

# MBI - Adnotacja DNA

Bartosz Latosek, Mateusz Krakowski

11 Kwiecień 2024

## 1 Wybór genomu

Numer indeksu - 310772 (mod 150 = **122**)

Do wykonania ćwiczenia został użyty *HDID\_scaffold0000113*.

## 2 Maskowanie genomu

Wykorzystuję program RepeatMasker:

```
sudo docker run -it --rm -v /tmp:/tmp
-w /tmp wkusmirek/repeatmasker RepeatMasker
--species arabidopsis /tmp/single_scaffold.fa
```

Wyjście programu:

```
analyzing file /tmp/single_scaffold.fa

Checking for E. coli insertion elements
identifying Simple Repeats in batch 1 of 3
identifying matches to arabidopsis sequences in batch 1 of 3
identifying Simple Repeats in batch 1 of 3

Checking for E. coli insertion elements
identifying Simple Repeats in batch 2 of 3
identifying matches to arabidopsis sequences in batch 2 of 3
identifying Simple Repeats in batch 2 of 3

Checking for E. coli insertion elements
identifying Simple Repeats in batch 3 of 3
identifying matches to arabidopsis sequences in batch 3 of 3
identifying Simple Repeats in batch 3 of 3
processing output:
cycle 1
cycle 2
cycle 3
cycle 4
cycle 5
cycle 6
cycle 7
cycle 8
cycle 9
cycle 10
Generating output...
masking
done
```

## 2.1 Ile nukleotydów zostało zamaskowanych?

Aby to obliczyć, wykorzystałem funkcję lupki w programie Visual Studio Code. W pliku single\_scaffold.fa znajduje się 480 symboli "N". W pliku single\_scaffold.fa.masked znajduje się 1126 symboli "N". Dzięki obliczeniu różnicy 1126 - 480 okazuje się, że zamaskowanych zostało 646 nukleotydów.

## 3 Czy zamaskowane nukleotydy były pojedynczymi nukleotydami, czy ciągami nukleotydów

Zamaskowane nukleotydy są ciągami.

### 3.1 Kolejnym etapem ćwiczenia będzie zmapowanie sekwencji mRNA i białek na genom z zamaskowanymi sekwencjami repetytywnymi. W jaki sposób maskowanie sekwencji repetytywnych może wpłynąć na wynik mapowania?

Maskowanie sekwencji repetytywnych może przyspieszyć dalszą pracę.

## 4 Mapowanie znanych sekwencji i adnotacja strukturalna

Generuję plik konfiguracyjny komendą:

```
sudo docker run --rm -v /tmp:/tmp -w /tmp wkusmirek/maker maker -CTL
```

Dodaję ścieżki do plików w konfiguracji

```
#-----Genome (these are always required)
genome= /tmp/single_scaffold.fa.masked #genome sequence (fasta file or fasta
    embedded in GFF3 file)
organism_type=eukaryotic #eukaryotic or prokaryotic. Default is eukaryotic

#-----Re-annotation Using MAKER Derived GFF3
maker_gff= #MAKER derived GFF3 file
est_pass=0 #use ESTs in maker_gff: 1 = yes, 0 = no
altest_pass=0 #use alternate organism ESTs in maker_gff: 1 = yes, 0 = no
protein_pass=0 #use protein alignments in maker_gff: 1 = yes, 0 = no
rm_pass=0 #use repeats in maker_gff: 1 = yes, 0 = no
model_pass=0 #use gene models in maker_gff: 1 = yes, 0 = no
pred_pass=0 #use ab-initio predictions in maker_gff: 1 = yes, 0 = no
other_pass=0 #passthrough anything else in maker_gff: 1 = yes, 0 = no

#-----EST Evidence (for best results provide a file for at least one)
est= /tmp/hymenolepis_diminuta.PRJEB507.WBPS10.mRNA_transcripts.fa #set of
    ESTs or assembled mRNA-seq in fasta format
altest= #EST/cDNA sequence file in fasta format from an alternate organism
est_gff= #aligned ESTs or mRNA-seq from an external GFF3 file
altest_gff= #aligned ESTs from a closely relate species in GFF3 format

#-----Protein Homology Evidence (for best results provide a file for at
    least one)
protein= /tmp/hymenolepis_diminuta.PRJEB507.WBPS10.protein.fa #protein
    sequence file in fasta format (i.e. from mutiple oransisms)
protein_gff= #aligned protein homology evidence from an external GFF3 file
```

Wykonuję program komendą:

```
sudo docker run --rm -v /tmp:/tmp -w /tmp wkusmirek/maker maker
```

```
## Pierwsze 10 linii pliku \\wsl.localhost\Ubuntu-22.04\tmp\single_scaffold.
fa_maker.output\single_scaffold.fa_datastore\93\9B\HDID_scaffold0000113
##gff-version 3
HDID_scaffold0000113      .      contig 1      123769      .      .      .
      ID=HDID_scaffold0000113;Name=HDID_scaffold0000113
###
HDID_scaffold0000113      repeatmasker      match      7009      7080      230      +
      .      ID=HDID_scaffold0000113:hit:0:1.3.0.0;Name=species:Copia
-14_ALY-I|genus:LTR%2FCopia;Target=species:Copia-14_ALY-I|genus:LTR%2
FCopia 2688 2760 +
HDID_scaffold0000113      repeatmasker      match_part      7009      7080      230
+      .      ID=HDID_scaffold0000113:hsp:0:1.3.0.0;Parent=
HDID_scaffold0000113:hit:0:1.3.0.0;Target=species:Copia-14_ALY-I|genus:
LTR%252FCopia 2688 2760 +
HDID_scaffold0000113      repeatmasker      match      52270      52308      238      +
      .      ID=HDID_scaffold0000113:hit:1:1.3.0.0;Name=species:Copia-4
_LH-I|genus:LTR%2FCopia;Target=species:Copia-4_LH-I|genus:LTR%2FCopia
1556 1594 +
HDID_scaffold0000113      repeatmasker      match_part      52270      52308      238
+      .      ID=HDID_scaffold0000113:hsp:1:1.3.0.0;Parent=
HDID_scaffold0000113:hit:1:1.3.0.0;Target=species:Copia-4_LH-I|genus:LTR
%252FCopia 1556 1594 +
HDID_scaffold0000113      repeatmasker      match      55164      55265      245      +
      .      ID=HDID_scaffold0000113:hit:2:1.3.0.0;Name=species:
Helitron-1B_ALy|genus:RC%2FHelitron;Target=species:Helitron-1B_ALy|genus:
RC%2FHelitron 3188 3291 +
HDID_scaffold0000113      repeatmasker      match_part      55164      55265      245
+      .      ID=HDID_scaffold0000113:hsp:2:1.3.0.0;Parent=
HDID_scaffold0000113:hit:2:1.3.0.0;Target=species:Helitron-1B_ALy|genus:
RC%252FHelitron 3188 3291 +
HDID_scaffold0000113      repeatmasker      match      118808      119905      1010      +
      .      ID=HDID_scaffold0000113:hit:3:1.3.0.0;Name=species:Mariner
-6_ACe|genus:DNA%2FTcMar-Tc1;Target=species:Mariner-6_ACe|genus:DNA%2
FTcMar-Tc1 371 1095 +
```

#### 4.1 Jakie informacje można odczytać z wygenerowanego pliku .gff?

W pliku .gff znajdują się informacje o (informacja - przykład):

- wersja gff - gff-version 3
- nazwa sekwencji - HDID\_scaffold0000113
- źródło danych - repeatmasker
- typ zdarzenia - match, match\_part
- identyfikator rozpoczęcia podsekwencji - 7009
- identyfikator zakończenia podsekwencji - 7080
- ocena podsekwencji - 230
- typ nici - "+" do przodu, "-" do tyłu
- pozycja w ramce - . (brak informacji)
- dodatkowe informacje o podsekwencji -

## 4.2 Oblicz ilość wygenerowanych zdarzeń typu `expressed_sequence_match` i `protein_match`. Co oznaczają wymienione typy zdarzeń?

Znów wykorzystam funkcję lupki w Visual Studio Code. sprawdziłem, że zdarzeń typu `expressed_sequence_match` jest 20, a `protein_match` jest 26.

Zdarzenie `"expressed_sequence_match"` sygnalizuje, że w analizowanym pliku zawierającym transkrypty odnaleziono sekwencję, która pokryła się z pewnym obszarem zmaskowanego kontigu.

Natomiast zdarzenie `"protein_match"` oznacza, że w badanym pliku zawierającym białka znaleziono konkretną sekwencję białka, która powstała w wyniku translacji fragmentu sekwencji z zmaskowanego kontigu.

## 5 Adnotacja funkcjonalna

Do badania adnotacji wybrano wiersz:

```
HDID_scaffold0000113      blastn      expressed_sequence_match      2528
11911      318      +      .      ID=HDID_scaffold0000113:hit:5:3.2.0.0;
Name=HDID_0000784001-mRNA-1
```

The screenshot displays the NCBI BLAST results interface. On the left, a sidebar contains metadata for the query: RID (QS4YM4CM016), Program (BLASTX), Database (nr), Query ID (lcl|Query\_10844510), Description (None), Molecule type (dna), Query Length (9384), and Other reports. The main panel shows the 'Sequences producing significant alignments' table. The table has columns for Description, Scientific Name, Max Score, Total Score, Query Cover, E value, Per. Ident, Acc. Len, and Accession. The first five rows are highlighted in blue, indicating they are the most similar to the query. The first row is 'unnamed protein product [Hymenolepis diminuta]' with an E value of 2e-59. The second row is 'major facilitator superfamily domain containing protein [Hymenolepis microstoma]' with an E value of 2e-41. The third row is 'hypothetical transcript [Hymenolepis microstoma]' with an E value of 3e-41. The fourth row is 'unnamed protein product [Rodentolepis nana]' with an E value of 3e-39. The fifth row is 'unnamed protein product [Rodentolepis nana]' with an E value of 6e-21. The sixth row is 'unnamed protein product [Hymenolepis diminuta]' with an E value of 3e-33. The seventh row is 'hypothetical transcript [Hymenolepis microstoma]' with an E value of 3e-26. The eighth row is 'unnamed protein product [Taenia asiatica]' with an E value of 2e-07.

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
unnamed protein product [Hymenolepis diminuta]	Hymenolepis diminuta	224	763	14%	2e-59	89.26%	415	VDL60156.1
major facilitator superfamily domain containing protein [Hymenolepis microstoma]	Hymenolepis microstoma	164	286	7%	2e-41	65.29%	194	CDS34941.1
hypothetical transcript [Hymenolepis microstoma]	Hymenolepis microstoma	164	272	7%	3e-41	65.29%	200	CDS34940.1
unnamed protein product [Rodentolepis nana]	Rodentolepis nana	158	266	7%	3e-39	61.98%	204	VDN98927.1
unnamed protein product [Rodentolepis nana]	Rodentolepis nana	115	731	13%	6e-21	70.27%	957	VDO05606.1
unnamed protein product [Hymenolepis diminuta]	Hymenolepis diminuta	141	380	6%	3e-33	100.00%	208	VUZ40186.1
hypothetical transcript [Hymenolepis microstoma]	Hymenolepis microstoma	124	449	9%	3e-26	79.73%	295	CUU98258.1
unnamed protein product [Taenia asiatica]	Taenia asiatica	71.2	263	6%	2e-07	47.30%	804	VDK35326.1

Rysunek 1: Wynik porównania wybranej podsekwencji

Jak widać, najbardziej sekwencja jest podobna do organizmu `hymenolepis_diminuta`, czyli naszego organizmu. Pięć najbardziej podobnych sekwencji do badanej, podobieństwo i E-Value widoczne jest na Rysunku 1:

- unnamed protein product [Hymenolepis diminuta] (organizm badany)
- major facilitator superfamily domain containing protein [Hymenolepis microstoma]
- hypothetical transcript [Hymenolepis microstoma]
- unnamed protein product [Rodentolepis nana]
- unnamed protein product [Rodentolepis nana]

## 5.1 Co oznacza oraz jak interpretować wartość E-Value?

Wartość E-Value należy rozumieć, jako miarę prawdopodobieństwa, że wynik sekwencji został uzyskany przypadkowo. Im mniejsza wartość E-value, tym bardziej istotne jest dopasowanie sekwencji, czyli szansa że dopasowanie jest przypadkowe jest wprost proporcjonalna do E-Value.

## 5.2 Zinterpretuj listę uzyskanych organizmów (w ćwiczeniu pracujemy na genomie tasiemca szczurzego *Hymenolepis diminuta*)

Pierwszym organizmem jest badany przez nas Tasiemiec szczurzy, *Hymenolepis microstoma* i następnie *Rodentolepis nana* czyli tasiemiec karłowaty. Najwidoczniej są to organizmy najbardziej podobne genetycznie do badanego przez nas tasiemca szczurzego.

## 6 Zadanie implementacji

Skrypt:

```
import argparse
from Bio import SeqIO
from Bio.Seq import Seq
from Bio.SeqRecord import SeqRecord

translation_dict = {
    "A": "GCU", "R": "CGU", "N": "AAU", "D": "GAU", "C": "UGU",
    "Q": "CAA", "E": "GAA", "G": "GGU", "H": "CAU", "I": "AUU",
    "L": "UUG", "K": "AAA", "M": "AUG", "F": "UUU", "P": "CCU",
    "S": "AGU", "T": "ACU", "W": "UGG", "Y": "UAU", "V": "GUU",
    "*": "UAA"
}

def retranslate_protein_to_mRNA(input_filename, output_filename):
    with open(input_filename) as input_handle:
        with open(output_filename, "w") as output_handle:
            for seq_record in SeqIO.parse(input_handle, "fasta"):
                translated_seq = ''.join(translation_dict.get(letter, 'NNN')
                                         for letter in seq_record.seq)
                mRNA_seq_record = SeqRecord(Seq(translated_seq), id=
                    seq_record.id, description="Retranslated sequence")
                SeqIO.write(mRNA_seq_record, output_handle, "fasta")

if __name__ == "__main__":
    parser = argparse.ArgumentParser(
        prog='mbi2',
        description='Script to retranslate protein to mRNA. The program requires filenames (input and output) as arguments.'
    )
    parser.add_argument('input_filename', help='Input FASTA file containing protein sequences')
    parser.add_argument('output_filename', help='Output FASTA file to store retranslated mRNA sequences')
    args = parser.parse_args()

    retranslate_protein_to_mRNA(args.input_filename, args.output_filename)
```

Na potrzebę zadania stworzony został słownik, w którym dla danego aminokwasu przypisujemy sekwencję nukleotydów. Jeśli danego oznaczenia aminokwasu nie ma w słowniku, to podmieniamy go na sekwencję "NNN".

Skrypt dodatkowo korzysta z biblioteki argparse, aby go uruchomić należy z linii komend wywołać skrypt z dwoma argumentami w następujący oznaczającymi plik wejściowy i wyjściowy w sposób:

```
python .\translate.py .\hymenolepis_diminuta.PRJEB507.WBPS10.protein.fa  
output.fa
```

Pierwsza sekwencja pliku wejściowego:

```
>HDID_0000000001-mRNA-1 transcript=HDID_0000000001-mRNA-1 gene=  
HDID_0000000001  
MPISVRQFLVVMLFGATLALASFSPESAKEHLEERMLEEDENFDGPGGEFIGELGFGVPYI  
KKNANFWKKSRFWKRANPQFWKRGGSRFW
```

Wynik działania skryptu:

```
>HDID_0000000001-mRNA-1 Retranslated sequence  
AUGCCUAUUAGUGUUCGUCAAUUUUUGGUUGUUAUGUUGUUUGGUGCUACUUUGGCUUUG  
GCUAGUUUUAGUCCUGAAAGUGCUAAAGAACAUUUGGAAGAACGUAUGUUGGAAGAAGAU  
GAAAAUUUUGAUGGUCCUGGUGAAUUUAUUGGUGAAUUGGGUUUUGGUGUCCUUAUAUU  
AAAAAAAAAUGCUAUUUUUUGGAAAAAAGUCGUUUUUGGAAACGUGCUAUCCUCAUUUU  
UGGAAACGUGGUGGUAGUCGUUUUUGG
```