Luiza Schmidt 281954 Ciência da Computação

1a) A subsequência ACCTGGTGTTCCCA 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 óriginal: 5082797
ACCTGGTGTTCCGA posição inicial: 6467967
G posição caracter original: 6467979
AGCTGGŤGTTCCCA posição inicial: 10038319
G posição caracter original: 10038320
ACATGGTGTTCCCA posição inicial: 10399167
A posição caracter original: 10399169
ACCTGGTGATCCCA posição inicial: 20966057
A posição caracter óriginal: 20966065
ACCTAGTGTTCCCA posição inicial: 35162825
A posição caracter original: 35162829
ACTTGGŤGTTCCCA posição inicial: 35745840
T posição caracter original: 35745842
ACCTGGTGTTCCTA posição inicial: 36705952
T posição caracter óriginal: 36705964
ACCTGGŤGTTTCCA posição inicial: 37453236
T posição caracter original: 37453246
ACCTGGŤGCTCCCA posição inicial: 42746544
 posição caracter óriginal: 42746552
ACCTGGTGTTCCGA posição inicial: 43076289
G posição caracter original: 43076301
ACCTGCŤGTTCCCA posição inicial: 43703887
C posição caracter original: 43703892
ACCTGGÁGTTCCCA posição inicial: 44397996
A posição caracter óriginal: 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 = 'ACCTGGTGTTCCCA' # 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 = 'ACCTGGTGTTCCCA'
```

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

caractere da vez for igua

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

1b) Um palíndromo é uma palavra, frase ou qualquer outra sequência de unidades (como uma cadeia de DNA, Enzima de Restrição, etc.) que tenha a propriedade de poder ser lida da mesma maneira da direita para a esquerda e vice-versa. Determine quantos diferentes palíndromos de tamanho 9 podem ser encontrados.

Entrada: sequência do cromossomo 21

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

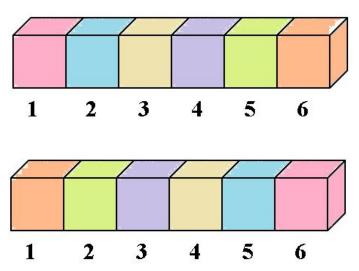
Saída: quantidade de palíndromos de tamanho 9, número de ocorrências de cada palíndromo.

Retorna 1021 strings de palíndromos diferentes e a quantidade de vezes que cada um aparece na sequência do cromossomo.

GAAGAGAAG: 553	TCTCGCTCT: 913	TGAAAAAGT: 873 ACTATATCA: 185	AATGAGTAA: 307	HIGHGIGHT: 405
TCAGAGACT: 269	ACCCTCCCA: 303	ACTATATCA: 185	ACTCACTCA: 397	GTGACAGTG: 250
CGAAAAAGC: 22	AAAGTGAAA: 736	AGTAAATGA: 387	ACATGTACA: 352	AACTATCAA: 215
GAAAAAAAG: 1143	GGTCCCTGG: 248	TTGTGTGTT: 405	GTGAGAGTG: 212	AGAATAAGA: 421
GTCGGGCTG: 41	GGTCTCTGG: 206	GTGACAGTG: 250	AGGTGTGGA: 197	CAACTCAAC: 138
GTCGGGCTG: 41 TGAGGGAGT: 170	GGTCTCTGG: 206 TTAAGAATT: 484	AACTATCAA: 215	GTTTATTTG: 520	GGTAAATGG: 143
GCAAGAACG: 41	GCCTGTCCG: 41	AGAATAAGA: 421	CCACCCACC: 853	CATGTGTAC: 193
	CAGGAGGAC: 234	CAACTCAAC: 138	TCACTCACT: 394	CAGGGGGAC: 116
CCTGAGICC: 202 TTGTTTGTT: 2066 GTTGTTTGT: 1172 TCTCACTCT: 1365 ACCATACCA: 113 CAATTTAAC: 211 GACCACCAG: 126 AGTCCCTGA: 200 AATGTGTAA: 369 CCTGCGTGC: 20 TTCCACCTT: 307 ACCAGAGCCA: 127	GTAGCGATG: 19 CAGAAAGAC: 292 TTGGGGGTT: 187 GGTGGGTGG: 790	GGTAAATGG: 143	TCACTCACT: 394	ACCGGGCCA: 34 CACCGCCAC: 79
GITTGTTTG: 1172	CAGAAAGAC: 292	CATGTGTAC: 193	TCACCCACT: 226	CACCGCCAC: 79
TCTCACTCT: 1265	TTGGGGGTT: 187	CAGGGGGAC: 116	ATTCACTTA: 314	GACTTTCAG: 190
ACCATACCA: 112	GGTGGGTGG: 790	CAGGGGGAC: 116 ACCGGGCCA: 34 CACCGCCAC: 79	TACCTCCAT: 154	AACCCCCAA: 198
CAATTTAAC. 211	AACATACAA: 473	CACCGCCAC: 79	CTTTTTTC: 1085	AATAGATAA: 454
CAGGAGGAG: 136	CACCCCCAC: 1230	GACTTTCAG: 190	GTCTCTCTG: 383	AACTCTCAA: 300
ACTCCCTCA: 200	GAGGCGGAG: 1230 AGTGGGTGA: 194 TAAATAAAT: 3013	AACCCCCAA: 198	ATTCCCTTA: 564	GGATTTAGG: 163
AGTCCCTGA: 200	TAAATAAAT. 2012	AATAGATAA: 454	CCAGCGACC: 37	GCTGTGTCG: 48
AATGIGIAA: 369	AATAAATAA: 3677	AACTCTCAA: 300	TGGGGGGGT: 133	TGGACAGGT: 168
CGTGCGTGC: 20	CACCACCAC: 1300	GGATTTAGG: 163	ATGGAGGTA: 170	CACCTCCAC: 282
TICCACCIT: 307	CACCACCAC: 1389 TAAAAAAAT: 1832	GCTGTGTCG: 48	CAGGTGGAC: 200	CACCTCCAC: 282 AGACACAGA: 599
ACCAGACCA: 127	TAAAAAAAT: 1652	TGGACAGGT: 168	CTCAGACTC: 264	CCTTCTTCC: 447
GAACCCAAG: 163	AAAAGAAAA: 4965 AGAGAGAGA: 3488 GTGGCGGTG: 74 CAAAAAAAC: 757	CACCTCCAC: 282 AGACACAGA: 599	TTGCACGTT: 44	TACATACAT: 653
GITGGGITG: 98	AGAGAGAGA: 3488	AGACACAGA: 599	GTCCTCCTG: 257	CCCAAACCC: 260
GAGGTGGAG: 1244	GIGGCGGIG: 74	CCTTCTTCC: 447		TGACTCAGT: 230
CGCGCGCGC: 126	CAAAAAAC: /5/	TACATACAT: 653	AATCACTAA: 254	TECTETECT. 257
GAACCCAAG: 163 GTTGGGTTG: 98 GAGGTGGAG: 1244 CGCGCGCGCG: 126 GCGCGCGCGCG: 132	CGGGTGGGC: 69	CCCAAACCC: 260 TGACTCAGT: 230	CGGGCGGC: 73	TGGTGTGGT: 257
CATCCCTAC: 106	TGTGAGTGT: 355	TGACTCAGT: 230	GGCGGGCGG: 392	GGGTATGGG: 76
CAGACAGAC: 251	CGTCGCTGC: 12 GAGACAGAG: 1745	TGGTGTGGT: 257	TCTCCCTCT: 730	GGCTTTCGG: 29
TGCTGTCGT: 35	GAGACAGAG: 1745	GGGTATGGG: 76	AAAACAAAA: 4232	CTTGGGTTC: 174
CATCCCTAC: 106 CAGACAGAC: 251 TGCTGTCGT: 35 AAATATAAA: 1854 TCCCGCCCT: 70	GTCACACTG: 207	GGCTTTCGG: 29	TAATCTAAT: 267	TGTGTGTGT: 10454
TCCCGCCCT: 70	CAGAGAGAC: 294 GGACCCAGG: 284	CTTGGGTTC: 174 TGTGTGTGT: 10454	CTCCCCCTC: 278	CGTTTTTGC: 37
	GGACCCAGG: 284	TGTGTGTGT: 10454	TGACACAGT: 212	GCTCTCTCG: 43
TGCCTCCGT: 58	GGGAAAGGG: 416	CGTTTTTGC: 37	GGCTATCGG: 10	CGCGGGCGC: 39
CTTTATTTC: 748	CTCTCTCTC: 4285	GCTCTCTCG: 43	AAATCTAAA: 661	CAGCGCGAC: 7
TGCCTCCGT: 58 CTTTATTTC: 748 AGGAGAGGA: 460 ATAAGAATA: 436	GGGAAAGGG: 416 CTCTCTCTC: 4285 AGACCCAGA: 258	CGCGGGCGC: 39	TAGGAGGAT: 167	CTCGTGCTC: 48
ATAAGAATA: 436	GGAGTGAGG: 247	CAGCGCGAC: 7	TAGGAGGAT: 167 TATCTCTAT: 288	ATCCTCCTA: 179
CCAAGAACC: 180	GGGCCCGGG: 144	CTCGTGCTC: 48	CCTAAATCC: 160	AGCCGCCGA: 7
CCAAGAACC: 180 TACCCCCAT: 109	GGGAGAGGG: 490	ATCCTCCTA: 179	ACCAAACCA: 232	TTTGGGTTT: 531 CAACGCAAC: 18
TTCCCCCTT: 329	CTCTTTCTC: 909	AGCCGCCGA: 7		CAACGCAAC: 18
TCTCTCTCT: 5158	GTCCCCCTG: 118	TTTGGGTTT: 531	TTTCCCTTT: 813	TTACCCATT: 211
AGGGTGGGA: 332	GTCCCCCTG: 118 AGGTCTGGA: 192	CAACGCAAC: 18	AGGACAGGA: 309	TCCGTGCCT: 57
AGGGTGGGA: 332 CACTTTCAC: 248	CACTGTCAC: 263	TTACCCATT: 211	GGCAGACGG: 40	AATCCCTAA: 152 ATAATAATA: 1821
GCAGAGACG: 112	CACCCCCAC: 377	TCCGTGCCT: 57	GTCTGTCTG: 443	ATAATAATA: 1821
GACGTGCAG: 52	CACCCCCAC: 377 GCCTCTCCG: 61 CCAGGGACC: 260	AATCCCTAA: 152	TCTTCTTCT: 939	CCAGTGACC: 151
GACGTGCAG: 52 GAGCGCGAG: 13 TCTTATTCT: 409 TGCCACCGT: 73	CCAGGGACC: 260	AATCCCTAA: 152 ATAATAATA: 1821 CCAGTGACC: 151	GCCCTCCCG: 77	AGCCACCGA: 52
TCTTATTCT: 409	TCCAGACCT: 191	CCAGTGACC: 151	GAAGGGAAG: 506	AGTAGATGA: 157 GATCACTAG: 70 GGTGAGTGG: 300
TCCCACCCT: 72	GGGTCTGGG: 248	AGCCACCGA: 52	GTTCTCTTG: 198	GATCACTAG: 70
GGCCACCGG: 45	GGGTCTGGG: 248	AGTAGATGA: 157 GATCACTAG: 70 GGTGAGTGG: 300	CGTGTGTGC: 122	GGTGAGTGG: 300
AAGTGTGAA: 284	TCTTTTTCT: 1862	GATCACTAG: 70	CCTGTGTCC: 501	GGATCTAGG: 88
TCTCCCTCT. 20	TCTTTTTCT: 1862 AACAAACAA: 2137	GGTGAGTGG: 300	TCCACACCT: 229	ATGACAGTA: 178
TGTCGCTGT: 30	CAAACAAAC: 1245	GGATCTAGG: 88	TGCCCCCGT: 43	CATTITIAC: 445
GGCATACGG: 28 CGGAAAGGC: 27	AAGAAAGAA: 3454	GGATCTAGG: 88 ATGACAGTA: 178	TGTGGGTGT: 369	AGGTTTGGA: 188
CGGAAAGGC: 27 GGGTTTGGG: 212	AAGAAAGAA: 3454 GAAAGAAAG: 2499	CATTITTAC: 445 AGGTTTGGA: 188 TGAACAAGT: 158	GTGCCCGTG: 51	TCAACAAGT: 158
GGGCGCGGG: 212	ACGGTGGCA: 66	AGGTTTGGA: 188	GAATCTAAG: 124	TGGTCTGGT: 126 ATTGGGTTA: 144
GGGCGCGGG: 90	TGCTCTCGT: 47	TGAACAAGT: 158	TAAGGGAAT: 188	ATTGGGTTA: 144
TCAACAACT: 184	TGCTCTCGT: 47 ATAAAAATA: 2380		CTTCTCTTC: 171	GTGGAGGTG: 268
GIGAAAGIG: 213	AATACATAA: 555	ATTGGGTTA: 144	CTTGTGTTC: 171 CTGGAGGTC: 212	TGGAGAGGT: 248
TCAACAACT: 184 GTGAAAGTG: 213 GCACCCACG: 50 AGATCTAGA: 172	AGAGCGAGA: 810	GTGGAGGTG: 268	CIGGAGGIC: 212	CCGTCTGCC: 45
AGAICIAGA: 1/2	GAGTTTGAG: 1221	ATTGGGTTA: 144 GTGGAGGTG: 268 TGGAGAGGT: 248	CTGTGTGTC: 359	TTGTATGTT: 435
CTTCACTTC: 262	AGAGGGAGA: 638	CCGTCTGCC: 45	GGAAGAAGG: 398	GTCAGACTG: 134
GAGTGTGAG: 195	ACAACAACA: 1140	TTGTATGTT: 435	GACAAACAG: 237	TCTGTGTCT: 660
GCAGGGACG: /4	TITGTGTTT: 1057	GTCAGACTG: 134	TAGGGGGAT: 85	TGAAGAAGT: 307
GGACGCAGG: 61	TTACACATT: 365	TCTGTGTCT: 660	CAGCTCGAC: 19	AGCAGACGA: 30
CGTGGGTGC: 54	CCTTTTTCC: 341 GGGTGTGGG: 373 GCCCACCCG: 80	TGAAGAAGT: 307	ATCAGACTA: 139	TGAGTGAGT: 463
GGCGTGCGG: 29	GGGTGTGGG: 373	AGCAGACGA: 30	CCTGCGTCC: 65	ACCTGTCCA: 148
TCAGGGACT: 197	CCCCACCCG: 80	TGAGTGAGT: 463	CACGTGCAC: 120	CCGTGTGCC: 63
ACAAAAACA: 1452	CTCCTCCTC: 708	ACCTGTCCA: 148	GGGATAGGG: 91	CAACCCAAC: 103
AAAGAGAAA: 1717 TGACCCAGT: 136 AAGGAGGAA: 539	CTCCTCCTC: 798 CTTCTCTTC: 562	CCGTGTGCC: 63	GGAGCGAGG: 65	GATACATAG: 154
TGACCCAGT: 136	GGACTCAGG: 219	CAACCCAAC: 103	TCTGGGTCT: 279	ACCCGCCCA: 49
AAGGAGGAA: 539	ACAGAGACA: 511	GATACATAG: 154	CCATGTACC: 88	CAGTGTGAC: 188
ACIGAGICA: 221		ACCCCCCCA · 49	CCAACAACC: 122	CAGTGTGAC: 188 GCAACAACG: 27
ATACACATA: 745	TAAACAAAT: 792 AAAATAAAA: 5242	CAGTGTGAC: 188 GCAACAACG: 27	AAATTTAAA: 1741	CTAACAATC: 87
TGCAGACGT: 54 ACGTGTGCA: 242 TACACACAT: 567	CTTTCTTTC: 2506	GCAACAACG: 27	GAGATAGAG: 334	TGTCTCTGT: 627
ACGTGTGCA: 242	TTCCTCCTT: 500	CTAACAATC: 87	TATTATTAT: 1852	TTGAAAGTT: 358
TACACACAT: 567	TTCCTCCTT: 598	TGTCTCTGT: 627	GAACTCAAG: 180	ACGATAGCA: 23
ACGAGAGCA: 43	CCCGGGCCC: 112	TTGAAAGTT: 358		ACGATAGCA: 23
CACAAACAC: 450	CCCGGGCCC: 112 ATTTGTTTA: 742 GTGTTTGTG: 452	ACGATAGCA: 23	ATTAAATTA: 813	CTTCGCTTC: 40 CTGCCCGTC: 59
AACCACCAA: 185	AGAAGAAGA: 894	CTTCGCTTC: 40	ACATCTACA: 299	TATELOCULE: 59
AACACACAA: 415	AGAAGAAGA: 894	CTTCGCTTC: 40 CTGCCCGTC: 59	CTCTGTCTC: 1967	TATCACTAT: 221
TTGGTGGTT: 195	GGTGTGTGG: 330	TATCACTAT: 221	CAGCCCGAC: 48	CACGAGCAC: 17
AACACACAA: 415 TTGGTGGTT: 195 TCTTGTTCT: 420	CCACTCACC: 229	CACGAGCAC: 17	CGACACAGC: 56	GAGCACGAG: 49
TGAAAAAGT: 873	TTCTTTCTT: 3510 CGCCACCGC: 118	GAGCACGAG: 49	CTGAGAGTC: 182	CTTATATTC: 187
ACTATATCA: 185	CGCCACCGC: 118	CTTATATTC: 187	GTGTCTGTG: 566	CGGAGAGGC: 59
AGTAAATGA: 387	CCCACACCC: 305	CGGAGAGGC: 59	GAAACAAAG: 461	TGGTTTGGT: 275

A estratégia utilizada para montar esse programa foi salvar a sequência do cromossomo em uma string, dividí-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.

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.

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 estivá 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