Bioinformatyka - zadania programistyczne

Wykorzystanie Pythona w bioinformatyce - część I

Kurs:	Bioinformatyka
Język programowania:	Python
Termin zwrotu	23.05.2024

Ogólne wytyczne

Poniższe zadania są oparte na problemach. Ważne jest, aby rozwiązać dany 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 fragment kodu został użyty i jak działa.

Wymagania dot. oprogramowania

Python 3.X wraz z biblioteką Biopython.

Najprostszym sposobem na uzyskanie działającego środowiska Python jest zainstalowanie oprogramowania o nazwie Anaconda ze strony Anaconda Distribution Webpage.

Zadanie I

Cele **Poziom:** Początkujący - średniozawansowany

Moduł: Biopython - Seq Objects

Studenci zdobędą praktyczne doświadczenie w manipulowaniu danymi sekwencji biologicznych przy użyciu obiektów `Seq` Biopythona. Zadanie obejmuje ładowanie, tworzenie, manipulowanie i analizowanie sekwencji DNA.

Podpowiedzi

- Zapoznaj się z dokumentacją Biopython dla obiektów `Seq` w sekcji 3 samouczka Biopython: Sequence objects.
- Użyj modułu `SeqUtils` w Biopython do obliczania zawartości GC.
- Pracę można rozpocząć od przykładowego kodu w pliku `script_1.py`.

1. **Konfiguracja środowiska:**.
   - Upewnij się, że Python jest zainstalowany.
   - Zainstaluj Biopython za pomocą pip: `pip install biopython`.
   - Jeśli używasz środowiska Anaconda skorzystaj z instalatora graficznego lub polecenia `conda install -c conda-forge biopython`
2. **Załaduj sekwencję DNA:**.
   - Użyj następującej sekwencji DNA:

```
AAGAAATTCCAAGTCCAGGGATACACAAACAGGTGTACAGC \
AAATCATGTAGGTGGTACTTTTCCCCTAAGTTATAATATT
```

3. **Utwórz obiekt Seq:**
   - Utwórz obiekt `Seq` w Biopython używając wybranej sekwencji.
4. **Przeanalizuj sekwencję:**
   - Zaimplementuj następujące metody lub funkcje, aby przeanalizować sekwencję:
     - Policz wystąpienia każdego nukleotydu (A, T, C, G).
     - Oblicz zawartość GC.
     - Dokonaj transkrypcji sekwencji do sekwencji RNA.
     - Przetłumacz sekwencję DNA na sekwencję białkową.

5. **Sekwencja antyrównoległa:**
   - Utwórz nowy obiekt `Seq` z odwrotną sekwencją komplementarną .
6. **Zapis do pliku:**
   - Zapisz oryginalną sekwencję DNA, transkrybowaną sekwencję RNA, przetłumaczoną sekwencję białka i sekwencję antyrównoległą do pliku tekstowego.
7. **Przykładowe dane wyjściowe:**
   - Zastosuj następujący format dla danych wyjściowych (ostrzeżenie: użyto przykładowej sekwencji DNA):

```
Oryginalna sekwencja DNA: AGTACACTGGT
Liczba nukleotydów:
  A: 2
  T: 3
  C: 2
  G: 2
Zawartość GC: 36.36%
Transkrybowany RNA: AGUACACUGGU
Translowane białko: STG
Odwrotne dopełnienie: ACCAGTGTACT
```

   2Wzór rozwiązania znajduje się w pliku `example.txt`.
8. **Wytyczne dotyczące przesyłania:**
   - Prześlij skrypt Pythona jako plik `script_1.py` wraz z poprawnym wyjściowym plikiem tekstowym o nazwie `sequence_analysis.txt` (informacje w tym pliku muszą odnosić się do sekwencji DNA podanej w punkcie 2).

Zadanie II

Poziom: Średnio zaawansowany

Moduł: Biopython - Seq Objects

Cel Studenci poznają bardziej zaawansowane funkcje obiektów `Seq` i modułu `SeqIO` Biopythona. Zadanie to obejmuje odczytywanie sekwencji biologicznych z pliku FASTA, wykonywanie manipulacji sekwencjami i implementację algorytmu wyszukiwania motywów.

Wskazówki:

- Zapoznaj się z dokumentacją Biopython dla obiektów `Seq` i `SeqIO`: SequenceIO.

1. **Konfiguracja środowiska:**
   - Upewnij się, że Python jest zainstalowany.
   - Zainstaluj Biopython za pomocą pip: `pip install biopython`.
   - Jeśli używasz środowiska Anaconda skorzystaj z instalatora graficznego lub polecenia `conda install -c conda-forge biopython`
2. **Załaduj sekwencje z pliku FASTA:**.
   - Do pracy użyj dostarczonego pliku `ls_orchid.fasta`.
   - Odczytaj sekwencje za pomocą modułu `SeqIO` Biopythona.
3. **Analiza wielu sekwencji:**.
   - Dla każdej sekwencji zaimplementuj metody lub funkcje aby:
     – Policzyć wystąpienia każdego nukleotydu (A, T, C, G).
     – Obliczyć zawartości GC.
     – Znaleźć odwrotną nić komplementarną .
4. **Znajdź wspólne motywy:**.

- Użyj listy motywów: `ATG`, `TATA`, `GAATTC`.
- Zaimplementuj funkcję identyfikującą pozycje początkowe każdego motywu w każdej sekwencji.
- Zidentyfikuj wszystkie motywy dla wszystkich sekwencji.

5. **Przeprowadź translację sekwencji:**.
   - Przetłumacz każdą sekwencję DNA na białko przy użyciu wszystkich "reading frames" (do przodu i do tyłu).
   - Zidentyfikuj wszystkie kodony stop i podaj długość translacji.
   - W przypadku problemów zobacz dodatkowe informacje na końcu dokumentu.
6. **Eksport wyników do CSV:**
   - Zapisz wyniki analizy do pliku CSV:
     - Identyfikator sekwencji.
     - Liczba nukleotydów.
     - Zawartość GC.
     - Pozycje motywów.
     - Odwrotne dopełnienie.
     - Długość przetłumaczonych białek.
7. **Przykładowe dane wyjściowe:**
   - Przykładowy plik CSV może wyglądać następująco:

File: **example.csv**

```
SeqID,Nucleotide_Counts,GC_Content,Motif_Positions,Reverse_Complement,Translation_Lengths
seq1,"{'A': 10, 'T': 15, 'C': 8, 'G': 7}",40.0%,"motif1: [1, 15], motif2: [3]",TGCACGATCG,"{Frame1: 45, Frame2: 47, Frame3: 44}"
seq2,"{'A': 12, 'T': 10, 'C': 10, 'G': 9}",45.0%,"motif1: [2], motif2: []",ACGTACGTTG,"{Frame1: 34, Frame2: 33}"
```

Rysunek 1: Przykładowy plik csv.

   - W razie kłopotów zajrzyj do pliku `help.csv`, który zawiera pierwsze trzy linie pliku csv będącego rozwiązaniem zadania.
8. **Wytyczne dotyczące przesyłania:**
   - Prześlij skrypt Python jako plik `script_2.py`.
   - Dołącz wygenerowany plik CSV o nazwie `sequence_analysis.csv`.

Dodatkowe informacje

Translacja W biologii molekularnej termin "reading frames" odnosi się do różnych sposobów, w jakie sekwencja DNA może być odczytywana i tłumaczona na sekwencje białkowe. "Reading frame" jest zasadniczo sposobem podziału sekwencji nukleotydów na kolejne triplety (kodony), które reprezentują aminokwasy lub sygnały zatrzymania. Koncepcja sześciu "reading frames" wynika z dwóch kluczowych czynników:

1. **Kierunek nici:** DNA jest dwuniciowy, z dwiema komplementarnymi nićmi zorientowanymi w przeciwnych kierunkach (antyrównległymi). Translacja może zachodzić na obu niciach:
   - Jedna z nici w kierunku od 5' do 3'.
   - Komplementarna nić w przeciwnym kierunku od 5' do 3'.
2. **Punkty początkowe przesunięcia:** Dla każdej nici, translacja może rozpocząć się w trzech różnych pozycjach lub przesunięciach w sekwencji, znanych jako "reading frames":
   - "**Ramka 1:**" Zaczynając od pierwszego nukleotydu.
   - "**Ramka 2:**" Począwszy od drugiego nukleotydu.
   - "**Ramka 3:**" Począwszy od trzeciego nukleotydu.

Prowadzi to w sumie do sześciu ramek odczytu:

- **Nić wiodąca 5' - 3':**.
  - Ramka 1 (oryginalna nić, zaczynająca się od pierwszej zasady)
  - Ramka 2 (zaczynająca się od drugiej zasady)

  - Ramka 3 (zaczynająca się od trzeciej zasady)
- **Nić komplementarna**
  - Ramka 4 (nić antyrównoległa, zaczynająca się od pierwszej zasady)
  - Ramka 5 (nić antyrównoległa, zaczynająca się od drugiej zasady)
  - Ramka 6 (nić antyrównoległa, zaczynająca się od trzeciej zasady)

Każda ramka odczytu daje unikalną sekwencję białka lub nie daje żadnego białka, jeśli zawiera kodony stop we wczesnej części ramki odczytu.