Analiza transkryptomu

semestr zimowy 2023/2024

Projekt zaliczeniowy

Hubert Kokoszka

1. Wstęp

Celem niniejszego projektu jest przeprowadzenie analizy RNA-Seq na plikach odczytów sekwencjonowania bioprojektu o numerze PRJNA313294. W tym bioprojekcie wykorzystano dwa różne instrumenty sekwencjonujące w celu zbadania różnic w transkryptomie komórek progenitorowych tkanki nerwowej zakażanych wirusem Zika (próba badawca) oraz komórek niezakażonych (kontrola, *mock-infected*). Poszczególne przebiegi sekwencjonowania zostały scharakteryzowane w Tabeli 1.

Tabela 1. Charakterystyka przebiegów sekwencjonowania.

Przebieg	Średnia długość odczytu	Próba	Instrument	Biblioteka
SRR3191542	150 nt	Badana	Illumina MiSeq	Sparowana
SRR3191543				
SRR3191544		Kontrola		
SRR3191545				
SRR3194428	75 nt	Badana	NextSeq 500	Pojedyncza
SRR3194429				
SRR3194430		Kontrola		
SRR3194431				

W celu realizacji projektu zaliczeniowgo napisano odpowiedni skrypt *X_project_script.sh* w języku bash, którego zadaniem było wygenerowanie macierzy zliczeń odczytów oraz skrypt *de_analysis.R* w jeżyku R, którego zadaniem było przeprowadzenie analizy *Differential expression* oraz wygenerowanie odpowiednich grafik.

2. Pobieranie danych

Do pobierania danych zastosowano poniższy kod:

project_script.sh – zebranie numerów przebiegów

```
esearch -db sra -query "$project_number" \
      | efetch -format runinfo -mode xml \
      | xtract -pattern SraRunInfo -element Run \
      | awk '{for (i=1; i<=NF; i++) print $i}' > SRR_list.txt

while IFS= read -r run_number; do
      fastq-dump -X 2000 "$run_number" --split-files -O Projekt/reads
done < SRR_list.txt
```

Na podstawie zadanego numeru bioprojektu zostały wyszukane numery przebiegów. Wyjściowo są one zapisywane w pliku tekstowym jako jeden wiersz. Tutaj zastosowano przekazanie do obróbki tak, aby zostały zapisane w pliku *SRR_list.txt* w formie kolumny. Następnie plik ten jest przekierowany do pętli, która za pomocą komendy `fastq-dump` pobiera pliki z odczytami.

Pliki SRR zostały dodatkowo rozdzielone na dwa foldery ze względu na to, czy były one sparowane czy nie wg poniższego kodu.

project_script.sh – ściąganie plików SRR

```
for file in ./Projekt/reads/*_1.fastq; do
    if [ -e "$file" ]; then
        read_name=$(basename "$file" "_1.fastq")
        if [ -e "Projekt/reads/${read_name}_2.fastq" ]; then
            mv "$file" "Projekt/reads/pair_end/"
            mv                          "Projekt/reads/${read_name}_2.fastq"
"Projekt/reads/pair_end/"
        else
            mv "$file" "Projekt/reads/single_end/"
        fi
    fi
done
```

3. Kontrola jakości i filtrowanie

Przeprowadzono kontrolę jakości odczytów oraz odfiltrowanie odczytów niezadowalających wg poniższego kodu. Raporty kontroli jakości zostały również wygenerowane po filtrowaniu

project_script.sh – filtrowanie odczytów o złej jakości

```
for raw_read in Projekt/reads/pair_end/*.fastq; do
      read_name=$(basename "$raw_read" _1.fastq)
      java -jar /bioapp/Trimmomatic-0.39/trimmomatic-0.39.jar PE \
            "Projekt/reads/pair_end/${read_name}_1.fastq"
"Projekt/reads/pair_end/${read_name}_2.fastq" \
            "Projekt/trimmed_reads/pair_end/${read_name}_trimmed_1.fastq"
"Projekt/trimmed_reads/pair_end/${read_name}_1_unpaired_1.fa" \
            "Projekt/trimmed_reads/pair_end/${read_name}_trimmed_2.fastq"
"Projekt/trimmed_reads/pair_end/${read_name}_2_unpaired.fa" \
            ILLUMINACLIP:pe_adapter.fa:5:20:5 SLIDINGWINDOW:4:20
done

for raw_read in Projekt/reads/single_end/*.fastq; do
      read_name=$(basename "$raw_read" _1.fastq)
      java -jar /bioapp/Trimmomatic-0.39/trimmomatic-0.39.jar SE \
            "Projekt/reads/single_end/${read_name}_1.fastq" \
      "Projekt/trimmed_reads/single_end/${read_name}_trimmed_1.fastq" \
            ILLUMINACLIP:se_adapter.fa:2:20:5 SLIDINGWINDOW:4:20
done
```

Do filtrowania wykorszystano pliki z adapterami

se_adapter.fa

```
>Adapter
AGATCGGAAGAGCACACGTCTGAACTCCAGTCA
```

pe_adapter.fa

```
>PE/1
AGATCGGAAGAGCACACGTCTGAACTCCAGTCA
>PE/2
AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT
```

project_script.sh – kontrola jakości

```
for raw_read in Projekt/reads/pair_end/*.fastq; do
    fastqc -o Projekt/fastqc/raw "$raw_read"
done

for raw_read in Projekt/reads/single_end/*.fastq; do
    fastqc -o Projekt/fastqc/raw "$raw_read"
done


for trimmed_read in Projekt/trimmed_reads/pair_end/*.fastq; do
    fastqc -o Projekt/fastqc/trimmed "$trimmed_read"
done

for trimmed_read in Projekt/trimmed_reads/single_end/*.fastq; do
    fastqc -o Projekt/fastqc/trimmed "$trimmed_read"
done
```

4. Mapowanie odczytów

Stworzono indeks genomowy *hisat2* na podstawie transkryptomu hg19.

project_script.sh – indeks genomowy

```
wget https://hgdownload.cse.ucsc.edu/goldenpath/hg19/bigZips/hg19.fa.gz \
    -O Projekt/ref/hg19.fa.gz
gunzip Projekt/ref/hg19.fa.gz
hisat2-build -p 2 Projekt/ref/hg19.fa hg19_index
```

Odczyty zmapowano. Otrzymane pliki .sam zostały przekonwertowane do plików .bam.

project_script.sh – indeks genomowy

```
for read in Projekt/trimmed_reads/pair_end/*_1.fastq; do
    read_name=$(basename "$read" _trimmed_1.fastq)

    hisat2 -q -x hg19_index \
        -1 "Projekt/trimmed_reads/pair_end/${read_name}_trimmed_1.fastq" \
        -2 "Projekt/trimmed_reads/pair_end/${read_name}_trimmed_2.fastq" \
        -S "Projekt/bams/pair_end/${read_name}.sam"

    samtools view -Sb -@ 2 "Projekt/bams/pair_end/${read_name}.sam" \
        > "Projekt/bams/pair_end/${read_name}.bam"
    rm "Projekt/bams/pair_end/${read_name}.sam"
done


for read in Projekt/trimmed_reads/pair_end/*_1.fastq; do
    read_name=$(basename "$read" _trimmed_1.fastq)
    echo "hisat2 alignment for $read_name started"
    hisat2 -q -x hg19_index \
        -U "Projekt/trimmed_reads/single_end/${read_name}_trimmed_1.fastq" \
        -S "Projekt/bams/single_end/${read_name}.sam"

    samtools view -Sb -@ 2 "Projekt/bams/single_end/${read_name}.sam" \
        > "Projekt/bams/single_end/${read_name}.bam"
    rm "Projekt/bams/single_end/${read_name}.sam"
done
```

5. Zliczanie odczytów

Odczyty zliczono oraz zapisano macierze zliczeń jako pliki txt, które zostały następnie przekazane skryptowi R.

project_script.sh – indeks genomowy

```
wget https://ftp.ebi.ac.uk/pub/databases/gencode/Genco... \
    -O Projekt/ref/hg19.gtf.gz
gunzip Projekt/ref/hg19.gtf.gz

featureCounts -a Projekt/ref/hg19.gtf \
      -g "gene_name" \
      -o "Projekt/MiSeq_counts.txt" \
    Projekt/bams/pair_end/*.bam

echo "Commencing feature count for single-end reads..."
featureCounts -a Projekt/ref/hg19.gtf \
      -g "gene_name" \
      -o "Projekt/NextSeq_counts.txt" \
    Projekt/bams/single_end/*.bam

Rscript de_analysis.R "Projekt/MiSeq_counts.txt"
Rscript de_analysis.R "Projekt/NextSeq_counts.txt"
```

6. Analiza *Differential Expression*

Przeprowadzono analizę z wykorzystaniem paczki DESeq2.

de_analysis.R – wczytanie danych

```
args <- commandArgs(trailingOnly=TRUE)
counts_file <- args[1]
instrument = sub("\\_counts.txt$", "", counts_file)

count_matrix <- read.table(counts_file, header=TRUE, row.names=1)

count_data <- count_matrix[,6:length(count_matrix[1,])]
colnames(count_data) <-  sub("\\.bam$", "", colnames(count_data))
```

Następnie przygotowano tabelę, która ma służyć, jako metadane dla funkcji tworzącej obiekt DESeq2

de_analysis.R – metadane

```
sample_names <- names(count_data)

condition <- as.factor(
  c("mock", "mock",
    "zika", "zika"))

meta_data <- data.frame(samples=sample_names,
                        condition=condition)
```

Stworzono obiekt DESeq2, oszacowano współczynniki normalizacji oraz znormalizowano dane.

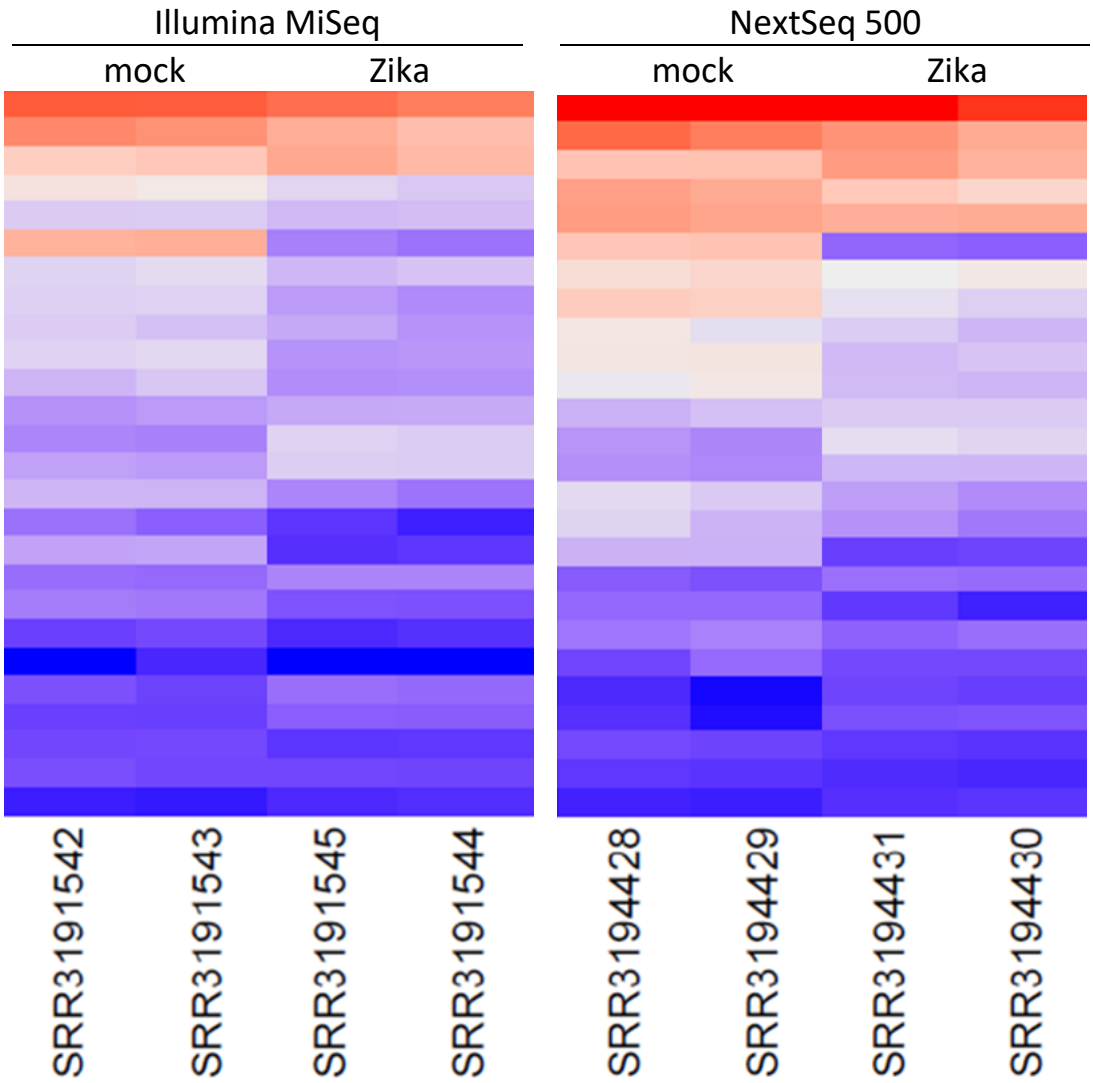
de_analysis.R – dds i normalizacja

```
dds <- DESeqDataSetFromMatrix(countData=count_data,
                              colData=meta_data,
                              design = ~ condition)
dds <- DESeq(dds)
dds <- estimateSizeFactors(dds)

log_data <- rlog(dds)
norm_data_matrix <- assay(log_data)
norm_data <- as.data.frame(norm_data.matrix)
```

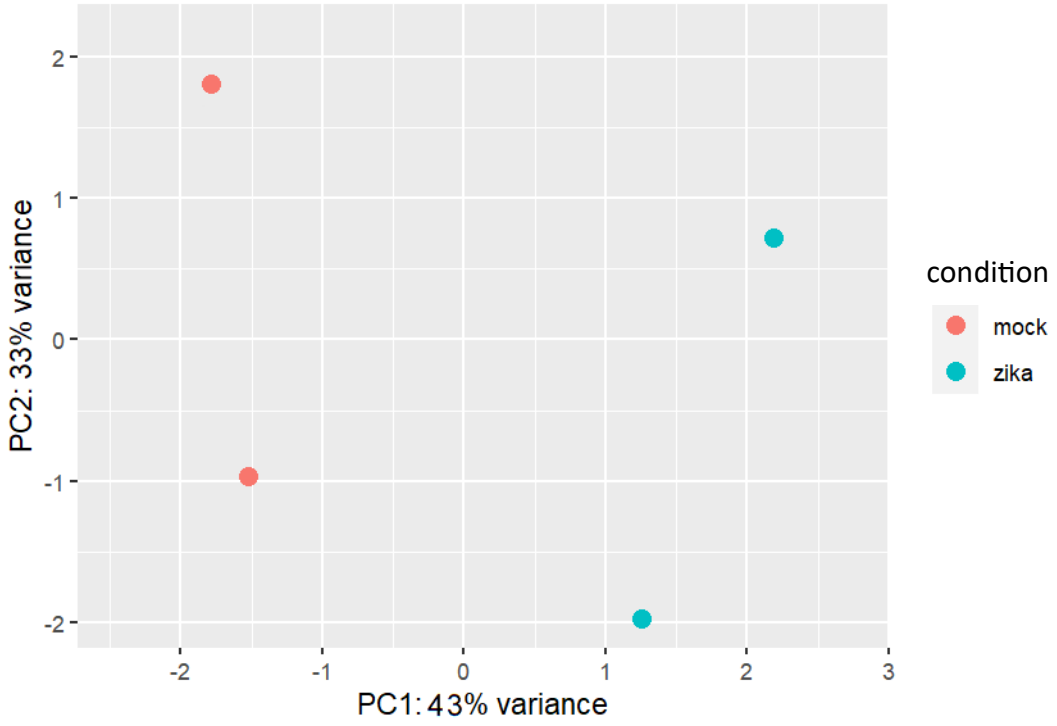
7. Heatmapy

Wygenerowano heatmapy wykazującą różnice w ekspresji pomiędzy próbkami

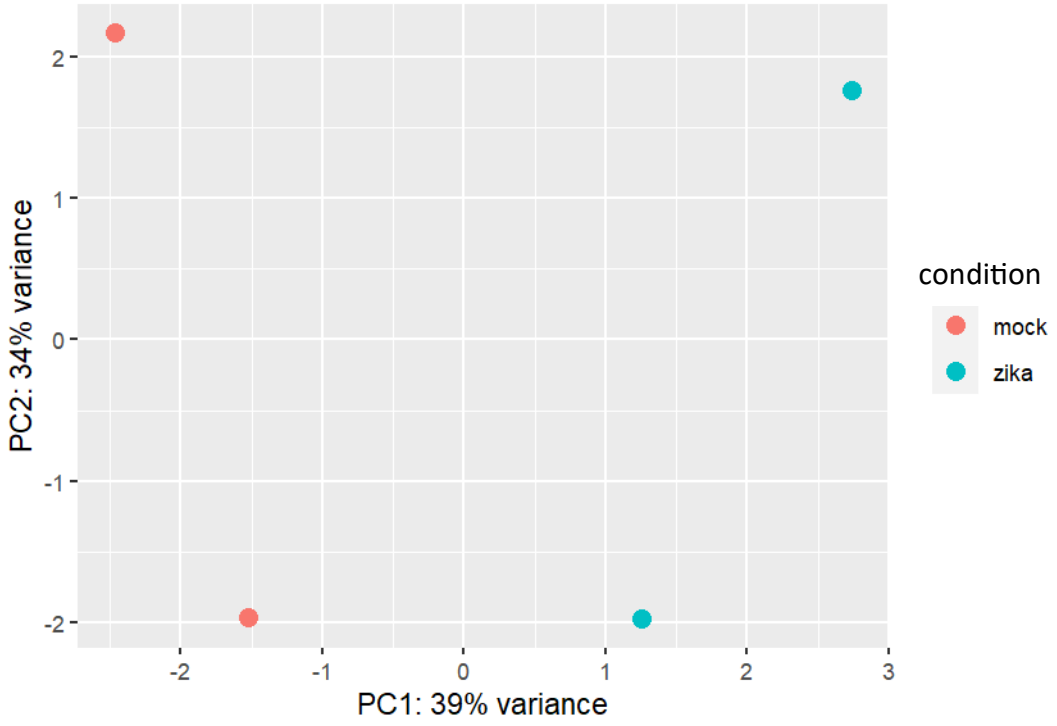


8. Analiza składników głównych *PCA*

Przeprowadzono PCA i wygenerowano wykresy.



Wykres PCA dla instrumentu Illumina MiSeq (sparowane odczyty)



Wykres PCA dla instrumentu NextSeq500 (pojedyncze odczyty)

9. Wnioski

Oba instrumenty do sekwencjonowania dały bardzo podobne wyniki. Widać to zarówno na heatmapach oraz wykresach PCA.