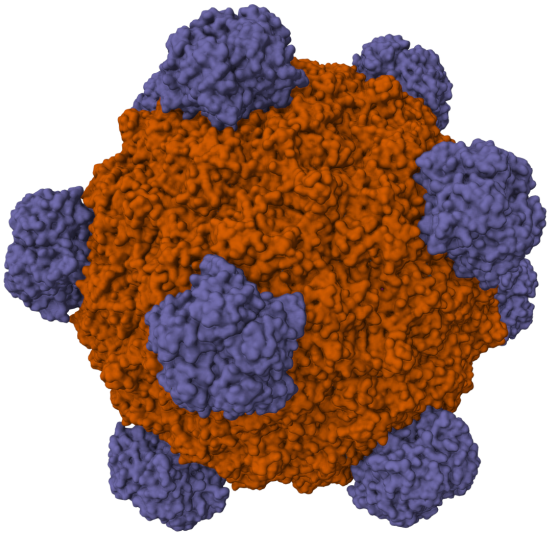
Lista de exercícios 3

Esta lista é composta por um único exercício. Você deve implementar a solução usando Python. Cada parte do código deve estar em um módulo separado. A nota desta lista é de até 20 pontos (a pontuação dependerá de quão completa sua solução estiver). Você deve executar seu programa com o comando "`python seu_programa.py -i genoma.fasta`" (dê um nome legal para o seu programa). Inclua um arquivo de texto chamado "`README.md`" explicando como usar o seu programa (você pode usar a linguagem *markdown* para formatar esse documento). A solução deve ser submetida em um único arquivo zip na plataforma Moodle.

1) Estudando o genoma de um vírus

A figura a seguir apresenta a estrutura 3D do capsídeo de um vírus de fita simples de DNA. O capsídeo é uma estrutura proteica que envolve o material genético de um vírus. A função principal do capsídeo é proteger e encapsular o material genético viral durante a transmissão e infecção.



*Estrutura tridimensional do vírus. Em roxo podemos ver a proteína **spike**.*

Observe o genoma completo desse mesmo vírus:

genoma.fasta

```
>genoma_virus
GAGTTTTATCGCTTCCATGACGCAGAAGTTAACACTTTCGGATATTTCTGATGAGTCGAAAAATTATCTT
GATAAAGCAGGAATTACTACTGCTTGTTTACGAATTAAATCGAAGTGGACTGCTGGCGGAAAATGAGAAA
ATTCGACCTATCCTTGCGCAGCTCGAGAAGCTCTTACTTTGCGACCTTTCGCCATCAACTAACGATTCTG
TCAAAAACTGACGCGTTGGATGAGGAGAAGTGGCTTAATATGCTTGGCACGTTCGTCAAGGACTGGTTTA
GATATGAGTCACATTTTGTTCATGGTAGAGATTCTCTTGTTGACATTTTAAAAGAGCGTGGATTACTATC
```

TGAGTCCGATGCTGTTCAACCACTAATAGGTAAGAAATCATGAGTCAAGTTACTGAACAATCCGTACGTT
TCCAGACCGCTTTGGCCTCTATTAAGCTCATTCAGGCTTCTGCCGTTTTGGATTTAACCGAAGATGATTT
CGATTTTCTGACGAGTAACAAAGTTTGGATTGCTACTGACCGCTCTCGTGCTCGTCGCTGCGTTGAGGCT
TGCGTTTATGGTACGCTGGACTTTGTGGGATACCCTCGCTTTCCTGCTCCTGTTGAGTTTATTGCTGCCG
TCATTGCTTATTATGTTCATCCCGTCAACATTCAAACGGCCTGTCTCATCATGGAAGGCGCTGAATTTAC
GGAAAACATTATTAATGGCGTCGAGCGTCCGGTTAAAGCCGCTGAATTGTTCGCGTTTACCTTGCGTGTA
CGCGCAGGAAACACTGACGTTCTTACTGACGCAGAAGAAAACGTGCGTCAAAAATTACGTGCGGAAGGAG
TGATGTAATGTCTAAAGGTAAAAAACGTTCTGGCGCTCGCCCTGGTCGTCCGCAGCCGTTGCGAGGTACT
AAAGGCAAGCGTAAAGGCGCTCGTCTTTGGTATGTAGGTGGTCAACAATTTTAATTGCAGGGGCTTCGGC
CCCTTACTTGAGGATAAATTATGTCTAATATTCAAACTGGCGCCGAGCGTATGCCGCATGACCTTTCCCA
TCTTGGCTTCCTTGCTGGTCAGATTGGTCGTCTTATTACCATTTCAACTACTCCGGTTATCGCTGGCGAC
TCCTTCGAGATGGACGCCGTTGGCGCTCTCCGTCTTTCTCCATTGCGTCGTGGCCTTGCTATTGACTCTA
CTGTAGACATTTTTACTTTTTATGTCCCTCATCGTCACGTTTATGGTGAACAGTGGATTAAGTTCATGAA
GGATGGTGTTAATGCCACTCCTCTCCCGACTGTTAACACTACTGGTTATATTGACCATGCCGCTTTTCTT
GGCACGATTAACCCTGATACCAATAAAATCCCTAAGCATTTGTTTCAGGGTTATTTGAATATCTATAACA
ACTATTTTAAAGCGCCGTGGATGCCTGACCGTACCGAGGCTAACCCTAATGAGCTTAATCAAGATGATGC
TCGTTATGGTTTCCGTTGCTGCCATCTCAAAAACATTTGGACTGCTCCGCTTCCTCCTGAGACTGAGCTT
TCTCGCCAAATGACGACTTCTACCACATCTATTGACATTATGGGTCTGCAAGCTGCTTATGCTAATTTGC
ATACTGACCAAGAACGTGATTACTTCATGCAGCGTTACCATGATGTTATTTCTTCATTTGGAGGTAAAAC
CTCTTATGACGCTGACAACCGTCCTTTACTTGTCATGCGCTCTAATCTCTGGGCATCTGGCTATGATGTT
GATGGAACTGACCAAACGTCGTTAGGCCAGTTTTCTGGTCGTGTTCAACAGACCTATAAACATTCTGTGC
CGCGTTTCTTTGTTCCTGAGCATGGCACTATGTTTACTCTTGCGCTTGTTCGTTTTCCGCCTACTGCGAC
TAAAGAGATTCAGTACCTTAACGCTAAAGGTGCTTTGACTTATACCGATATTGCTGGCGACCCTGTTTTG
TATGGCAACTTGCCGCCGCGTGAAATTTCTATGAAGGATGTTTTCCGTTCTGGTGATTCGTCTAAGAAGT
TTAAGATTGCTGAGGGTCAGTGGTATCGTTATGCGCCTTCGTATGTTTCTCCTGCTTATCACCTTCTTGA
AGGCTTCCCATTCATTCAGGAACCGCCTTCTGGTGATTTGCAAGAACGCGTACTTATTCGCCACCATGAT
TATGACCAGTGTTTCCAGTCCGTTCAGTTGTTGCAGTGGAATAGTCAGGTTAAATTTAATGTGACCGTTT
ATCGCAATCTGCCGACCACTCGCGATTCAATCATGACTTCGTGATAAAAGATTGAGTGTGAGGTTATAAC
GCCGAAGCGGTAAAAATTTTAATTTTTGCCGCTGAGGGGTTGACCAAGCGAAGCGCGGTAGGTTTTCTGC
TTAGGAGTTTAATCATGTTTCAGACTTTTATTTCTCGCCATAATTCAAACTTTTTTTCTGATAAGCTGGT
TCTCACTTCTGTTACTCCAGCTTCTTCGGCACCTGTTTTACAGACACCTAAAGCTACATCGTCAACGTTA
TATTTTGATAGTTTGACGGTTAATGCTGGTAATGGTGGTTTTCTTCATTGCATTCAGATGGATACATCTG
TCAACGCCGCTAATCAGGTTGTTTCTGTTGGTGCTGATATTGCTTTTGATGCCGACCCTAAATTTTTTGC
CTGTTTGGTTCGCTTTGAGTCTTCTTCGGTTCCGACTACCCTCCCGACTGCCTATGATGTTTATCCTTTG
AATGGTCGCCATGATGGTGGTTATTATACCGTCAAGGACTGTGTGACTATTGACGTCCTTCCCCGTACGC
CGGGCAATAACGTTTATGTTGGTTTCATGGTTTGGTCTAACTTTACCGCTACTAAATGCCGCGGATTGGT
TTCGCTGAATCAGGTTATTAAAGAGATTATTTGTCTCCAGCCACTTAAGTGAGGTGATTTATGTTTGGTG
CTATTGCTGGCGGTATTGCTTCTGCTCTTGCTGGTGGCGCCATGTCTAAATTGTTTGGAGGCGGTCAAAA
AGCCGCCTCCGGTGGCATTCAAGGTGATGTGCTTGCTACCGATAACAATACTGTAGGCATGGGTGATGCT
GGTATTAAATCTGCCATTCAAGGCTCTAATGTTCCTAACCCTGATGAGGCCGCCCCTAGTTTTGTTTCTG
GTGCTATGGCTAAAGCTGGTAAAGGACTTCTTGAAGGTACGTTGCAGGCTGGCACTTCTGCCGTTTCTGA
TAAGTTGCTTGATTTGGTTGGACTTGGTGGCAAGTCTGCCGCTGATAAAGGAAAGGATACTCGTGATTAT
CTTGCTGCTGCATTTCCTGAGCTTAATGCTTGGGAGCGTGCTGGTGCTGATGCTTCCTCTGCTGGTATGG
TTGACGCCGGATTTGAGAATCAAAAAGAGCTTACTAAAATGCAACTGGACAATCAGAAAGAGATTGCCGA
GATGCAAAATGAGACTCAAAAAGAGATTGCTGGCATTCAGTCGGCGACTTCACGCCAGAATACGAAAGAC
CAGGTATATGCACAAAATGAGATGCTTGCTTATCAACAGAAGGAGTCTACTGCTCGCGTTGCGTCTATTA
TGGAAAACACCAATCTTTCCAAGCAACAGCAGGTTTCCGAGATTATGCGCCAAATGCTTACTCAAGCTCA
AACGGCTGGTCAGTATTTTACCAATGACCAAATCAAAGAAATGACTCGCAAGGTTAGTGCTGAGGTTGAC
TTAGTTCATCAGCAAACGCAGAATCAGCGGTATGGCTCTTCTCATATTGGCGCTACTGCAAAGGATATTT
CTAATGTCGTCACTGATGCTGCTTCTGGTGTGGTTGATATTTTTCATGGTATTGATAAAGCTGTTGCCGA
TACTTGGAACAATTTCTGGAAAGACGGTAAAGCTGATGGTATTGGCTCTAATTTGTCTAGGAAATAACCG
TCAGGATTGACACCCTCCCAATTGTATGTTTTCATGCCTCCAAATCTTGGAGGCTTTTTTATGGTTCGTT
CTTATTACCCTTCTGAATGTCACGCTGATTATTTTGACTTTGAGCGTATCGAGGCTCTTAAACCTGCTAT
TGAGGCTTGTGGCATTTCTACTCTTTCTCAATCCCCAATGCTTGGCTTCCATAAGCAGATGGATAACCGC
ATCAAGCTCTTGGAAGAGATTCTGTCTTTTCGTATGCAGGGCGTTGAGTTCGATAATGGTGATATGTATG
TTGACGGCCATAAGGCTGCTTCTGACGTTCGTGATGAGTTTGTATCTGTTACTGAGAAGTTAATGGATGA
ATTGGCACAATGCTACAATGTGCTCCCCCAACTTGATATTAATAACACTATAGACCACCGCCCCGAAGGG
GACGAAAAATGGTTTTTAGAGAACGAGAAGACGGTTACGCAGTTTTGCCGCAAGCTGGCTGCTGAACGCC
CTCTTAAGGATATTCGCGATGAGTATAATTACCCCAAAAAGAAAGGTATTAAGGATGAGTGTTCAAGATT
GCTGGAGGCCTCCACTATGAAATCGCGTAGAGGCTTTGCTATTCAGCGTTTGATGAATGCAATGCGACAG
GCTCATGCTGATGGTTGGTTTATCGTTTTTGACACTCTCACGTTGGCTGACGACCGATTAGAGGCGTTTT
ATGATAATCCCAATGCTTTGCGTGACTATTTTCGTGATATTGGTCGTATGGTTCTTGCTGCCGAGGGTCG
CAAGGCTAATGATTCACACGCCGACTGCTATCAGTATTTTTGTGTGCCTGAGTATGGTACAGCTAATGGC
CGTCTTCATTTCCATGCGGTGCACTTTATGCGGACACTTCCTACAGGTAGCGTTGACCCTAATTTTGGTC
GTCGGGTACGCAATCGCCGCCAGTTAAATAGCTTGCAAAATACGTGGCCTTATGGTTACAGTATGCCCAT
CGCAGTTCGCTACACGCAGGACGCTTTTTCACGTTCTGGTTGGTTGTGGCCTGTTGATGCTAAAGGTGAG
CCGCTTAAAGCTACCAGTTATATGGCTGTTGGTTTCTATGTGGCTAAATACGTTAACAAAAAGTCAGATA
TGGACCTTGCTGCTAAAGGTCTAGGAGCTAAAGAATGGAACAACTCACTAAAAACCAAGCTGTCGCTACT
TCCCAAGAAGCTGTTCAGAATCAGAATGAGCCGCAACTTCGGGATGAAAATGCTCACAATGACAAATCTG
TCCACGGAGTGCTTAATCCAACTTACCAAGCTGGGTTACGACGCGACGCCGTTCAACCAGATATTGAAGC
AGAACGCAAAAAGAGAGATGAGATTGAGGCTGGGAAAAGTTACTGTAGCCGACGTTTTGGCGGCGCAACC
TGTGACGACAAATCTGCTCAAATTTATGCGCGCTTCGATAAAAATGATTGGCGTATCCAACCTGCA

A seguir, construa um *pipeline* em Python que atenda aos seguintes requisitos. O *pipeline* deve ter o nome de sua preferência, mas deve ter cinco módulos (**a.py, b.py, c.py, d.py** e **e.py**), que podem ser importados pelo controlador principal (exceto o módulo "e", que poderá ser chamado separadamente). Os arquivos dos módulos devem estar em uma pasta chamada "lib". Cada módulo deverá resolver o respectivo problema indicado em cada uma das letras a seguir:

a) Leia o arquivo completo do genoma presente no arquivo "genoma.fasta". Imprima na tela o tamanho da sequência e qual o conteúdo GC do genoma.
b) Divida a sequência em fragmentos de tamanho k = 31 usando a técnica de janela deslizante e salve em um arquivo multi-FASTA chamado "reads.fasta" dentro de uma pasta chamada "saida" (cada sequência deve ser identificada por um número e deve ter uma linha de cabeçalho iniciada com ">"). Quantas sequências serão armazenadas nesse arquivo? **Opcional**: tente reconstruir o genoma original com base no arquivo "reads.fasta".
c) Identifique todas as regiões codificantes (CDS) deste genoma. Considere que a região codificante começa com uma metionina e termina com um stop códon; considere as seis janelas de codificação (não há problema seu programa retornar falsos-positivos). Salve cada possível CDS em um arquivo no formato FASTA (os arquivos devem ser salvos em uma pasta chamada "cds").
d) Identifique o gene que codifica a proteína SPIKE (pesquise a importância dessa proteína para vírus no ChatGPT ou em um buscador) e salve em um arquivo chamado "spike.fasta" dentro da pasta "saida". **Dica**: esta proteína possui uma região de grande importância com cinco aminoácidos em sequência - o primeiro é uma glicina (G), seguida de dois aminoácidos com carga polar positivo (R, K ou H), seguido por um aminoácido polar negativo (D ou E) e, por fim, uma outra glicina (G).
e) Realize a modelagem da estrutura 3D da proteína (spike.fasta) usando ColabFold: https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb (use os parâmetros padrão). *Opcional*: *abra a estrutura em um programa de visualização de proteínas (como o PyMOL ou o ChimeraX).* Por fim, crie um script que gera um mapa de distâncias e um mapa de contatos (ligações de hidrogênio) desta proteína (esse script não precisa estar conectado diretamente no seu *pipeline*). Salve os mapas no formato PNG ou PDF dentro da pasta "saida".