1  
TCACGTGCTTCAATCTTTAAATCAGAAAAATAAATCA : 1  
CACGTGCTTCAATCTTTAAATCAGAAAAATAAATCTT : 1  
ACGTGCTTCAATCTTTAAATCAGAAAAATAAATCTTC : 1  
CGTGCTTCAATCTTTAAATCAGAAAAATAAATCTTCT : 1  
GTGCTTCAATCTTTAAATCAGAAAAATAAATCTTCTG : 1  
TGTCTTCAATCTTTAAATCAGAAAAATAAATCTTCTGC : 1  
GCTTCAATCTTTAAATCAGAAAAATAAATCTTCTGCT : 1  
CTTCAATCTTTAAATCAGAAAAATAAATCTTCTGCTG : 1  
TCAATCTTTAAATCAGAAAAATAAATCTTCTGCTGA : 1  
TCAATCTTTAAATCAGAAAAATAAATCTTCTGCTGAA : 1  
CAATCTTTAAATCAGAAAAATAAATCTTCTGCTGAAG : 1  
AATCTTTAAATCAGAAAAATAAATCTTCTGCTGAAGT : 1  
ATCTTTAAATCAGAAAAATAAATCTTCTGCTGAAGTA : 1  
TCTTTAAATCAGAAAAATAAATCTTCTGCTGAAGTAC : 1  
CTTTAAATCAGAAAAATAAATCTTCTGCTGAAGTACA : 1  
TTTAAATCAGAAAAATAAATCTTCTGCTGAAGTACAT : 1  
TTAAATCAGAAAAATAAATCTTCTGCTGAAGTACATT : 1  
TAAATCAGAAAAATAAATCTTCTGCTGAAGTACATTA : 1  
AATCAGAAAAATAAATCTTCTGCTGAAGTACATTAA : 1  
AATCAGAAAAATAAATCTTCTGCTGAAGTACATTAA : 1  
ATCAGAAAAATAAATCTTCTGCTGAAGTACATTAA : 1  
TCAGAAAAATAAATCTTCTGCTGAAGTACATTAA : 1  
CAGAAAAATAAATCTTCTGCTGAAGTACATTAA : 1  
AGAAAAATAAATCTTCTGCTGAAGTACATTAA : 1  
GAAAAATAAATCTTCTGCTGAAGTACATTAA : 1  
AAAAATAAATCTTCTGCTGAAGTACATTAA : 1  
AAATAAATCTTCTGCTGAAGTACATTAA : 1  
AATAAATCTTCTGCTGAAGTACATTAA : 1  
ATAAATCTTCTGCTGAAGTACATTAA : 1  
TAAATCTTCTGCTGAAGTACATTAA : 1  
AAATCTTCTGCTGAAGTACATTAA : 1  
AACTTCTGCTGAAGTACATTAA : 1  
ACTTCTGCTGAAGTACATTAA : 1  
CTTCTGCTGAAGTACATTAA : 1  
TCTGCTGAAGTACATTAA : 1  
TCTGCTGAAGTACATTAA : 1  
TGCTGAAGTACATTAA : 1  
TCTGAAGTACATTAA : 1  
CTGAAGTACATTAA : 1  
TGAAGTACATTAA : 1  
GAAGTACATTAA : 1  
AGTACATTAA : 1  
AGTACATTAA : 1  
GTACATTAA : 1  
TACATTAA : 1  
ACATTAA : 1  
CATTAA : 1  
ATTAACAAAGCTCCATGAACATCTCATTAGCCAGAAG : 1  
TTAACAAAGCTCCATGAACATCTCATTAGCCAGAAGT : 1  
TAACAAAGCTCCATGAACATCTCATTAGCCAGAAGTG : 1  
AACAAAGCTCCATGAACATCTCATTAGCCAGAAGTGT : 1  
ACAAAGCTCCATGAACATCTCATTAGCCAGAAGTGT : 1  
CAAAGCTCCATGAACATCTCATTAGCCAGAAGTGTTC : 1  
AAAGCTCCATGAACATCTCATTAGCCAGAAGTGTTC : 1  
AAGCTCCATGAACATCTCATTAGCCAGAAGTGTTCAC : 1  
AGCTCCATGAACATCTCATTAGCCAGAAGTGTTCACA : 1  
GCTCCATGAACATCTCATTAGCCAGAAGTGTTCACAC : 1
```

Pseudocódigo:

Cria dicionário para sequências de tamanho 37

Enquanto a string de entrada não tiver terminado

 Divide ela em pedaço de tamanho 37, avançando de 1 em 1

 Se a string já estiver registrada no dicionário

 Incrementa sua quantidade

 Senão

 Adiciona string ao dicionário, com valor 1

Imprime todas as sequências do dicionário junto com seu valor (quantidade de vezes que aparece)

1d) Contabilize o número de ocorrências de cada um dos quatro nucleotídeos. Estime o GC%. Existe um caractere diferente na sequência?

Entrada: sequência do cromossomo 21.

```
if __name__ == '__main__':  
    f = open(sys.argv[1])  
  
    s = ''  
    for line in f.readlines()[1:]:  
        s += line #pula essa linha  
                #>NC_000021.9 Homo sapiens chromosome 21, GRCh38.p7 Primary Assembly  
  
    s = s.replace("\n", "") #s contém seq do cromossomo 21
```

Saída: Número de ocorrências para cada nucleotídeo, GC%, “sim” ou “não” para caractere diferente e identificar, caso exista.

```
N: 6621361  
G: 8226381  
A: 11820664  
T: 11856330  
C: 8185244  
M: 2  
R: 1  
16411625  
percentual de GC: 35.13515515516244%  
caracteres diferentes de 'A' 'T' 'C' ou 'G':  
N  
M  
R
```

Pseudocódigo:

Cria dicionário para bases

Enquanto a string de entrada não tiver terminado

Divide ela em pedaço de tamanho 1, avançando de 1 em 1

Se o caractere já estiver registrado no dicionário

Incrementa sua quantidade

Senão

Adiciona ao dicionário, com valor 1

Imprime cada caractere e seu número de ocorrências

Soma o número de ocorrências de G com C e divide pelo número total de bases

Imprime o percentual de GC

Imprime os caracteres diferentes