

MBI - Sprawozdanie Asemblacja de novo DNA

Bartosz Latosek, Mateusz Krakowski

March 2024

1 Wybór genomu

Numer indeksu - 310790 (mod 150 = **140**)

Do wykonania ćwiczenia został użyty genom bakterii *Campylobacter jejuni*.

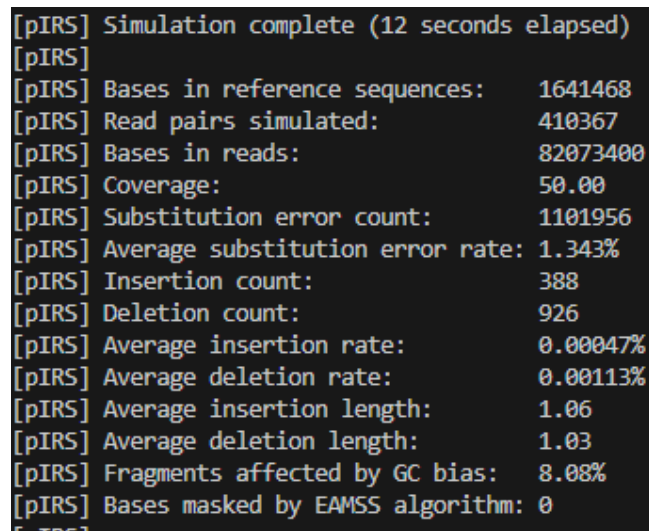
(Plik *GCF_002209025.1_ASM220902v1/GCF_002209025.1_ASM220902v1_genomic.fna.gz*)

2 Generowanie odczytów

Wykonano polecenie:

```
docker run --rm -v ./:/tmp -w /tmp wkusmirek/pirs pirs simulate -x 50 -m 400  
-v 20 -l 100 --error-rate=0.01 ./GCF_002209025.1_ASM220902v1_genomic.fna
```

Uzyskane wyniki:



```
[pIRS] Simulation complete (12 seconds elapsed)  
[pIRS]  
[pIRS] Bases in reference sequences: 1641468  
[pIRS] Read pairs simulated: 410367  
[pIRS] Bases in reads: 82073400  
[pIRS] Coverage: 50.00  
[pIRS] Substitution error count: 1101956  
[pIRS] Average substitution error rate: 1.343%  
[pIRS] Insertion count: 388  
[pIRS] Deletion count: 926  
[pIRS] Average insertion rate: 0.00047%  
[pIRS] Average deletion rate: 0.00113%  
[pIRS] Average insertion length: 1.06  
[pIRS] Average deletion length: 1.03  
[pIRS] Fragments affected by GC bias: 8.08%  
[pIRS] Bases masked by EAMSS algorithm: 0  
[pIRS]
```

Rysunek 1: Uzyskane wyniki

Pytania:

2.1 Ile zostało wygenerowanych odczytów? Jakiej długości?

Zostało wygenerowanych **410367** par odczytów o długości **100** par zasad.

2.2 Oblicz wygenerowaną głębokość pokrycia genomu odczytami. Czy wynik jest przybliżony do zakładanego poziomu 50x?

$$depth = \frac{bases.in.reads}{bases.in.reference.sequence} = \frac{82073400}{1641468} \approx 50 \quad (1)$$

Wynik jest zbliżony do poziomu 50.

2.3 W jaki sposób można znaleźć odczyty, które zawierają błędy?

Informacje o odczytach zawierających błędy znajdują się w pliku *Sim_100_400.read.info*.

```
1 # File "/sim_100_400.read.info": generated at Wed Mar 13 16:32:00 2024 by the command:
2 # pirs simulate -x 50 -m 400 -v 20 -l 100 --error-rate=0.01 /GCF_002209025.1_ASM220902v1_genomic.fna
3 #
4 # This file is a log of every read that was simulated. It shows
5 # exactly where each read came from and the substitution errors,
6 # insertions, and deletions (if any) that were made to it.
7 #
8 # The following lists the parameters of the simulation:
9 #
10 # Input reference sequence files: /GCF_002209025.1_ASM220902v1_genomic.fna
11 # Read length: 100
12 # Insert length mean: 400
13 # Insert length standard deviation: 20
14 # Coverage: 50
15 # Diploid: no
16 # Cylized (jumping library): no
17 # Simulate substitution errors: yes
18 # Substitution error rate: 0.01
19 # Base-calling profile: /usr/local/share/pirs/Base-Calling_Profiles/humNew.PE100.matrix.gz
20 # Substitution error algorithm: quality transition mode algorithm
21 # Simulate InDel errors: yes
22 # InDel error profiles: /usr/local/share/pirs/InDel_Profiles/phixv2.InDel.matrix
23 # Simulate GC content bias: yes
24 # GC bias profile: /usr/local/share/pirs/GC-depth_Profiles/humNew.gcdep_200.dat
25 # Output type: text
26 # Output directory: -
27 # Indiv name: -
28 # Simulate quality values: yes
29 # ASCII shift of quality value 33
30 # Mode of mask quality: None
31 # Random seed: 1718347529
32 # Number of simulator threads: 2
33 #
34 # readid referencefile contig/scaffold/chromosome position orientation insertSize maskIndex substitutions insertions deletions
35 read_400_1/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1085295 + 414 0 - - -
36 read_400_1/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1085295 414 0 31,A->C;32,T->A; - - -
37 read_400_2/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1165192 + 391 0 14,A->G;16,G->A;62,C->G; - - -
38 read_400_2/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1165192 391 0 - - -
39 read_400_3/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1559542 + 387 0 - - -
40 read_400_3/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1559542 387 0 - - -
41 read_400_4/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1559542 387 0 86,T->G;88,A->C; - - -
42 read_400_4/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1015249 + 419 0 - - -
43 read_400_5/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1015249 419 0 37,T->A;80,T->A; - - -
44 read_400_5/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1251268 + 379 0 5,T->C;52,C->A; - - -
45 read_400_6/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1251268 379 0 - - -
46 read_400_6/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1224072 + 379 0 99,C->A; - - -
47 read_400_7/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1224072 379 0 20,A->C;27,C->A;100,T->G; - - -
48 read_400_7/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 644502 - 375 0 - - -
49 read_400_8/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 644502 + 375 0 5,T->A; - - -
50 read_400_8/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 151969 432 0 73,A->C; - - -
51 read_400_9/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 151969 + 432 0 - - -
52 read_400_9/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 983818 - 363 0 15,A->C;17,A->C;99,A->T;100,T->C; - - -
53 read_400_10/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 983818 + 363 0 20,T->C;85,T->G; - - -
54 read_400_10/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 821459 + 391 0 - - -
55 read_400_11/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1361084 422 0 2,T->A;88,A->C;98,A->T; - - -
56 read_400_11/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1361084 + 422 0 13,G->C;80,A->T;94,T->C; - - -
57 read_400_12/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1340722 + 374 0 - - -
58 read_400_12/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1340722 374 0 3,T->A;84,T->G;90,A->C;93,T->G;95,T->C;96,G->A; - - -
59 read_400_13/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 541844 + 426 0 - - -
60 read_400_13/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 541844 426 0 - - -
61 read_400_14/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 509792 - 366 0 - - -
62 read_400_14/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 509792 366 0 - - -
63 read_400_15/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1211393 + 357 0 13,T->C; - - -
64 read_400_15/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 1211393 357 0 - - -
65 read_400_16/1 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 933509 - 385 0 52,C->T;82,T->G; - - -
66 read_400_16/2 /GCF_002209025.1_ASM220902v1_genomic.fna NZ_CP022077.1 Campylobacter jejuni strain FDAARGOS_263 chromosome, complete genome 933509 + 385 0 94,C->T; - - -
```

Rysunek 2: Fragment pliku *Sim_100_400.read.info*

2.4 Odczytaj z wygenerowanych plików odległości pomiędzy sparowanymi odczytami. Czy wartości zgadzają się z ustawianymi parametrami aplikacji?

Dane na temat długości odczytu znajdują się w pliku *Sim_100_400.insert.len.distr*, a sama statystyka dotycząca całości jest dostępna w nagłówku:

```
# This file shows the length distribution of the simulated inserts.
# We were trying to simulate inserts with a mean length of 400 and a
# standard deviation of 20. The actual mean is 399.526, and the actual
# standard deviation is 20.0181.
```

Rysunek 3: Nagłówek pliku *Sim_100_400.insert.len.distr*

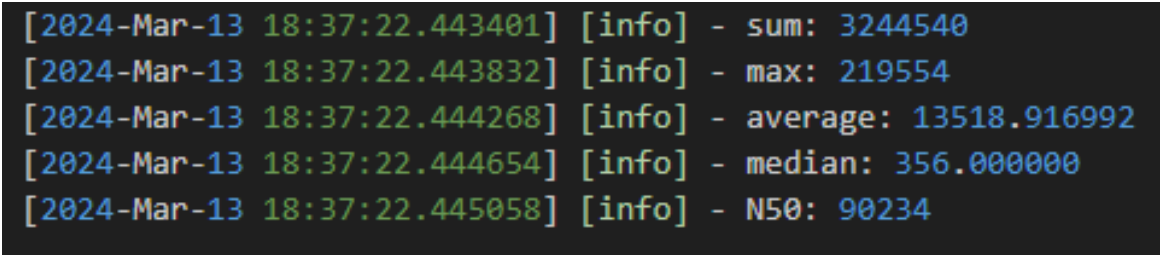
Z tego wynika, że średnia i odchylenie standardowe są bliskie zadanym w parametrach wywołania.

3 Asemblacja de novo

Wykonano polecenie:

```
docker run --rm -v ./:/tmp -w /tmp wkusmirek/dnaasm dnaasm -assembly -k 55
-genome_length 650000 -insert_size_mean_inward 400
-insert_size_std_dev_inward 20 -single_edge_counter_threshold 5
-i1_1 Sim_100_400_1.fq -i1_2 Sim_100_400_2.fq
-output_file_name contigs.fa
```

Uzyskane wyniki widoczne są w pliku *dnaasm/dnaasm_calc_0.log*:



```
[2024-Mar-13 18:37:22.443401] [info] - sum: 3244540
[2024-Mar-13 18:37:22.443832] [info] - max: 219554
[2024-Mar-13 18:37:22.444268] [info] - average: 13518.916992
[2024-Mar-13 18:37:22.444654] [info] - median: 356.000000
[2024-Mar-13 18:37:22.445058] [info] - N50: 90234
```

Rysunek 4: Zawartość pliku *dnaasm/dnaasm_calc_0.log*

Pytania:

3.1 Czy suma długości wygenerowanych sekwencji jest w przybliżeniu równa długości badanego genomu? Dlaczego?

Suma długości jest ok. dwukrotnie większa od długości badanego genomu. Wynika to z nadmiarowości sekwencjonowania. Wiemy, że genom referencyjny ma 1 chromosom, ale uzyskano 293 sekwencje, których nie udało się złączyć w jeden kontig. Nadmiarowość sekwencjonowania jest stosowana w celu umożliwienia nałożenia na siebie sufiksów i prefiksów odczytów. Większą długość wynika z tego, że nie wszystkie sufiksy udało się połączyć z prefiksami.

3.2 Czy plik z sekwencjami wynikowymi jest w formacie FASTA czy FASTQ?

Plik zapisany jest w formacie **FASTA** (Brak informacji o jakości danych).

3.3 Czy możliwa jest konwersja pliku w formacie FASTA na FASTQ? Jaką informację należy wówczas sztucznie wygenerować?

Jest możliwe. Wymagane jest wygenerowanie informacji o jakości danych.

3.4 Czy możliwa jest konwersja pliku w formacie FASTQ na FASTA? Jaka informacja przy takiej konwersji zostaje utracona?

Jest możliwa. Przy takiej konwersji tracimy informację o jakości danych.

4 Sprawdzenie wyników

Wykonano polecenie:

```
docker run --rm -v ./:/tmp -w /tmp/mbi
wkusmirek/quast quast.py -R ./GCF_002209025.1_ASM220902v1_genomic.fna
contigs.fa
```

Uzyskane wyniki dostępne są w pliku `quast_results/results_< cur_date >/report.txt`:

```
1 All statistics are based on contigs of size >= 500 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs).
2
3 Assembly                contigs
4 # contigs (>= 0 bp)      240
5 # contigs (>= 1000 bp)   90
6 # contigs (>= 5000 bp)   66
7 # contigs (>= 10000 bp)  54
8 # contigs (>= 25000 bp)  36
9 # contigs (>= 50000 bp)  24
10 Total length (>= 0 bp)   3250990
11 Total length (>= 1000 bp) 3214738
12 Total length (>= 5000 bp) 3162759
13 Total length (>= 10000 bp) 3072860
14 Total length (>= 25000 bp) 2791182
15 Total length (>= 50000 bp) 2368133
16 # contigs                108
17 Largest contig           219554
18 Total length             3226653
19 Reference length         1641468
20 GC (%)                   30.47
21 Reference GC (%)        30.55
22 N50                     90234
23 NG50                    122021
24 N75                     47244
25 NG75                    102888
26 L50                     13
27 LG50                    5
28 L75                     26
29 LG75                    9
30 # misassemblies         1
31 # misassembled contigs   1
32 Misassembled contigs length 102888
33 # local misassemblies    0
34 # unaligned mis. contigs 0
35 # unaligned contigs      0 + 0 part
36 Unaligned length         0
37 Genome fraction (%)      98.136
38 Duplication ratio        2.003
39 # N's per 100 kbp       0.00
40 # mismatches per 100 kbp 0.12
41 # indels per 100 kbp     0.06
42 Largest alignment        219554
43 Total aligned length     3226079
44 NA50                     90234
45 NGA50                    122021
46 NA75                     47244
47 NGA75                    96438
48 LA50                     13
49 LGA50                    5
50 LA75                     26
51 LGA75                    9
52
```

Rysunek 5: Zawartość pliku `quast_results/results_< cur_date >/report.txt`

Pytania:

4.1 Czy asemblacja de novo genomu pozwoliła uzyskać satysfakcjonujące wyniki? Dlaczego?

Asemblacja de novo przyniosła zadowalające rezultaty. Nie ma żadnych poważnych błędów (np. *missassemblies* czy *unaligned contigs*), a liczba mniej istotnych niedopasowań jest minimalna: jedynie **0.06** na każde **100** tysięcy par zasad.

4.2 Czym są translokacje w genomie? Czy aplikacja QUAST dostarcza informacji o liczbie translokacji w wynikach asemblacji de novo względem genomu referencyjnego? Jeśli tak, to skąd mogą być takie informacje odczytane?

Translokacje to przemieszczenia się kompletnych fragmentów chromosomów do innych lokalizacji w tym samym chromosomie lub do innego chromosomu. **Quast** dostarcza informacji o translokacjach w odniesieniu do genomu referencyjnego, które można znaleźć w pliku `quast_results/results_< cur_date >/contigs_reports/misassemblies_report.txt`.

```

1 All statistics are based on contigs of size >= 500 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs)
2
3 Assembly          contigs
4 ✓ # misassemblies  1
5   # relocations    1
6   # translocations  0
7   # inversions     0
8   # misassembled contigs 1
9   Misassembled contigs length 102888
10  # local misassemblies 0
11  # unaligned mis. contigs 0
12  # mismatches      2
13 ✓ # indels        1
14   # indels (<= 5 bp) 0
15   # indels (> 5 bp)  1
16   Indels length      84
17

```

Rysunek 6: Zawartość pliku `quast_results/results_< cur_date >/contigs_reports/misassemblies_report.txt`

5 Zadanie implementacyjne

5.1 Treść

Proszę zapoznać się z tematyką par GC(GC-content). Proszę odczytać z aplikacji QUASt zawartość par GC w badanym podczas ćwiczenia genomie referencyjnym. Następnie proszę zaimplementować prosty skrypt umożliwiający odczytać zawartość par GC w pliku w formacie FASTA. Proszę wykorzystać biblioteki dedykowane do przetwarzania danych genomowych, np.:

- SeqAn
- Biopython
- BioJava

Proszę porównać wyniki dostarczane przez aplikację QUASt oraz zaimplementowany skrypt.

5.2 Rozwiązanie

GC-content (zawartość GC) jest miarą proporcji zasad azotowych guaniny (G) i cytozyny (C) w sekwencji DNA lub RNA. Tematyka dotycząca zawartości GC obejmuje analizę różnych aspektów związanych z występowaniem tych zasad w genomach organizmów.

Zawartość GC można uzyskać ze wzoru:

$$GC = \frac{G + C}{A + T + G + C} \times 100\% \quad (2)$$

Dla każdego kontigu i mamy zadany stosunek GC, który możemy wyliczyć przy pomocy biblioteki Biopython. Sumaryczna zawartość GC dla kontigów jest obliczana poprzez zsumowanie zawartości GC dla każdego kontigu, uwzględniając ich długości. Istnieje również możliwość agregacji sekwencji w celu obliczenia zawartości GC dla całej sekwencji, jednak istnieje ryzyko, że może to być bardziej kosztowne, szczególnie gdy mamy już obliczone wartości GC dla podsekwencji o znanych długościach.

Stworzony skrypt prezentuje się w następujący sposób:

```

1 import argparse
2 from Bio import SeqIO, SeqUtils
3
4
5 def gc_content(filename: str) -> float:
6     """
7     Calculate the GC content of a FASTA file.
8     :param filename: str: Path to the FASTA file
9     :return: float: GC content in percentage
10    """
11    with open(filename) as f:
12        top, bottom = 0, 0
13        for record in SeqIO.parse(f, "fasta"):
14            top += SeqUtils.gc_fraction(record.seq) * len(record.seq)
15            bottom += len(record.seq)
16
17    return (top / bottom) * 100
18
19
20 if __name__ == "__main__":
21     parser = argparse.ArgumentParser()
22     parser.add_argument('--filename', help='Input file in FASTA format', type=str, required=True)
23     args = parser.parse_args()
24
25     result = gc_content(args.filename)
26     print(f"GC-Content: {result:0.2f} %")
27

```

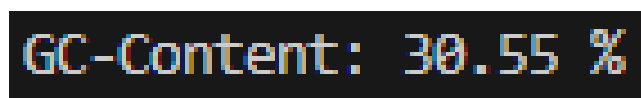
Rysunek 7: Stworzony skrypt *gc_content.py*

5.3 Wyniki

Wywołana komenda dla *GCF_002209025.1_ASM220902v1_genomic.fna*:

```
python3 .\gc_content.py --filename .\GCF_002209025.1_ASM220902v1_genomic.fna
```

Wynik:



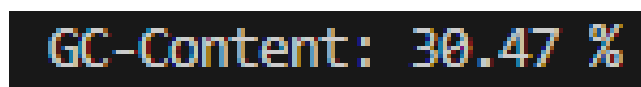
```
GC-Content: 30.55 %
```

Rysunek 8: Wynik działania skryptu dla pliku *GCF_002209025.1_ASM220902v1_genomic.fna*

Wywołana komenda dla :

```
python3 .\gc_content.py --filename ./contigs.fa
```

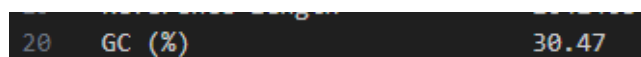
Wynik:



```
GC-Content: 30.47 %
```

Rysunek 9: Wynik działania skryptu dla pliku *contigs.fa*

Wynikowe wartości dla obydwu plików pokrywają się z wartością zwróconą przez **QUAST**.



```
20 GC (%) 30.47
```

Rysunek 10: Wartość zwrócona przez **QUAST**