Universidade Federal de Pelotas

Centro de Desenvolvimento Tecnológico

Programa de Pós-Graduação em Biotecnologia

Disciplina “Bioinformática Avançada”

**Aula 2: Introdução à Linguagem Python – Atividade**

1. Crie um notebook na plataforma Google Colab.
2. Renomeie o notebook para “Aula 2: Introdução à Linguagem Python”
3. Crie uma célula de texto e nela adicione:
   1. Um cabeçalho de nível 1 com o texto “Aula 2: Introdução à Linguagem Python”
   2. Um parágrafo resumindo, com duas palavras, o porquê de a linguagem Python ser utilizada na bioinformática.
4. **Círculo.** Crie uma célula de código e nela defina as seguintes variáveis:

pi = 3.14

radius = 5

Utilizando as operações matemáticas do Python vistas em aula, calcule o perímetro e a área de um círculo com o raio igual ao radius, armazenando os valores calculados, respectivamente, nas variáveis circle\_perimeter e circle\_area.

1. **Baskara – parte I.** Crie uma célula de código e nela defina as seguintes variáveis:

a = -5

b = 2

c = 3

Utilizando as operações matemáticas do Python vistas em aula, calcule as raízes (x\_1 e x\_2) de uma equação do segundo grau para as variáveis acima.

1. **Transcrição – parte I**. Crie uma célula de código e nela defina uma variável dna\_sequence, nela armazenando uma sequência de DNA à sua escolha. Utilize o método .replace para produzir a sequência de RNA correspondente, armazenando o resultado em uma variável rna\_sequence.
2. ***Short Sequence Repeat* (SSR).** Crie uma célula de código e nela a seguinte variável:

ssr\_repeat = “GCATCGAC”

Utilizando as operações matemáticas para *string*, produza duas novas variáveis que contenham a sequência do alelo de indivíduos com 5 e 7 repetições neste locus. O nome das variáveis deve ser ssr\_allele\_5 e ssr\_allele\_7.

1. **Primeiro códon – parte I**. Crie uma célula de código e nela uma variável chamada dna\_sequence e nela insira uma sequência de DNA. Utilizando as operações de *slicing*, selecione apenas o primeiro códon da sequência a armazene ele em uma variável chamada codon.
2. **Primeiro códon – parte II**. Replique o código do exercício anterior em uma nova célula de código, adicionando ao final uma estruturação if para checar se o primeiro códon da sequência é um ATG. Se for, a mensagem “É um códon de iniciação” deve ser mostrada. Caso contrário, a mensagem “Não é um códon de iniciação” deve ser mostrada.
3. **Fim de transcrição.** Crie uma nova células de código e nela as seguintes variáveis:

sequence\_codons = [‘ATG’, ‘GGG’, ‘CCC’, ‘AAA’, ‘TAA’, ‘GGC’]

stop\_codons = [‘TAA’, ‘TGA’, ‘TAG’]

Percorra os códons da lista sequence\_codons utilizando a estrutura for e cheque, utilizando uma estrutura if, se o códon está na lista de códons de terminação (stop\_codons). Se estivar, a mensagem “códon de terminação encontrado” deve ser mostrada e o for deve ser interrompido com um break. Caso contrário, mostrar a sequência do códon com um print.

1. **Complementariedade de base**. Crie uma nova célula de código e nela um dicionário chamado base\_complements que armazene os dados de complementariedade de base e uma variável dna\_sequence que armazene uma sequência de DNA à sua escolha. Utilizando a estrutura for, percorra as bases da sequência armazenada na dna\_sequence e mostre a sua respectiva base complementar.
2. **Baskara – parte II.** Crie uma nova célula de código e nela uma função que receba os parâmetros a, b e c de uma função de segundo grau e retorne as raízes x\_1 e x\_2, de modo similar à atividade 5. Teste a função utilizando os mesmos valores utilizados na atividade 5.
3. **Transcrição – parte II**. Cria uma nova célula de codigo e nela função que receba através do parâmetro dna\_sequence uma sequência de DNA e retorne a sequência de RNA correspondente, de modo similar à atividade 6.
4. Compartilhe o *notebook* produzido com o e-mail [fred.s.kremer@gmail.com](mailto:fred.s.kremer@gmail.com) através da função “Compartilhar”. A data final para entrega (compartilhamento) é o dia 28 de março de 2021, às 23:59.