

Biologia Computacional 2019/2 - Lista de Exercícios 1

Aluno: Leonardo Oliveira Wellausen

1a) Pseudo-código:

```
cromo = open(sequence.fasta)
target_subseq = 'CAATTGAATAATTG'
possible_nucs = 'CATG'
trg_len = len(target_subseq)
mutated_subseq = ''

for i to trg_len:
    for cada nuc em possible_nucs:
        if target_subseq[i] == nuc:
            continua # ignora este laço pois não há mutação
        mutated_subseq = target_subseq
        mutated_subseq[i] = nuc # realiza mutação
        occ = cromos.count(mutated_subseq) # número de ocorrências da mutação
        if occ == 1: # se mutação aparece uma única vez
            retorna posição da mutação, mutação realizada e outras informações
```

Ao final encontramos a subsequência dada sem a mutação que ela continha. Como mais de uma mutação foi encontrada, todas são retornadas..

Resultados:

Subsequência não mutada: CAATAGAATAATTG . Encontrada com início na posição 20843949 do cromossomo. Mutação realizada: nucleotídeo T pelo nucleotídeo A na posição 4 da subsequência.

Subsequência não mutada: CAATTCAATAATTG . Encontrada com início na posição 48205636 do cromossomo. Mutação realizada: nucleotídeo G pelo nucleotídeo C na posição 5 da subsequência.

Subsequência não mutada: CAATTAAATAATTG . Encontrada com início na posição 35769315 do cromossomo. Mutação realizada: nucleotídeo G pelo nucleotídeo A na posição 5 da subsequência.

Subsequência não mutada: CAATTTAATAATTG . Encontrada com início na posição 47003091 do cromossomo. Mutação realizada: nucleotídeo G pelo nucleotídeo T na posição 5 da subsequência.

Subsequência não mutada: CAATTGAATGATTG . Encontrada com início na posição 3630527 do cromossomo. Mutação realizada: nucleotídeo A pelo nucleotídeo G na posição 9 da subsequência.

Subsequência não mutada: CAATTGAATATTTG . Encontrada com início na posição 58013515 do cromossomo. Mutação realizada: nucleotídeo A pelo nucleotídeo T na posição 10 da subsequência.

Subsequência não mutada: CAATTGAATAACTG . Encontrada com início na posição 36409919 do cromossomo. Mutação realizada: nucleotídeo T pelo nucleotídeo C na posição 11 da subsequência.

Subsequência não mutada: CAATTGAATAATTT . Encontrada com início na posição 8010831 do cromossomo. Mutação realizada: nucleotídeo G pelo nucleotídeo T na posição 13 da subsequência.

1b) Pseudo-código:

```
cromo = open(sequence.fasta)
size1 = 9
size2 = 11
pal1 = {}
pal2 = {}
total1 = 0
total2 = 0
auxseq = "
```

for i sendo cada posição em cromos:

```
    auxseq = subsequência de tamanho size1 começando em i
    ## insere auxseq no dicionário, tendo como valor seu número de ocorrências
    if auxseq é palíndromo
        pal1[auxseq] ++
    auxseq = subsequência de tamanho size2 começando em i
    if auxseq é palíndromo
        pal2[auxseq] ++
write(pal1, pal2, file=e1-1b.txt)
```

Ao final do algoritmo, pal1 e pal2 são dicionários onde as chaves são todos os palíndromos de tamanho 9 e 11, respectivamente, e seus respectivos valores são o número de ocorrência de cada palíndromo.

Resultados:

Total de palíndromos de tamanho 9: 720286 sendo 1025 únicos (diferentes entre si).

Total de palíndromos de tamanho 11: 363615 sendo 3713 únicos (diferentes entre si).

Listagem completa de cada palíndromo e seu número de ocorrências no arquivo e1-1b.txt.

Alguns exemplos de linhas do arquivo:

TTAATAATT : 1135

TTGTATGTT : 722

TAAAGAAAT : 1399

GAGTTTGAG : 2177
AAAAAAAAAA : 70789
TTGTCGCTGTT : 4
CAAAAAAAAAAC : 192
CCGCCGCCGCC : 106
CTCACCCACTC : 38
TTTTTTTTTTTT : 46712

1c) Pseudo-código:

```
cromo = open(sequence.fasta)
subseqs = {}
subseq_size = 37
auxseq = ''
```

for i sendo cada posição de cromó:

```
    auxseq = subsequência de tamanho subseq_size começando em i
    ## atualiza chave auxseq no dicionário incrementando sua contagem
    subseqs[auxseq] ++
```

```
write(subseqs, file=e1-1c.txt)
```

Ao final o dicionário subseqs contém todas as diferentes subsequências de tamanho 37 como chave e seus respectivos valores são seus números de ocorrência.

Resultados:

Número de subsequências diferentes de tamanho 37 encontradas: 56024903 .

Listagem completa de cada subsequência e seu número de ocorrências no arquivo e1-1c.txt.

Começando gravação de arquivo. Processo demorado e arquivo gigante!

Alguns exemplos de linhas do arquivo:

TAACCCTAACCCTAACCCTAACCCTAACCCTAACCCT : 10
AACCCTAACCCTAACCCTAACCCTAACCCTAACCCTA : 11
ACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAA : 11
CCCTAACCCTAACCCTAACCCTAACCCTAACCCTAAC : 11

1d) Pseudo-código:

```
cromo = open(sequence.fasta)
possible_nucs = list('CATG')
nucleotideos = {}
```

for i sendo cada posição de cromô:

```
    ## atualiza posição do dicionário correspondente ao caracter i de cromô
    nucleotideos[cromo[i]] ++
```

Ao final o dicionário nucleotideos contém todos os diferentes caracteres encontrados em cromô como chaves tendo seus números de ocorrências como valor.

Resultados:

Número de ocorrências do nucleotídeo C : 11918693 .

Número de ocorrências do nucleotídeo A : 17146584 .

Número de ocorrências do nucleotídeo T : 17132531 .

Número de ocorrências do nucleotídeo G : 11909449 .

Caracter diferente encontrado: N . Sua contagem: 2396 .

1e) Pseudo-código:

```
cromo = open(sequence.fasta)
```

for i sendo cada posição de cromô:

```
    if cromô[i] == 'A':
        cromô[i] = 'T'
    elif cromô[i] == 'T':
        cromô[i] = 'A'
    elif cromô[i] == 'C':
        cromô[i] = 'G'
    elif cromô[i] == 'G':
        cromô[i] = 'C'
```

```
write(cromo, file=e1-1e.txt)
```

Resultados:

Arquivo salvo com a sequência complementar.

Algumas linhas do arquivo:

GATTGGGATTGGGATTGGGATTGGGATTGGGATTGGGATTGGGATTGGGATTGG
GG
GATTGGGATTGGGATTGGGATTGTTGGGATTGGGATTGGGATTGGGATTGGGATTTGG
AT
TGGGATTGGGATTGGGATTGTTGGGATTGGGATTGGGATTGGGATTGGGATTGTTGGG
AT
TGGGATTGTTGGGATTGGGATTGGGATTGGGATTGGGATTGGGATTGGGATTGGGATT
GG
GTTGTAAGTTTTCGACTCGTCCCGAAATTTGCATAGAATAATTATTAATAAAGACATAAC

Comparado com o original:

CTAACCCCTAACCCCTAACCCCTAACCCCTAACCCCTAACCCCTAACCCCTAACCCCTAAC
C
CTAACCCCTAACCCCTAACCCCTAACAACCCTAACCCCTAACCCCTAACCCCTAACCCCTAACCCCTAAACCTA
ACCCTAACCCCTAACCCCTAACAACCCTAACCCCTAACCCCTAACCCCTAACCCCTAACCCCTAACCAACCCT
A
ACCCTAACCAACCCTAACCCCTAACCCCTAACCCCTAACCCCTAACCCCTAACCCCTAACCCCTAACCCCTAAC
C
CAACATTCAAAGCTGAGCAGGGCTTTAAAGCTATCTTATTAATAATTATTTCTGTATTG