Python w Bioinformatyce

Poniższe zadania mają chrakter problemowy. Ważne jest, aby rozwiązać podany problem przy użyciu dostępnych źródeł informacji. Sposób rozwiązania problemu jest drugorzędny i może być dowolny, ale musi wykorzystywać język Python i bibliotekę Biopython. Postaraj się napisać jak najprostszy kod, realizujący podane zadania. Jeśli korzystasz z narzędzi sztucznej inteligencji, umieść tę informację w swoim rozwiązaniu jako komentarz. Pamiętaj, że prowadzący może w każdej chwili poprosić Cię o wyjaśnienie, dlaczego dany kod został napisany i jak działa. Jeśli nie będziesz w stanie wyjaśnić, w jakim celu używasz wybranego fragmentu kodu, zadanie nie zostanie zaliczone.

Zadania - część pierwsza

Zwracane pliki rozwiązań:

Zadania	PLiki
1	script_1.py, sequences.txt
2	script_2.py
3	script_3.py, fig_3.py
4	script_4.py, fig_4.py
5	script_5.py

Zadanie 1

Napisz skrypt w języku Python, za pomocą którego można wygenerować `n` losowych ciągów nukleotydów o długości `k`. Wygenerowane ciągi powinny zostać zapisane w kolumnie w pliku o nazwie `sequences.txt`.

Wyniki swojej pracy zapisz w pliku o nazwie `script_1.py`.

Zadanie 2

Napisz skrypt w języku Python, za pomocą którego będziesz mógł wyszukiwać określone motywy w danej sekwencji DNA. Wynik działania skryptu powinien wyglądać następująco:

```
Szukany motyw: ACCTG
Pozycje: 7, 18, 45, 125
```

Szukany motyw jest przechowywany w zmiennej `motif`, sekwencja DNA jest ładowana z pliku. Użyj sekwencji zawartej w pliku `sequence.txt`. W swoim kodzie uwzględnij sytuację, w której szukany motyw nie występuje w analizowanej sekwencji.

Zapisz wyniki swojej pracy w pliku o nazwie `script_2.py`

Możesz użyć następującego fragmentu kodu, aby załadować sekwencję do obiektu `Seq`:

```python
from Bio.Seq import Seq

file_path = 'sequence.txt'
with open(file_path, 'r') as file:
    dna_sequence = file.read().strip()

sequence = Seq(dna_sequence)
```

Zadanie 3

Napisz skrypt w języku Python, aby wizualizować rozkład procentowy wybranego nukleotydu w danej sekwencji DNA. Zastanów się, w jaki sposób obliczana jest zawartość procentowa. Na początek można spróbować wykreślić liczbę nukleotydów danego typu w zależności od pozycji w sekwencji. Skrypt powinien zapisać wygenerowany wykres w pliku `fig_3.png`. Pamiętaj, aby odpowiednio nazwać osie wykresu.

Aby przeanalizować rozkład nukleotydów, użyj pliku o nazwie `sequence.txt` zawierającego dość długą sekwencję DNA.

Użyj biblioteki `matplotlib` wraz z modułem `pyplot`. Do generowania wykresów można użyć funkcji `scatter`, `plot` lub `barplot`.

Wyniki swojej pracy zapisz w pliku o nazwie `script_3.py`.

Zadanie 4

Zmodyfikuj skrypt z **Zadania 1** tak, aby można było ustawić wagi prawdopodobieństwa dla nukleotydów, które mają być wybierane losowo, co pozwoliłoby kontrolować zawartość poszczególnych nukleotydów w sekwencjach. Możesz użyć funkcji `choices()` z modułu `random`.

Użyj rozwiązania z **Zadania 3**, aby wykreślić rozkład nukleotydów w wygenerowanych sekwencjach. Na przykład, wygeneruj sekwencję 200 nukleotydów zawierającą 10% nukleotydu A, 30% nukleotydu C, 40% nukleotydu T i 20% nukleotydu G.

Zapisz wyniki swojej pracy w pliku o nazwie `script_4.py` i `fig_4.png`.

Zadanie 5

Napisz skrypt w języku Python zliczający liczbę par komplementarnych dla dwóch sekwencji DNA o równej długości. Załóżmy, że sekwencje nie są względem siebie "przesunięte". Przedstaw wyniki działania skryptu w następującej formie (dla dwóch przykładowych sekwencji):

```
Seq 1: ACTAGCTTGAGCCAGT
       | ||| | || -> 6
Seq 2: TAGACGATCAGATTCG
```

Zapisz wyniki swojej pracy w pliku o nazwie `script_5.py`.