Instytut Informatyki i Matematyki Komputerowej UJ,
opracowanie: Ewa Matczyńska, Jacek Śmietański.

NCBI, Entrez, Biopython

1. NCBI

National Center for Biotechnology Information (NCBI, http://www.ncbi.nlm.nih.gov) jest częścią *National Library of Medicine* (NLM) czyli największej na świecie biblioteki medycznej prowadzonej przez rząd federalny Stanów Zjednoczonych. Pod zarządem NCBI jest szereg najważniejszych biologicznych baz danych, m.in. bazy danych sekwencji DNA, sekwencji białkowych, struktur białkowych, czy publikacji medycznych. Zależności między poszczególnymi rekordami w bazach danych realizują się w ogromnej sieci powiązań.
NCBI udostępnia też system ENTREZ, pozwalający wyszukiwać informacje jednocześnie we wszystkich zgromadzonych bazach. Oprócz tego NCBI udostępnia szereg narzędzi, np. do analizy sekwencji.
Na poprzednich zajęciach tworzyliśmy sekwencję komplementarną do sekwencji hemoglobiny oraz tłumaczyliśmy mRNA hemoglobiny na białko. Znajdziemy teraz w NCBI podaną wcześniej sekwencję genu hemoglobiny człowieka wraz z jego mRNA.

Zadanie 1.
1. Wejdź na stronę **NCBI** (http://www.ncbi.nlm.nih.gov) lub **Entrez** (https://www.ncbi.nlm.nih.gov/gquery/) – pierwszy link zaprowadzi Cię na główną stronę NCBI, gdzie możesz się zapoznać z dostępnymi narzędziami; drugi wyświetla listę baz danych indeksowanych przez Entrez.
2. W pole wyszukiwania na górze strony wpisz: ***hemoglobin human*** - pokażą się rezultaty we wszystkich bazach NCBI dla tego zapytania (w ilu bazach danych zostały znalezione informacje?).
3. Wybierz bazę **Gene**, a w wynikach pierwszy znaleziony rekord o nazwie HBB - pojawia się strona podsumowująca informacje o tym genie. Czego możemy się dowiedzieć o hemoglobinie w kolejnych sekcjach?
4. Znajdź sekcję NCBI Reference Sequences (**RefSeq**). Co wyróżnia bazę RefSeq?
5. Pobierz sekwencję genomową formacie FASTA w podsekcji **Genomic**.

> FASTA jest prostym formatem zapisu sekwencji nukleotydowych jak i aminokwasowych. Plik w formacie FASTA rozpoczyna się znakiem „>”, po czym następuje opis sekwencji, najczęściej jest to jej numer identyfikacyjny, informacje na temat tego z jakiego organizmu pochodzi sekwencja i na jakim chromosomie znajduje się jej źródło. Następnie, w liniach o ustalonej długości (najczęściej 70 znaków) jest zapisywana sekwencja. Kody zarówno dla nukleotydów, jak i aminokwasów są jednoliterowe.

6. W podsekcji **mRNA and Protein** przejdź do linka oznaczonego jako **NM_000518.5** (NM oznacza identyfikator dla mRNA, a NP dla sekwencji białkowych). Wyświetli się podsumowanie tej sekwencji w formacie GenBank.

Genbank – format przechowywania sekwencji bardziej skomplikowany od FASTA. Zawiera za to dużo więcej informacji.

7. W prawym górnym rogu możesz zapisać sekwencję w formacie FASTA wybierając Send (zaznacz opcję Coding Sequences).

2. Biopython – wprowadzenie

Biopython jest zbiorem narzędzi pythona pomocnych przy zadaniach bioinformatycznych. Posiada moduły, które pozwalają łatwo przetwarzać dane z plików formatów występujących w bioinformatycznych bazach danych oraz plików wynikowych z najpopularniejszych programów używanych do analizy sekwencji bądź struktur. Ułatwia komunikację z tymi bazami, umożliwia obliczenia równoległe, jest zintegrowany z modelem BioSQL (modelem bazy danych do opisu i adnotacji sekwencji), zawiera również wiele innych funkcji.
Szczegóły i dokumentacja na stronie projektu: http://biopython.org/wiki/Main_Page

Zadanie 2. Operacje na sekwencjach.

```
from Bio.Seq import Seq
sequence = Seq("TGAGGCCCTGGGCAGGCTGCTGGTGGTCTACCCT")
print(sequence)

# obiekt Seq zachowuje się jak string:
print(sequence[0])
print(sequence[-1])
print(sequence.count('A'))
print(len(sequence))
print(sequence+'ATTTTTTA')
print(sequence[0::3])  # pierwszy nukleotyd z kodonu
print(sequence[1::3])  # drugi nukleotyd z kodonu
print(sequence[2::3])  # trzeci nukleotyd z kodonu

# mozemy przekształcić sekwencje do stringa:
print(str(sequence))
```

Zadanie 3. Alfabet.

Obiekt Seq, reprezentujący sekwencje, zachowuje się tak samo jak obiekt typu string, dodatkowo możemy ustawić w nim alfabet, czyli sprecyzować czy jest to sekwencja nukleotydowa czy może sekwencja złożona z aminokwasów. Nie można łączyć sekwencji o różnych alfabetach:

```
from Bio.Seq import Seq
from Bio.Alphabet import IUPAC
sequenceN=Seq("GATCGATGGGCCTATATAGGATCGAAAATCGC")

# alfabet obiektu
print(sequenceN.alphabet)

sequenceN = Seq("GATCGATGGGCCTATATAGGATCGAAAATCGC", IUPAC.unambiguous_dna)
print(sequenceN.alphabet)

sequenceAA = Seq("GATCGATGGGCCTATATAGGATCGAAAATCGC", IUPAC.protein)
print(sequenceN.alphabet)
```

```
print(sequenceAA.alphabet)

# nie można łączyć sekwencji o różnych alfabetach!
print(sequenceN+sequenceAA)
```

Zadanie 4. Nić komplementarna

```
from Bio.Seq import Seq
from Bio.Alphabet import IUPAC
nt_seq=Seq("ATGGCCATTGTAATGGGCCGCTGAAAGGGTGCCCGATAG",IUPAC.unambiguous_dna)
print(nt_seq.complement()) #uwaga ta sekwencja jest w kierunku 3' do 5'

prot_seq=Seq("MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFE",IUPAC.protein)
print(prot_seq.complement()) # sekwencje białkowe nie mają komplementarnych
```

Zadanie 5. Transkrypcja i translacja

Zwróć uwagę na automatyczną zmianę alfabetu

```
from Bio.Seq import Seq
from Bio.Alphabet import IUPAC
coding_dna=Seq("ATGGCCATTGTAATGGGCCGCTGAAAGGGTGCCCG",IUPAC.unambiguous_dna)
print(coding_dna, coding_dna.alphabet)

#transkrypcja
messenger_rna = coding_dna.transcribe()

#translacja z mRNA
protein = messenger_rna.translate()

#translacja prosto z DNA
protein2 = coding_dna.translate()

print(messenger_rna, messenger_rna.alphabet)
print(protein, protein.alphabet)
print(protein2, protein2.alphabet)
```

Zadanie 6. Obiekt SeqRecord

```
from Bio.Seq import Seq
from Bio.SeqRecord import SeqRecord
simple_seq = Seq("TGAGGCCCTGGGCAGGCTGCTGGTGGTCTACCC")
simple_seq_r = SeqRecord(simple_seq, id="AC12345", name="seqX",
                         description="Homo sapiens, chr11")
print(simple_seq_r)
```

Do obiektu SeqRecord możemy bardzo łatwo wczytać plik typu FASTA

```
from Bio import SeqIO
record = SeqIO.read("hemoglobin.fa", "fasta")
print(record.id)
print(record.name)
print(record.description)
print(record.annotations)
print(record.seq)
```

3. Entrez Biopython - pakiet Entrez

Entrez „widzi” bazy danych jako węzły (nodes), czyli kolekcje danych, które są tego samego typu oraz są razem indeksowane unikalnym w danym węźle identyfikatorem (unique ID (UID)). Typowe zapytanie przeszukujące wszystkie bazy zwróci nam listę węzłów, każdemu z nich będzie przyporządkowana lista rekordów z danej bazy wyszukana dla danego zapytania. Każdy rekord odpowiada obiektowi w bazie o określonym UID. Oprócz tego Entrez zapewnia połączenia między węzłami, tak jak w interfejsie webowym.

Korzystaliśmy już z systemu Entrez z poziomu przeglądarki, teraz użyjemy do tego pakietu Entrez Biopythona. Opiera się on na narzędziach dostępu do danych - Entrez Programming Utilities - udostępnianych przez NCBI, są to:

EInfo	Provides field index term counts, last update, and available links for each database
ESearch	Searches and retrieves primary IDs (for use in EFetch, ELink, and ESummary) and term translations and optionally retains results for future use in the user's environment
EPost	Posts a file containing a list of primary IDs for future use in the user's environment to use with subsequent search strategies
ESummary	Retrieves document summaries from a list of primary IDs or from the user's environment
EFetch	Retrieves records in the requested format from a list of one or more primary IDs or from the user's environment
ELink	Checks for the existence of an external or Related Articles link from a list of one or more primary IDs. Retrieves primary IDs and relevancy scores for links to Entrez databases or Related Articles, creates a hyperlink to the primary LinkOut provider for a specific ID and database, or lists LinkOut URLs and Attributes for multiple IDs
EGQuery	Provides Entrez database counts in XML for a single search using Global Query
ESpell	Retrieves spelling suggestions

Uwaga! NCBI narzuca pewne warunki korzystania z Entrez o których można przeczytać na powyższej stronie, chodzi głównie o to aby nie przeciążać NCBI zwłaszcza w godzinach pracy w USA, oraz aby przedstawiać się adresem mailowym.
Format pliku typowo zwracany przez EUtils to XML, który możemy odczytywać wbudowanym do Biopythona parserem **Bio.Entrez.Parser**, bądź korzystając z innych parserów. Wszystkie parametry dla wywołań narzędzi Entrez można znaleźć na stronie http://www.ncbi.nlm.nih.gov/books/NBK25497/

Zadanie 7. Zobacz jakie bazy danych są dostępne przez Bio.Entrez

```
from Bio import Entrez
Entrez.email = "your_email@uj.edu.pl" # przedstawiamy sie
handle = Entrez.einfo()
result = handle.read()
print(result)     # otrzymujemy XML
handle = Entrez.einfo()
```

```
result = Entrez.read(handle)  # uzywajac parsera z Bio.Entrez
print(result)     # otrzymujemy obiekt Pythona - slownik z lista baz
```

Zadanie 8. Zobacz informacje o konkretnej bazie

```
handle = Entrez.einfo(db="nucleotide")
result = Entrez.read(handle)

print(result["DbInfo"].keys())
print(result["DbInfo"]["Description"])
print(result["DbInfo"]["Count"])
print(result["DbInfo"]["LastUpdate"])

for field in result["DbInfo"]["FieldList"]:
      print("%(Name)s, %(FullName)s, %(Description)s" % field)
```

Sprawdź informacje dla innych baz np. *protein, structure* bądź *pubmed.*

Zadanie 9. Wyszukaj rekordy z bazy *nucleotide,* które posiadają związek z wirusem świńskiej grypy

```
handle = Entrez.esearch(db="nucleotide",term="swine influenza",retmax="10")
result = Entrez.read(handle)
print(result["IdList"])
```

Parametr „retmax” określa maksymalną liczbę zwracanych rezultatów.

Wyszukaj te same rekordy przez przegladarkę, zobacz jakiemu polu rekordu odpowiadają unikalne numery sekwencji nukleotydowych, zwrócone w tym zapytaniu.

Używając AND, OR i NOT oraz pól z FieldList znalezionych wcześniej przez **EInfo**, możemy rozbudować zapytanie, np.:

```
term = "influenza AND H1N1[TITL] AND (hemagglutinin[PROT] OR
neuraminidase[PROT])"
```

wyszuka wszystkie rekordy czyli sekwencje nukleotydowe, które w którymkolwiek polu zawierają słowo „influenza”, zawierają w tytule „H1N1” i kodują białka hemagglutinin bądź neuraminidase (są to dwa bardzo ważne białka obecne na powierzchni wirusa, są one często celem leków przeciwwirusowych, od ich typów wirusy grypy biorą swoje nazwy - np. H1N1).

Wymyśl i wypróbuj jakieś swoje zapytanie, używając spójników i pól wyszukiwania z **FieldList** (może być dla innej bazy niż nucleotide).

Zadanie 10. Za pomocą ESummary pobierz informację podsumowującą o rekordzie

Zmodyfikuj poprzedni przykład używając:

```
...
handle = Entrez.esummary(db="nucleotide", id="326579398")  # pojedynczy
rekord
```

Można też pobrać info o wielu rekordach, podając na wejściu listę identyfikatorów.

```
handle = Entrez.esummary(db="nucleotide", id=",".join(result["IdList"]))
#lista
                                                                rekordow
```

Otrzymujemy wówczas listę słowników – w każdym słowniku jest informacja o jednym rekordzie, którą można wypisać np. tak:

```
result = Entrez.read(handle)
for record in result:
    print("%(Title)s, %(Gi)s, %(Length)s %(CreateDate)s" % record)
```

Zadanie 11. Pobierz cały rekord

Aby pobrać cały rekord korzystamy z **EFetch**, możemy też wybrać inny format niż XML, np. podstawowe formaty jak genbank, fasta, genpept dla bazy Protein (dokładny opis formatów: www.ncbi.nlm.nih.gov/entrez/query/static/efetchseq_help.html)

```
handle = Entrez.efetch(db="nucleotide",
id="326579398",rettype="gb",retmode="text")
print(handle.read())
```

Plik w odpowiednim formacie możemy od razu wczytać do obiektu sekwencji SeqRecord:

```
from Bio import SeqIO

handle = Entrez.efetch(db="nucleotide",
id="326579398",rettype="gb",retmode="text")
record = SeqIO.read(handle, "genbank")

print(record.name)
print(record.description)
print(record.seq)
```

Zadanie 12. Historia sesji

NCBI zapewnia możliwość korzystania z historii sesji, stąd można usprawnić proces wyszukiwania i ściągania sekwencji za pomocą **ESearch** i **EFetch**, tak by robić do bez podawania listy uzyskanych ID, wykonując **ESearch** użyjmy opcji *usehistory*, dostaniemy dwie nowe pozycje - klucz zapytania i ciasteczko sesji:

```
handle = Entrez.esearch(db="protein",term="Homo sapiens[orgn] and
rhodopsin",
                        usehistory="y")
result = Entrez.read(handle)
querykey = result['QueryKey']
webenv = result['WebEnv']
```

Możemy ich teraz użyć bezpośrednio w **EFetch**, ponieważ z reguły wyniki takiego wyszukiwania będą bardzo duże, możemy wybrać zakres rekordów, który chcemy obejrzeć przez *retstart* i *retmax.*

```
handle = Entrez.efetch(db="protein", WebEnv=webenv, query_key=querykey,
                       retstart=0, retmax=2, rettype="gp", retmode="text")
print(handle.read())
```

Zadanie 13. Elink

Elink pozwala na wyszukanie linków pomiędzy bazami, np. mając ID genu chcemy przejść do sekwencji białkowej

```
handle = Entrez.elink(dbfrom="gene", db="protein", id=3043) # ID genu
hemoglobiny czlowieka
result = Entrez.read(handle)

for link in result[0]["LinkSetDb"][0]["Link"]:
    print(link["Id"])  #wypisze listę powiązanych identyfikatorów z bazy
protein
```

W pętli *for* pierwsze [0] dlatego, że szukamy linków dla pierwszego id (w naszym przypadku jest tylko jeden, ale parametrem elink może być również lista identyfikatorów). Drugie [0] dlatego, że szukamy dla pierwszej bazy docelowej (w naszym przypadku „protein”; uwaga j.w. – tu jest jedna baza docelowa, ale może być więcej). Dodatkowe informacje o parametrach Elinka znajdziesz na stronie **NCBI Eutils**.

Zadanie 14. (4 pkt)

Rozwiązanie zadanie prześlij mailem do wtorku, **29.10.2019** włącznie, na adres:
jacek.smietanski@ii.uj.edu.pl
Temat wiadomości proszę opatrzyć przedrostkiem **[Bio] Lab 03**. Rozwiązaniem ma być **tylko jeden plik** – skrypt lub notebook zgodny z Pythonem w wersji 3.x**,** zawierający wszystkie niezbędne funkcje oraz procedurę wykonawczą. Proszę o nazwanie pliku wg schematu**: Imie.Nazwisko.03.py**.

Napisz skrypt, który przy wykonuje kolejno następujące zadania:

1. Wyszukaj geny powiązane z nazwą BRCA1 u człowieka. Wypisz pola: ***Name, Chromosome, MapLocation i Description***
2. Zachowaj ***id*** genu o nazwie BRCA1
3. Wypisz tytuły rekordów bazy OMIM związane z tym genem
4. Znajdź białka powiązane z tym genem (baza *protein*): wypisz pola numeru identyfikacyjnego ***Gi*** dla wyszukanych sekwencji
5. Pobierz białko o numerze identyfikacyjnym **121949022** w formacie *genbank*, wypisz jego nazwę oraz sekwencję aminokwasów
6. Znajdź wszystkie mutacje w obrębie genu BRCA1 u człowieka (w *elink* do bazy SNP użyj *term="Homo sapiens"*), wypisz ile mutacji znaleziono, dla pierwszych 50 wyświetl ich numer identyfikacyjny ***SNP_ID***, klasę ***SNP_CLASS***, gen ***GENE*** i pozycję ***CONTIGPOS***
7. Na podstawie przeprowadzonych wyżej poszukiwań, sformułuj odpowiednie wnioski (jakie informacje udało Ci się uzyskać? jakie jest ich znaczenie?). Treść wniosku wyświetl na ekranie (*print*) lub zawrzyj w komentarzu na końcu pliku ze skryptem.

Wskazówka:

Wykorzystaj poznane na ćwiczeniach metody z modułu **Bio.Entrez** (**einfo, esearch, esummary, efetch, elink**).

Dodatkowe materiały:
- dokumentacja modułu Bio.Entrez: http://biopython.org/DIST/docs/api/Bio.Entrez-module.html
- przewodnik po Biopythonie, rozdział 9: http://biopython.org/DIST/docs/tutorial/Tutorial.html#chapter%3Aentrez