

MBI - Sprawozdanie Resekwencjonowanie genomu człowieka

Bartosz Latosek, Mateusz Krakowski

April 2024

1 Mapowanie

Wykonano polecenie w celu zaindeksowania pliku:

```
sudo docker run -v ./:/data quay.io/biocontainers/bwa:0.7.17--hed695b0_7 bwa index data/chr1.fa
```

Następnie poniższym poleceniem wykonano mapowanie, co utworzyło plik **SAM**:

```
sudo docker run -v ./:/data quay.io/biocontainers/bwa:0.7.17--hed695b0_7 bwa mem -t 4 /data/chr1.fa /data/coriell_chr1.fq -o /data/coriell_chr1.sam
```

Uzyskane wyniki dostępne są w pliku *SAM*:

```
1 @SQ SN:chr1 LN:249258621
2 @PG ID:bwa PN:bwa VN:0.7.17-r1188 CL:bwa mem -t 4 /data/chr1.fa /data/coriell_chr1.fq -o /data/coriell_chr1.sam
3 M01956:55:000000000-BDTHK:1:2114:23661:3523 16 chr1 14883 60 73M * 0 0 CAACCACTTGAGCAAACTCCAAGACATCTTCTACCCCAACAC
4 M01956:55:000000000-BDTHK:1:1112:29336:15225 0 chr1 17434 60 76M * 0 0 ACATCATCAGGGGCACAGCGTGACGTGGGGTCCAG
5 M01956:55:000000000-BDTHK:1:1112:29336:15225 16 chr1 17620 60 75M * 0 0 GTTGAAGAGCAGCAAGGAGCTGACAGAGCTGATGTTC
6 M01956:55:000000000-BDTHK:1:2117:6824:15111 0 chr1 19244 60 76M * 0 0 TCCACGTGCAGAGCAGCTCAGCACTCACCAGGACAGGCGA
7 M01956:55:000000000-BDTHK:1:2117:6824:15111 16 chr1 19319 60 76M * 0 0 GCACGACTGGGGTTTCAGGAGAGGGCAGGAGGGGTGTGG
8 M01956:55:000000000-BDTHK:1:2106:12104:23462 0 chr1 20941 60 76M * 0 0 TAATTTGCCAGGAGCTCACTGCTGCTGCTGCTGGGCA
9 M01956:55:000000000-BDTHK:1:2106:12104:23462 16 chr1 21161 60 76M * 0 0 AGTGCCACAGGAGGGGCAAGTGAGGAGGAGAGGTGGC
10 M01956:55:000000000-BDTHK:1:2114:5181:12460 0 chr1 28825 60 75M * 0 0 CGGAGTTACAGGCTCGCTGTAGGCTCCCGGGAACCCACCGG
11 M01956:55:000000000-BDTHK:1:2114:5181:12460 16 chr1 28938 60 76M * 0 0 CGAGGCTTCCCAGAACCCGGAAGGGGCGGAAGACGACGAGAG
12 M01956:55:000000000-BDTHK:1:1111:23811:8767 0 chr1 32721 60 76M * 0 0 AGGAAGCCTCTGCAGCCAGGAACCTCCCTTATCGGAATG
13 M01956:55:000000000-BDTHK:1:1111:23811:8767 16 chr1 32841 60 76M * 0 0 CAGGAGAGTATAGTGGTTACTAGGAAGGGAAGGACTTTGGGA
14 M01956:55:000000000-BDTHK:1:1106:5887:4936 0 chr1 34881 60 76M * 0 0 ATTTATAGCACTAAATGCCCAAGAGACCTCTGCTGAGAA
15 M01956:55:000000000-BDTHK:1:1106:5887:4936 16 chr1 34283 60 76M * 0 0 CGACTTGCTCTCACATCTCTTGGCCAGCACTGGACACACA
16 M01956:55:000000000-BDTHK:1:1119:13255:1151 0 chr1 35769 60 76M * 0 0 ATGCATCCTCTCGGGGCAGCACTGCTGCTCCGAGGTGAGA
17 M01956:55:000000000-BDTHK:1:1119:13255:1151 16 chr1 35912 60 75M * 0 0 CTGCCCCGTCTGCAACTTTGGAGGAGAAATGGCGTGAAGGG
18 M01956:55:000000000-BDTHK:1:2117:18564:4004 0 chr1 38015 60 75M * 0 0 GGGACGGTGTATGTGTAGTCCAGTAACACAGCCAGACCTT
19 M01956:55:000000000-BDTHK:1:2117:18564:4004 16 chr1 38075 60 13M1162M * 0 0 ACAGGAGCCTCTCAAACTGCTCTCTGCTTCCAT
20 M01956:55:000000000-BDTHK:1:2106:26386:22085 0 chr1 40663 60 73M * 0 0 CAGGTTCTCTGCTGCACTCCAGCTGGAGTGCAGTGACATG
21 M01956:55:000000000-BDTHK:1:2106:26386:22085 4 * 0 0 * CTGGTCTGCTGCTGGACTTCAGGGGTGCCGAGCTTTGGGTAGCA
22 M01956:55:000000000-BDTHK:1:1106:5090:19094 0 chr1 43663 60 75M * 0 0 TATCGTACTAAAGTCTAGCCAGGACAATTAGACAAATAA
23 M01956:55:000000000-BDTHK:1:1106:5090:19094 16 chr1 43752 60 45M1D30M * 0 0 ACAGATAACATAATCTTATATGTAGAAACCCCTT
24 M01956:55:000000000-BDTHK:1:1104:4586:14107 0 chr1 44216 60 75M * 0 0 TGAATAGCTAAAGCAATCTTGAGTAAGATGAACACACTGGA
25 M01956:55:000000000-BDTHK:1:1104:4586:14107 16 chr1 44349 60 76M * 0 0 GGAATGGAATAAAGAGCACAGAATAAGTCCACACATTTACA
26 M01956:55:000000000-BDTHK:1:1106:27774:10855 0 chr1 46677 60 76M * 0 0 TCACAATTGAGTTACATTAGCCCTGCAATCATGTAG
27 M01956:55:000000000-BDTHK:1:1106:27774:10855 16 chr1 46822 60 76M * 0 0 GATGAAGATAGATAGGATGGTGCTCTACACATACCTT
28 M01956:55:000000000-BDTHK:1:2117:19311:17634 0 chr1 57529 60 76M * 0 0 AGCTTCCTTTCCAATATGAAGAATCTTATATAGCTT
29 M01956:55:000000000-BDTHK:1:2117:8005:19869 0 chr1 57529 60 76M * 0 0 AGCTTCCTTTCCAATATGAAGAATCTTATATAGCTTGTCT
30 M01956:55:000000000-BDTHK:1:2117:19311:17634 16 chr1 57635 60 76M * 0 0 CCTATGACTGAAAGACAGGTAAAGAGATGCTTTTAA
31 M01956:55:000000000-BDTHK:1:2117:8005:19869 16 chr1 57635 60 76M * 0 0 CCTATGACTGAAAGACAGGTAAAGAGATGCTTTTAA
32 M01956:55:000000000-BDTHK:1:2112:12793:14341 0 chr1 60225 60 76M * 0 0 TCTTATTACTCTATAATGTTCCCGGGTTCAATTCCT
33 M01956:55:000000000-BDTHK:1:2112:12793:14341 16 chr1 60316 60 76M * 0 0 GTAAGAAGTCCAGGACAGCACTGAATGAAGTGAAA
34 M01956:55:000000000-BDTHK:1:2112:9420:9180 0 chr1 61304 60 75M * 0 0 AATAAACCATTTATTCCTCAACTCTTATGCTCAATTTGATG
35 M01956:55:000000000-BDTHK:1:2112:9420:9180 16 chr1 61431 60 75M * 0 0 AAAAAACAAAGCACTACTGTTAATTAACATGTTGACTAT
36 M01956:55:000000000-BDTHK:1:1104:18027:4183 0 chr1 62124 60 75M * 0 0 GTGCTGGGATAACAGGCGTGAACACCACTGCCGGCTGTAA
37 M01956:55:000000000-BDTHK:1:1104:18027:4183 16 chr1 62236 60 62M4110M * 0 0 ACATACACACACACACACATATCTGTATATAC
38 M01956:55:000000000-BDTHK:1:2113:12853:5511 0 chr1 71322 60 75M * 0 0 GAGTCAGATGATAAGAGGGTCAAAATATGTTTATCTAGGA
39 M01956:55:000000000-BDTHK:1:2113:12853:5511 16 chr1 71409 60 76M * 0 0 AAGTAGTAATAATAAGCAGATGTTCAAACTAGTCAGGAG
40 M01956:55:000000000-BDTHK:1:1108:15145:16409 0 chr1 81512 23 76M * 0 0 TAGACATTTGTGTAGATTATTTGACCACTGAAGTCT
41 M01956:55:000000000-BDTHK:1:1108:15145:16409 16 chr1 81597 29 76M * 0 0 ATTTTCGGTGTGCTAAATGCAATTTTAACTATAGATAT
42 M01956:55:000000000-BDTHK:1:1106:4402:20687 0 chr1 96027 0 76M * 0 0 TGCATCTGCCCCCTGGGACTCTCTGTCAGAGGCTGAGAGCT
43 M01956:55:000000000-BDTHK:1:1106:4402:20687 16 chr1 96186 8 75M * 0 0 ATCAACTAGAAAATTTAATAAATAGTGCAGATTTGTAGAC
44 M01956:55:000000000-BDTHK:1:1113:4230:18528 0 chr1 108586 9 76M * 0 0 GAAGGATCTTATCTATCATGTGCTCACTCCCAAGAGGGGAC
45 M01956:55:000000000-BDTHK:1:1113:4230:18528 16 chr1 108697 0 76M * 0 0 GAGCAAAAATATGCTCAGCTTATTAATATGGATCTTAAT
46 M01956:55:000000000-BDTHK:1:1111:5182:7752 0 chr1 108832 10 75M * 0 0 TATCTGGTAGAGATGAGGCAATGATAGGAATGGGAAGCAG
47 M01956:55:000000000-BDTHK:1:1111:5182:7752 16 chr1 108942 0 75M * 0 0 TCAGCATTAGAGATGCCAGCCCTGATAGATATAAAACAGGAA
48 M01956:55:000000000-BDTHK:1:1114:15990:12251 0 chr1 260436 0 76M * 0 0 TGAGACACACTGAAAGTAAAGAGCAGGAGGAAACAAA
49 M01956:55:000000000-BDTHK:1:1114:15990:12251 0 chr1 348927 0 76M * 0 0 TTGAAGGCACACAGATTGTGTGAGTTCCAGGCTGAAC
50 M01956:55:000000000-BDTHK:1:2104:5830:15244 0 chr1 646416 0 76M * 0 0 ATAAAAATTCAGTTGTTTGTATACAGATAGAAATGGCCCTTG
```

Rysunek 1: Zawartość pliku *SAM*

Pytania:

1.1 Sprawdź zawartość wygenerowanego pliku SAM. Jaka jest typowa długość odczytów?

Średnia długość odczytów to **73 - 76**.

2 Sortowanie

Wykonano polecenie w celu sortowania pliku i wygenerowania pliku **BAM**:

```
sudo docker run -v ./:/data biocontainers/samtools:v1.9-4-deb_cv1 \samtools  
sort -O BAM -o coriell_chr1.bam coriell_chr1.sam
```

Pytania:

2.1 Jak jest różnica w wielkości plików FASTQ, BAM, SAM?

```
-rwxrwxrwx 1 root root 14137450 Mar 14 18:27 coriell_chr1.bam  
-rwxrwxrwx 1 root root 56984022 Mar 14 17:54 coriell_chr1.fq  
-rwxrwxrwx 1 root root 75895193 Mar 14 18:05 coriell_chr1.sam
```

Rysunek 2: Porównanie wielkości plików FASTQ, BAM, SAM

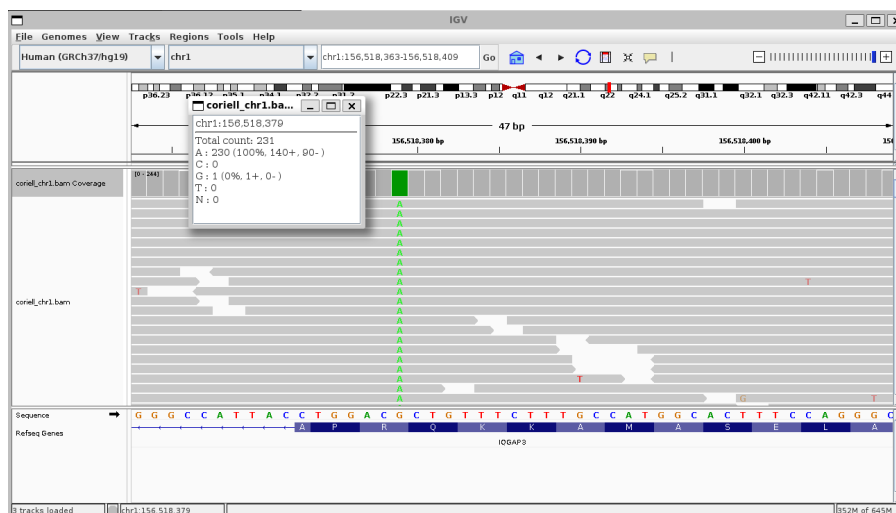
Największy jest plik **SAM**, następnie **FASTQ** a potem **BAM**. Wiąże się to z tym, że plik **BAM** jest zakodowany binarnie, co wpływa na mniejszy rozmiar.

3 Wizualizacja zawartości pliku BAM w programie IGV

Poniższą komendą wykonano indeksowanie pliku **BAM**:

```
sudo docker run -v ./:/data biocontainers/samtools:v1.9-4-deb_cv1 samtools  
index coriell_chr1.bam
```

Za pomocą programu **IGV** wyszukano gen **IQGAP3** i znaleziono wariant o pokryciu całkowitym powyżej **10x**:



Rysunek 3: Zrzut ekranu z programu **IGV**

Pytania:

3.1 Jaka jest pozycja tego wariantu?

156,518,379

3.2 Ile odczytów wskazuje na wariant a ile na referencje?

Na wariant wskazuje **230** (A), a na referencje **1** odczyt (G).

3.3 Czy jest to wariant homo czy heterozygotyczny?

Wariant wskazujący na **230** (A) i referencję **1** odczyt (G) wskazuje na to, że wariant ten jest heterozygotyczny.

4 Wykrywanie wariantów

Poniższą komendą wygenerowano plik **BCF**:

```
sudo docker run -v ./:/data biocontainers/samtools:v1.9-4-deb_cv1 samtools  
mpileup -Ou -f chr1.fa coriell_chr1.bam > coriell_chr1.bcf
```

Następnie przy pomocy poniższej komendy wygenerowano plik **VCF**:

```
sudo docker run -v ./:/data biocontainers/bcftools:v1.9-1-deb_cv1 bcftools  
call -mv coriell_chr1.bcf > coriell_chr1.vcf
```

Pytania:

4.1 Ile wariantów zawiera plik?

Nagłówek pliku **vcf** (nieodfiltrowanego) ma **29** linijek. Po odjęciu ich od ogólnej liczby linijek w pliku (**6079**) otrzymujemy **6050** wariantów.

4.2 Ile wariantów zostało po filtracji? Jakich innych parametrów możemy użyć do dalszej filtracji liczby wariantów?

Za pomocą poniższej komendy dokonano filtrowania wariantów z niskim pokryciem:

```
sudo docker run -v ./:/data biocontainers/bcftools:v1.9-1-deb_cv1 bcftools  
filter -i "INFO/DP>10" coriell_chr1.vcf > coriell_chr1_filtered.vcf
```

Nagłówek pliku po filtracji ma **31** linijek. Po odjęciu ich od ogólnej liczby linijek w pliku (**272**) otrzymujemy **241** wariantów.

Do dalszej filtracji liczby wariantów możemy wykorzystać:

- **Jakość (QUAL)**: Wartość jakościowa, która jest miarą pewności, że wariant jest prawdziwy.
- **Głębokość pokrycia (DP)**: Głębokość pokrycia odczytów wariantu, czyli liczba odczytów mapujących się do danego regionu genomowego.
- **Allelic Depth (AD)**: Liczba odczytów obsługujących alternatywny allel.

5 Adnotacje wariantów

Do adnotacji przefiltrowanego pliku **VCF** wykorzystano narzędzie *VEP* w wersji online.

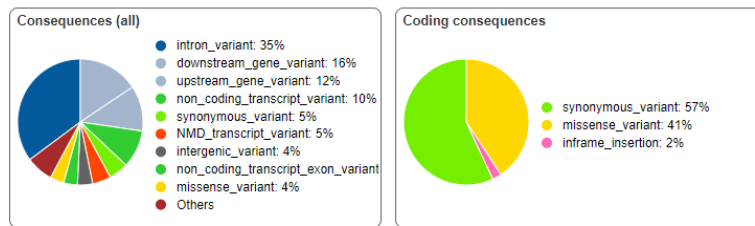
Wyniki adnotacji przedstawione są poniżej:

Variant Effect Predictor results ?

Job details

Summary statistics

Category	Count
Variants processed	241
Variants filtered out	0
Novel / existing variants	17 (7.1) / 224 (92.9)
Overlapped genes	37
Overlapped transcripts	131
Overlapped regulatory features	15



Rysunek 4: Zrzut ekranu z programu VEP

Pytania:

5.1 Jaki typ wariantu przeważa?

Przeważa typ *intron_variant* (35%).

5.2 Pobierz zaadnotowaną listę wariantów w formacie txt (Download TXT). Wyszukaj wariant, który wcześniej zidentyfikowałeś ręcznie w programie IGV. Załącz do sprawozdania wiersze odpowiadające temu wariantowi. Czy jest to wariant w części kodującej?

1:156513579-156513579	T	upstream_gene_variant	MODIFIER	IQGAP3	ENSG00000183856	Transcript	ENST00000498755.1	processed_transcript	-	-	-
1:156513579-156513579	T	regulatory_region_variant	MODIFIER	-	-	RegulatoryFeature	ENSR00001506378	enhancer	-	-	-
1:156518379-156518379	A	missense_variant	MODERATE	IQGAP3	ENSG00000183856	Transcript	ENST00000361170.2	protein_coding	17/38	-	-
1:156518379-156518379	A	missense_variant,NMD_transcript_variant	MODERATE	IQGAP3	ENSG00000183856	Transcript	ENST00000491900.1	no	-	-	-
1:156520173-156520173	A	intron_variant	MODIFIER	IQGAP3	ENSG00000183856	Transcript	ENST00000361170.2	protein_coding	-	15/37	-

Rysunek 5: Wiersze odpowiadające wariantowi IQGAP3

Wpis *protein_coding* sugeruje, że jest to wariant w części kodującej.

6 Zadanie implementacyjne

6.1 Treść

Proszę zapoznać się z formatem pliku refFlat:

<https://genome.ucsc.edu/goldenPath/gbdDescriptions.html>,

oraz pobrać jego zawartość z:

<https://hgdownload.cse.ucsc.edu/goldenpath/hg19/database/refFlat.txt.gz>

Proszę napisać skrypt, który wyliczy ile wariantów (z pliku *coriell_chr1.vcf*) znajduje się w poszczególnych genach, których współrzędne znajdują się w pliku refFlat. Jako początek i koniec genu należy przyjąć kolumny *txStart* i *txEnd*. Skrypt powinien zwracać tabelę z dwiema kolumnami (symbol genu, liczba wariantów). Należy dokonać implementacji w języku R z wykorzystaniem pakietu *GenomicRanges*:

<https://bioconductor.org/packages/release/bioc/html/GenomicRanges.html>

lub w języku python z wykorzystaniem biblioteki *pyranges*

<https://github.com/biocompare/pyranges>

<https://github.com/biocompare/pyranges>.

6.2 Rozwiązanie

Dane z plików wejściowych są przetwarzane w ramki **PyRanges**. Obiekt ramki jest następnie filtrowany zgodnie zadaną nazwą chromosomu. Wyodrębniona lista unikalnych genów umożliwia sprawdzanie w pętli liczbę wariantów dla każdego genu. Po zakończeniu pętli wyniki są zapisywane do pliku wyjściowego a początkowe wiersze (zgodnie z zadanym parametrem) są wyświetlane w terminalu.

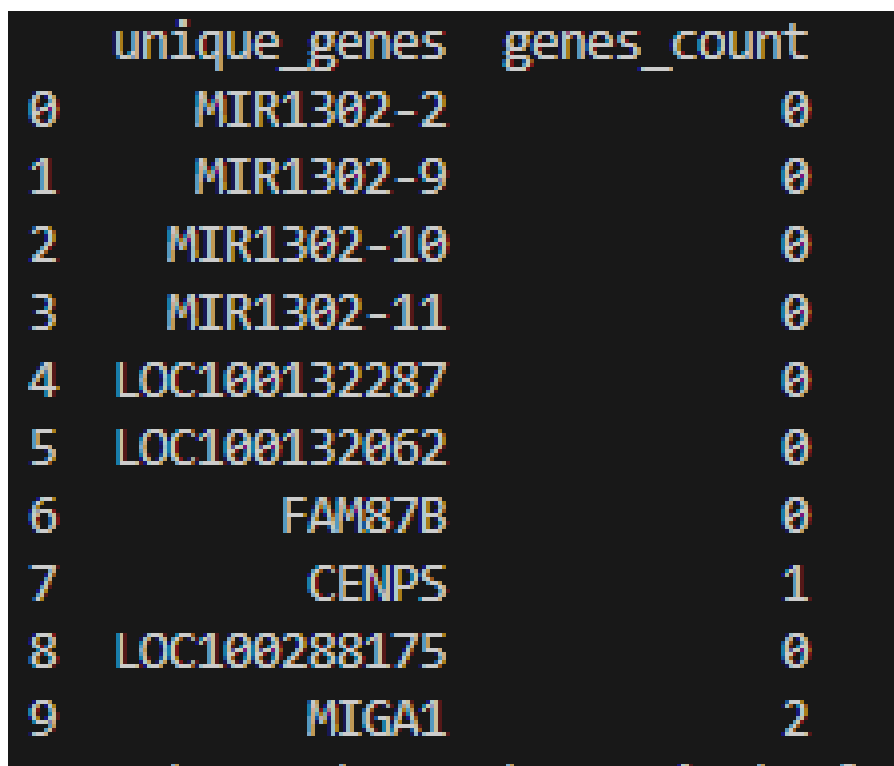
Ze względu na oszczędność miejsca w sprawozdaniu, nie został w nim zamieszczony skrypt. Został on natomiast dołączony do plików laboratoryjnych.

6.3 Wyniki

Po wywołaniu komendy:

```
python3 .\variant_counter.py --reffile ./refFlat.txt --chromosome chr1 --vcffile .\coriell_chr1.vcf --output ./output.csv --result_count 10
```

Na ekranie widzimy 10 początkowych genów wynikowych, a całość dostępna jest w pliku *output.csv*.



	unique_genes	genes_count
0	MIR1302-2	0
1	MIR1302-9	0
2	MIR1302-10	0
3	MIR1302-11	0
4	LOC100132287	0
5	LOC100132062	0
6	FAM87B	0
7	CENPS	1
8	LOC100288175	0
9	MIGA1	2

Rysunek 6: Informacja zwrotna zwracana przez skrypt.