

# NGS\_final\_project

Małgorzata\_Bujak

2025-06-16

## Opis projektu

Kompleksowa bioinformatyczna analiza danych z badania - dane pobrane z Sequence Read Archive:  
**SRR1536581**

SRX669648: GSM1465030: DKO1 H3K4me3 replicate 1 ChIP-seq; Homo sapiens; ChIP-Seq 1 ILLUMINA (Illumina HiSeq 2000)

## Study

Global loss of DNA methylation uncovers intronic enhancers in genes

## Abstract

We used HCT116 colorectal cancer cells with and without mutations in DNA methyltransferases (resulting in a 95% reduction in global DNA methylation levels) to study the relationship between DNA methylation, histone modifications, and gene expression. (The double knockout cell line is called DKO1) Overall design: Examination of DNA methylation, two histone modifications, RNA polymerase II ChIP-seq and RNA expression in two cell line (HCT116 and DKO1). One of replicates of HCT116 RNA P II ChIP-seq is from GSM970210. Histone modification data(H3K27ac, H3K4me3) of HCT116 are from GSM945304, GSE31755. Two replicates of RNA-seq data in HCT116 are from GSM1266733 and GSM1266734.

## Analiza

```
# Aktywacja środowiska
conda activate SRA_tools

# Pobranie plików fastq
fastq-dump SRR1536581 --split-3 --skip-technical --gzip

# Analiza jakości odczytów programem FastQC
# Tworzenie katalogu na raporty
mkdir final_project_quality_reports

# Uruchomienie FastQC na pliku fastq z użyciem 4 wątków
fastqc -t 4 -o final_project_quality_reports/ *.fastq.gz
```

Wygenerowano raport jakości - plik w formacie HTML o nazwie **SRR1536581\_fastqc**

## Opis raportu jakości pobranych danych z sekwencjonowania (raw):

- odczyty mają długość 50 pz
- wszystkie są dobrej jakości (0 ocenionych na słabą jakość)
- średnia zawartość par GC to 50%, co odbiega od standardowej zawartości par CG w próbkach - ale przy analizie epigenetycznej nie jest to alarmujące ani nic dziwnego
- odczyty zawierają zanieczyszczenie adapterami (Overrepresented sequences), ale adaptery nie są wykrywane w swoich pozycjach (np. na końcu czy początku sekwencji, stąd Adapter Content jest zielone)

```
# Usuwanie sekwencji niskiej jakości programem Trimmomatic (automatyzacja za pomocą skryptu w Bash)

#!/bin/bash

source activate trimed

# Lista próbek
samples=("SRR1536581")

# Liczba rdzeni
THREADS=4

# Parametry programu Trimmomatic
PARAMS="LEADING:8 TRAILING:8 SLIDINGWINDOW:5:20 MINLEN:40"

# Pętla po wszystkich próbkach
for sample in "${samples[@]}"
do
echo "Przycinam próbki: $sample"

trimomatic SE -phred33 -threads $THREADS \
${sample}.fastq.gz \
${sample}_trimmed.fastq.gz \
$PARAMS
echo "Zakończono: $sample"
done

# Raport jakości z programu FastQC po przycięciu
fastqc -t 4 -o final_project_quality_reports/ *trimmed.fastq.gz
```

Wygenerowano raport: **SRR1536581\_trimmed\_fastqc**

**Komunikat z programu Trimmomatic:** Input Reads: 30681927 Surviving: 27757634 (90.47%) Dropped: 2924293 (9.53%)

Te dane wskazują, że 9,53% odczytów zostało odrzuconych, bo nie spełniło ustawionych parametrów.

```
# Usuwanie adapterów programem Cutadapt
cutadapt -a AGATCGGAAGAGC -o SRR1536581_cutadapt.fastq.gz SRR1536581_trimmed.fastq.gz

# Raport jakości z programu FastQC po usuwaniu adapterów
fastqc -t 4 -o final_project_quality_reports/ *cutadapt.fastq.gz
```

Wygenerowano raport: plik o nazwie **SRR1536581\_cutadapt.fastqc**

**Podsumowanie Cutadapt:** Total reads processed: 27,757,634 Reads with adapters: 3,908,991 (14.1%)  
Reads written (passing filters): 27,757,634 (100.0%)

Total basepairs processed: 1,379,279,334 bp Total written (filtered): 1,212,145,745 bp (87.9%)

```
# Tworzenie folderu do zapisania wyników z programu MultiQC  
mkdir multiqc_raports  
  
# Raport zbiorczy wygenerowany programem MultiQC  
multiqc -o final_project_quality_reports/multiqc_raports -f -v final_project_quality_reports
```

## Porównanie statystyk odczytów przed i po przycinaniu i usuwaniu adapterów

- zmniejszyła się długość odczytów od 0 do 50 pz
- nadal wszystkie są dobrej jakości (0 ocenionych na słabą jakość)
- średnia zawartość par GC zmniejszyła się do 48% - wykres lepiej wygląda, mniejsze odchylenia
- nie ma zanieczyszczeń sadapterami
- poprawiła się statystyka Per base sequence content
- zmieniła się długość odczytów, co widać na wykresie Sequence Length Distribution

Raport multiqc - plik o nazwie **multiqc\_report**

```
# Mapowanie przyciętych odczytów przy użyciu BWA MEM  
  
# Aktywacja środowiska  
conda activate mapping  
  
# Utworzenie folderu genome_index  
mkdir -p genome_index  
  
# Rozpakowanie genomu referencyjnego  
gunzip hg38.fa.gz  
  
# Indeksowanie genomu referencyjnego  
bwa index hg38.fa      # za mało RAM, więc skopiowałem gotowy indeks: Homo_sapiens_assembly38.fasta  
  
# Mapowanie przyciętych odczytów SE (single-end)  
bwa mem -t 4 "/mnt/c/Users/magor/25_PJATK_Genomika/NGS_final_project/genome_index/Homo_sapiens_assembly38.fasta" Galaxy3_human_mapped_reads.bam -o Human_rep.sorted.bam
```

Na Galaxy zmapowałam odczyty oraz uzyskałam indeks. Ale przeszłam jeszcze lokalnie sortowanie i indeksowanie

```
# Sortowanie pliku BAM według koordynatów  
samtools sort Galaxy3_human_mapped_reads.bam -o Human_rep.sorted.bam  
  
# Nadanie indeksu BAM (uzyskasz plik typu .bai)  
samtools index Human_rep.sorted.bam  
  
# Statystyki mapowania  
samtools flagstat Human_rep.sorted.bam > mapping_stats.txt
```

**Statystki mapowania w pliku o nazwie:** **mapping\_stats** 85.59% odczytów zostało zmapowanych.  
Mapowanie wygląda dobrze, brak duplikatów.

```

# Usuń duplikaty
samtools markdup -r Human_rep.sorted.bam Human_rep.sorted_dedup.bam

# Nadaj indeks posortowanemu plikowi BAM (uzyskasz plik typu .bai)
samtools index Human_rep.sorted_dedup.bam

```

Korzystam z pliku po sortowaniu Human\_rep.sorted.bam

```

# Nadanie indeksu genomowi ref
samtools faidx Homo_sapiens_assembly38.fasta

# Wyświetlenie wariantów w programie IGV
igv_hidpi

# Wykonaj identyfikację wariantów
#Aktywacja środowiska
conda activate BCF_tools

# Oszacowanie stopnia pokrycia programem BCFtools
bcftools mpileup -O b -o raw_1.bcf -f "/mnt/c/Users/magor/25_PJATK_Genomika/NGS_final_project_human/genome/Homo_sapiens_assembly38.fasta"

# Identyfikacja wariantów pojedynczych nukleotydów (SNV) za pomocą BCFtools
bcftools call -m -v --ploidy 1 -o final_project_bcf_variants_1.vcf raw_1.bcf

# Statystyki
bcftools stats final_project_bcf_variants_1.vcf > _final_project_bcf_variant_stats.txt

less _final_project_bcf_variant_stats.txt

# Wyświetlenie info o SNP, InDel
bcftools view -v snps final_project_bcf_variants_1.vcf | grep -v "^#" | wc -l

bcftools view -v indels final_project_bcf_variants_1.vcf | grep -v "^#" | wc -l

```

**SNPs:** 545327 **InDel:** 27024

Fragment pliku ze statystykami po identyfikacji wariantów, pełne wyniki w pliku \*\*\_final\_project\_bcf\_variant\_stats.txt\*\*  
 SN [2]id [3]key [4]value SN 0 number of samples: 1 SN 0 number of records: 572351 SN 0 number of no-ALTs:  
 0 SN 0 number of SNPs: 545327 SN 0 number of MNPs: 0 SN 0 number of indels: 27024 SN 0 number of others: 0 SN 0 number of multiallelic sites: 0 SN 0 number of multiallelic SNP sites: 0

```

# Filtrowanie wariantów
Wyfiltruj warianty według określonych kryteriów (np. minimalna jakość, głębokość pokrycia)
bcftools filter -s LOWQUAL -e '%QUAL<20 || DP<10' final_project_bcf_variants_1.vcf -Ov -o filtered_variants.vcf

# Porównanie liczb wariantów przed i po filtrowaniu
bcftools stats filtered_variants.vcf > variant_stats.txt

bcftools view -v snps filtered_variants.vcf | grep -v "^#" | wc -l

bcftools view -v indels filtered_variants.vcf | grep -v "^#" | wc -l

```

## Uzasadnij wybrane kryteria filtrowania

- minimalna jakość < 20 , aby pozbyć się odczytów o słabej jakości
- głębokość pokrycia < 10, aby uniknąć fałszywie pozytywnych wyników (czyli mieć pewność, że dany wariant jest prawdą, a nie błędem)

Przed filtrowniem: **SNPs:** 545327 **InDel:** 27024

Po filtrowaniu: **SNPs:** 545327 **InDel:** 27024

Wyniki przed i po filtrowaniu są takie same dla SNP i InDel

## Adnotacja wariantów programem SnpEff na Galaxy

Wykorzystany plik: filtered\_variants.vcf

## Konsekwencje funkcjonalne wariantów

Warianty międzygenowe stanowią około 25%, natomiast tylko 1,88% znajduje się w regionach eksonowych, które kodują geny i mogą wpływać na budowę czy funkcję białek.

Występują też warianty w regionach UTR oraz w miejscach splicingu, które mogą wpływać na regulację genów i składanie RNA.

70% wariantów występuje w regionach nikodujących, czyli intronach. Mogą one wpływać na poprawność składania RNA i stabilność transkryptów. Ich efekt często jest trudniejszy do przewidzenia niż wariantów w eksonach.

Analiza konsekwencji funkcjonalnych wariantów wykazała, że zdecydowana większość, bo ok. 98,5% zidentyfikowanych mutacji należy do kategorii MODIFIER, czyli lokalizuje się w regionach o nieznanym lub minimalnym wpływie na funkcję białka. Wśród wariantów z potencjalnym skutkiem na strukturę białka (HIGH, MODERATE, LOW impact) największą grupę stanowią mutacje missense (61,5%), które mogą wpływać na zmianę aminokwasów i potencjalnie funkcję białek. Warianty o wysokim wpływie (0,066%), takie jak mutacje nonsensowne i frameshift, choć stanowią niewielki procent, są szczególnie istotne z punktu widzenia ich potencjalnej roli w chorobach genetycznych.”