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

BLAST

1. Blast on-line

Zadanie 1.

1. wejdź na główną stronę NCBI, po prawej w dziale *popular resources* wybierz BLAST,
2. zobacz jakie główne odmiany blasta mamy do dyspozycji. Do czego służą? Jakie sekwencje porównują ze sobą?
3. poszukaj z jakiego genu pochodzi następująca sekwencja:

```
GTACCTTGATTTCGTATTCTGAGAGGCTGCTGCTTAGCGGTAGCCCCTTGGTTTCCGTGGCAACGGAAAA
GCGCGGGAATTACAGATAAATTAAAACTGCGACTGCGCGGCGTGAGCTCGCTGAGACTTCCTGGACGGGG
GACAGGCTGTGGGGTTTCTCAGATAACTGGGCCCCTGCGCTCAGGAGGCCTTCACCCTCTGCTCTGGGTA
AAGTTCATTGGAACAGAAAGAAATGGATTTATCTGCTCTTCGCGTTGAAGAAGTACAAAATGTCATTAAT
GCTATGCAGAAAATCTTAGAGTGTCCCATCTGTCTGGAGTTGATCAAGGAACCTGTCTCCACAAAGTGTG
ACCACATATTTTGCAAATTTTGCATGCTGAAACTTCTCAACCAGAAGAAAGGGCCTTCACAGTGTCCTTT
ATGTAAGAATGATATAACCAAAAGGAGCCTACAAGAAAGTACGAGATTTAGTCAACTTGTTGAAGAGCTA
TTGAAAATCATTTGTGCTTTTCAGCTTGACACAGGTTTGGAGTATGCAAACAGCTATAATTTTGCAAAAA
AGGAAAATAACTCTCCTGAACATCTAAAAGATGAAGTTTCTATCATCCAAAGTATGGGCTACAGAAACCG
TGCCAAAAGACTTCTACAGAGTGAACCCGAAAATCCTTCCTTGCAGGAAACCAGTCTCAGTGTCCAACTC
TCTAACCTTGGAACTGTGAGAACTCTGAGGACAAAGCAGCGGATACAACCTCAAAAGACGTCTGTCTACA
TTGAATTGGGATCTGATTCTTCTGAAGATACCGTTAATAAGGCAACTTATTGCAGTGTGGGAGATCAAGA
ATTGTTACAAATCACCCCTCAAGGAACCAGGGATGAAATCAGTTTGGATTCTGCAAAAAAGGCTGCTTGT
GAATTTTCTGAGACGGATGTAACAAATACTGAACATCATCAACCCAGTAATAATGATTTGAACACCACTG
AGAAGCGTGCAGCTGAGAGGCATCCAGAAAAGTATCAGGGTAGTTCTGTTTCAAACTTGCATGTGGAGCC
ATGTGGCACAAATACTCATGCCAGCTCATTACAGCATGAGAACAGCAGTTTATTACTCACTAAAGACAGA
ATGAATGTAGAAAAGGCTGAATTCTGTAATAAAAGCAAACAGCCTGGCTTAGCAAGGAGCCAACATAACA
GATGGGCTGGAAGTAAGGAAACATGTAATGATAGGCGGACTCCCAGCACAGAAAAAAAGGTAGATCTGAA
TGCTGATCCCCTGTGTGAGAGAAAAGAATGGAATAAGCAGAAACTGCCATGCTCAGAGAATCCTAGAGAT
ACTGAAGATGTTCCTTGGATAACACTAAATAGCAGCATTCAGAAAGTTAATGAGTGGTTTTCCAGAAGTG
ATGAACTGTTAGGTTCTGATGACTCACATGATGGGGAGTCTGAATCAAATGCCAAAGTAGCTGATGTATT
GGACGTTCTAAATGAGGTAGATGAATATTCTGGTTCTTCAGAGAAAATAGACTTACTGGCCAGTGATCCT
CATGAGGCTTTAATATGTAAAAGTGAAAGAGTTCACTCCAAATCAGTAGAGAGTAATATTGAAGACAAAA
TATTTGGGAAAACCTATCGGAAGAAGGCAAGCCTCCCCAACTTAAGCCATGTAACTGAAAATCTAATTAT
AGGAGCATTTGTTACTGAGCCACAGATAATACAAGAGCGTCCCCTC
```

Jako bazę do przeszukania wybierz *refseq_genomic*. Obejrzyj wyniki, zwróć uwagę na klikalne podsumowanie graficzne wyszukanych dopasowań, obejrzyj wyszukane dopasowania, zwróć uwagę na wartości *score, e-value, identities, gaps.*

4. wyszukaj białko o następującej sekwencji:

```
MKSILDGLADTTFRTITTDLLGSPFQEKMTAGDNPQLVPADQVNITEFYNKSLSSFKENEENIQCGENFM
DIECFMVLNPSQQLAIAVLSLTLGTFTVLENLLVLCVILHSRSLRCRPSYHFIGSLAVADLLGSVIFVYS
FIDFHVFHRKDSRNVFLFKLGGVTASFTASVGSLFLTAIDRYISIHRPLAYKRIVTRPKAVVAFCLMWTI
AIVIAVLPLLGWNCEKLQSVCSDIFPHIDETYLMFWIGVTSVLLLFIVYAYMYILWKAHSHAVRMIQRGT
QKSIIIHTSEDGKVQVTRPDQARMDIRLAKTLVLILVVLIICWGPLLAIMVYDVFGKMNKLIKTVFAFCS
```

```
MLCLLNSTVNPIIYALRSKDLRHAFRSMFPSCEGTAQPLDNSMGDSDCLHKHANNAASVHRAAESCIKST
VKIAKVTMSVSTDTSAEAL
```

Rozwiń menu *Algorithm parameters*, zobacz jak można zmieniać macierze substytucji, dla jakiej macierzy domyślnie wykonywany jest algorytm?
Jaka jest rola tego białka w organizmie człowieka? Na którym chromosomie występuje? Jakie inne zwierzęta posiadają dokładnie taką samą, bądź bardzo zbliżoną sekwencję tego białka? (nazwy organizmów łacińskie są w nawiasach kwadratowych w tabelce podsumowującej przeszukiwanie, zobacz w google co to za organizmy).

5. użyjemy jeszcze jednej odmiany BLASTA: tblastx, który może odnaleźć bardzo odległe ewolucyjnie powiązania, ponieważ tłumaczy sekwencję zapytania (*query)* w 6 ramkach odczytu i dopasowuje ją względem bazy danych również przetłumaczonej w 6 ramkach odczytu, stąd jest to najwolniejsza z odmian BLASTA.

Znajdź poprzez Entrez sekwencję rhodopsyny - światłoczułego barwnika występującego w siatkówce oka u człowieka (NM_000539.3), dla tej sekwencji użyj tblastx do znalezienia sekwencji podobnych – może to trwać dość długo, nawet kilka minut, w wynikach zwróć uwagę na wypisaną ramkę odczytu dla dopasowania.

2. BLAST w biopythonie

Zadanie 2.

```
from Bio.Blast import NCBIWWW
print(help(NCBIWWW.qblast))
```

Musimy podać jakiego typu BLASTA chcemy użyć, bazę, względem której będziemy przeszukiwać, oraz sekwencję, względnie numer identyfikacyjny GI:

```
from Bio import SeqIO

result_handle = NCBIWWW.qblast("blastn", "nr", "23527284")
print(result_handle.read())
s = SeqIO.read(open("sequence.fa"), format="fasta")
result_handle = NCBIWWW.qblast("blastn", "nt", s.seq)
```

Aby ograniczyć liczbę zwracanych wyników można ustawić parametr *hitlist_size*:

```
s = SeqIO.read(open("sequence.fa"), format="fasta")
result_handle = NCBIWWW.qblast("blastn", "nr", s.seq, hitlist_size=1)
```

Podstawowe bazy:
nr – nieredundantna, dla sekwencji białkowych,
nt – nieredundantna, dla sekwencji nukleotydowych
(inne możliwe bazy – opis:
https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=ProgSelectionGuide)

Domyślnie wynik zwracany jest w postaci XML'a. Możemy go sparsować przygotowanym do tego narzędziem.

```
from Bio.Blast import NCBIXML
blast_records = NCBIXML.parse(result_handle)
blast_record = next(blast_records)
```

Jeśli wyszukiwaliśmy większą liczbę sekwencji, będziemy mieć dla każdej sekwencji jeden wynik blasta, jeśli szukaliśmy tylko jednej to bierzemy pierwszy obiekt zwrócony przez iterator `blast_records`.
Blast record jest klasą stworzoną do obsługi wyników blasta:
biopython.org/DIST/docs/api/Bio.Blast.Record-module.html

Przeglądnijmy uzyskane dopasowania :

```
for alignment in blast_record.alignments:
    print('Alignment----------------------------------')
    print('title:', alignment.title)
    print('length:', alignment.length)
for hsp in alignment.hsps:
    print('HSP : ')
    print('e value:', hsp.expect)
    print(hsp.query[0:75] + '...')
    print(hsp.match[0:75] + '...')
    print(hsp.sbjct[0:75] + '...')
```

Można również uruchamiać BLASTA lokalnie, trzeba do tego ściagnąć z NCBI binarkę. Do komunikacji z lokalnym blastem można użyć wrappera z Biopython'a: *Bio.Blast.Applications* (dokumentacja http://biopython.org/DIST/docs/api/Bio.Blast.Applications-module.html). Lokalny BLAST będzie na pewno szybszy, możemy dla niego tworzyć również własne bazy danych do przeszukania.

Zadanie 3 (4pkt):

Rozwiązanie zadania prześlij mailem do poniedziałku, **19.11.2019** włącznie, na adres:
jacek.smietanski@ii.uj.edu.pl
Temat wiadomości proszę opatrzyć przedrostkiem **[Bio] Lab 06**. Rozwiązaniem ma być **tylko jeden plik** – skrypt zgodny z Pythonem w wersji 3.x**,** zawierający wszystkie niezbędne funkcje oraz procedurę wykonawczą. Proszę o nazwanie pliku wg schematu**: Imie.Nazwisko.06.py**.

Pamiętaj, że Twój kod powinien być czytelny i zrozumiały (nazewnictwo funkcji i zmiennych, docstringi, prezentacja wyników) oraz powinien dać się łatwo uruchomić dla sekwencji innych niż podane w zadaniu (parametry wywołania).

Wykonaj zadanie 1, punkty 3-5 przy użyciu biopythona.